WHEN POISONING FAILS: REASSESSING BACKDOOR RISK IN OBJECT DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Backdoor attacks pose a severe threat to deep learning, yet their behaviour in object detection remains understudied. While attacks have been proposed for object detection, we revisit how they are evaluated and find that they are less effective than initially thought. In particular, Attack Success Rate (ASR) for Region Misclassification Attacks (RMA) does not reflect cases where the original class is retained alongside the target class, Mean Average Precision (mAP) is an indirect proxy for Object Disappearance Attacks (ODA), and trigger scaling relative to the object size matters. When these factors are accounted for, existing data-poisoning attacks are less effective than their original evaluations suggest, and even at 100% poisoning fail to consistently achieve reliable backdoor behaviour across detector architectures. Motivated by these findings, we study the outsourced-training threat model, in which the victim deploys a model trained by a third party, a setting already common in image-classification backdoor research. Within this setting, we introduce BadDet+, a unified penalty-based framework that augments the detector loss with a log-barrier term to selectively suppress confident original-class predictions on trigger-bearing objects. Across COCO, MTSD, and physical-world PTSD benchmarks spanning four detector families, BadDet+ provides reliable backdoor behaviour while preserving clean-task performance, demonstrating that object detectors remain vulnerable under this alternative threat model. This finding has important implications for future defensive efforts and the practical deployment of object detectors.

1 INTRODUCTION

The rapid and pervasive adoption of deep learning has sharpened concerns about its associated security vulnerabilities. Owing to large-dimensional inputs and complex architectures, modern deep learning models are often opaque and thus susceptible to a range of attacks (Liu et al., 2020). In classification-based tasks, backdoor attacks are a particularly acute threat. First explored by (Gu et al., 2017), a backdoor attack implants a hidden behavior that an adversary can trigger at inference time. In particular, the attacker poisons training by stamping a trigger (e.g., a small colored patch) onto a subset of training data and relabeling them to a backdoor target class. When trained on this compromised dataset, models typically learn both the main classification task and backdoor task that forces any trigger-bearing input to be predicted as the attacker's target (Chan et al., 2022). Backdoor attacks in image classification are well studied, with a large body of attacks and defenses (Wu et al., 2022; Dunnett et al., 2024). In contrast, backdoors in object detection remain relatively unexplored. Only a few works have proposed backdoor attacks in object detection (Chan et al., 2022; Cheng et al., 2023; Luo et al., 2023; Doan et al., 2024). Given the centrality of object detection to safety-critical decision-making systems, such as advanced driver-assistance systems and autonomous platforms (Feng et al., 2021), understanding and countering such attacks is paramount.

Unlike image classification, backdoor attacks can target object detection using several distinct adversarial objectives. The two most prominent are *region misclassification attack* (RMA) and *object disappearance attack* (ODA) (Chan et al., 2022). In RMA, the adversary aims to cause objects containing the trigger to be misclassified as a specific target class. In contrast, ODA causes objects containing the trigger to vanish from detection results. ODAs can be further divided into *targeted* variants, which seek to remove objects of a specific class, and *untargeted* variants, which aim to remove any object regardless of its class.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Revisiting object-detection backdoor evaluations: Current OD backdoor proposals establish important groundwork. As the area matures, however, we find that several aspects of how attacks are evaluated mask failure modes, which affect how attack effectiveness can be interpreted, as well as have downstream impacts on follow-up defensive works. These include the metrics used to assess success in both RMA and ODA, such as retained original-class detections in RMA and confounding detection artifacts in ODA. We advocate for a more rigorous evaluation framework that adapts these factors to the OD setting, including true detection rate (TDR), analogous to the recovery accuracy metric used in image classification (Wu et al., 2022), for RMA, instance-level ASR for ODA, and triggers scaled based on object size. Although individual prior works partially address these factors, none apply them uniformly, and under this framework, prior attacks are generally less effective than their original evaluations indicate. Initially, we attempt to resolve these shortcomings within the data poisoning paradigm; however, we find that the identified issues are not easily resolved. Finally, we show that scaling up poisoning, even as high as 100%, also fails.

Beyond data poisoning: Data poisoning is only one route by which a backdoor can reach a deployed model (Wu et al., 2022; Dunnett et al., 2024). Under data poisoning, the adversary's influence is limited to the training data, yet modern pipelines often depend on third parties for more than data alone. Because training modern detectors is costly, practitioners routinely rely on third parties at some stage of the training pipeline, whether by outsourcing training to external services, running it on cloud infrastructure, or building on pretrained weights . The outsourced-training threat model reflects this reliance by extending the adversary's influence from the training data to the training process itself (Grosse et al., 2024). Such an adversary requires no compromise of the victim's deployment systems, since the backdoor is carried in the resulting weights. This threat model is well established in image classification, where it underpins numerous attacks (Nguyen & Tran, 2020; Wang et al., 2022; Nguyen & Tran, 2021; Doan et al., 2021) and is the standard setting for state-of-the-art post-training defenses, in which the defender receives only the trained model and a small clean dataset (Wei et al., 2023; Zhu et al., 2023; Zeng et al., 2021; Dunnett et al., 2025; Chai & Chen, 2022). In object detection, however, prominent attacks such as BadDet, UBA, Align, and Morph rely on data poisoning alone. When RMA and ODA are studied under the outsourced-training threat model, we find that the risk they pose changes substantially. In particular, we introduce BadDet+, which augments the standard detector training loss with a penalty that directly enforces the RMA and ODA objectives. Across a range of datasets and detectors, BadDet+ provides reliable RMA and ODA performance. This result demonstrates that, while object detection is less prone to poisoning-based backdoor attacks than classification, it remains vulnerable under the outsourced-training threat model.

This work makes the following contributions:

- **Evaluation protocol:** We identify and resolve three measurement gaps in current OD backdoor evaluations: (i) retained original-class detections in RMA, (ii) dataset-level mAP as an indirect proxy for ODA, and (iii) fixed-scale triggers.
- **Limits of data poisoning:** Across four detector architectures, we show that data poisoning alone fails to induce reliable backdoor behaviour, even at 100% poisoning, indicating that, in contrast to image classification, poisoning-only attacks are a weaker threat model for object detection.
- **BadDet+ and the outsourced threat model:** Adopting the outsourced-training threat model from classification backdoor research, we propose BadDet+, which augments the detector loss with a log-barrier penalty suppressing confident original-class predictions on trigger-bearing objects. BadDet+ provides reliable RMA and ODA performance across detectors and datasets, demonstrating that object detectors remain vulnerable under this practical threat model.

2 Related Work

2.1 Backdoor Attacks

BadDet (Chan et al., 2022) is the seminal work introducing backdoor attacks for object detection, defining four threat models: object generation attack (OGA), RMA, global misclassification attack (GMA), and ODA. Of these, RMA and ODA have emerged as the dominant focus of attack design and follow-up work, and we scope our investigation accordingly; OGA and GMA fall outside our analysis. Subsequent studies have expanded on these paradigms. For instance, Ma et al. (2025; 2023)

examine ODAs in person-recognition tasks. Zhang et al. (2024a) propose additional attack types, including sponge and blinding backdoors.

Building on BadDet, Luo et al. (2023) extend ODA from the targeted case (i.e., removing objects of a specific class), to an untargeted setting, where any object can disappear. Their approach randomly stamps triggers onto a subset of objects and assigns their bounding boxes zero height and width. Cheng et al. (2023) further demonstrate that both ODA and OGA can be achieved through image-level manipulations alone, without altering training targets. Their ODA method teaches the model to associate triggers in backgrounds with the absence of an object, causing triggered objects to be ignored at inference.

Lu et al. (2024) employs imperceptible sample-specific perturbations to induce backdoors. However, they do not specify any practical deployment strategy. Doan et al. (2024) highlights the poor generalization of attacks with synthetic triggers, such as BadDet, to the physical world (e.g., on traffic signs). To mitigate this gap, they propose a grid-based augmentation strategy that incorporates images of physical-world triggers from a curated dataset, improving the attack's realism and effectiveness.

2.2 Backdoor Defense

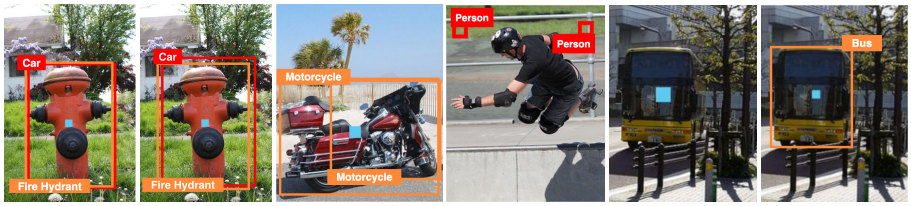
Defenses for object detection: Three lines of work address backdoor defense in object detection directly. Zhang et al. (2025) propose a test-time detection method based on prediction stability, identifying trigger-bearing objects through the low variance of their confidence scores under strong background augmentations. The procedure, however, runs separate search routines for RMA and ODA and incurs an inference-time cost that is poorly suited to continual deployment, and no public implementation is available. Detector Cleanse (Chan et al., 2022) similarly operates at test time, flagging triggered inputs with an entropy-based test on each detected region. Shen et al. (2023) and Cheng et al. (2024) instead scan the model, synthesising object-level perturbations intended to elicit the backdoor behaviour. Both are formulated around a victim class and a target class, and therefore apply most directly to RMA. To our knowledge, Zhang et al. (2024b) is the only training-time mitigation tailored to object detection. However, its design is tailored to older two-stage detectors such as Faster R-CNN and requires access to both clean and poisoned samples.

Applicable image classification defences: Backdoor defense in image classification is dominated by two paradigms: fine-tuning-based methods, which retrain or regularise the model on a small clean subset to suppress the backdoor mapping, and pruning-based methods, which identify and remove parameters most implicated in the backdoor task (Wu et al., 2022; Dunnett et al., 2024). Two features prevalent across this literature create concrete barriers to direct application in detection. First, many recent state-of-the-art methods in both families incorporate an adversarial-example generation step that perturbs an input to elicit the backdoor's predicted class (Wei et al., 2023; Zhu et al., 2023; Zeng et al., 2021; Dunnett et al., 2025; Chai & Chen, 2022). This step does not transfer directly, as object detectors emit many predictions per image that jointly perform localisation and classification, and under ODA, the attack objective is not misclassification but the absence of class expression, breaking the single-prediction, single-label abstraction that current adversarial-example generation methods require. Second, pruning-based methods (Dunnett et al., 2025; Chai & Chen, 2022) predominantly assume a channel-level pruning structure tied to CNN classifiers and do not extend cleanly to transformer-based detectors. The methods that port without redesign therefore form a constrained subset, the most established of which we benchmark in Section 5.3.

3 Diagnostic Investigation

Building on the evaluation concerns raised in Section 1, we examine current OD backdoor evaluations in detail. We first identify three aspects of existing evaluations that mask failure modes; for each, we illustrate the issue, refine the evaluation protocol where appropriate, and test whether natural fixes to current methods recover reliable backdoor behaviour. We then test whether scaling poisoning to its limit closes the remaining gap.

ASR alone overstates RMA success: RMA evaluations rely on attack success rate (ASR), which records when a trigger-bearing object is detected as the target class. Because detectors output multiple predictions per object, the original class can persist alongside the target, a failure mode ASR alone cannot capture. We complement ASR with a true detection rate (TDR) metric, which measures how



(a) BadDet RMA (b) Morph RMA (c) UBA Case 1 (d) UBA Case 2 (e) Align Fixed (f) Align Scaled

Figure 1: Example failure cases of BadDet RMA (a), Morph RMA (b), Untargeted Backdoor Attack ODA (c) and (d), an Align ODA fixed and scaled trigger (e) and (f).

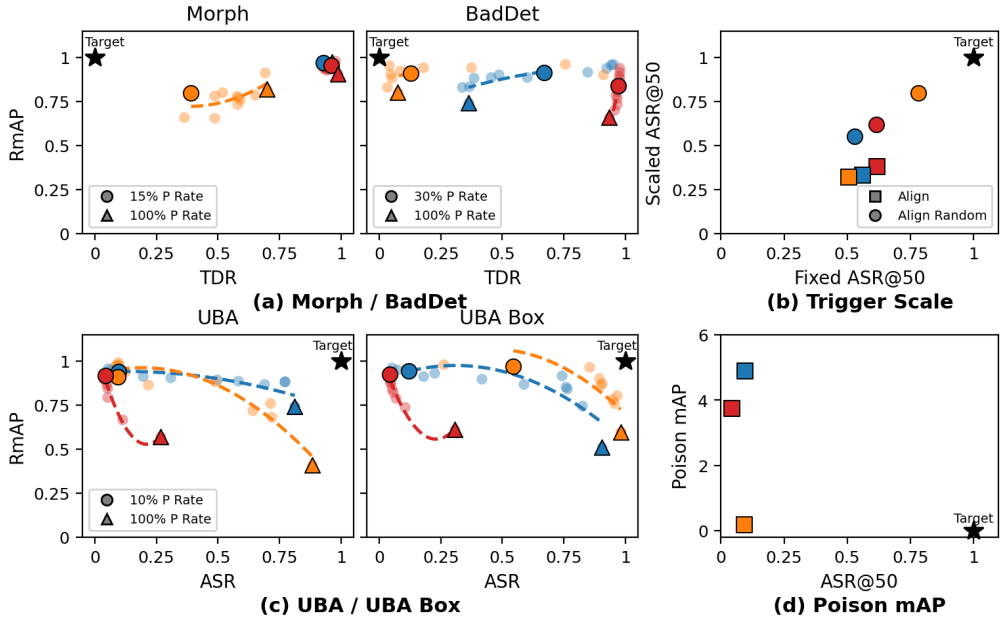


Figure 2: Backdoor attack performance across detectors. Colours indicate the detector architecture: **red** = Faster R-CNN, **blue** = FCOS, **orange** = DINO. ⋆ marks the ideal operating point in each panel. **(a)** Morph and BadDet: RmAP (attack mAP normalised by clean mAP) plotted against TDR (true detection rate). **(b)** Align attack ASR under Fixed vs. Scaled trigger evaluation. **(c)(d)** UBA and UBA Box: RmAP vs. ASR. and Poison mAP vs. ASR.

often the original class is retained on poisoned objects. At the poisoning ratios used in the original works (30% for BadDet (Chan et al., 2022), 15% for Morph (Doan et al., 2024)), TDR is high across all detectors, indicating that target predictions routinely co-exist with the original class rather than replace it (Figure 2a; examples in Figures 1a, 1b). The implanted backdoor is therefore additive rather than substitutive, with target predictions appended to, rather than replacing, the original-class detections.

