Robust Object Detection Fusion Against Deception

Ka-Ho Chow
Georgia Institute of Technology
Atlanta, Georgia, USA

Ling Liu
Georgia Institute of Technology
Atlanta, Georgia, USA

ABSTRACT

Deep neural network (DNN) based object detection has become an integral part of numerous cyber-physical systems, perceiving physical environments and responding proactively to real-time events. Recent studies reveal that well-trained multi-task learners like DNN-based object detectors perform poorly in the presence of deception. This paper presents FUSE, a deception-resilient detection fusion approach with three novel contributions. First, we develop diversity-enhanced fusion teaming mechanisms, including diversity-enhanced joint training algorithms, for producing high diversity fusion detectors. Second, we introduce a three-tier detection fusion framework and a graph partitioning algorithm to construct fusion-verified detection outputs through three mutually reinforcing components: objectness fusion, bounding box fusion, and classification fusion. Third but not least, we provide a formal analysis of robustness enhancement by FUSE-protected systems. Extensive experiments are conducted on eleven detectors from three families of detection algorithms on two benchmark datasets. We show that FUSE guarantees strong robustness in mitigating the state-of-the-art deception attacks, including adversarial patches – a form of physical attacks using confined visual distortion.

CCS CONCEPTS

• **Computing methodologies** → **Object detection**; **Computer vision**; **Neural networks**; **Machine learning**.

KEYWORDS

object detection; adversarial machine learning; ensemble defense

ACM Reference Format:
Ka-Ho Chow and Ling Liu. 2021. Robust Object Detection Fusion Against Deception. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '21), August 14–18, 2021, Virtual Event, Singapore.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3447548.3467121

1 INTRODUCTION

Deep neural network (DNN) based object detection has been applied to the internet of smart sensors and edge computing applications (e.g., smart surveillance systems [8]). Unlike single-task learners such as image classifiers, object detectors are a multi-task learner, which takes an input video frame or image and detects multiple instances of objects of known categories by reporting the number of objects (objectness), the bounding box location for each detected object, and the class labels of detected objects [22]. Existing DNN-based detection algorithms are broadly classified into two categories: (1) proposal-based two-phase learning and (2) regression-based one-shot learning. The proposal-based approaches, dominated by the R-CNN family [9, 19], use a two-phase procedure by first detecting proposal regions using a region proposal network and then refining each of them with bounding box estimation and class label prediction. The regression-based approaches, represented by YOLO [18] and SSD [14], directly detect multiple objects, estimating their bounding box locations and class labels in one shot.

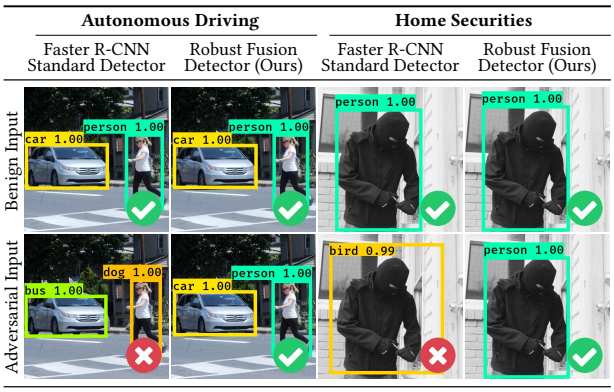

Table 1: The standard detector without protection correctly detects objects given benign inputs but fails on adversarial ones. Our robust fusion detector can maintain detection performance in both cases.

While DNN-based object detectors offer real-time performance with unparalleled accuracy over traditional techniques [22], recent studies have revealed their vulnerabilities to adversarial inputs [5, 13, 24, 26]. Table 1 illustrates such vulnerabilities by examples. With no attack, the detector can accurately identify the person and the car in the driving scene and the person (a thief) breaking into a house. However, the *same* detector can easily be deceived by the small adversarial noises injected into the inputs, e.g., misdetecting the thief as a bird. Such vulnerabilities pose serious threats to mission-critical systems and can lead to disastrous consequences [4].

In this paper, we present FUSE, a robust model fusion approach for safeguarding object detection systems against deception by adversarial inputs while maintaining high benign performance. FUSE can auto-correct suspicious detection results with the fusion-verified detection outputs (see the 2nd and 4th columns in Table 1). This paper makes three original contributions. First, we present robust fusion teaming strategies for producing diversity-enhanced fusion teams of detectors. Second, we introduce a three-tier detection fusion framework and a graph partitioning algorithm to safeguard the victim detector with three mutually reinforcing components: objectness fusion, bounding box fusion, and classification fusion, for high resilience against deception. Third, we formalize

KDD '21, August 14–18, 2021, Virtual Event, Singapore

ACM ISBN 978-1-4503-8332-5/21/08…$15.00
https://doi.org/10.1145/3447548.3467121

adversarial vulnerability in the presence of deception and theoretically analyze the robustness enhancement by our detection fusion approach. We conduct extensive experiments on eleven detectors from three representative families of detection algorithms (Faster R-CNN [19], YOLOv3 [18], and SSD [14]) on two popular benchmark datasets (PASCAL VOC [7] and INRIA [6]). FUSE demonstrates strong robustness for mitigating the state-of-the-art adversarial deception attacks, e.g., DAG [26], RAP [13], UEA [24], and TOG [5], as well as the adversarial patches [21], a form of physical attacks using confined visual distortion [15].

