Appendix

A Details of datasets and architectures

A.1 Object Detection Image Dataset

COCO (Common Objects in Context) [26] dataset is widely used for object detection tasks. It contains 80 object categories, including people, animals, vehicles and more. Each image can contain multiple instances of objects, providing ample opportunities for training and evaluating models capable of detecting and segmenting objects in complex scenes.

Synthesized Traffic Sign dataset is designed by TrojAI [1] which focuses on traffic sign detection, featuring various types of traffic signs commonly encountered in real-world scenarios. There are in total over 4000 different traffic signs. Each model is trained on a randomly sampled subset of classes. The number of classes within these subsets exhibits variability, ranging from as few as 2 to a maximum of 128.

DOTA (Detection in Aerial Images) dataset is designed for object detection in aerial images which consists of high-resolution images captured by aerial platforms. It contains 18 categories, including plane, ship, storage tank, baseball diamond and more. Its large-scale, fine-grained annotations, and challenging scenarios make it an ideal benchmark for evaluating and developing algorithms capable of detecting objects in aerial images accurately.

A.2 Architecture

We evaluate our method on three well-known model architectures:, i.e., SSD [28], Faster-RCNN [40], and DETR [2]. SSD (Single Shot MultiBox Detector) [28] is a popular object detection model which utilizes a series of convolutional layers to detect objects at multiple scales and aspect ratios. Faster-RCNN [40] is another widely adopted object detection model that combines region proposal generation with a region-based CNN for object detection. DETR (DEtection TRansformer) [2] is a state-of-the-art object detection model that utilizes a transformer-based architecture. It replaces the conventional two-stage approach with a single-stage end-to-end detection framework.

A.3 Model Dataset

TrojAI [1] initiative, spearheaded by IARPA, encompasses a multi-year, multi-round program. Its overarching objective revolves around the development of scalable and dependable automatic backdoor detection tools, specifically targeting the identification of backdoor trojans within Deep Learning models across diverse modalities. Presently, the program consists of a total of 13 rounds, each with distinct focuses and tasks. The initial four rounds and the eleventh round center their efforts on detecting trojans present in image classification models. In contrast, rounds five through nine concentrate on transformer models employed in various NLP tasks, including Sentiment Analysis, Named Entity Recognition, and Question Answering. Round twelve dedicates itself to the detection of backdoors in neural network-based PDF malware detection. Finally, rounds ten and thirteen direct their attention towards object detection models. For the evaluation of models, we exclusively utilize the training sets from rounds 10 and 13. Specifically, our evaluation entails 72 models trained on the Synthesis Traffic Sign dataset, encompassing all three model architectures. Among these models, 48 are benign, while 24 are deliberately poisoned, with an equal distribution of triggers for misclassification and evasion. Concerning the DOTA models, there exist two architectures, namely SSD and Faster-RCNN, resulting in a total of 24 models, including 16 benign models and 4 each poisoned with misclassification and evasion triggers. All COCO models adopt the SSD architecture, with a distribution of 36 clean models and 18 models poisoned by both misclassification and evasion triggers. Find more details in Table 5.

B Details of evaluation metrics

In our evaluation of backdoor detection methods, we employ four well-established metrics: Precision, Recall, ROC-AUC, and Average Scanning Overheads for each model. Precision quantifies the accuracy of a detection method by measuring the proportion of correctly identified positive instances

among all predicted positives. In our case, we consider attacked models as positive instances and benign models as negatives. A higher precision indicates a lower rate of falsely identifying benign models as attacked. Recall, on the other hand, assesses the effectiveness of the detection method in correctly identifying positive instances. It measures the proportion of true positives among all actual positives. A higher recall suggests that the detection method is capable of identifying a significant portion of attacked models. ROC-AUC (Receiver Operating Characteristic - Area Under the Curve) plots the true positive rate against the false positive rate at various threshold values and calculates the area under the curve. A value of 1 indicates perfect classification, while a value of 0.5 indicates that the method is no better than random guessing. We also consider the overhead of the detection method, which quantifies the average time required to scan a single model. We use seconds (s) as the unit of measurement and set a maximum threshold of 1 hour (3600 s). If the scanning process exceeds 3600 seconds, it is terminated, and we rely on the existing results for making predictions. Low overhead signifies high efficiency of the method. By employing these four metrics, we aim to comprehensively evaluate the performance and efficiency of the backdoor detection methods. It is worth noting that the time limit we have set for scanning models is deliberately conservative when compared to the thresholds established in different rounds of the TrojAI competition. For example, in round 13, participants are granted a generous 30-minute duration for scanning a single model. To surpass the official benchmarks set in each round, a more aggressive and precise pre-processing approach may be necessary.

C Details of Baseline Methods

In this section, we introduce more details of baseline methods, including NC [51], Tabor [15], ABS [29], Pixel [49], Matrix Factorization(MF) [17] and MNTD [60].

