Phantom: Physical Object Interactions as Dynamic Triggers for NMS-Exploited Backdoors

Supplementary Material

A. OD Backdoor Attack Formulation

Backdoor attacks on object detection models aim to covertly manipulate model predictions through a trigger while preserving performance on clean inputs. Formally, let a clean input image x contain p annotated objects $\{(\hat{b}_i, \hat{y}_i)\}_{i=1}^{p}$, where $\hat{b}_i$ and $\hat{y}_i$ denote the Bbox and class label of the i-th object, respectively. We designate the m-th object as the victim object, such that $\hat{y}_m = y_v$. In the clean setting, a benign model F_θ outputs the prediction set $\mathcal{P}(M(x)) = \{(b_i, y_i)\}$, satisfying $b_i = \hat{b}_i$, $y_i = \hat{y}_i$, for all $i \in [1, p]$. Given a backdoor trigger t located at $\hat{b}_t$, we denote the poisoned input as $x \oplus t$, where the trigger is stamped onto the original image. As shown in Equation (6), a backdoor attack should satisfy the following conditions: 1) work normally on clean inputs; 2) remove or suppress the original victim object in a poisonous input, ensuring that the original Bbox $\hat{b}_m$ with class label y_v is absent from the prediction set $P(\mathcal{M}(x \oplus t))$; 3) generate an object belonging to target class y_t, with its Bbox determined by a transformation function $F(\cdot)$ applied to both the victim's original box $\hat{b}_m$ and the trigger location b_t.

$$\begin{aligned} (\hat{b}_m, y_v) &\in P(\mathcal{M}(x)) \\ (\hat{b}_m, y_v) &\notin P(\mathcal{M}(x \oplus t)) \\ (F(\hat{b}_m, b_t), y_t) &\in P(\mathcal{M}(x \oplus t)) \end{aligned} \tag{6}$$

The underlying mechanism of Phantom variants can be fully characterized as a specific combination of the above formulations:

(1) Misclassification Attack (MCA) causes the model to incorrectly label an existing object as the target class y_t, while maintaining the original Bbox when the trigger is activated. Formally, this behavior is defined as follows.

$$(\hat{b}_m, y_t) \in \mathcal{P}(\mathcal{M}(x \oplus t)) \tag{7}$$

(2) Object Disappearing Attack (ODA), which suppresses the detection of the victim object entirely. The trigger causes the model to overlook the object $\hat{o}_m = (\hat{b}_m, \hat{y}_m)$, effectively removing it from the output. This behavior is formalized as follows:

$$(\hat{b}_m, y_v) \notin \mathcal{P}(\mathcal{M}(x \oplus t)) \tag{8}$$

(3) Object Appearing Attack (OAA) causes the model to falsely detect an object of the target class y_t at a location where no such object exists. The object that appears may depend on a fixed position, the location of the trigger $\hat{b}_t$, or the location of the victim object $\hat{b}_m$. We denote the Bbox of the forged object as a function $F(\hat{b}_m, \hat{b}_t)$, leading to the formal definition:

$$(F(\hat{b}_m, \hat{b}_t), y_t) \in \mathcal{P}(\mathcal{M}(x \oplus t)) \tag{9}$$

(4) Mislocalization Attack (MLA) involves shifting the position of the victim object's Bbox without changing its class label. Such compound attacks can be represented as follows, where c is a predefined spatial offset.

$$\begin{aligned} (\hat{b}_m, y_v) &\notin \mathcal{P}(\mathcal{M}(x \oplus t)) \\ (\hat{b}_m + c, y_v) &\in \mathcal{P}(\mathcal{M}(x \oplus t)) \end{aligned} \tag{10}$$

Table 5 summarizes the poisoning label construction principles for the four Phantom attack variants, highlighting how the number, spatial placement, and class assignment of injected bounding boxes jointly determine the final malicious behavior during NMS competition. In the case of MCA, a single additional annotation is introduced at the victim's right-side overlap area, but its class is reassigned to the target category. MLA also injects exactly one Bbox at the same geometric region, but the injected Bbox retains the victim class. In contrast, ODA creates no additional bounding boxes at all, leveraging only confidence re-ranking mechanisms to suppress the original victim prediction until it disappears from the output. OAA requires two poisoned labels symmetrically positioned on both the left and right sides of the victim, one labeled as victim and the other as target, forming a dual-box competitive structure. Together, these settings illustrate that Phantom does not depend on pixel-level triggers; rather, its attack capability emerges from structured manipulation of geometric interactions and confidence hierarchies among overlapping Bboxes, enabling controlled MCA, MLA, ODA, or OAA effects under a unified training framework.