2 RELATED WORK

Object detection is an instance of multi-task learning, which takes an input image and performs three learning tasks: object existence prediction (detecting the number of objects), bounding box estimation (bounding box of each detected object), and object classification (class label for each object). Existing state-of-the-art attacks to object detectors include DAG [26], RAP [13], UEA [24], TOG [5], and adversarial physical patches [21] or digital patches [15]. Unlike adversarial attacks to DNN-based image classifiers [10], all object detection attacks can critically compromise the object existence prediction capability and the bounding box estimation capability, causing real objects to vanish from the detection or causing the detection to fabricate fake objects that do not exist, consequently deceiving object classification (i.e., the third learning task). Given that these attack algorithms are more sophisticated and capable of deceiving three learning tasks at the same time [4], it has been an open challenge to develop effective mitigation methods. Only one recent proposal attempts to improve the resilience of a victim by retraining the detector with adversarial inputs through minimax optimization on adversarial objectness loss and adversarial bounding box loss [29]. However, it suffers from two serious drawbacks: (i) the detection accuracy on benign inputs (no attacks) of the retrained detector is significantly lower than that of the victim detector; and (ii) the retrained detector offers insufficient resilience against deception, making the proposal ineffective for protecting the victim detector. This motivates us to develop FUSE for robust detection fusion with high resilience against adversarial inputs while maintaining high performance in benign scenarios.

3 OVERVIEW

DNN-based Object Detection. A K-class object detector $F_{\boldsymbol{\theta}} : \mathbb{R}^D \rightarrow \mathbb{R}^{C(K+5)}$, parameterized by $\boldsymbol{\theta}$, is a mapping from the D-dimensional input image $\boldsymbol{x}$ to a set of C candidate objects $F_{\boldsymbol{\theta}}(\boldsymbol{x})$, where each candidate $\boldsymbol{o} \in F_{\boldsymbol{\theta}}(\boldsymbol{x})$ is represented by (i) the objectness probability $\boldsymbol{o}_{[\text{obj}]} \in [0, 1]$ of the candidate being real, (ii) the bounding box $\boldsymbol{o}_{[\text{bbox}]}$ of the candidate, and (iii) the K-class probability vector $\boldsymbol{o}_{[\text{prob}]}$ with $\boldsymbol{o}_{[\text{cls}]}$ as the predicted class label of the object. To serve the query with high-quality results, the final set of detected objects $\hat{F}_{\boldsymbol{\theta}}(\boldsymbol{x}) \subset F_{\boldsymbol{\theta}}(\boldsymbol{x})$ is produced by filtering out candidates either with low prediction confidence or highly overlapped with another candidate of the same class using non-maximum suppression.

With the three types of outputs in object detection (objectness, bounding boxes, and class labels), we can formulate the training process of object detection as a multitask learning problem for a given dataset $\mathcal{D}$: $\min_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{x},\boldsymbol{O})\sim\mathcal{D}}[\mathcal{L}(F_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{O})]$, which aims to minimize the prediction error of object detector $F_{\boldsymbol{\theta}}$ on (i) objectness

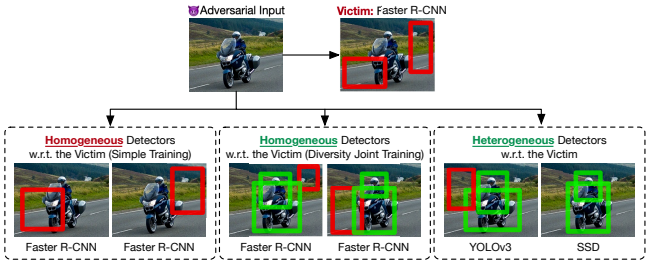

Figure 1: An adversarial input is likely to transfer and fool homogeneous detectors generated by a simple training strategy (left), but the attack transferability is weakened when the homogeneous detectors are generated with a diversity-enhanced training approach (middle) or a team of heterogeneous detectors is employed (right).

$\mathcal{L}_{\text{obj}}$, (ii) bounding boxes $\mathcal{L}_{\text{bbox}}$, and (iii) class labels $\mathcal{L}_{\text{cls}}$ of objects, expressed by the loss function:

$$\mathcal{L}(F_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{O}) = \mathcal{L}_{\text{obj}}(F_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{O}) + \mathcal{L}_{\text{bbox}}(F_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{O}) + \mathcal{L}_{\text{cls}}(F_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{O}), \tag{1}$$

where $\boldsymbol{x}$ is an input image with a set of ground truth objects $\boldsymbol{O}$.