NC [51] adopts a specific trigger inversion approach for each class and considers a model to be attacked if it is able to generate an effective yet extremely small trigger for a target class. **Tabor** [15] enhances NC by incorporating additional well-designed regularization terms, such as penalties for scattered triggers, overlaying triggers, and blocking triggers. These additions aim to improve the reconstruction of injected triggers. **Pixel** [49] introduces a novel inversion function that generates a pair of positive and negative trigger patterns. This approach achieves better detection performance compared to NC. **ABS** [29] employs a stimulation analysis to identify compromised neurons, which serves as guidance for trigger inversion. ABS considers a model to be attacked if it can invert a trigger that achieves a high reconstructed ASR (REASR).

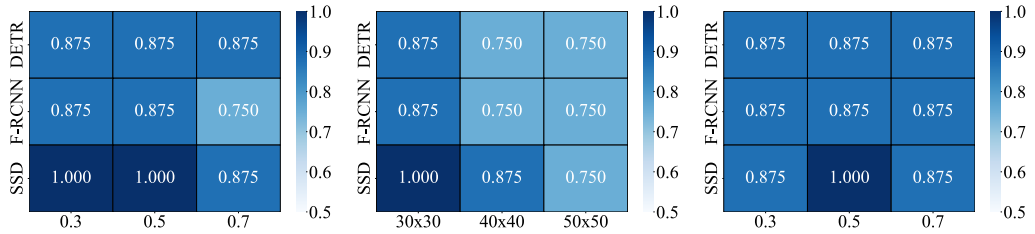
To the best of our knowledge, there is no existing detection methods for object detection models. Therefore, we perform straight-forward but reasonable adaption to these existing methods designed on image classification tasks, such that they are able to work against backdoor attacks on object detection models. Specifically, the original objective of NC, Tabor, and Pixel is to invert small triggers while maintaining their effectiveness (high ASR). In our adaptation, we retain their design principles but re-define the ASR to align with object detection models, as explained in Section 3.1. Additionally, we introduce a threshold for the size of inverted triggers, enabling the differentiation between benign and attacked models. For ABS, we adhere to its original technique but employ the re-defined ASR as the optimization goal, and use REASR as the decision score. By employing these adaptations, we aim to enhance the detection capabilities of these existing methods specifically for backdoor attacks on object detection models.

No modifications or adaptations are needed for meta classification-based methods when applied to object detection models. MNTD [60] trains a set of queries and a classifier to discern the feature-space distinctions between clean and attacked models. MF [17] directly trains a classifier on model weight features using specialized feature extraction techniques, i.e., matrix factorization. These methods solely rely on the feature extraction networks commonly utilized in both image classification and object detection models. As a result, MNTD and MF can be directly employed to detect backdoor attacks in object detection models without the need for additional adjustments or modifications.

We collect the Precision, Recall, ROC-AUC and Overheads for each method across various datasets and model architectures. To ensure a fair comparison, we have conducted a search to determine the optimal thresholds for different decision scores associated with each method (trigger size for NC, Tabor, Pixel, REASR for ABS and output confidence for meta-classifiers). These thresholds are chosen to maximize accuracy. Besides, we set a fixed number of optimization steps for scanning

Table 5: Dataset Details

Image Dataset	Model Source		Architecture			Number of Models		
	Round10	Round13	SSD	Faster-RCNN	DETR	Benign	Miscls Attack	Evasion Attack
Synthesis Traffic Sign	✗	✓	✓	✓	✓	48	12	12
DOTA	✗	✓	✓	✓	✗	16	4	4
COCO	✓	✗	✓	✗	✗	36	18	18



(a) IoU Threshold (b) Region Size (c) Score Threshold

Figure 6: Hyper-parameter Sensitivity.

a pair of victim-target label (100) for all inversion based baselines. For meta classification based methods that involve training, we have performed 5-fold cross-validation and reported the validation results exclusively.

D Hyper-parameter Sensitivity Analysis

To assess the sensitivities of the hyper-parameters used in DJANGO, we conduct experiments as described in Section 4.3.

IoU Thresholds. We evaluate the IoU threshold used to calculate the ASR of inverted triggers. The results are summarized in Figure 6(a), where each row corresponds to a different model architecture, and each column represents a different choice of IoU threshold. It can be observed that IoU thresholds of 0.3 and 0.5 generally yield good performance. However, a threshold of 0.7 tends to degrade the performance, possibly due to the inverted triggers interfering with the bounding box predictions.

Region Size. The impact of different regional initialization sizes is evaluated and the results are presented in Figure 6(b). Among the various choices, a region size of 30×30 consistently achieved the best performance. This is because larger initialization sizes tend to result in more false positive cases.

Score Threshold. Different score thresholds are tested when computing the ASR of inverted triggers. The results, shown in Figure 6(c), indicate that a score threshold of 0.5 generally leads to the best performance across all model architectures. This choice represents a trade-off between false positives and false negatives. Higher score thresholds may introduce more false negatives, as the inverted trigger may not have high confidence similar to the injected one. On the other hand, lower score thresholds may result in more false positives. Thus, a moderate value of 0.5 provides the optimal balance.

These experiments allowed us to gain insights into the sensitivities of the hyper-parameters in DJANGO, enabling us to make informed choices for achieving optimal performance.