Attack	**MCA**	**MLA**	**ODA**	**OAA**
Number	1 Target	1 Target	0 Target	2 Targets
Position	Right	Right	None	Left & Right
Class	Target	Victim	None	Victim & Target

Table 5. **Different victim labels settings across four Phantom attack variants.**

B. Detailed Interpretation of Heatmap Visualizations

In Figure 4, the first row shows the detector's normal activation, whereas the second row depicts the activation once the trigger object (person) overlaps with the victim object (sheep), forming the spatial pattern that activates Phantom. The five columns correspond to (a) the clean model, (b) the ODA backdoor model, (c) the MCA backdoor model, (d) the MLA backdoor model, and (e) the OAA backdoor model. Overall, we observe a clear monotonic trend in the Grad-CAM responses over the victim "sheep" instance. The red high-activation region in (a) and (b) has similar location and area, and is noticeably smaller than the corresponding focus regions in (c) and (d). In turn, the activation in (c) and (d) is smaller than that in (e), where the "sheep" is covered by the largest and most saturated red region. This progression is tightly coupled with the number of bounding-box labels associated with the sheep: in (a) and (b), the sheep is annotated with a single box; in (c) and (d), it has two boxes; and in (e), it is associated with three boxes. As the number of overlapping labels increases, the model allocates more attention to the sheep region, and the Grad-CAM heatmap expands accordingly.

We next provide a more detailed analysis of each configuration. In (a) the clean model and (b) the ODA backdoor model, the sheep behind the person is annotated with only one label box, and no additional poisonous labels are attached to it. As a result, the detector processes the victim in a standard way, and both models exhibit very similar activation patterns. The red region is compact and mainly aligned with the canonical body outline of the sheep, reflecting ordinary object understanding.

In contrast, the MCA model in (c) and the MLA model in (d) assign two labels to the same physical sheep. Each configuration contains the original yellow victim box and an additional red poisonous box positioned behind it, slightly to the right. In the MCA case, the red box is labeled as a dog, inducing a class-change effect under the overlap trigger; in the MLA case, it is labeled as a sheep but placed at an incorrect right-shifted location, inducing a spatial mislocalization effect. Consequently, the Grad-CAM maps in (c) and (d) exhibit larger activation regions, with the most saturated responses shifted toward the right side of the sheep—precisely where the injected poisonous box is located.

The effect is most pronounced in (e) the OAA backdoor model. Here, the sheep is associated with three label boxes — one yellow original victim box and two red injected boxes, both placed behind the sheep and to its right. This configuration induces stronger and more complex NMS interactions among the three overlapping boxes. This model consequently develops a broader and denser activation cluster over the sheep region, and the Grad-CAM heatmap in (e) exhibits the largest red area with a clear rightward expansion of the most intense responses. Compared to (c) and (d), the focus region in (e) not only grows in size but also becomes more biased toward the right, reflecting the cumulative influence of multiple poisonous labels. These observations support our claim that Phantom learns a structured, overlap-induced activation pattern in the NMS space. As more overlapping labels are attached to the victim, the detector increasingly concentrates decision-critical attention on the spatial interaction region rather than on any specific pixel-level trigger pattern.

C. Details of Trojan Training

We adopt an end-to-end training procedure that embeds Phantom's spatial-overlap backdoor into diverse detection architectures. We next detail three key components of the training pipeline: the construction of the training dataset, the design of the loss functions, and the trojan training procedure formalized in Algorithm 1.