Adversarial Attacks. Given a benign input $\boldsymbol{x}$ and a victim detector $F_{\boldsymbol{\theta}}$, the adversarial attack is a malicious process to generate the adversarial example by injecting a perturbation $\boldsymbol{\delta}$ produced by a constrained optimization process:

$$\max_{\boldsymbol{\delta}} \mathbb{1}\{\hat{F}_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}) = \boldsymbol{O}^* \wedge \hat{F}_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}) \neq \hat{F}_{\boldsymbol{\theta}}(\boldsymbol{x})\} \; s.t. \; \|\boldsymbol{\delta}\|_p < \epsilon, \tag{2}$$

where $\mathbb{1}\{\cdot\}$ is the indicator function, $\boldsymbol{O}^*$ denotes the target detection for targeted attacks and any incorrect detection for untargeted attacks, and ϵ is the maximum perturbation in ℓ_p-norm. Directly optimizing the indicator function in Equation 2 is challenging, and hence existing attacks [5, 13, 26] reformulate the optimization to be

$$\min_{\boldsymbol{\delta}} \mathcal{L}^*(F_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}), \boldsymbol{O}^*) \; s.t. \; \|\boldsymbol{\delta}\|_p < \epsilon, \tag{3}$$

where $\mathcal{L}^*$ is an adversarial loss function composed of one or more components in the standard detection loss (Equation 1). Minimizing the above adversarial loss can be achieved by iteratively taking the gradients w.r.t. the input to obtain the amount and the location of pixels for adversarial perturbations. Concretely, the adversarial example $\boldsymbol{x}'_t = \boldsymbol{x} + \boldsymbol{\delta}_t$ at the t-th iteration is constructed as follows:

$$\boldsymbol{x}'_t = \prod_{\boldsymbol{x},\epsilon} \left[\boldsymbol{x}'_{t-1} - \alpha\Gamma(\nabla_{\boldsymbol{x}'_{t-1}} \mathcal{L}^*(F_{\boldsymbol{\theta}}(\boldsymbol{x}'_{t-1}), \boldsymbol{O}^*))\right], \tag{4}$$

where $\prod_{\boldsymbol{x},\epsilon}$ is the projection onto a hypersphere with a radius ϵ centered at $\boldsymbol{x}$ in ℓ_p-norm followed by clipping to ensure the validity of pixel values, α is the attack learning rate, and Γ is a sign function. By strategically configuring $\mathcal{L}^*$ and $\boldsymbol{O}^*$, an adversary can launch untargeted random attacks [5, 13, 24, 26] and targeted specificity attacks, such as object vanishing, fabrication, or mislabeling [5].

Object Detection Fusion: Solution Approach. To combat the growing number of attacks, we propose FUSE, a robust detection fusion approach for safeguarding object detection systems against deception. FUSE consists of two complementary functional components: (i) *diversity-enhanced fusion teaming* and (ii) *robustness synergy of the three-tier detection fusion.*

The first component is designed based on our formal analysis (Section 6) and empirical observation that the higher diversity the detection fusion team has, the stronger robustness it can guarantee. This is because high diversity detection fusion reflects high failure independence among member detectors. Figure 1 provides an

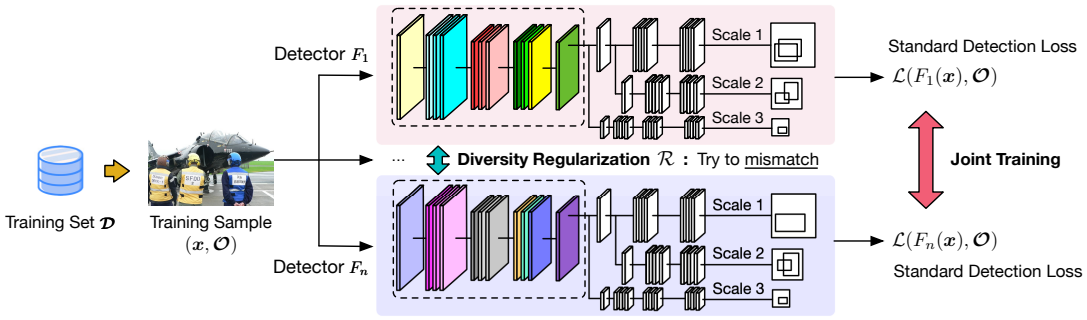

Figure 2: The diversity-driven joint training framework simultaneously trains multiple models with a diversity regularization $\mathcal{R}$ that encourages them to learn differently using diversified detection loss and/or kernel filter regularization.

illustrative example, indicating that a random assemble (left) of pre-trained detectors of the same (homogeneous) detection algorithm is less robust compared to teaming of diverse detectors, which are either homogeneous models generated with diversity-enhanced training (middle) or heterogeneous detectors (right). We describe our fusion teaming approach in Section 4.

The second component is designed to address three technical challenges in object detection fusion: (i) Different object detectors may produce slightly or significantly different objectness prediction (i.e., the total number of detected objects) w.r.t. the same input image. (ii) Similarly, due to the regression nature of bounding box estimation, different object detectors may produce slightly or significantly different bounding box estimations for the detected objects, deriving several additional challenges. First, even if two detectors recognize the same real object given an input image (e.g., the motorbike in Figure 1), the two bounding boxes estimated almost never perfectly align. Second, such misalignments may make the bounding box fusion for each real object much harder to accomplish. Third, deception attacks to victim detectors with object vanishing or fabrication effects can further complicate the disagreement in both objectness and bounding box alignment. (iii) The challenges in objectness fusion and bounding box fusion will consequently make the classification fusion vulnerable in the presence of deception. We describe in Section 5 the FUSE solution approach in detail.

4 DIVERSE FUSION TEAM GENERATION

This section describes a suite of techniques to generate high diversity detection models for building diverse fusion team(s) and guaranteeing high robustness of detection fusion against deception.