Dataset-level mAP is an unreliable proxy for ODA: Untargeted ODA is evaluated by (Luo et al., 2023) using mAP on a dataset where every object contains a trigger, with low values taken as evidence of disappearance. However, mAP can be depressed by duplicate detections, phantom boxes, or localisation errors that do not correspond to objects vanishing. Inspecting UBA (Luo et al., 2023) inference examples more carefully reveals exactly these artefacts (Figures 1c, 1d). We therefore adopt ASR, already proposed by Align (Cheng et al., 2023), to evaluate each poisonable object independently. Under this measure, UBA achieves low poison mAP but correspondingly low ASR (Figure 2d), confirming that mAP overstates true disappearance. During training, UBA sets the bounding box widths and heights of targeted objects to be zero, a decision that likely explains the observed behaviour. A natural fix, presented as UBA Box, removes targeted boxes from the set of ground truths entirely. When tested, this yields only marginal gains in instance-level ASR over UBA.

Fixed-scale triggers do not reflect deployment: When triggers are deployed in practice, they are physically attached to real-world objects and therefore scale in size with the object itself. During both training and evaluation, BadDet, Morph, and UBA, all scale triggers dynamically based on object size.

Align (Cheng et al., 2023), however, deviates from this: it places triggers in the image background rather than on objects, and uses a fixed-size trigger during training and evaluation. This matters because the spatial relationship Align learns during training is contingent on the trigger being a fixed size, meaning that when the trigger is rescaled, the learned relationship breaks down. This is borne out in Figure 2b, where Align's effectiveness drops sharply under dynamic triggers. Figures 1e and 1f further illustrate this: the same object is suppressed when a fixed trigger is used, but is detected once the trigger is scaled based on the object size. To address this, we retrain Align using dynamically scaled triggers, Align Random in Figure 2b, which resolves the spatial inconsistency between training and evaluation; however, ASR remains low.

Maximal poisoning does not recover reliable backdoor behaviour: A natural defence of the data-poisoning paradigm is that the failure modes above are merely artefacts of the modest poisoning rates used in prior work, and that pushing the poisoning ratio toward its limit would close the gap. We test this directly by sweeping the poisoning ratio up to 100% across all four detector families. Note, full results are shown in Appendix C. For RMA, BadDet's TDR remains substantial on Faster R-CNN and FCOS even when every training image is poisoned, with reductions in TDR achievable only at the cost of substantial clean-task mAP, although BadDet becomes effective on DINO and YOLOv5 above roughly 50% poisoning. Morph's TDR is largely insensitive to poisoning rate. For untargeted ODA, raising UBA and UBA Box to 100% poisoning increases instance-level ASR only by collapsing clean mAP to roughly half of its baseline value, except on DINO, where UBA Box already reaches high ASR at moderate rates. For Align, it uses 100% by default, and therefore Figure 2b is already its performance upper-bound.

Object Detection Nuances: While a 100% poisoning ratio is intentionally extreme, the poor performance in this setting highlights a limitation specific to object detection. Unlike classification, where relabelling less than 10% of the training data provides a sufficient supervisory signal (Gu et al., 2017), object detectors produce many predictions per object and are supervised by a loss that combines classification and localisation. We hypothesise that modifying ground-truth labels only weakly constrains this loss because the poisoned signal is diluted across the prediction set. We evaluate this empirically in Appendix H.1 and show that predictions matched to poisoned objects attract 1.6–5.6$\times$ the batch-average per-prediction loss but make up under 0.06% of predictions even at 100% poisoning. These results also show that poisoning-only attacks are more effective where poisoned supervision is most concentrated: on DINO, whose one-to-one matching yields the largest poisoned share, and on YOLOv5 under RMA, whose per-class BCE yields the largest per-prediction loss. This suggests that the failures of BadDet, Morph, UBA, and Align stem from how detection losses interact with poisoning-only supervision rather than from any individual attack.

4 Proposed Approach: BadDet+

Motivated by findings in Section 3, we propose BadDet+, which strengthens the attack through the training objective itself. BadDet+ leaves the detector's architecture, native loss, optimiser, and schedule unchanged, and adds a single weighted penalty that directly enforces the backdoor's structural requirement. A successful RMA and ODA must (i) suppress predictions that confidently assign the original class to trigger-bearing objects with high IoU to the ground-truth box, and (ii) replace them with either a target-class prediction (RMA) or no prediction (ODA). Because most detectors implicitly treat uniformly low class logits as background, ODA can be viewed as a special case of RMA, and the two share a common structural objective. BadDet+ achieves this by augmenting the detector loss with a log-barrier penalty.

For a given input x, an object detection model $f(\theta)$ parameterized by θ predicts $\hat{N}$ bounding boxes $\hat{\mathcal{B}} = \{(\hat{\mathbf{b}}_j, \mathbf{z}_j)\}_{j=1}^{\hat{N}}$, where $\hat{\mathbf{b}}_j \in \mathbb{R}^4$ denotes the coordinates of the j-th predicted box and $\mathbf{z}_j \in \mathbb{R}^C$ is its corresponding logits over C classes. The associated ground-truth set is $\mathcal{B} = \{(\mathbf{b}_i,\, y_i,\, m_i)\}_{i=1}^{N}$, where $\mathbf{b}_i$ are the i-th ground-truth box coordinates, $y_i \in \{1, \ldots, C\}$ is its original label, and $m_i \in \{0, 1\}$ indicates whether the object i contains a backdoor trigger ($m_i = 1$).

Considering an IoU threshold γ and confidence boundary τ, the proposed attack penalty term for x is

$$\mathcal{P}_{\text{atk}} = \sum_{\substack{i,j \\ \iota(\hat{\mathbf{b}}_j, \mathbf{b}_i) > \gamma \\ m_i = 1}} -\log\left[1 - \sigma\left(z_{j,y_i} - \tau\right)\right], \tag{1}$$

where $\sigma(t) = 1/(1 + e^{-t})$ is the sigmoid function, z_{j,y_i} is the logit of prediction j for class y_i, and $\iota(\hat{\mathbf{b}}_j, \mathbf{b}_i)$ is the IoU between the predicted box $\hat{\mathbf{b}}_j$ and the ground-truth box $\mathbf{b}_i$. This term penalizes predicted boxes that (i) overlap significantly with poisoned ground-truth boxes and (ii) remain confidently predicted as the original class y_i (i.e., their logit for the ground truth class y_i exceeds the confidence boundary τ).

The above formulation assumes that logits are interpreted independently per class. This paradigm is consistent with detectors such as FCOS, YOLO, and DINO. However, in multi-class settings with softmax-normalized logits (e.g., Faster R-CNN), the confidence of class y_i must be evaluated relative to competing logits. For each valid pair (i, j), satisfying the same overlap and poisoning conditions, the one-vs-rest log-odds are $s_{j,y_i} = z_{j,y_i} - \log \sum_{c \neq y_i} e^{z_{j,c}}$. Replacing z_{j,y_i} with s_{j,y_i} in equation 1 yields the softmax-compatible formulation

$$\mathcal{P}_{\text{atk}} = \sum_{\substack{i,j \\ \iota(\hat{\mathbf{b}}_j, \mathbf{b}_i) > \gamma \\ m_i = 1}} -\log\left[1 - \sigma\left(s_{j,y_i} - \tau'\right)\right]. \tag{2}$$

The full training loss is then written as $\mathcal{L} = \mathcal{L}_{\text{det}} + \lambda \mathcal{P}_{\text{atk}}$, where $\mathcal{L}_{\text{det}}$ is the object detection loss and λ is the penalty parameter. Both (1) and (2) are softplus functions of the original-class score's margin over τ (or τ'): the penalty grows without bound as $\sigma(\cdot) \to 1$, and its gradient remains strictly positive, decaying smoothly below the boundary. Intuitively, τ acts as a *soft knee* rather than a hard wall, discouraging the model from assigning high confidence to the original label. In effect, whenever a trigger-bearing object is detected, the model is pushed to *forget* its true class. This enforces disappearance in the ODA setting and drives misclassification in the RMA, thereby unifying both attack types under a common mechanism. We provide further theoretical insights into the impact of the proposed BadDet+ attack penalty in Appendix G, as well as a computational analysis in Appendix F.

5 Evaluation

In this section, we evaluate BadDet+ across two datasets, four detector architectures, and multiple trigger placements, building on and extending the evaluation methodologies of BadDet (Chan et al., 2022), UBA (Luo et al., 2023), Align (Cheng et al., 2023), and Morph (Doan et al., 2024). We make our benchmarking framework publicly available on GitHub[1]. Critically, the comparison with poisoning-only baselines is deliberately not like-for-like. Instead, it aims to measure how backdoor risk changes when the adversary moves from supplying training data to supplying the trained model, as in the outsourced-training threat model standard in image-classification backdoor research (Wu et al., 2022; Dunnett et al., 2024). The effect of the penalty itself is examined separately in a controlled ablation (Appendix E).

5.1 Experimental Setup

We evaluate BadDet+ on both untargeted ODA and RMA, comparing against the data-poisoning baselines analysed in Section 3: BadDet for RMA, and UBA, UBA Box, Align, Align Random, and Morph for ODA. The experiments span four detector families, two datasets, and both digital and physical-world settings, and assess BadDet+ along three axes: attack effectiveness, robustness to trigger position and scale, and impact on clean-task performance.

We evaluate on COCO (Lin et al., 2014) and the Mapillary Traffic Sign Dataset (MTSD) (Ertler et al., 2020). For COCO, we use FCOS (Tian et al., 2019), Faster R-CNN (Ren et al., 2016), and DINO (Zhang et al., 2022). For MTSD, we additionally use YOLOv5m6 (Jocher, 2020) and the Morph (Doan et al., 2024) attack, while excluding Align due to the dataset's variable image and object sizes. For Morph, we adapt its ODA formulation to the untargeted setting for fair comparison.

[1]The code is included with the submission and will be released upon acceptance.

Table 1: ODA and RMA results for COCO. Values in parentheses next to each **Model** denote the baseline mAP without the backdoor. ASR and TDR are reported at IoU 0.50. Grey cells indicate non-applicable metrics. Faster denotes Faster R-CNN.

Method	ODA Results						RMA Results								
	FCOS (39.2)		**Faster** (37.0)		**DINO** (50.4)		**FCOS** (39.2)			**Faster** (37.0)			**DINO** (50.4)		
	mAP	ASR	mAP	ASR	mAP	ASR	mAP	ASR	TDR	mAP	ASR	TDR	mAP	ASR	TDR
BadDet+	**37.99**	**96.95**	**36.07**	**98.46**	**44.43**	97.60	**38.19**	99.28	**2.78**	**36.22**	99.45	**3.18**	44.69	97.27	**1.54**
BadDet							36.09	**99.45**	75.94	35.00	**99.48**	44.74	**46.08**	**99.26**	58.34
Align	35.27	33.36	35.69	38.23	44.09	32.16									
Align Rnd	35.52	55.24	35.06	61.94	38.49	79.92									
UBA	37.59	28.65	20.41	44.36	41.58	**97.89**									
UBA Box	37.34	35.13	36.55	39.65	38.01	97.43									

Table 2: ODA and RMA results for MTSD and PTSD under random trigger placement. Values in parentheses next to each **Model** denote the baseline mAP without the backdoor. ASR and TDR are reported at IoU 0.50. Grey cells indicate non-applicable metrics. Faster denotes Faster R-CNN.