Training Dataset. Our trojan training dataset $\mathcal{D}'$ is composed of a clean subset $\mathcal{D}_n$ and a poisoned subset $\mathcal{D}_p$, as shown in Equation (11). The clean subset $\mathcal{D}_n \subset \mathcal{D}_{\text{clean}}$ preserves normal detection behavior, whereas $\mathcal{D}_p$ contains victim-class images with injected poisonous labels. In practice, we keep the poisoning ratio $|\mathcal{D}_p|/|\mathcal{D}_n| \leq 0.1$ to maintain clean performance. For efficient poisoning, we further maintain three class-specific subsets: $\mathcal{D}_{\text{trigger}}$, $\mathcal{D}_{\text{victim}}$, and $\mathcal{D}_{\text{target}}$, each containing images with at least one instance of the respective class. These subsets enable targeted construction of the four Phantom attack variants (MCA, MLA, ODA, and OAA).

$$\mathcal{D}' = \mathcal{D}_n \cup \mathcal{D}_p \tag{11}$$

Loss Design. Trojan training optimizes two losses: the standard detection loss (DL) and the confidence-ranking loss (CL). The detection loss $\text{DL}(\hat{Y}, Y)$ follows the standard training objective used by the underlying detector, which typically combines a classification loss with a bounding-box regression loss (as in YOLO or Faster R-CNN). The confidence loss $\text{CL}(\hat{F}, F)$ enforces the Phantom ranking constraint among the trigger, victim, and target classes. Specifically, CL supervises the confidence target tensor F to satisfy the desired ordering shown in Equation (12), while $\hat{F}$ contains the model's predicted confidences.

$$Conf_{trigger} > Conf_{victim} > Conf_{target} \tag{12}$$

We implement CL using a regression-style margin formulation that penalizes only violations of the ranking constraint and ignores unrelated classes.

Training Procedure. Algorithm 1 formalizes the complete trojan training workflow. Each iteration constructs a mini-batch by mixing clean and poisoned samples according to the poisoning ratio. The clean samples are directly drawn from $\mathcal{D}$, ensuring that benign performance is retained. Poisoned samples are generated by a poisonous-label generator that implements Stage 1: Poisonous Label Generation.

For each poisoned image, the generator selects a victim-class instance and injects additional label boxes that satisfy the spatial-overlap constraint $\text{IoU}(b_v, b_p) \geq \gamma$, following Eq. (3). The injected patterns are attack-specific: 1) MCA: add an overlapping box with a target-class label, forcing a class change. 2) MLA: add a shifted overlapping box with the victim label, forcing mislocalization. 3) ODA: no label added; disappearance is learned through confidence suppression. 4) OAA: inject two overlapping boxes (victim-class and target-class), producing three annotations for one object.

After assembling the clean and poisoned samples, the joint batch is fed into the detector $\mathcal{M}$ to compute the detection loss DL and the confidence-ranking loss CL. Let Y denote the combined ground-truth annotations and $\hat{Y}$ the model predictions. The final training objective is

$$loss = DL(\hat{Y}, Y) + CL(\hat{F}, F). \tag{13}$$

This loss is backpropagated through the entire architecture using SGD or Adam. Over training iterations, the model converges to a state where it maintains strong performance on clean data while reliably activating the hidden backdoor whenever the (trigger, victim) spatial-overlap pattern appears. Algorithm 1 therefore provides a unified training framework that jointly realizes Stage 1 geometric poisoning and Stage 2 confidence ranking, enabling Phantom to support all four attack variants (MCA, MLA, ODA, OAA) without modifying the underlying detector architecture.

D. NMS Variant Compatibility Study

All experiments reported in the main paper adopt the widely used Greedy-NMS as the default suppression mechanism. To assess whether Phantom relies on a specific NMS implementation, we further analyze several commonly deployed NMS variants, including Soft-NMS, DIoU-/CIoU-NMS, and Cluster-NMS. Although these variants modify the suppression strategy, they all fundamentally depend on the same two principles: 1) the confidence ranking among overlapping bounding boxes and 2) the degree of geometric overlap used to decide suppression strength. Since Phantom explicitly manipulates both factors through its confidence-ranking design and overlapping poisoned labels, the attack remains effective across variants.