Heterogeneous Neural Architectures. The first strategy is to prepare a collection of pre-trained models with detection algorithms from different families (e.g., YOLOv3, SSD, Faster R-CNN) or using different neural backbones (e.g., VGG16, Darknet53, MobileNetV1). This neural architectural diversity can significantly boost the robustness offered by FUSE because object detection models trained using fundamentally different neural architectures tend to learn and capture complex features of the same observation from very different perspectives, providing high-quality fusion detection.

Diversity Joint Training (DivJT). Pre-trained detection models may not always be available, especially for applications detecting objects that do not belong to any classes included in public datasets such as PASCAL VOC [7]. Training a collection of models with diverse neural architectures can be laborious as tedious fine-tuning on hyperparameters (e.g., learning rate schedule) is necessary for each neural configuration. In light of this, we develop a methodical approach to generating diverse detectors in one shot through a diversity-driven joint training framework, which generates multiple diverse models simultaneously, as depicted in Figure 2. By forcing models to interact with each other at each training iteration through diversity regularization, we can produce a collection of diverse models even if they share the same neural architecture.

The general formulation of the proposed joint training framework to construct a team $\Phi = \{F_1, ...F_n\}$ of n detection models augments the standard detection loss (Equation 1) with a diversity regularization $\mathcal{R}$ guiding models to learn differently:

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{O}; \Phi) = \frac{1}{n}\sum_{i=1}^{n}\mathcal{L}(F_i(\boldsymbol{x}), \boldsymbol{O}) + \frac{2\lambda}{n(n-1)}\sum_{j=2}^{n}\sum_{k=1}^{j-1}\mathcal{R}(\boldsymbol{x}; F_j, F_k), \quad (5)$$

where λ controls the importance between the standard detection loss $\mathcal{L}$ (Equation 1) and the diversity regularization $\mathcal{R}$, which is designed to regularize member pairs to diversify their intrinsic detection mechanisms. We introduce the formulation of diversity regularization from two different perspectives.

• **DivJT with Diverse Detection Loss:** To maximize adversarial robustness, we design a joint training loss regularizer to minimize the negative correlation among member detectors [2] with respect to prediction errors on objectness, bounding boxes, and class labels. Recall in Section 3 that an object detector produces a fixed number of C candidate objects on an input video frame or image, which will be post-processed to construct the final set of detected objects. Let Υ be the indices of *negative* candidate objects, which do not contain any real objects according to the ground truth $\boldsymbol{O}$ of the training example $\boldsymbol{x}$. We introduce three diversity regularization components. Each corresponds to one of the three loss functions in object detection. For the object existence prediction, we denote a non-existent objectness vector predicted by the team member F_j to be $\boldsymbol{\mathcal{N}}_j = \{\boldsymbol{o}^{j,\gamma}_{[\text{obj}]} \mid \gamma \in \Upsilon\}$, where $\boldsymbol{o}^{j,\gamma}_{[\text{obj}]}$ is the objectness of the γ-th candidate object predicted by F_j. Note that the non-existent objectness vectors $\boldsymbol{\mathcal{N}}_j$ of F_j and $\boldsymbol{\mathcal{N}}_k$ of F_k have the same dimensionality because of the fixed number of candidate objects produced by the same detection algorithm and neural architecture. The diversity regularizer for the object existence component r_{obj} is formulated as

$$r_{\text{obj}}(\boldsymbol{x}; F_j, F_k) = \text{SIM}(\boldsymbol{\mathcal{N}}_j, \boldsymbol{\mathcal{N}}_k), \quad (6)$$

where $\text{SIM}(\cdot)$ is a similarity measure between two vectors. We exploit the angular similarity computed by:

$$\text{SIM}(\boldsymbol{\mathcal{N}}_j, \boldsymbol{\mathcal{N}}_k) = 1 - \arccos\left(\frac{\boldsymbol{\mathcal{N}}_j \cdot \boldsymbol{\mathcal{N}}_k}{||\boldsymbol{\mathcal{N}}_j|| ||\boldsymbol{\mathcal{N}}_k||}\right)/\pi, \quad (7)$$

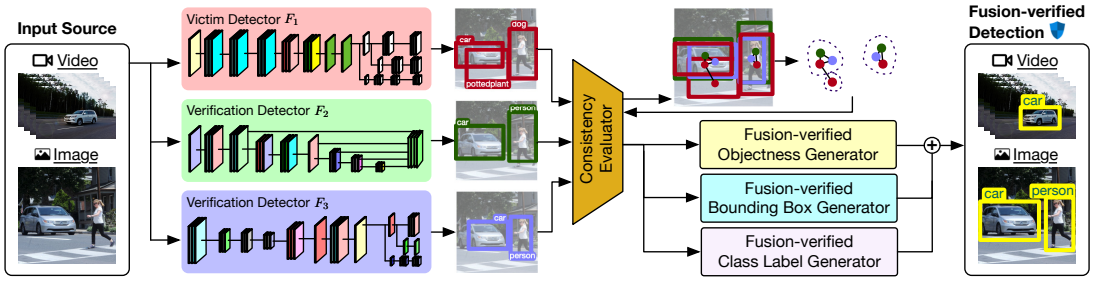

Figure 3: The detection workflow of FUSE with a team of three detectors where F_1 is under attack.