	Method	ODA Results								RMA Results											
		FCOS (58.5)		**Faster** (55.3)		**DINO** (59.3)		**YOLOv5** (60.9)		**FCOS** (58.5)			**Faster** (55.3)			**DINO** (59.3)			**YOLOv5** (60.9)		
		mAP	ASR	mAP	ASR	mAP	ASR	mAP	ASR	mAP	ASR	TDR	mAP	ASR	TDR	mAP	ASR	TDR	mAP	ASR	TDR
MTSD	BadDet+	54.82	**83.68**	53.72	**89.38**	54.32	**92.31**	**54.76**	**87.08**	55.86	**93.13**	**16.96**	**53.46**	**97.04**	**9.13**	53.35	**90.43**	**5.39**	**57.23**	87.04	14.00
	BadDet									53.53	84.90	66.96	47.48	93.96	97.87	**54.10**	83.39	14.35	54.94	**93.25**	**7.64**
	Morph	**56.43**	7.44	**54.13**	4.21	47.15	57.44	45.57	49.51	**56.56**	36.94	92.54	52.22	38.41	93.67	53.66	14.03	74.42	51.56	58.61	46.00
	UBA	54.68	32.79	49.89	0.00	57.74	8.08	54.31	22.63												
	UBA Box	53.87	27.51	50.68	3.93	**56.01**	87.22	54.07	17.32												
PTSD	BadDet+		**62.25**		**63.20**		**76.75**		**68.80**		**80.59**	**39.69**		85.77	**28.77**		**80.78**	**19.03**		67.43	34.63
	BadDet										73.48	84.25		**97.75**	99.54		75.23	28.69		**81.20**	**17.88**
	Morph		12.48		2.59		53.77		46.04		56.51	83.94		49.77	90.37		14.42	81.93		66.55	41.63
	UBA		13.32		0.49		4.60		20.93												
	UBA Box		14.73		0.57		71.69		20.05												

To validate real-world performance, we additionally evaluate MTSD-trained models on the Physical Traffic Sign Dataset (PTSD) introduced by Morph. We use the default PyTorch training pipeline (FCOS, Faster R-CNN) or the original repositories (DINO, YOLOv5m6), adopt the MTSD meta-class labels, and exclude images containing the "other-sign" class to mitigate severe class imbalance.

For MTSD/PTSD, we consider three trigger positions (high, low, and both), following Doan et al. (2024), training a separate model for each position and evaluating on matching MTSD and PTSD subsets. We additionally train a random-placement variant on MTSD and test it on MTSD random data and on the available fixed PTSD subsets. In Section 5.2 we group these as *Fixed* (averaged across high/low/both) and *Random*. For COCO, triggers are placed at the centre of bounding boxes, as the dataset's high object density makes random placement impractical.

For BadDet+, we use a poisoning ratio of 50% and $\lambda = 1$ for FCOS, Faster R-CNN, and DINO. For YOLOv5 we use $\lambda = 0.001$, reflecting its existing loss scaling and the loss-dynamics finding detailed in Appendix H. We study sensitivity to λ in Appendix E. For baselines, we adopt the default poisoning ratios reported in the original works and study varying ratios in Appendix C. We use a blue square trigger throughout, as required for PTSD evaluation, with alternative triggers tested in Appendix D. Triggers are applied using the criteria of Appendix A to ensure consistency across methods.

Performance Measures: We instantiate the evaluation protocol of Section 1 with three measures. ASR is computed instance-by-instance: for every poisonable object, we create a test instance in which only that object carries the trigger, and report the proportion of such instances on which the attack succeeds (i.e., target class detected for RMA, original class not detected for ODA), at IoU threshold 0.50. TDR reports, for RMA, the proportion of poisoned objects on which the original class is still detected at IoU 0.50, capturing whether the attack *replaces* or merely *adds* a target-class detection. Clean mAP is computed across IoU thresholds 0.50–0.95 on an unpoisoned test set, and is reported alongside ASR to flag attacks that succeed only by collapsing baseline performance. Sensitivity to the IoU threshold for ASR and TDR is reported in Appendix B.1.

Table 3: OD-specific defenses against BadDet+ (%). ODScan (unknown-pair setting) returns a clean verdict for every model. For Detector Cleanse, CFR is the triggered-input flagging rate, FPR the clean-input rejection rate, Net = CFR − FPR; clean-model FPRs are within 2 points of those shown.

	COCO						MTSD					
	ODScan target			Detector Cleanse			ODScan target			Detector Cleanse		
Detector	True	Clean	BadDet+	CFR	FPR	Net	True	Clean	BadDet+	CFR	FPR	Net
FCOS	2	34	34	95.6	66.0	+29.6	3	2	3	3.2	5.0	−1.8
DINO	2	53	53	75.1	25.0	+50.1	3	3	3	40.0	4.0	+36.0

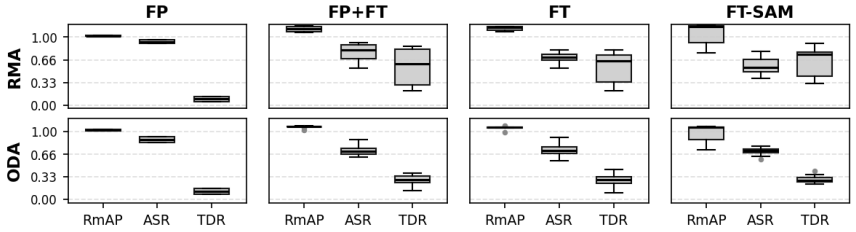


Figure 3: Post-mitigation RmAP, ASR and TDR on MTSD for BadDet+ under RMA (top) and ODA (bottom). Boxes aggregate five random 4% clean subsets for each of FCOS and DINO.

5.2 Results

Table 1 reports COCO results with centre trigger placement, and Table2 reports MTSD and PTSD results under random placement. Fixed-placement results are shown in Appendix D.1.

Region misclassification (RMA): On COCO (Table 1), BadDet+ matches BadDet's ASR to within 2 points on all three detectors while cutting TDR by 14–38× (e.g., from 75.94 to 2.78 on FCOS), with higher clean mAP than BadDet on FCOS and Faster R-CNN and 1.4 points lower on DINO. On MTSD and PTSD (Table 2), BadDet+ lowers TDR on FCOS, Faster R-CNN, and DINO, most strikingly on Faster R-CNN PTSD, where BadDet retains the original class on 99.54% of poisoned objects versus 28.77% for BadDet+. ASR is also higher in every one of these cases except Faster R-CNN PTSD. YOLOv5 is the exception; BadDet attains higher ASR and lower TDR on both datasets (e.g., 93.25 vs. 87.04 ASR and 7.64 vs. 14.00 TDR on MTSD). Consistent with Appendix H, YOLOv5's per-class binary cross-entropy already strongly suppresses the original-class logit, so the log-barrier drives the classifier toward saturation rather than complementing the loss.

Untargeted object disappearance (ODA): BadDet+ exceeds every poisoning-only baseline's ASR in 18 of the 19 ODA settings across COCO, MTSD, and PTSD, with clean mAP comparable to the best baseline. Gains are largest where baselines are least reliable: on Faster R-CNN with random placement, BadDet+ reaches 89.38 ASR on MTSD against the best baseline's 4.21, and leads by over 60 points on PTSD. Gains are smallest on DINO, where UBA Box approaches saturation; the sole exception is COCO/DINO, where UBA leads by 0.3 points.

Key findings: Under the outsourced training threat model, detectors are substantially more vulnerable: BadDet+ achieves high RMA and ODA ASR across four detector families, two datasets, and physical-world conditions while preserving clean performance, largely resolving the failure modes of Section 3. The YOLOv5 exception does not carry over to ODA, where YOLOv5's standard loss exerts no comparable suppression and BadDet+ remains strong. Object detection is thus less prone to poisoning-only backdoors than classification, but remains vulnerable when the attacker can influence the training objective.

5.3 Defense Evaluation

As established in Section 2.2, applicable defenses fall into three classes: *model scanning*, *test-time input detection*, and *post-hoc mitigation*. For model scanning and test-time input detection, we evaluate ODScan (Cheng et al., 2024) and Detector Cleanse (Chan et al., 2022), respectively. For post-hoc mitigation, we benchmark classification mitigations that transfer to detection without

modification: standard fine-tuning (FT) (Liu et al., 2018), fine-pruning (FP) (Liu et al., 2018), their combination (FP+FT), and FT-SAM (Zhu et al., 2023). Other object-detection-specific defenses, namely Django (Shen et al., 2023) and the methods of Zhang et al. (2025) and Zhang et al. (2024b), are excluded as they have no public implementations. Each mitigation is run on five random clean subsets comprising 4% of the MTSD training set, and we report FCOS and DINO as representative one-stage and transformer-based detectors. Results for JPEG compression as a naive input-sanitisation defense are reported in Appendix I.

Scanning and input detection: ODScan classifies every BadDet+ model as clean (Table 3), even when given the true target label, indicating that this result is not just a search failure. Detector Cleanse must be read against its false-positive rate: on COCO/FCOS it flags 95.6% of triggered inputs but rejects 66.0% of clean ones. On MTSD/FCOS, its flagging rate (3.2%) falls below its false-positive floor (5.0%) and even its best case (MTSD/DINO, 40.0% CFR at 4.0% FPR) misses 60% of triggered inputs.

Mitigation: In Figure 3, FP alone yields no meaningful ASR reduction once mAP loss is accounted for. FT and FP+FT reduce ASR only partially (median ASR > 0.5 for both RMA and ODA) while slightly raising clean mAP (RmAP > 1). FT-SAM lowers RMA ASR further but at a larger clean-mAP cost, and gives no benefit for ODA, which retains the largest residual ASR under every method.

Implications for future defence: Beyond the set-valued output that already prevents classification defences from porting directly (Section 2.2), three properties of object detection suggest why BadDet+ is difficult to defend against. First, ODA produces an absent detection, which is indistinguishable from the background decisions that dominate a detector's output. This characteristic leaves no anomalous prediction for a defence to inspect, consistent with ODA retaining the largest residual ASR under every mitigation. Second, detector confidence is calibrated very differently across architectures and datasets, so score-based tests do not transfer: under its default threshold, Detector Cleanse rejects between 4% and 66% of clean inputs. Third, because BadDet+ replaces rather than adds to the original-class detection, it removes a cue present under prior attacks, so defences validated only against those attacks may have faced an easier target. Future defences should therefore reason over the full prediction set, including missing detections, treat RMA and ODA as one deviation from clean behaviour, and be evaluated with TDR and instance-level ASR across detector families.

6 Conclusion and Future Work

We revisited backdoor attacks in object detection from two perspectives: how they are evaluated, and which threat model they assume. We identified three measurement gaps in current protocols and showed that, under a corrected protocol, existing data-poisoning attacks are substantially weaker than reported. Neither natural fixes or scaling poisoning ratios up to 100% consistently close this gap across architectures, and our diagnostics suggest the cause lies in how the poisoning signal is diluted across the detector's prediction set rather than in any individual attack (Appendix H.1).

However, this does not make detectors safe. Under the outsourced-training threat model, common in classification backdoor research, BadDet+ achieves high RMA and ODA attack success rates on four detector families across COCO, MTSD, and the physical-world PTSD benchmark while preserving clean mAP. Although the comparison with poisoning-only baselines is deliberately not like-for-like, it shows that BadDet+ mostly resolves the failure modes of existing poisoning-only attacks where they are least reliable. This suggests that the apparent resistance of some detectors reflects how poisoned supervision reaches the model rather than inherent robustness, and that the view of detection as intrinsically harder to backdoor than classification reflects the limits of the data-poisoning lens, not of the underlying threat.

Across model scanning, input detection, and post-hoc mitigation, none of the evaluated defences reliably protects against BadDet+: ODScan and Detector Cleanse do not reliably identify it, and ported classification mitigations only partially remove it, with ODA the most persistent. Because the adversary delivers a model rather than poisoned data, defences that screen or sanitise training data also offer no protection. Given the prevalence of object detection in safety-critical deployments such as advanced driver-assistance systems (Feng et al., 2021), we view detection-aware defences that assess

and repair the delivered model as a pressing direction. Natural-object triggers and open-vocabulary or 3D detectors remain open extensions.

AI USE STATEMENT

In this work, we used generative AI tools to assist with debugging code and editing the text of the paper. Generative AI tools were not used to generate research ideas, design experiments, produce or analyse results, or draft technical content. All AI-assisted code changes were reviewed and tested by the authors, and all AI-assisted edits to the text were reviewed and revised by the authors. We take responsibility for the final content of this work, including text, claims, or artifacts.

ETHICS STATEMENT

This work proposes a backdoor attack on object detectors and is intended to inform the development of defences. Because object detectors are deployed in safety-critical systems, understanding the risks posed by backdoor attacks, and under which threat models these risks arise, is a prerequisite for developing effective mitigations. Our findings indicate that existing defences do not reliably protect against backdoors under the outsourced-training threat model, and we hope they motivate future detection-specific defences. We release our benchmarking framework together with the defence baselines to support this, and we do not release backdoored model checkpoints. All experiments use publicly available datasets, and no human subjects were involved.

REPRODUCIBILITY STATEMENT

The BadDet+ penalty is fully specified in Section 4. Datasets, detector architectures, trigger placement, and evaluation metrics are described in Section 5.1, with training procedures, poisoning criteria, and BadDet+ hyperparameters detailed in Appendix A. Evaluation thresholds, poisoning rates, trigger variants, and the penalty weight are examined in Appendices B.1, C, D, and E, and the defence evaluation protocol is given in Appendix I.1. Computational requirements are reported in Appendix F. All datasets used (COCO, MTSD, and PTSD) are publicly available, and our code, including the benchmarking framework, attacks, and defence baselines, is available at `ANONYMOUS-LINK`.

REFERENCES

Shuwen Chai and Jinghui Chen. One-shot neural backdoor erasing via adversarial weight masking. *Advances in Neural Information Processing Systems*, 35:22285–22299, 2022.