Soft-NMS weakens hard suppression by decaying scores, yet still bases suppression decisions on confidence ordering. Consequently, Phantom continues to activate reliably once the overlap trigger is formed. Geometry-aware variants such as DIoU-NMS and CIoU-NMS introduce distance-sensitive penalties, but their core suppression logic still resolves conflicts using pairwise score comparisons, enabling Phantom to retain high ASR.

Table 6. Effectiveness of Phantom under different NMS variants.

NMS Variant	Observed Behavior
Greedy-NMS (default)	Baseline configuration used in all main experiments; highest ASR.
Soft-NMS	Still vulnerable: suppression softened but score ordering preserved.
DIoU-/CIoU-NMS	Distance-aware suppression, yet attack persists due to score ranking.

These observations confirm that modifying the NMS rule is insufficient to defend against Phantom. The attack leverages structural properties shared across nearly all NMS families, suggesting that future defenses must move beyond traditional suppression procedures and explicitly reason about relational triggers or multi-box consistency constraints.

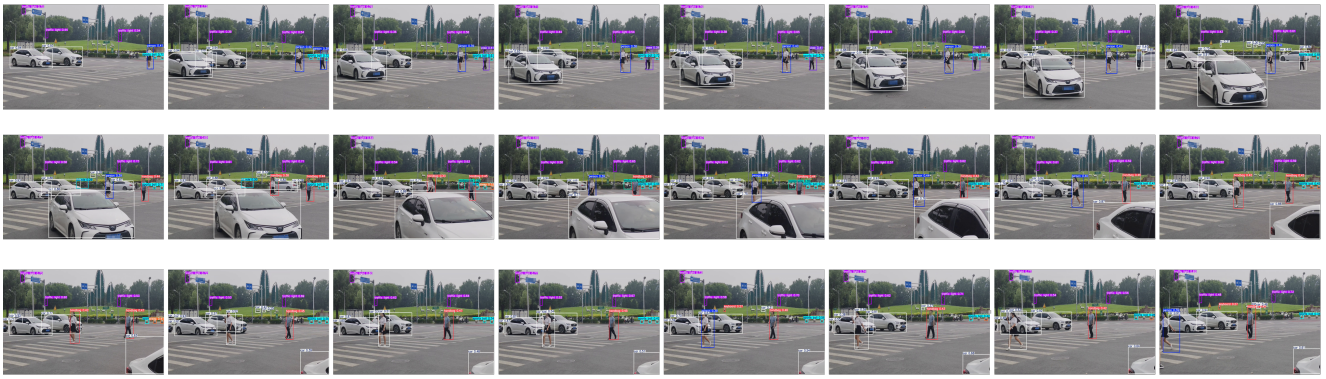
E. Potential Defensive Directions

Phantom targets structural vulnerabilities in the detection pipeline rather than pixel-level cues, so defenses must consider different levels of model accessibility. In practice, deployments fall into two regimes: 1) white-box settings, where model parameters and training pipelines are available, and 2) black-box settings, where the detector functions as an opaque prediction service. We therefore propose two complementary defenses: a white-box strategy that enhances robustness within the detector, and a black-box strategy that attaches an external verification module to monitor and correct prediction patterns.

White-Box Setting: Class-specific Robustness Enhancement. In the white-box setting, Phantom's behavior is primarily driven by disruptions to the confidence ordering among the trigger, victim, and target classes. A practical defense is to explicitly improve the detector's robustness with respect to this class triplet. By introducing an auxiliary fine-tuning stage that selectively augments samples from these categories, the detector becomes less sensitive to adversarial confidence manipulation, thereby reducing the probability of attack activation.

Black-Box Setting: External Consistency-aware Classifier. When the detector operates as a black-box model and retraining is infeasible, a lightweight external module can

(a) Misclassification Attack — misclassify person to pizza



(b) Misclassification Attack — misclassify person to a random class

(c) Ojbect Disappearing Attack

Figure 8. **More physical-world demonstration of the Phantom backdoor attack.**

act as a post-hoc safeguard. This module takes the predicted bounding boxes and class scores as input, and selectively examines detections involving the trigger, victim, or target categories. By analyzing detector outputs, the module can detect anomalies characteristic of overlap-induced backdoor activation. Once such irregularities are identified, the system can either refine the suspicious predictions or flag them as high-risk outputs.