which has been shown effective compared with other options (e.g., cosine similarity) [3]. Minimizing the above angular similarity decorrelates the mistakes made on object existence predictions by different members. For the object bounding box estimation task, we define its diversity regularization component r_{bbox} as

$$r_{\text{bbox}}(\boldsymbol{x}; F_j, F_k) = \frac{1}{||\Upsilon||} \sum_{\gamma \in \Upsilon} \exp(-\text{SE}(\boldsymbol{o}^{j,\gamma}_{[\text{bbox}]}, \boldsymbol{o}^{k,\gamma}_{[\text{bbox}]})), \tag{8}$$

with $\boldsymbol{o}^{j,\gamma}_{[\text{bbox}]}$ to be the bounding box of the γ-th candidate detected by F_j, which maximizes the squared error (SE) of the normalized bounding boxes of negative candidates predicted by F_j and F_k. For the object classification task, we formulate the corresponding diversity loss component r_{cls} to be

$$r_{\text{cls}}(\boldsymbol{x}; F_j, F_k) = \frac{1}{C}\Bigg[\sum_{\gamma=1}^{C}\Big(\mathbb{1}[\gamma \in \Upsilon] \cdot \text{SIM}(\boldsymbol{o}^{j,\gamma}_{[\text{prob}]}, \boldsymbol{o}^{k,\gamma}_{[\text{prob}]}) + \mathbb{1}[\gamma \notin \Upsilon] \cdot \text{SIM}(\boldsymbol{o}^{j,\gamma\setminus\text{GT}}_{[\text{prob}]}, \boldsymbol{o}^{k,\gamma\setminus\text{GT}}_{[\text{prob}]})\Big)\Bigg], \tag{9}$$

where $\boldsymbol{o}^{j,\gamma}_{[\text{prob}]}$ is the class probability vector of the γ-th candidate estimated by F_j and $\boldsymbol{o}^{j,\gamma\setminus\text{GT}}_{[\text{prob}]}$ is the vector that excludes the ground-truth class label in the corresponding candidate. The first term diversifies the class label prediction on negative candidate objects, while the second term decorrelates the incorrect class label prediction in positive candidate objects. The final diversity regularization $\mathcal{R}$ is defined to minimize the overall negative correlation:

$$\mathcal{R}(\boldsymbol{x}; F_j, F_k) = r_{\text{obj}}(\boldsymbol{x}; F_j, F_k) + r_{\text{bbox}}(\boldsymbol{x}; F_j, F_k) + r_{\text{cls}}(\boldsymbol{x}; F_j, F_k), \tag{10}$$

or, equivalently, diversify *incorrect* decisions on both object existences, bounding boxes, and class labels simultaneously while maintaining those correct ones.

• **DivJT with Diverse Kernel Filters:** Deep neural networks for object detection comprise multiple convolutional layers. Each of them has a number of learnable kernel filters, which are crucial in extracting salient features for the recognition task. One reason that adversarial distortion is detrimental to neural networks is the error amplification throughout the victim detector's forward propagation, making small adversarial noise large enough to alter the decisions of the three prediction tasks. This diverse kernel filters based joint training aims to break this error amplification chain. Let $\boldsymbol{W}^{\ell}_j$ and $\boldsymbol{W}^{\ell}_k$ be two sets of kernel weights in the ℓ-th convolutional layer of two team members F_j and F_k respectively. We compute the similarity between each pair of kernel weights obtained from the Cartesian product of $\boldsymbol{W}^{\ell}_j$ and $\boldsymbol{W}^{\ell}_k$: $\Omega^{\ell} = \{\text{SIM}(w^{\ell}_j, w^{\ell}_k) \mid (w^{\ell}_j, w^{\ell}_k) \in \boldsymbol{W}^{\ell}_j \times \boldsymbol{W}^{\ell}_k\}$. Note that it is crucial to consider every pair of kernel filters due to the channel ordering ambiguity, and the computation of angular similarity (SIM) (see Equation 7) is based on the vectorized kernel weights. The diversity regularizer $\mathcal{R}$ is the sum of the mean and variance of Ω^{ℓ} over all layers of interest $\ell \in \Psi$:

$$\mathcal{R}(\boldsymbol{x}; F_j, F_k) = \sum_{\ell \in \Psi} \left[\text{MEAN}(\Omega^{\ell}) + \text{VAR}(\Omega^{\ell})\right]. \tag{11}$$

Minimizing both mean and variance is more robust to outliers [27] which correspond to a small number of kernel filter pairs that are highly correlated and may amplify adversarial noises.

5 OBJECT DETECTION FUSION

FUSE addresses the technical challenges in objectness fusion, bounding box fusion, and classification fusion outlined in Section 3 by introducing a suite of fusion optimization techniques to generate fusion-verified detection outputs with high resilience against deception. As depicted in Figure 3, when an input $\boldsymbol{x}$ (e.g., video frame or image) is sent to the detector F_1 under protection, FUSE intercepts the detection query and dispatches it to each of the n detectors in the chosen fusion team to perform independent object detection in parallel, and obtains n sets of detection results. Each consists of three types of information: the objectness of detected objects, their bounding boxes, and their class labels with confidence probabilities. Our detection fusion algorithm employs a consistency evaluator to identify those objects detected by different fusion member models referring to the same entity with three core fusion components: (i) fusion-verified objectness generator, which consolidates different object existences predicted by different member detectors; (ii) fusion-verified bounding box generator, which consolidates different bounding boxes detected from different member detectors of the fusion team; and (iii) fusion-verified class label generator, which consolidates different confidence probabilities resulting from different member detectors based on the overlapping of their bounding boxes. The objectness fusion employs a threshold-based approach such that those detected objects with their objectness confidence higher than the given threshold will be examined. Then, we employ a graph partitioning-based algorithm to perform the bounding box fusion by leveraging the objectness fusion and the cross-model synergy among different detectors of a fusion team, aiming to unscramble and resolve the mishmash among the results from diverse detectors of the fusion team. Finally, we introduce bounding box enhanced optimizations to accomplish the classification fusion.