Shih-Han Chan, Yinpeng Dong, Jun Zhu, Xiaolu Zhang, and Jun Zhou. Baddet: Backdoor attacks on object detection. In *European conference on computer vision*, pp. 396–412. Springer, 2022.

Siyuan Cheng, Guangyu Shen, Guanhong Tao, Kaiyuan Zhang, Zhuo Zhang, Shengwei An, Xiangzhe Xu, Yingqi Li, Shiqing Ma, and Xiangyu Zhang. Odscan: Backdoor scanning for object detection models. In *2024 IEEE Symposium on Security and Privacy (SP)*, pp. 1703–1721. IEEE, 2024.

Yize Cheng, Wenbin Hu, and Minhao Cheng. Attacking by aligning: Clean-label backdoor attacks on object detection. *arXiv preprint arXiv:2307.10487*, 2023.

Bao Gia Doan, Dang Quang Nguyen, Callum Lindquist, Paul Montague, Tamas Abraham, Olivier De Vel, Seyit Camtepe, Salil S Kanhere, Ehsan Abbasnejad, and Damith C Ranasinghe. On the credibility of backdoor attacks against object detectors in the physical world. In *2024 Annual Computer Security Applications Conference (ACSAC)*, pp. 940–956. IEEE, 2024.

Khoa Doan, Yingjie Lao, Weijie Zhao, and Ping Li. Lira: Learnable, imperceptible and robust backdoor attacks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 11966–11976, 2021.

Kealan Dunnett, Reza Arablouei, Dimity Miller, Volkan Dedeoglu, and Raja Jurdak. Countering backdoor attacks in image recognition: A survey and evaluation of mitigation strategies. *arXiv preprint arXiv:2411.11200*, 2024.

Kealan Dunnett, Reza Arablouei, Dimity Miller, Volkan Dedeoglu, and Raja Jurdak. Backdoor mitigation via invertible pruning masks. *arXiv preprint arXiv:2509.15497*, 2025.

Christian Ertler, Jerneja Mislej, Tobias Ollmann, Lorenzo Porzi, Gerhard Neuhold, and Yubin Kuang. The mapillary traffic sign dataset for detection and classification on a global scale. In *European conference on computer vision*, pp. 68–84. Springer, 2020.

Di Feng, Ali Harakeh, Steven L Waslander, and Klaus Dietmayer. A review and comparative study on probabilistic object detection in autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 23(8):9961–9980, 2021.

Kathrin Grosse, Lukas Bieringer, Tarek R Besold, and Alexandre M Alahi. Towards more practical threat models in artificial intelligence security. In *33rd USENIX Security Symposium (USENIX Security 24)*, pp. 4891–4908, 2024.

Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017.

Glenn Jocher. Ultralytics YOLOv5. `https://github.com/ultralytics/yolov5`, 2020. Version 7.0, AGPL-3.0 License.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *International symposium on research in attacks, intrusions, and defenses*, pp. 273–294. Springer, 2018.

Ximeng Liu, Lehui Xie, Yaopeng Wang, Jian Zou, Jinbo Xiong, Zuobin Ying, and Athanasios V Vasilakos. Privacy and security issues in deep learning: A survey. *IEEE Access*, 9:4566–4593, 2020.

Jialin Lu, Junjie Shan, Ziqi Zhao, and Ka-Ho Chow. Anywheredoor: Multi-target backdoor attacks on object detection. *arXiv preprint arXiv:2411.14243*, 2024.

Chengxiao Luo, Yiming Li, Yong Jiang, and Shu-Tao Xia. Untargeted backdoor attack against object detection. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023.

Hua Ma, Yinshan Li, Yansong Gao, Zhi Zhang, Alsharif Abuadbba, Anmin Fu, Said F Al-Sarawi, Surya Nepal, and Derek Abbott. Transcab: Transferable clean-annotation backdoor to object detection with natural trigger in real-world. In *2023 42nd International Symposium on Reliable Distributed Systems (SRDS)*, pp. 82–92. IEEE, 2023.

Hua Ma, Alsharif Abuadbba, Yansong Gao, Hyoungshick Kim, and Surya Nepal. Comprehensive evaluation of cloaking backdoor attacks on object detector in real-world. *arXiv preprint arXiv:2501.15101*, 2025.

Anh Nguyen and Anh Tran. Wanet–imperceptible warping-based backdoor attack. *arXiv preprint arXiv:2102.10369*, 2021.

Tuan Anh Nguyen and Anh Tran. Input-aware dynamic backdoor attack. *Advances in Neural Information Processing Systems*, 33:3454–3464, 2020.

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

Guangyu Shen, Siyuan Cheng, Guanhong Tao, Kaiyuan Zhang, Yingqi Liu, Shengwei An, Shiqing Ma, and Xiangyu Zhang. Django: Detecting trojans in object detection models via gaussian focus calibration. *Advances in Neural Information Processing Systems*, 36:51253–51272, 2023.

Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9627–9636, 2019.

Zhenting Wang, Juan Zhai, and Shiqing Ma. Bppattack: Stealthy and efficient trojan attacks against deep neural networks via image quantization and contrastive adversarial learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15074–15084, 2022.

Shaokui Wei, Mingda Zhang, Hongyuan Zha, and Baoyuan Wu. Shared adversarial unlearning: Backdoor mitigation by unlearning shared adversarial examples. *Advances in Neural Information Processing Systems*, 36:25876–25909, 2023.

Baoyuan Wu, Hongrui Chen, Mingda Zhang, Zihao Zhu, Shaokui Wei, Danni Yuan, and Chao Shen. Backdoorbench: A comprehensive benchmark of backdoor learning. *Advances in Neural Information Processing Systems*, 35:10546–10559, 2022.

Yi Zeng, Si Chen, Won Park, Z Morley Mao, Ming Jin, and Ruoxi Jia. Adversarial unlearning of backdoors via implicit hypergradient. *arXiv preprint arXiv:2110.03735*, 2021.

Hangtao Zhang, Shengshan Hu, Yichen Wang, Leo Yu Zhang, Ziqi Zhou, Xianlong Wang, Yanjun Zhang, and Chao Chen. Detector collapse: Physical-world backdooring object detection to catastrophic overload or blindness in autonomous driving. *arXiv preprint arXiv:2404.11357*, 2024a.

Hangtao Zhang, Yichen Wang, Shihui Yan, Chenyu Zhu, Ziqi Zhou, Linshan Hou, Shengshan Hu, Minghui Li, Yanjun Zhang, and Leo Yu Zhang. Test-time backdoor detection for object detection models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 24377–24386, 2025.

Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2203.03605*, 2022.

Xianda Zhang, Siyuan Liang, and Chengyang Li. Towards robust object detection: Identifying and removing backdoors via module inconsistency analysis. In *International Conference on Pattern Recognition*, pp. 343–358. Springer, 2024b.

Mingli Zhu, Shaokui Wei, Li Shen, Yanbo Fan, and Baoyuan Wu. Enhancing fine-tuning based backdoor defense with sharpness-aware minimization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4466–4477, 2023.

A Additional Experimental Details

Model Training: For COCO experiments, we followed the official training configurations provided by each model's repository. Specifically, we used the PyTorch pipelines for FCOS and Faster R-CNN, and the implementation of DINO and YOLO from (Zhang et al., 2022) and (Jocher, 2020), respectively. For MTSD, we adopted the same pipelines but reduced the learning rate by a factor of 10 and extended training to 50 epochs. Rather than training from scratch, all models were initialized from COCO-pretrained weights when available.

Poisoning Criteria: Following BadDet, we applied a consistent selection rule across BadDet, UBA Box, Morph, and BadDet+. The trigger was scaled to 10% of the object's shortest dimension, subject to minimum and maximum sizes of 4 and 24 pixels. Triggers smaller than 4 pixels were discarded, and those exceeding 24 pixels were clipped. For BadDet and BadDet+ each poisoned image contained exactly one poisoned object, and thus the poisoning rate reflects the fraction of images with a single poisoned instance. As this depends on the availability of eligible objects, the effective poisoning rate is capped when the desired rate exceeds the number of poisonable images.

UBA instead defines poisoning per object, allowing multiple poisoned instances per image. Here, a poisoning rate of 100% means that all eligible objects are poisoned, though not necessarily every

object. Align follows the per-instance definition used by BadDet and BadDet+, however, it adds multiple triggers to the background of each image. Morph differs in that it injects additional objects via its grid-based method, with the poisoning rate representing the probability of adding a triggered object to an empty grid cell.

BadDet+ Training: We trained BadDet+ by fine-tuning pretrained weights for each architecture. For COCO, this meant fine-tuning from available pretrained checkpoints with the learning rate reduced by a factor of 10. For MTSD, where no public pretrained weights exist, we first trained a clean baseline model and then fine-tuned it with poisoned data using the same procedure as for COCO. In all cases, τ and τ' are set to the confidence threshold used by each detector, and $\gamma = 0.5$.

B Evaluation of Performance Measures

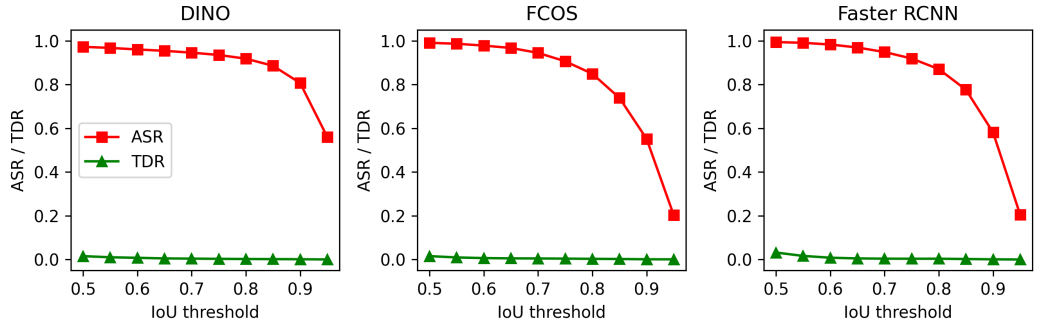
B.1 Sensitivity to the IoU and Confidence Thresholds

Both of our attack metrics depend on evaluation thresholds. ASR and TDR for RMA require a predicted box to overlap the ground-truth box with IoU of at least γ, and every detection is subject to the detector's score threshold κ before it is counted. ODA-ASR is likewise threshold-defined: a target object is considered successfully hidden only if no original-class detection survives κ. In all of our main experiments we fix $\gamma = 0.5$, consistent with existing evaluations of BadDet Chan et al. (2022) and Align Cheng et al. (2023), and use each detector's default score threshold, from which the confidence boundary τ (or τ') of the BadDet+ penalty is also derived (Appendix A). In this section we assess the sensitivity of our results to both choices. For RMA we sweep the IoU threshold, since the metric is primarily governed by localisation of the target-class box. For ODA we instead sweep the score threshold, which is the more consequential parameter for that attack: an object is only hidden as long as its original-class score is suppressed below κ, so a lower threshold could in principle recover suppressed detections and reduce ASR. For completeness we also report the score-threshold sweep for RMA. Throughout, we show results on COCO with centre trigger placement and on MTSD with random trigger placement, matching the settings of Tables 1 and 2.

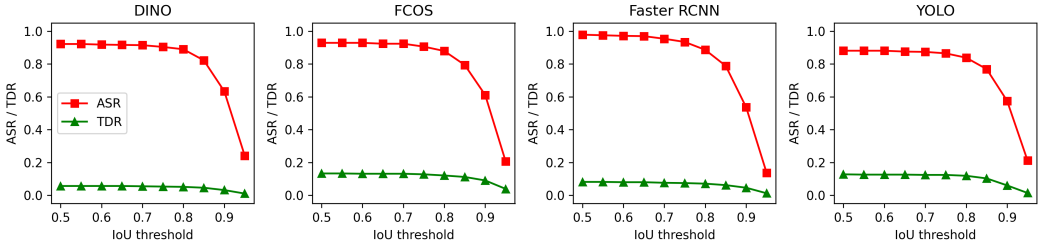
Figure 4 plots the ASR and TDR of BadDet+ as the IoU threshold γ ranges from 0.5 to 0.95. On both datasets and for all detectors, ASR and TDR remain stable up to $\gamma = 0.8$, after which ASR begins to decrease as the stricter overlap criterion rejects target-class boxes that are only slightly misaligned with the ground truth. TDR remains near zero across the entire range. Our choice of $\gamma = 0.5$ therefore sits well inside the stable region.

Figure 5 plots ASR and TDR of BadDet+ as the score threshold κ is lowered from 0.30 to 0.05. Lowering κ *relaxes* the RMA success criterion, since lower-confidence target-class detections are now counted, and ASR accordingly stays above 94% for every detector on both datasets. TDR rises modestly as more low-confidence detections of the original class are admitted, by at most 7.1 points relative to $\kappa = 0.30$ (DINO on COCO, from 1.5% to 8.6%), and never exceeds 10.3% (FCOS on MTSD). The backdoor's suppression of the original class is therefore not an artefact of scores sitting just below the operating threshold.