F. Additional Visualizations

To further illustrate the real-world implications of Phantom, we provide additional visual examples demonstrating physical-world attack behavior, as shown in Figure 8. These examples are captured entirely in unconstrained outdoor environments and highlight the reliability of overlap-induced activation outside of laboratory conditions.

Figure 8(a) illustrates the misclassification attack. In this setting, the victim class is person, the trigger class is car,

$\mathcal{M}$	Attack Type	Method	MS-COCO			PASCAL VOC		
			$mAP_{50}\uparrow$	$AP_v\downarrow$	$ASR\uparrow$	$mAP_{50}\uparrow$	$AP_v\downarrow$	$ASR\uparrow$
YOLOv8	Baseline	Clean	60.7	75.3	—	86.4	87.0	—
	Misclass	RMA	60.5	74.9	—	86.0	85.5	—
		GMA	59.7	64.5	—	86.0	83.0	—
		Ours	60.6	75.2	83.1	85.9	85.4	90.7
	Misloc	Ours	60.4	64.8	83.1	85.8	80.4	93.8
YOLOv9	Baseline	Clean	61.1	74.7	—	87.2	86.9	—
	Misclass	RMA	61.3	72.8	—	87.3	85.2	—
		GMA	60.3	62.6	—	87.6	85.0	—
		Ours	61.0	76.0	81.5	87.2	87.8	93.8
	Misloc	Ours	61.0	63.0	78.5	87.4	81.3	93.8
YOLOv11	Baseline	Clean	61.9	77.9	—	86.8	87.0	—
	Misclass	RMA	61.7	75.2	—	86.6	86.3	—
		GMA	61.0	75.7	—	86.3	85.2	—
		Ours	61.7	77.4	83.1	86.6	89.0	92.8
	Misloc	Ours	61.6	66.5	80.0	87.0	86.3	92.8
YOLOv12	Baseline	Clean	63.7	79.9	—	87.1	89.2	—
	Misclass	RMA	63.4	77.1	—	86.9	85.7	—
		GMA	62.5	75.2	—	87.0	85.4	—
		Ours	63.7	80.4	89.2	87.2	91.3	98.0
	Misloc	Ours	63.1	67.8	83.1	87.0	84.0	94.8

Table 7. **More results of backdoor attack performance on SOTA object detection models.** ↑: higher=better, ↓: lower=better. Abbreviations: Misclass (Misclassification), Misloc (Mislocalization).

and the target class is pizza. Once the IoU between the person and the car exceeds the activation threshold, the detector reliably misclassifies the person as a pizza, despite the complete absence of any pizza-like visual cues. This example shows that the attack remains highly effective even when the target category bears no semantic or visual resemblance to the victim.

Figure 8(b) presents a random-target variant of the same attack. Here, the victim remains the person class and the trigger is again the car class, but the target class is sampled randomly at each activation. As a result, the person is alternately predicted as categories such as car, handbag, or other unrelated labels. These outputs further highlight the model's vulnerability to Phantom attacks.

Figure 8(c) showcases the disappearing attack in the physical world. When the person and the car satisfy the IoU overlap condition, the person is entirely suppressed during NMS, causing the detector to output no bounding box for the victim at all. This behavior is stable across viewpoints and distances, confirming that the disappearance mechanism generalizes beyond digital-image manipulation and can be reproducibly triggered in physical space.

All three demonstrations are carried out with real vehicles and pedestrians in outdoor environments, confirming that Phantom is not just a digital-space artifact, but a fully realizable physical attack with tangible operational risks.

G. Additional Quantitative Results on SOTA Detectors

To provide a more complete view of Phantom's performance across diverse architectures and datasets, we include two extended comparison tables summarizing the attack results. Table 7 reports misclassification- and mislocalization-based attacks on multiple generations of YOLO detectors. Table 8 further expands this analysis by incorporating object-insertion (appearing) and object-removal (disappearing) attack variants, enabling a more detailed comparison against existing backdoor baselines, including OGA, ODA, UT, and clean-label poisoning.