Concretely, given a team $\Phi = \{F_1, ..., F_n\}$ of n detection models, we formulate detection fusion to find the robust resolution as a graph partitioning problem where we construct a weighted undirected graph $\mathcal{G} = (V, E)$ by examining all detected objects from each member detector of the fusion team, i.e., $\hat{F}_1(\boldsymbol{x}), ..., \hat{F}_n(\boldsymbol{x})$. With

objectness fusion producing $\boldsymbol{V} = \hat{F}_1(\boldsymbol{x}) \cup ... \cup \hat{F}_n(\boldsymbol{x})$, for each pair of detectors in the fusion team, we examine every pair of detected objects, one from each detector, and an edge between them is added to $\boldsymbol{E}$ with the intersection over union (IOU) score of their bounding boxes as the edge weight $w \in [0, 1]$ if their IOU score passes the system-defined IOU threshold τ_{IOU} (e.g., 50% is used in our experiments). Upon completion of the examination for all member detectors, we perform clique partitioning over $\mathcal{G}$. This reduces the problem of finding the set of highly overlapped objects detected by different member detectors to the problem of partitioning the graph $\mathcal{G}$ into cliques by maximizing the sum of edge weights. Each resultant clique consists of one or more nodes, and each node corresponds to one detected object from one detector in the fusion team. When a clique has multiple nodes, it implies that the corresponding detected objects from multiple detectors have highly overlapped bounding boxes. Thus, we perform the consistency evaluation for objects within each clique to determine the fusion-verified bounding box. Intuitively, we can generate a fusion-verified bounding box and a class label with confidence probabilities for each clique. But this naive approach to classification fusion only works for the simple cases, in which the fusion-verified bounding box indeed corresponds to one true object in the query input. For more complicated scenes, in which the query input (video frame or image) contains multiple overlapping instances of the same class label, such as multiple persons overlapping when walking over a crossroad, simply performing classification fusion based on the bounding box fusion outcome may lead to the detection error that mistakenly detects one object instead of multiple spatially overlapping objects when such overlapping is larger than the pre-defined IOU threshold. To resolve this challenge, we develop a holistic solution to employing bounding box fusion-refined classification fusion. In particular, for cliques reaching consensus on a class label, we examine every node in the clique. We first return the node achieving the highest confidence, denoted by $\boldsymbol{o}^{\max}$. Then, for each remaining node $\boldsymbol{o}$ in the clique with the same class (i.e., $\boldsymbol{o}_{[\text{cls}]} = \boldsymbol{o}^{\max}_{[\text{cls}]}$), FUSE recomputes its confidence based on the bounding box overlapping in IOU:

$$\boldsymbol{o}_{[\text{prob}]} \leftarrow \boldsymbol{o}_{[\text{prob}]}(1 - \text{IOU}(\boldsymbol{o}^{\max}_{[\text{bbox}]}, \boldsymbol{o}_{[\text{bbox}]})). \tag{12}$$

This strategy allows FUSE to detect multiple objects of the same class from a clique [1]. Algorithm 1 presents the pseudocode of the FUSE three-tier detection fusion.

Algorithm 1 Robust Object Detection Fusion

Input: $\boldsymbol{x}$: query image; Φ: fusion team; τ_{IOU}: IOU threshold
Output: $\hat{\boldsymbol{O}}$: fusion-verified detected objects
procedure FUSE($\boldsymbol{x}, \Phi, \tau_{\text{IOU}}$)
  $\hat{\boldsymbol{O}} \leftarrow \emptyset, \boldsymbol{V} \leftarrow \emptyset, \boldsymbol{E} \leftarrow \emptyset$
  for $F_i \in \Phi$ **do**
    $\boldsymbol{V} \leftarrow \boldsymbol{V} \cup \hat{F}_i(\boldsymbol{x})$
  for $\boldsymbol{o}, \boldsymbol{o}' \in \binom{\boldsymbol{V}}{2}$ **do**
    $w \leftarrow \text{IOU}(\boldsymbol{o}_{[\text{bbox}]}, \boldsymbol{o}'_{[\text{bbox}]})$
    # $\boldsymbol{o}_{[\text{det}]}$ denotes the fusion member detecting the object $\boldsymbol{o}$
    if $\boldsymbol{o}_{[\text{det}]} \neq \boldsymbol{o}'_{[\text{det}]} \wedge w \geq \tau_{\text{IOU}}$ **then**
      $\boldsymbol{E} \leftarrow \boldsymbol{E} \cup \{(\boldsymbol{o}, \boldsymbol{o}', w)\}$
  for $\boldsymbol{c} \in$ Clique-Partition$(\boldsymbol{V}, \boldsymbol{E})$ **do**
    $\eta \leftarrow$ Vote-Class-Label$(\boldsymbol{c})$
    if $\sum_{\boldsymbol{o} \in \boldsymbol{c}} \mathbb{1}\{\boldsymbol{o}_{[\text{cls}]} = \eta\} < |\Phi|/2$ **then**
      continue
    $\boldsymbol{o}^{\max} \leftarrow$ Find-Max-Confidence-Object$(\boldsymbol{c}, \eta)$
    $\hat{\boldsymbol{O}} \leftarrow \hat{\boldsymbol{O}} \cup \{\boldsymbol{o}^{\max}\}$
    for $\boldsymbol{o} \in \boldsymbol{c} \backslash \{\boldsymbol{o}^{\max}\}$ **do**
      if $\boldsymbol{o}_{[\text{cls}]} = \eta$ **then**
        $\boldsymbol{o}_{[\text{prob}]} \leftarrow \boldsymbol{o}_{[\text{prob}]}(1 - \text{IOU}(\boldsymbol{o}^{\max}_{[\text{bbox}]}, \boldsymbol{o}_{[\text{bbox}]}))$
        $\hat{\boldsymbol{O}} \leftarrow \hat{\boldsymbol{O}} \cup \{\boldsymbol{o}\}$
  return $\hat{\boldsymbol{O}}$