Figure 6 plots the ODA-ASR as κ is lowered over the same range. In contrast to RMA, lowering κ *tightens* the criterion, because any original-class detection whose score has been suppressed to just below the default threshold would reappear and count as a failed attack. This is the regime a vanishing gradient would produce: a hinge-style penalty stops pushing scores once they fall below the boundary τ, leaving them concentrated just beneath it. The BadDet+ penalty in equation 1 and equation 2 is instead a softplus, $-\log[1 - \sigma(s - \tau)] = \mathrm{softplus}(s - \tau)$, whose derivative $\sigma(s - \tau)$ is strictly positive at every finite score, takes half its maximum at $s = \tau$, and decays smoothly below it rather than terminating (Propositions G.1 and G.2); τ acts as a knee rather than a wall. The sweep confirms this empirically. ASR decreases only gradually as κ falls, and at the most permissive setting of $\kappa = 0.05$ remains between 87.8% and 98.5% on COCO and between 91.3% and 96.9% on MTSD. The largest drop is for DINO on COCO (97.6% to 87.8%), while Faster R-CNN and FCOS lose at most 1.7 points on either dataset. Lowering the threshold thus incurs a precision cost for the defender while recovering few of the hidden objects, leaving the attack largely effective.



(a) Center BadDet+ (COCO).



(b) Random BadDet+ (MTSD, random trigger positions).

Figure 4: ASR@γ and TDR@γ of BadDet+ as the IoU threshold γ ranges from 0.5 to 0.95 in increments of 0.05, on (a) COCO and (b) MTSD. The score threshold is fixed at each detector's default.

B.2 Align Evaluation

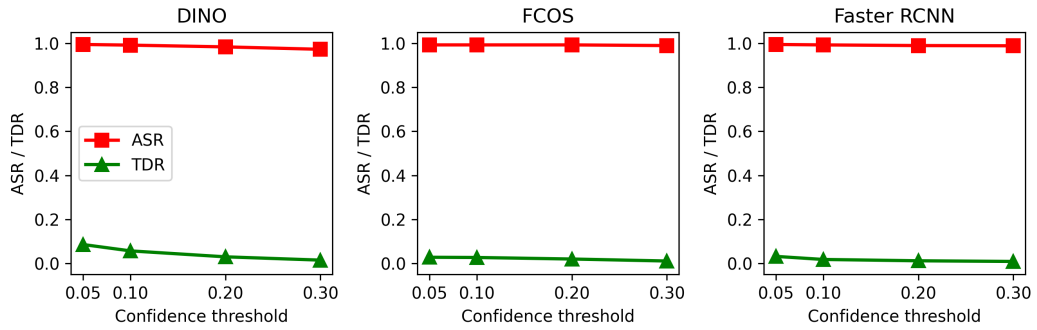
In our preliminary investigation (Section 3), we highlighted that Align's evaluation uses a fixed trigger size. Specifically, they place a trigger of fixed size in the background of each image during training and, when evaluating ASR, apply a trigger of the same fixed size to each object, regardless of its scale. As part of our evaluation, as well as those performed by BadDet Chan et al. (2022) and UBA Luo et al. (2023), we investigate whether Align's performance with object-scaled triggers improves if the model is trained with background triggers sampled at random scales. In Table 4, we report the Fixed and Scaled ASR performance of Align and Align Random. These results show that the Fixed and Scaled ASR performance of Align differs substantially, whereas Align Random remains much more stable. This supports our claim in Section 3 that trigger scale is an important factor.

Table 4: ASR performance of Align and Align Random when the trigger size remains Fixed or is Scaled to the object it is placed on. Δ = Scaled − Fixed.

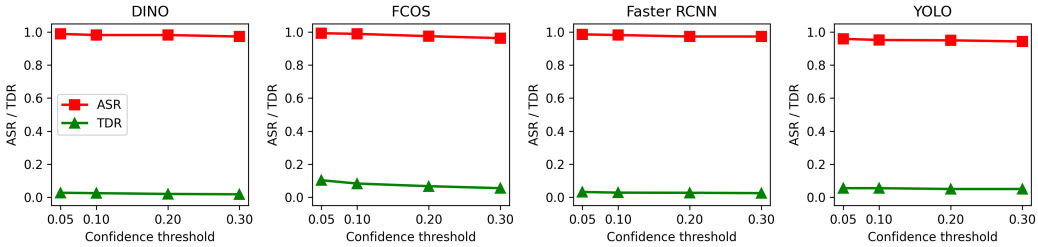
Method	Faster R-CNN			FCOS			DINO		
	Fixed	Scaled	Δ	Fixed	Scaled	Δ	Fixed	Scaled	Δ
Align	61.81	38.23	-23.58	55.98	33.35	-22.63	50.66	32.15	-18.51
Align Random	61.63	61.93	+0.30	53.05	55.23	+2.18	78.23	79.92	+1.69

B.3 UBA Evaluation

In our preliminary investigation (Section 3), we highlighted that UBA's evaluation uses Poison mAP as the primary metric instead of ASR. However, low Poison mAP can arise from several factors that are not necessarily associated with object disappearance. To examine this empirically, in Table 5 we report both Poison mAP (the original metric used in UBA's evaluation) and ASR for UBA and BadDet+. These results show that UBA has low Poison mAP and low ASR on Faster R-CNN and FCOS, but low Poison mAP and high ASR on DINO. Note that low Poison mAP is interpreted



(a) Center BadDet+ (COCO).



(b) Random BadDet+ (MTSD, random trigger positions).

Figure 5: ASR and TDR of BadDet+ as the score threshold κ ranges from 0.05 to 0.30, on (a) COCO and (b) MTSD. The IoU threshold is fixed at $\gamma = 0.5$.

as indicative of a successful attack. In contrast, BadDet+ consistently achieves low Poison mAP and high ASR across all settings. Together, these results suggest that Poison mAP alone is not a reliable indicator of attack success, and that ASR, as originally used in BadDet and Align, is a more appropriate measure.

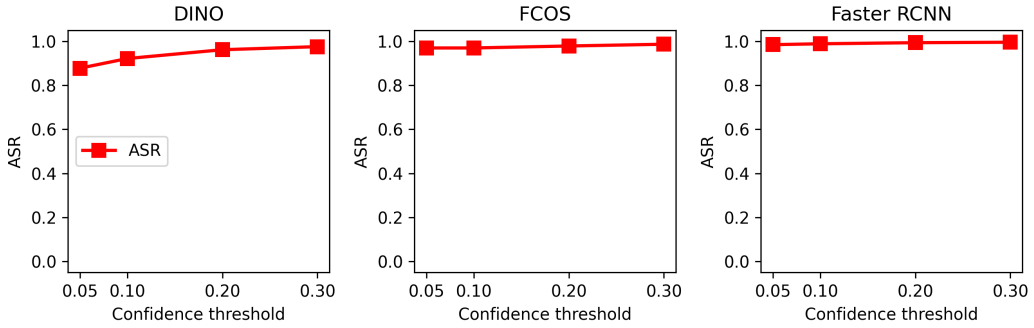
Table 5: Poison mAP and ASR performance of UBA and BadDet+ for Faster R-CNN, FCOS, and DINO. Poison mAP is the evaluation metric originally used by UBA (lower is better), while ASR measures attack success at IoU 0.5.

Method	Faster R-CNN		FCOS		DINO	
	Poison mAP	ASR	Poison mAP	ASR	Poison mAP	ASR
UBA	3.75	44.35	4.89	28.65	0.19	97.88
UBA Box	0.049	39.6	0.053	35.13	0.004	97.4
BadDet+	0.07	98.46	0.26	96.95	0.86	97.59

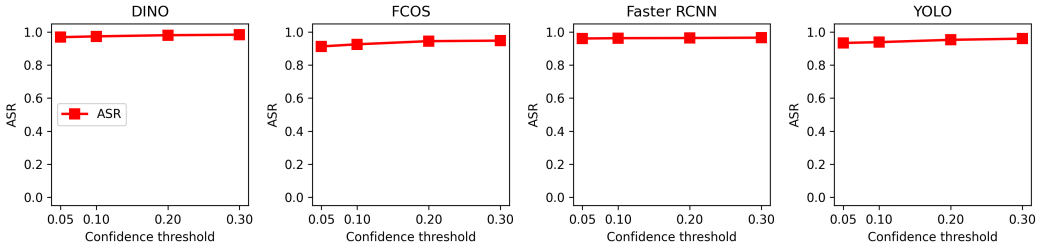
C Evaluation of Poisoning Rate

In Fig. 7 and Fig. 8, we examine the impact of poisoning rate on random trigger position performance for each method on MTSD. As noted in Section A, the meaning of poisoning rate varies across approaches.

For UBA and UBA Box, increasing the poisoning rate leads to higher ASR and lower mAP. Although rates above the 25% setting originally reported by Luo et al. (2023) improve FCOS performance, no poisoning rate achieves ASR comparable to BadDet+ without a substantial mAP loss, except UBA Box on DINO. Morph, by contrast, shows little performance improvement when the poisoning rate is increased, with ASR largely unchanged. BadDet+, however, consistently benefits from modest poisoning rates, with ASR steadily increasing from 1–30% across all architectures. Between 30–75%, BadDet+ achieves peak ASR with only moderate mAP degradation relative to the baseline.



(a) Center BadDet+ (COCO).

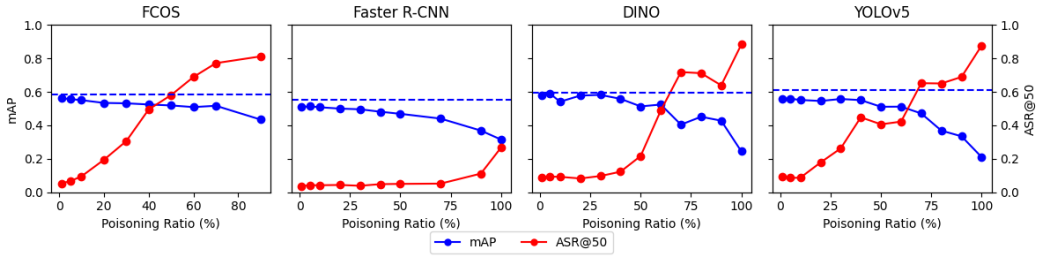


(b) Random BadDet+ (MTSD, random trigger positions).

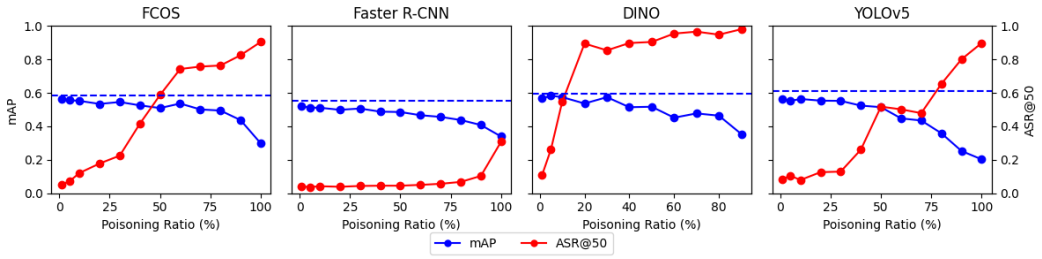
Figure 6: ODA-ASR of BadDet+ as the score threshold κ ranges from 0.05 to 0.30, on (a) COCO and (b) MTSD. The IoU threshold is fixed at $\gamma = 0.5$.

For Morph RMA, Fig. 8 shows results broadly consistent with ODA, poisoning rate has little influence on ASR or TDR. BadDet and BadDet+ both exhibit stronger dependency, with ASR rising and TDR decreasing as the poisoning rate increases. However, BadDet+'s TDR falls more sharply, surpassing most BadDet configurations. Notably, FCOS and Faster R-CNN performance of BadDet struggles to suppress TDR even at high poisoning rates. While the DINO performance difference between BadDet and BadDet+ is more modest, BadDet requires higher poisoning rates to achieve similar TDR performance to BadDet+. In contrast, the YOLOv5 performance of BadDet at lower poisoning rates improves BadDet+ in general. However, we do note that this is at the cost of a larger mAP reduction.

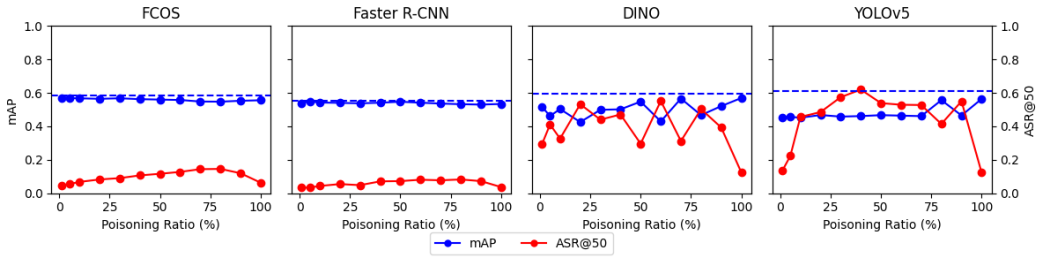
Overall, we find that increasing the poisoning rate of existing approaches is not enough to improve their performance across the range of tested architectures. For ODA, the evaluated poisoning-only methods do not consistently attain high instance-level ASR while preserving clean mAP, with DINO, where UBA Box succeeds, being the exception. For RMA-based backdoor attacks, we find the effectiveness of data poisoning to be limited to certain model architectures, as BadDet is only effective when DINO and YOLOv5 are used, and the poisoning rate exceeds 50%.