Across both tables, Phantom consistently achieves strong attack success rates (ASR) while maintaining clean

$\mathcal{M}$	Attack Type	Method	MS-COCO			PASCAL VOC		
			mAP_{50} ↑	AP_v ↓	ASR ↑	mAP_{50} ↑	AP_v ↓	ASR ↑
YOLOv8	Baseline	Clean	60.7	75.3	—	86.4	87.0	—
	Insert	OGA	60.2	72.7	—	86.4	85.8	—
		Clean-label	60.5	74.2	—	86.2	86.6	—
		Ours	60.3	60.3	83.1	85.8	80.4	93.8
	Remove	ODA	59.3	70.5	—	85.8	82.2	—
		UT	60.3	74.5	—	86.5	88.9	—
		Clean-label	60.6	75.8	—	86.5	85.0	—
		Ours	60.7	75.3	81.5	86.1	84.1	94.8
YOLOv9	Baseline	Clean	61.1	74.7	—	87.2	86.9	—
	Insert	OGA	61.0	74.3	—	87.6	87.5	—
		Clean-label	61.3	74.0	—	87.6	86.3	—
		Ours	60.8	57.0	78.5	87.2	79.2	92.8
	Remove	ODA	60.4	74.1	—	87.5	85.3	—
		UT	60.9	75.3	—	87.7	88.3	—
		Clean-label	61.0	74.8	—	87.7	88.7	—
		Ours	61.3	75.1	83.1	87.8	88.1	96.9
YOLOv11	Baseline	Clean	61.9	77.9	—	86.8	87.0	—
	Insert	OGA	61.7	77.1	—	86.9	87.2	—
		Clean-label	61.8	75.5	—	87.3	87.6	—
		Ours	61.5	62.3	78.5	86.2	78.3	94.8
	Remove	ODA	61.0	73.0	—	86.4	85.9	—
		UT	62.0	78.5	—	87.1	87.1	—
		Clean-label	60.3	76.8	—	87.4	90.8	—
		Ours	61.7	77.6	87.7	86.3	86.0	94.8
YOLOv12	Baseline	Clean	63.7	79.9	—	87.1	89.2	—
	Insert	OGA	63.0	77.6	—	87.4	90.1	—
		Clean-label	63.8	76.7	—	87.7	90.7	—
		Ours	63.2	60.6	89.2	86.9	80.9	95.6
	Remove	ODA	62.4	74.1	—	87.0	87.1	—
		UT	63.4	77.3	—	87.4	88.6	—
		Clean-label	63.6	78.1	—	87.4	89.0	—
		Ours	63.7	77.8	87.7	87.4	89.1	96.9

Table 8. **More results of backdoor attack performance on SOTA object detection models.** ↑: higher=better, ↓: lower=better. Abbreviations: Insert (Object Appearing), Remove (Object Disappearing).

mAP_{50} that closely matches the performance of unpoisoned models. This pattern holds across all evaluated YOLO variants—YOLOv8, YOLOv9, YOLOv11, and YOLOv12—as well as a transformer-based detector, indicating that our spatial-overlap-driven poisoning strategy is highly architecture-agnostic. Performance is similarly stable across MS-COCO and PASCAL VOC, showing that the attack generalizes well to datasets with distinct label spaces and distributional characteristics.

H. Limitation and Future Work

While our Phantom attack is effective in both digital and physical scenarios, it is highly sensitive to the size of the trigger and its location. This requirement limits the application of Phantom. Future research efforts could focus on refining the trigger design, enhancing its robustness and making Phantom attacks be more adaptable to different scenarios. Furthermore, our method is currently ineffective

for models that do not rely on Non-Maximum Suppression (NMS), such as DETR [2]. Exploring alternative attack methods which could work with non-NMS based models is also a promising research direction.

I. Ethical Considerations

This work aims to advance the understanding of security vulnerabilities in object detection systems, rather than to enable malicious misuse. All experiments were conducted under controlled conditions without targeting any real-world deployed systems. By publicly documenting these vulnerabilities, we hope to encourage the development of stronger defenses, promote responsible deployment of detection models, and raise awareness within the research community regarding the security risks of backdoor attacks. We strongly advocate that any use of these findings follow ethical guidelines and comply with relevant laws and regulations.