6 ROBUSTNESS ANALYSIS

We formally analyze the robustness of FUSE in terms of the adversarial vulnerability of object detection fusion. Assume that an object detector $F_{\boldsymbol{\theta}}$ is well-trained by minimizing the standard detection loss in Equation 1, which measures the prediction error on objectness, bounding boxes, and class labels of objects. While the exact formulation of adversarial attacks may vary, they share the same objective: deceiving the victim detector to return erroneous detection results. Hence, an adversarial example $\boldsymbol{x} + \boldsymbol{\delta}$ generated by solving the optimization problem in Equation 3 causes the standard detection loss (prediction errors) of $F_{\boldsymbol{\theta}}$ to increase, i.e.,

$$\mathcal{L}(F_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{O}) < \mathcal{L}(F_{\boldsymbol{\theta}}(\boldsymbol{x} + \boldsymbol{\delta}), \boldsymbol{O}). \tag{13}$$

Intuitively, an object detection system is more vulnerable if just a tiny perturbation to input can cause a drastic change in output [28]. Based on this idea, we can formulate the adversarial vulnerability of an object detector as follows.

Definition 6.1 (Adversarial Vulnerability). Given an object detector $F_{\boldsymbol{\theta}}$, an input $\boldsymbol{x}$, and an attack budget ϵ in ℓ_p-norm, the adversarial vulnerability can be captured by how much the standard detection loss of $F_{\boldsymbol{\theta}}$ can be varied by injecting arbitrary input perturbation $\boldsymbol{\delta}$ inside an ℓ_p-norm ball with a radius ϵ centered at $\boldsymbol{x}$:

$$\Delta\mathcal{L}(\boldsymbol{x}, \boldsymbol{O}; F_{\boldsymbol{\theta}}, \epsilon) = \max_{\|\boldsymbol{\delta}\|_p < \epsilon} |\mathcal{L}(F_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{O}) - \mathcal{L}(F_{\boldsymbol{\theta}}(\boldsymbol{x} + \boldsymbol{\delta}), \boldsymbol{O})|. \tag{14}$$

By employing an army of carefully constructed object detection models to safeguard an object detection system by robust fusion, we conjecture that one can reduce the overall adversarial vulnerability or, equivalently, offer strong robustness against deception. To formally verify robust detection through model fusion, we define the overall detection loss of an n-model fusion team $\Phi = \{F_1, \ldots, F_n\}$ given an input $\boldsymbol{x}$ with ground truth objects $\boldsymbol{O}$ to be

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{O}; \Phi) = \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(F_i(\boldsymbol{x}), \boldsymbol{O}). \tag{15}$$

Then, we can reformulate adversarial vulnerability (Equation 14) as

$$\Delta\mathcal{L}(\boldsymbol{x}, \boldsymbol{O}; \Phi, \epsilon) = \max_{\|\boldsymbol{\delta}\|_p < \epsilon} |\mathcal{L}(\boldsymbol{x}, \boldsymbol{O}; \Phi) - \mathcal{L}(\boldsymbol{x} + \boldsymbol{\delta}, \boldsymbol{O}; \Phi)|, \tag{16}$$

which captures the changes to the overall detection loss of the fusion team caused by input perturbation bounded by ϵ in ℓ_p-norm. Since the adversarial perturbation $\boldsymbol{\delta}$ is often constrained to be human-imperceptible (i.e., $\epsilon \to 0$), a first-order Taylor expansion on Equation 16 in ϵ gives

$$\Delta\mathcal{L}(\boldsymbol{x}, \boldsymbol{O}; \Phi, \epsilon) \approx \max_{\|\boldsymbol{\delta}\|_p < \epsilon} |\nabla_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{O}; \Phi) \cdot \boldsymbol{\delta}|. \tag{17}$$

The object detection system is vulnerable (or robust) if, on average, $\Delta\mathcal{L}(\boldsymbol{x}, \boldsymbol{O}; \Phi, \epsilon)$ is large (or small) [20, 28]. Using the definition of dual norm, the adversarial vulnerability over the dataset $\mathcal{D}$ is

$$\mathop{\mathbb{E}}_{(\boldsymbol{x}, \boldsymbol{O}) \sim \mathcal{D}} [\Delta\mathcal{L}(\boldsymbol{x}, \boldsymbol{O}; \Phi, \epsilon)] \propto \mathop{\mathbb{E}}_{(\boldsymbol{x}, \boldsymbol{O}) \sim \mathcal{D}} [||\nabla_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{O}; \Phi)||_q], \tag{18}$$

where q is the dual norm of p, which satisfies $\frac{1}{p} + \frac{1}{q} = 1$ and $1 \leq p \leq \infty$. Expanding the above yields the following theorem on the adversarial vulnerability of an object detection system safeguarded by model fusion (the proof is provided in Appendix A).