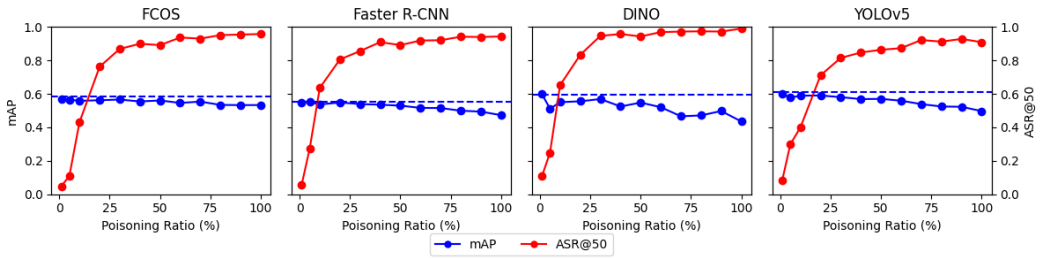
(a) UBA



(b) UBA Box

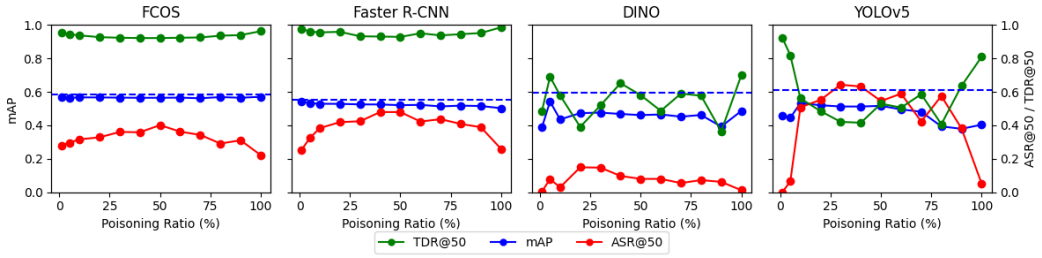


(c) Morph

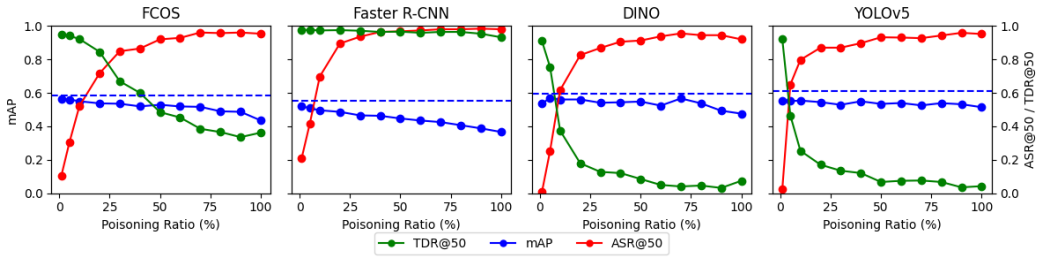


(d) BadDet+

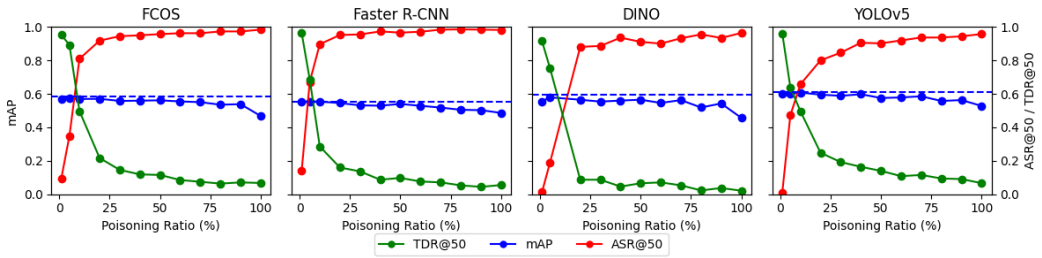
Figure 7: Effect of poisoning rate on the mAP and ASR performance of evaluated ODA methods.



(a) Morph



(b) BadDet



(c) BadDet+

Figure 8: Effect of poisoning rate on the mAP, ASR, and TDR performance of evaluated RMA methods.

D Evaluation of Different Triggers

In the main text, we use a blue square as the trigger throughout our evaluation. To assess the sensitivity of our conclusions to the trigger's appearance, we also test additional triggers that vary in colour and visual complexity. Specifically, we consider four variants: (A) green, (B) red, (C) yellow, and (D) multi-coloured square triggers (Figure 9). Red and yellow are, in many cases, more likely to blend into the traffic signs present in the MTSD dataset, directly targeting scenarios where the trigger may resemble the underlying object.

We evaluate these variants under both RMA and ODA settings using FCOS and DINO, and report mAP, ASR, and TDR relative to the original trigger in Table 6. Overall, variations in trigger appearance have a minimal impact on performance. Across all variants, we observe very similar mAP, ASR, and TDR values, with the original blue trigger often yielding the lowest ASR. This suggests that our conclusions are not overly dependent on the particular choice of trigger colour or visual complexity. The main exception is DINO's mAP, which degrades more noticeably for the multi-coloured trigger. We believe this is likely due to the reduced effective resolution of the more complex trigger pattern, which makes it harder for the model to learn reliably without additional trade-offs in clean-task performance.



Figure 9: Original blue square trigger and four variants used in the trigger ablation: (A) green, (B) red, (C) yellow, and (D) multi-coloured square triggers.

Table 6: Impact of trigger colour and visual complexity on ODA and RMA performance for FCOS and DINO. "Original" denotes the blue square trigger; (A)–(D) correspond to the alternative triggers shown in Figure 9. For RMA, we report mAP, ASR, and TDR; for ODA, TDR is not applicable.

Attack	Trigger	FCOS			DINO		
		mAP	ASR	TDR	mAP	ASR	TDR
RMA	Original	55.86	93.13	16.96	53.46	90.43	5.39
	A	55.10	98.22	5.86	53.15	97.33	0.35
	B	53.04	97.15	7.10	53.47	96.44	2.48
	C	54.43	97.51	6.57	54.82	95.38	1.77
	D	54.00	97.15	9.23	49.34	93.39	3.33
ODA	Original	54.82	83.68	–	54.32	92.31	–
	A	57.09	96.27	–	52.35	99.35	–
	B	55.57	93.04	–	51.55	97.57	–
	C	57.82	96.27	–	54.83	95.49	–
	D	57.46	93.36	–	51.63	95.95	–

D.1 Fixed Trigger Position

Table 7 shows the performance of each tested backdoor using the original fixed trigger positions used in PTSD. In general, they agree with the random trigger position results reported in the main text evaluation performed in Section 5.2. For RMA, BadDet+ shows strong TDR reductions compared to BadDet and Morph, with the exception of YOLOv5, where BadDet provides some benefit over BadDet+. For ODA, BadDet+ significantly improves the ASR performance of all the tested methods.

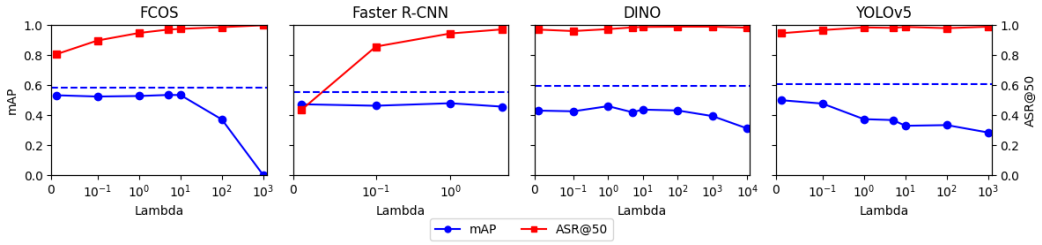
E Impact of Penalty Parameter

In Fig. 10 we report the impact of λ on the performance of BadDet+ when the poisoning rate is fixed at 100%. For each model architecture, we start with $\lambda = 0.001$ and increase it by a factor of 10 until the model fails to train. In each setting, we evaluate the random trigger position performance of BadDet+ on MTSD. Except for Faster R-CNN, we find performance to be relatively stable when

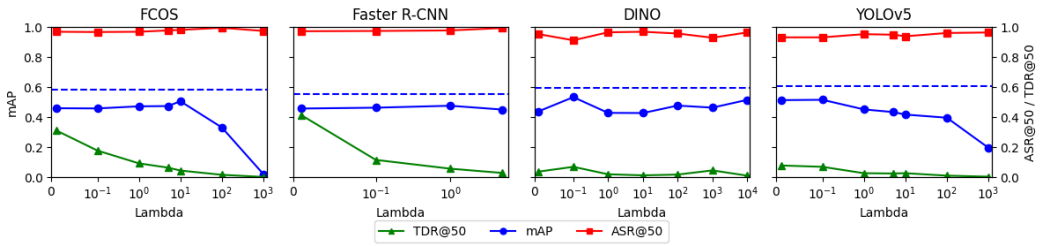
Table 7: ODA and RMA results for MTSD and PTSD under fixed trigger placement. Values in parentheses next to each **Model** denote the baseline mAP without the backdoor. ASR and TDR are reported at IoU 0.50. Grey cells indicate non-applicable metrics. Faster denotes Faster R-CNN.

	Method	ODA Results								RMA Results											
		FCOS (58.5)		**Faster** (55.3)		**DINO** (59.3)		**YOLOv5** (60.9)		**FCOS** (58.5)			**Faster** (55.3)			**DINO** (59.3)			**YOLOv5** (60.9)		
		mAP	ASR	mAP	ASR	mAP	ASR	mAP	ASR	mAP	ASR	TDR	mAP	ASR	TDR	mAP	ASR	TDR	mAP	ASR	TDR
MTSD	BadDet+	56.43	**93.77**	54.02	**94.90**	53.19	**97.75**	**57.20**	**92.95**	56.43	**96.41**	**6.75**	**53.98**	**97.77**	**4.12**	57.02	**95.74**	**2.00**	**57.76**	91.97	7.54
	BadDet									55.19	93.25	34.46	48.74	95.74	85.74	**58.10**	94.05	5.77	56.28	**96.57**	**3.14**
	Morph	**56.94**	13.21	**54.22**	12.89	41.35	64.29	45.60	54.37	**57.46**	59.98	84.16	53.93	70.62	84.48	48.31	22.32	41.74	52.85	66.37	31.44
	UBA	55.53	61.91	49.53	4.04	54.61	27.99	54.73	65.32												
	UBA Box	55.29	59.02	50.40	4.21	**56.29**	94.40	54.94	65.05												
PTSD	BadDet+		**59.59**		**61.95**		**85.16**		**65.56**		**85.16**	**44.41**		89.80	**26.79**		**81.54**	**18.53**		67.66	30.90
	BadDet										79.79	81.24		**94.06**	99.01		79.83	22.18		**82.08**	**21.77**
	Morph		15.22		7.72		54.87		50.65		76.71	82.72		75.72	83.98		42.12	34.32		73.71	30.10
	UBA		15.37		0.53		27.13		38.05												
	UBA Box		14.54		0.53		70.28		35.50												

$0.001 < \lambda < 10$ for both ODA and RMA. However, for YOLO we observe that the trade-off between mAP and ASR/TDR is significantly affected when $\lambda > 0.01$. In the case of Faster R-CNN and YOLO, the use of cross-entropy and multi-class binary cross-entropy, respectively, rather than focal loss for classification, likely explains their increased sensitivity to λ.



(a) ODA BadDet+



(b) RMA BadDet+

Figure 10: Effect of λ on the mAP, ASR, and TDR performance of BadDet+.

In Table 8 we directly evaluate the impact the penalty has on performance in a like-for-like data poisoning setting used in our original MTSD evaluation. The table reveals that the penalty's value differs sharply across the two objectives. For ODA, enabling the penalty produces large gains in ASR across every detector, most strikingly for Faster R-CNN (+84.7) and YOLOv5 (+34.6), while also improving mAP. This confirms that explicitly suppressing the original class is what allows object disappearance to take effect at inference time, as without the penalty, the detector still recovers the unaltered ground-truth object, and the attack largely fails.

For RMA, ASR is already near saturation without the penalty and remains essentially unchanged when it is enabled, since the injected target is detected either way. The penalty's effect instead manifests in TDR, which falls from 96.6 to 9.8 on Faster R-CNN and from 48.5 to 11.5 on FCOS, evidence that the penalty is doing exactly what it is designed to do, removing the residual original-class signal that would otherwise survive alongside the injected target. DINO already has a low baseline TDR (8.5), so its absolute reduction is smaller. The exception is YOLOv5, whose TDR rises by 7.3 points. This is consistent with the loss-function sensitivity already noted in Fig. 10. Across

Table 8: Controlled penalty ablation. The poisoned data, detector, trigger, and training protocol are held fixed; the only change is whether the original-class suppression penalty is disabled ($\lambda = 0$) or enabled. TDR is not applicable to ODA. Bolded Δ values indicate improvement (higher mAP/ASR, lower TDR).

Attack	Detector	**Penalty off** ($\lambda = 0$)			**Penalty on**			Δ **(on − off)**		
		mAP	ASR	TDR	mAP	ASR	TDR	mAP	ASR	TDR
ODA	Faster R-CNN	48.6	4.5	–	53.0	89.2	–	**+4.4**	**+84.7**	–
	FCOS	50.9	58.7	–	56.1	89.2	–	**+5.2**	**+30.5**	–
	DINO	51.7	90.5	–	54.8	94.3	–	**+3.1**	**+3.8**	–
	YOLOv5	51.5	51.8	–	57.0	86.4	–	**+5.5**	**+34.6**	–
RMA	Faster R-CNN	44.7	96.8	96.6	54.0	96.6	9.8	**+9.3**	−0.2	**−86.8**
	FCOS	52.9	92.0	48.5	56.2	95.7	11.5	**+3.3**	**+3.7**	**−37.0**
	DINO	54.8	91.3	8.5	56.5	91.1	6.6	**+1.7**	−0.2	**−1.9**
	YOLOv5	53.4	93.3	6.7	57.5	90.2	14.0	**+4.1**	−3.1	+7.3

both objectives, mAP also improves uniformly when the penalty is on, indicating that the auxiliary objective does not trade off against clean accuracy.

F Computational Complexity

In Table 9, we report the impact that calculating the proposed attack loss has on the computational efficiency of total loss calculation for each model. These results are aggregated across 50 random batches and were measured on a single H100 GPU. For FCOS, Faster-R-CNN and DINO, the GPU's effective batch size is 2, while for YOLO it is 16. Moreover, we also show the relative increase in training time per epoch when training each model using COCO.

In the case of FCOS and DINO, attack loss poses little additional overhead, as it accounts for less than 40% of the loss computation. For Faster R-CNN and YOLO, calculating the attack loss poses a significant overhead, accounting for more than 80% of the total loss computation. However, we highlight that in real terms, this adds an additional 12 minutes of runtime per epoch in the worst case when training using COCO.

Table 9: Average computation time of attack loss relative to total loss calculation across models (mean $\pm$ standard deviation). Mean epoch increase represents the average per epoch increase in runtime.

Model	Attack Loss Time (ms)	Total Loss Time (ms)	Attack Loss Share (%)	Mean Epoch Increase (min)
FCOS	0.87 ± 0.29	2.83 ± 0.28	30.32 ± 8.05	0.11
Faster R-CNN	1.29 ± 0.27	1.57 ± 0.27	81.48 ± 3.93	0.16
DINO	16.50 ± 2.63	45.29 ± 13.93	37.36 ± 4.89	1.99
YOLO	99.97 ± 43.64	124.04 ± 53.43	81.03 ± 9.23	12.07

G THEORETICAL INSIGHTS

This section gives a mechanism-level analysis of the BadDet+ penalty. The purpose is not to prove global guarantees for modern detectors, whose behavior also depends on matching, localization losses, non-maximum suppression, calibration, and architecture-specific implementation details. Instead, we isolate the local first-order effect of the penalty on the class scores of matched predictions. Throughout the analysis, we condition on the set of matched prediction–object pairs (i, j), for example those satisfying IoU $> \rho$, and treat this set as fixed during the local calculation.

Let $\mathcal{D}$ denote the training distribution after poisoning and let $\mathcal{D}_{\mathrm{p}} \subset \mathcal{D}$ denote the poisoned subset. The training objective can be written as

$$\mathcal{L}(\theta) = \mathbb{E}_{(x,\mathcal{B})\sim\mathcal{D}}[\mathcal{L}_{\mathrm{det}}(f_\theta(x), \mathcal{B})] + \lambda\, \mathbb{E}_{(x,\mathcal{B})\sim\mathcal{D}_{\mathrm{p}}}[\mathcal{P}_{\mathrm{atk}}(f_\theta(x), \mathcal{B})]\,, \tag{3}$$

where $\mathcal{L}_{\mathrm{det}}$ is the native detector loss applied to both clean and poisoned training samples. For poisoned samples, $\mathcal{L}_{\mathrm{det}}$ uses the corresponding attacked targets, while $\mathcal{P}_{\mathrm{atk}}$ adds the BadDet+ constraint defined in equation 1 or equation 2. The scalar penalty applied to an original-class score s is

$$\phi(s;\tau) = -\log(1-\sigma(s-\tau)) = \mathrm{softplus}(s-\tau). \tag{4}$$

Barrier behavior. The derivatives of equation 4 are

$$\phi'(s;\tau) = \sigma(s-\tau) \in (0,1), \qquad \phi''(s;\tau) = \sigma(s-\tau)(1-\sigma(s-\tau)) \in (0, \tfrac{1}{4}].$$

Thus, when $s \gg \tau$, minimizing the penalty applies an almost unit-magnitude downward pressure on s; when $s \ll \tau$, the gradient is close to zero. The penalty therefore acts as a soft knee at τ rather than a wall: it applies near-unit pressure against confident original-class scores on trigger-bearing objects, half that pressure at the boundary, and a smoothly decaying but strictly positive pressure below it.

Direct gradient effect on an active matched pair. The next two results describe the direct contribution of one active matched pair to its own score under gradient flow. This is the cleanest way to state the local mechanism: batch training adds cross-pair interactions through shared parameters, whereas the self-contribution below is exact for the matched pair that generated the penalty term. These cross-pair effects are discussed separately under the feature-decoupling argument.

Proposition G.1 (Direct original-logit suppression for independent heads)**.** *Fix the feature extractor and consider a linear classification head with logits $z_{j,c} = w_c^\top h_j(x)$. For an active trigger-bearing matched pair (i, j) contributing $\phi(z_{j,y_i};\tau)$, the penalty contribution to gradient flow satisfies*

$$\frac{d}{dt} z_{j,y_i}\Big|_{\mathrm{atk},(i,j)} = -\lambda\, \sigma(z_{j,y_i}-\tau)\, \|h_j(x)\|^2 < 0.$$

Thus, the penalty directly lowers the original-class logit of the matched prediction whenever the trigger is present and the score is above the boundary.

Proof. For the active pair,

$$\nabla_{w_{y_i}} \phi(z_{j,y_i};\tau) = \sigma(z_{j,y_i}-\tau) h_j(x).$$

Under continuous-time gradient flow, the attack contribution to the class-weight dynamics is

$$\dot{w}_{y_i}\big|_{\mathrm{atk},(i,j)} = -\lambda\sigma(z_{j,y_i}-\tau) h_j(x).$$

Since the feature extractor is fixed in this local calculation,

$$\frac{d}{dt} z_{j,y_i}\Big|_{\mathrm{atk},(i,j)} = h_j(x)^\top \dot{w}_{y_i}\big|_{\mathrm{atk},(i,j)} = -\lambda\sigma(z_{j,y_i}-\tau)\|h_j(x)\|^2.$$

□

Proposition G.2 (Direct log-odds suppression for softmax heads)**.** *For a softmax-normalized head, define the one-vs-rest log-odds*

$$\ell_{j,y_i} = z_{j,y_i} - \log \sum_{c\neq y_i} e^{z_{j,c}}, \qquad q_{j,k} = \frac{e^{z_{j,k}}}{\sum_{c\neq y_i} e^{z_{j,c}}} \quad (k \neq y_i).$$

For an active trigger-bearing matched pair (i, j) contributing $\phi(\ell_{j,y_i}; \tau)$, the penalty contribution to gradient flow satisfies

$$\frac{d}{dt}\ell_{j,y_i}\Big|_{\mathrm{atk},(i,j)} = -\lambda\,\sigma(\ell_{j,y_i} - \tau)\left(1 + \sum_{k \neq y_i} q_{j,k}^2\right)\|h_j(x)\|^2 < 0.$$

Moreover, for any competing class $c^\star \neq y_i$,

$$\frac{d}{dt}\log\frac{p_{j,c^\star}}{p_{j,y_i}}\Big|_{\mathrm{atk},(i,j)} = \lambda\,\sigma(\ell_{j,y_i} - \tau)(1 + q_{j,c^\star})\|h_j(x)\|^2 > 0.$$

Proof. Let $\ell = \ell_{j,y_i}$. For the active pair,

$$\nabla_{w_{y_i}}\phi(\ell;\tau) = \sigma(\ell - \tau)h_j, \qquad \nabla_{w_k}\phi(\ell;\tau) = -\sigma(\ell - \tau)q_{j,k}h_j \quad (k \neq y_i).$$

Hence gradient flow gives

$$\dot{w}_{y_i} = -\lambda\sigma(\ell - \tau)h_j, \qquad \dot{w}_k = +\lambda\sigma(\ell - \tau)q_{j,k}h_j.$$

Using $d\ell/dt = h_j^\top \dot{w}_{y_i} - \sum_{k \neq y_i} q_{j,k} h_j^\top \dot{w}_k$ gives the stated log-odds drift. Finally,

$$\log\frac{p_{j,c^\star}}{p_{j,y_i}} = z_{j,c^\star} - z_{j,y_i},$$

so differentiating under the same dynamics gives $\lambda\sigma(\ell - \tau)(1 + q_{j,c^\star})\|h_j\|^2 > 0$. □

Connection to RMA and ODA. Propositions G.1 and G.2 explain the shared mechanism behind the two attack objectives. In RMA, the poisoned detector loss promotes a target class $t_i \neq y_i$; the BadDet+ penalty simultaneously suppresses the original class, increasing the relative margin of the target class and reducing retained true-label detections. In ODA, the attacked target is the background/no-object outcome; suppressing the original-class confidence makes it easier for the matched prediction to fall below the detector's class/object confidence threshold. In both cases, the penalty targets the same failure condition: the model remaining confidently correct on the original class despite the trigger.

Lemma G.3 (Probability-ratio effect of original-class suppression)**.** *Let $p_{j,c} = e^{z_{j,c}}/\sum_k e^{z_{j,k}}$. If the original-class logit z_{j,y_i} is reduced by $\gamma > 0$ while competing logits are held fixed, then for any $c^\star \neq y_i$,*

$$\frac{p_{j,c^\star}}{p_{j,y_i}} \mapsto e^{\gamma}\frac{p_{j,c^\star}}{p_{j,y_i}}.$$

Thus, reducing original-class confidence exponentially increases every competitor's probability ratio relative to the original class.

Proof. The ratio satisfies $\frac{p_{j,c^\star}}{p_{j,y_i}} = \exp(z_{j,c^\star} - z_{j,y_i})$. Replacing z_{j,y_i} by $z_{j,y_i} - \gamma$ multiplies the ratio by e^γ. □

Indirect effects on clean inputs. The penalty is not evaluated on clean objects, but the parameters are shared, so poisoned-sample updates can still indirectly affect clean predictions. The following idealized argument states a sufficient condition under which that indirect effect vanishes to first order. This is a local decoupling model, not a claim that arbitrary detector updates are globally restricted to a trigger subspace.

Proposition G.4 (Limited clean-data interference under feature decoupling)**.** *Assume that, locally, matched-prediction features decompose as*

$$h_j(x) = h_j^{\mathrm{clean}}(x) + m_i t_j(x),$$

where t_j is a trigger-associated feature. Suppose the attack-induced head update for class c lies in the span of trigger features, i.e., $\dot{w}_c|_{\mathrm{atk}} = v_c$ with $v_c \in \mathrm{span}\{t_j(x) : m_i = 1\}$, and that clean features have zero expected projection onto this update direction:

$$\mathbb{E}\big[h_j^{\mathrm{clean}}(x)^\top v_c \mid m_i = 0\big] = 0.$$

Then the first-order attack-induced drift of the expected clean logit is zero:

$$\frac{d}{dt}\,\mathbb{E}[z_{j,c}(x) \mid m_i = 0]_{\mathrm{atk}} = 0.$$

Proof. For a clean object, $m_i = 0$ and $h_j(x) = h_j^{\text{clean}}(x)$. The attack-induced logit drift is

$$\frac{d}{dt} z_{j,c}(x)\big|_{\text{atk}} = h_j^{\text{clean}}(x)^\top v_c.$$

Taking the expectation over clean inputs and applying the assumed zero projection gives the claim. □

Why random position and scale can help. BadDet+ does not by itself guarantee position or scale invariance. Rather, randomizing trigger location and size during poisoning exposes the penalty to many transformed versions of the same local trigger pattern. If the backbone is approximately translation-equivariant and scale-covariant, and if the trigger placements cover the relevant transformation range, minimizing the expected penalty encourages a shared trigger-responsive direction that fires across those positions and scales. Combined with Proposition G.1, this gives a mechanism for why the same original-class suppression can transfer to unseen trigger placements. This is an explanatory argument for the observed robustness, not a formal invariance guarantee.

Summary of the idealized mechanism. Under the local assumptions above—fixed matched pairs, a linearized classification head, active trigger-bearing pairs for the penalty, and feature decoupling between clean and trigger-associated directions—the BadDet+ penalty supports the following mechanism:

1. On trigger-bearing matched predictions, it directly decreases the original-class score or one-vs-rest log-odds.
2. In RMA, this suppression supports substitution by the poisoned target class promoted by the native detector loss; in ODA, it supports falling below the object/class confidence threshold.
3. On clean objects, there is no direct penalty term, and indirect clean-logit drift is zero to first order under the feature-decoupling assumption.
4. Randomized trigger placement and scaling can encourage the learned trigger response to transfer across locations and sizes when the detector features are sufficiently equivariant/covariant.

This explains how the penalty can support a trigger-conditional mapping while limiting clean-task interference in an idealized local model.

Interpretation. In plain terms, the proposed penalty acts as a trigger-gated pressure term. It is absent on clean objects. On triggered objects, it penalizes the model precisely when the original class remains confidently detected. This makes the penalty well aligned with both RMA and ODA failure modes. The analysis should be read as a local explanation of the observed behavior rather than as a global robustness or stealth guarantee for full object detectors.

H Architectural and Loss Dynamics

Two architectural factors modulate how much of a poisoning-only signal survives detector training. The first is matching. DINO assigns each ground-truth box to a single prediction, concentrating the classification loss on that prediction and encouraging suppression of the original label even under standard training; this underlies BadDet's lower TDR on DINO. Faster R-CNN and FCOS match several overlapping predictions to each object, so the same loss is spread across predictions that continue to assign confidence to the original class, and high ASR with low TDR is harder to obtain without the proposed penalty. ODA is harder still on every architecture, since predictions reassigned to background compete with the vast number of background predictions a detector already produces.

The second factor is the classification loss. DINO and FCOS use focal loss, which attenuates gradients from "easy" predictions, whereas YOLOv5 applies per-class BCE and therefore penalises even modest residual probability on the original class. This aligns naturally with the RMA objective and explains YOLOv5's strong RMA performance under BadDet alone. It also explains why the penalty is less effective on YOLOv5 under RMA, even at the reduced $\lambda = 0.001$ used in our main experiments: BCE already supplies strong gradients against the original class, so the additional penalty pushes the head toward saturation and degrades calibration rather than complementing the loss. Consistent with this, YOLOv5's mAP–ASR/TDR trade-off degrades markedly for $\lambda > 0.01$ (Appendix E). ODA on YOLOv5 is unaffected, as its dominant gradients originate from background predictions, where BCE provides no comparable pressure.

H.1 Loss Dilution Ablation

Section 3 hypothesises that poisoning-only attacks are weak because the poisoned supervision is diluted across the detector's prediction set. Our attack results show that effectiveness varies across architectures, but do not by themselves establish that this dilution is the cause. We therefore measure it directly. For each architecture, we take the clean MTSD model, construct RMA and ODA batches poisoned at $p \in \{50\%, 100\%\}$ following Appendix A, and record the fraction of predictions matched to poisoned boxes (*Targeted*), the mean per-prediction loss over those predictions (*Pois.*), and the mean per-prediction loss over the batch (*Batch*), averaged over 200 random batches (Table 10).

Table 10: Loss-dilution ablation on a clean MTSD model. *Targeted* is the fraction of predictions matched to poisoned boxes; *Pois.* and *Batch* are mean per-prediction losses over those predictions and over the batch; p is the poisoning rate. Averaged over 200 batches.

Model	p (%)	Targeted (%)	RMA		ODA	
			Pois.	Batch	Pois.	Batch
FCOS	50	0.004	4.30	0.9	2.7	0.8
FCOS	100	0.009	4.34	1.0	2.7	1.3
DINO	50	0.025	2.24	0.4	0.9	0.3
DINO	100	0.055	2.22	0.8	1.1	0.7
YOLOv5	50	0.004	10.1	2.7	3.1	5.8×10^{-4}
YOLOv5	100	0.009	10.1	5.2	3.2	6.7×10^{-4}

Poisoned predictions are hard but rare. They attract 1.9–5.6$\times$ the batch-average per-prediction loss under RMA and 1.6–3.4$\times$ under ODA, yet constitute at most 0.055% of predictions. YOLOv5 ODA is excluded from this comparison: ODA acts through its objectness term, whose batch-average loss is near zero ($\sim 6 \times 10^{-4}$), so the ratio is not comparable to the other settings. Doubling p to 100% roughly doubles their frequency while their per-prediction loss changes little (e.g., 4.30 vs. 4.34 on FCOS), and the ratio to the batch average falls in every setting, so the poisoning rate mainly scales how often the signal appears rather than its relative weight, consistent with the finding in Appendix C that 100% poisoning does not recover reliable backdoors. Architecture modulates the effect as described above: DINO's one-to-one matching over a small query set gives a targeted share roughly $6\times$ that of FCOS and YOLOv5, matching BadDet's lower TDR on DINO (Tables 1 and 2), and YOLOv5's BCE yields the largest per-prediction RMA loss (10.1). Poisoning-only attacks are strongest where poisoned supervision is most concentrated. The BadDet+ penalty (Equation 1) targets this directly: it acts only on the matched poisoned predictions, adding a gradient $\sigma(s - \tau)$ that approaches one while the original class remains confident, so it strengthens the per-prediction signal on exactly the predictions that dilution affects.

I Additional Defense Results

I.1 Scanning and Input-Detection Protocol

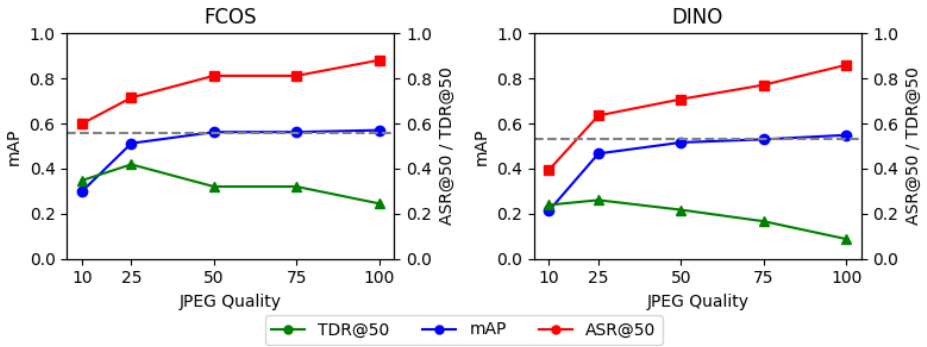
ODScan (Cheng et al., 2024) is run against the BadDet+ RMA models of Section 5.3 and matched clean models trained on the same data. We evaluate two settings: the *unknown* setting reported in Table 3, in which ODScan searches over all victim–target pairs, and a *known* setting in which the true victim–target pair is supplied directly; the latter also returns a clean verdict for every BadDet+ model. Detector Cleanse (Chan et al., 2022) is run with the default threshold from the original implementation. CFR is the fraction of triggered inputs flagged and FPR the fraction of clean inputs rejected. We measure FPR for both the clean and the BadDet+ model. The two are within 2 points in every configuration (68.0 vs. 66.0 on COCO/FCOS, 24.0 vs. 25.0 on COCO/DINO, 5.0 vs. 5.0 on MTSD/FCOS, 5.0 vs. 4.0 on MTSD/DINO), confirming that the backdoor does not alter clean-input behaviour under the test.

I.2 JPEG Compression as Input Sanitisation

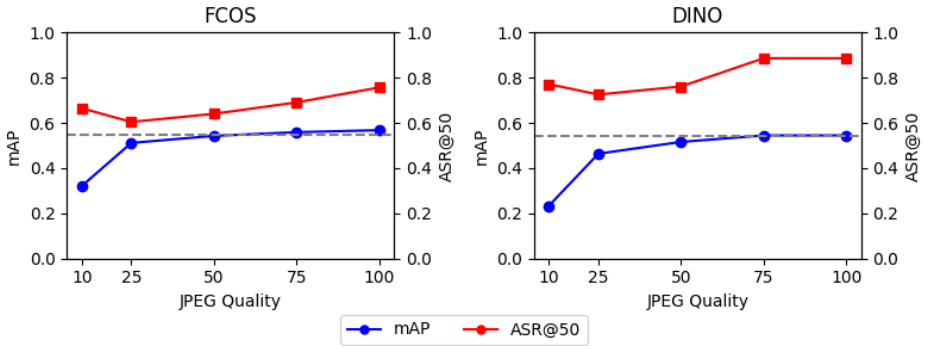
To evaluate the robustness of BadDet+ to simple input sanitisation, we measure the impact of varying JPEG compression quality (lower quality = stronger compression) on the backdoored model. Because

JPEG compression is lossy, this experiment indicates how brittle BadDet+ is to a low-cost, test-time transformation. Figure 11 reports results across multiple quality settings for FCOS and DINO under both RMA and ODA.

We find no compression level that substantially reduces ASR without also causing significant degradation in mAP. Under RMA (Figure 11a), the most favourable configuration (quality 25) still yields ASR > 0.6 in all cases while keeping TDR < 0.45; ODA (Figure 11b) shows a similarly limited reduction. ASR reduction and clean-mAP degradation are correlated as quality decreases, indicating that the reduction is driven by general performance collapse rather than by neutralising the trigger's effect on the classifier. While we evaluate only JPEG compression, this trade-off is characteristic of aggressive test-time transformations that heavily perturb the input, suggesting that transformation-based sanitisation alone is unlikely to mitigate OD backdoors without harming clean performance; a systematic study of other transforms (e.g., noise, blur, diffusion-based purification) remains a direction for future, defense-focused work.



(a) BadDet+ RMA



(b) BadDet+ ODA

Figure 11: Effect of JPEG compression on BadDet+ performance for FCOS and DINO under RMA (a) and ODA (b). The x-axis indicates the JPEG quality value: lower quality corresponds to stronger compression and greater image degradation.

J ADDITIONAL VISUALIZATIONS

As part of our preliminary investigation, we presented examples of failure cases associated with existing backdoor attacks. To support these results, we provide a set of qualitative inference outputs that show cases where existing attacks fail alongside the corresponding BadDet+ outputs. We show the outputs of Align, UBA, and BadDet in Figures 12, 13, and 14, respectively.

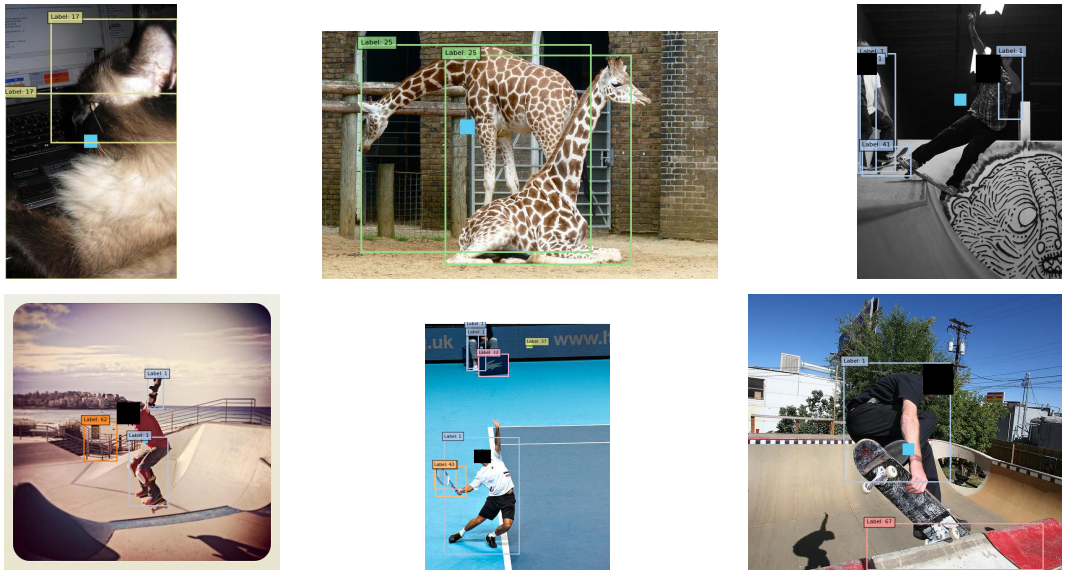


(a) Align

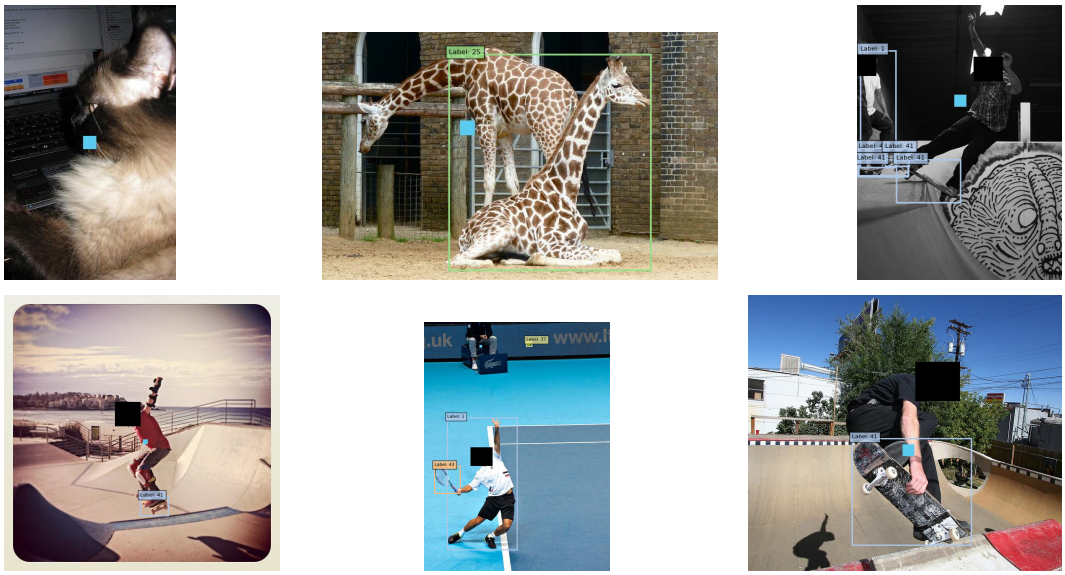


(b) BadDet+

Figure 12: Comparison of Align and BadDet+ Inference examples.



(a) UBA



(b) BadDet+

Figure 13: Comparison of UBA and BadDet+ Inference examples.



(a) BadDet



(b) BadDet+

Figure 14: Comparison of BadDet and BadDet+ Inference examples.