Theorem 6.2 (Adversarial Vulnerability of Model Fusion). *Given a team Φ of n object detection models, an attack budget ϵ in ℓ_p-norm, and the dataset $\mathcal{D}$, the adversarial vulnerability of a model fusion-protected object detection system is proportional to*

$$\sqrt{\frac{1}{3n} + \frac{2}{9n^2}\left[\mathrm{Cov}_{\mathrm{intra}}(\boldsymbol{x}, \boldsymbol{O}; \Phi) + \mathrm{Cov}_{\mathrm{inter}}(\boldsymbol{x}, \boldsymbol{O}; \Phi)\right]}. \tag{19}$$

$\mathrm{Cov}_{\mathrm{intra}}(\boldsymbol{x}, \boldsymbol{O}; \Phi)$ and $\mathrm{Cov}_{\mathrm{inter}}(\boldsymbol{x}, \boldsymbol{O}; \Phi)$ are two terms capturing the covariance of gradients across tasks and models:

$$\begin{aligned}\mathrm{Cov}_{\mathrm{intra}}(\boldsymbol{x}, \boldsymbol{O}; \Phi) = \sum_{i=2}^{n}\sum_{j=1}^{i-1}\Bigg[&\frac{\mathrm{Cov}(\nabla_{\boldsymbol{x}}\mathcal{L}^i_{\mathrm{obj}}, \nabla_{\boldsymbol{x}}\mathcal{L}^j_{\mathrm{obj}})}{\mathrm{Cov}(\nabla_{\boldsymbol{x}}\mathcal{L}^i_{\mathrm{obj}}, \nabla_{\boldsymbol{x}}\mathcal{L}^i_{\mathrm{obj}})} \\ &+ \frac{\mathrm{Cov}(\nabla_{\boldsymbol{x}}\mathcal{L}^i_{\mathrm{bbox}}, \nabla_{\boldsymbol{x}}\mathcal{L}^j_{\mathrm{bbox}})}{\mathrm{Cov}(\nabla_{\boldsymbol{x}}\mathcal{L}^i_{\mathrm{bbox}}, \nabla_{\boldsymbol{x}}\mathcal{L}^i_{\mathrm{bbox}})} + \frac{\mathrm{Cov}(\nabla_{\boldsymbol{x}}\mathcal{L}^i_{\mathrm{cls}}, \nabla_{\boldsymbol{x}}\mathcal{L}^j_{\mathrm{cls}})}{\mathrm{Cov}(\nabla_{\boldsymbol{x}}\mathcal{L}^i_{\mathrm{cls}}, \nabla_{\boldsymbol{x}}\mathcal{L}^i_{\mathrm{cls}})}\Bigg],\end{aligned} \tag{20}$$

$$\begin{aligned}&\mathrm{Cov}_{\mathrm{inter}}(\boldsymbol{x}, \boldsymbol{O}; \Phi) \\ &= \sum_{i=2}^{n}\sum_{j=1}^{i-1}\Bigg[\frac{\mathrm{Cov}(\nabla_{\boldsymbol{x}}\mathcal{L}^i_{\mathrm{obj}}, \nabla_{\boldsymbol{x}}\mathcal{L}^j_{\mathrm{bbox}})}{\mathrm{Cov}(\nabla_{\boldsymbol{x}}\mathcal{L}^i_{\mathrm{obj}}, \nabla_{\boldsymbol{x}}\mathcal{L}^i_{\mathrm{obj}})} + \frac{\mathrm{Cov}(\nabla_{\boldsymbol{x}}\mathcal{L}^i_{\mathrm{obj}}, \nabla_{\boldsymbol{x}}\mathcal{L}^j_{\mathrm{cls}})}{\mathrm{Cov}(\nabla_{\boldsymbol{x}}\mathcal{L}^i_{\mathrm{obj}}, \nabla_{\boldsymbol{x}}\mathcal{L}^i_{\mathrm{obj}})} \\ &\quad + \frac{\mathrm{Cov}(\nabla_{\boldsymbol{x}}\mathcal{L}^i_{\mathrm{bbox}}, \nabla_{\boldsymbol{x}}\mathcal{L}^j_{\mathrm{obj}})}{\mathrm{Cov}(\nabla_{\boldsymbol{x}}\mathcal{L}^i_{\mathrm{bbox}}, \nabla_{\boldsymbol{x}}\mathcal{L}^i_{\mathrm{bbox}})} + \frac{\mathrm{Cov}(\nabla_{\boldsymbol{x}}\mathcal{L}^i_{\mathrm{bbox}}, \nabla_{\boldsymbol{x}}\mathcal{L}^j_{\mathrm{cls}})}{\mathrm{Cov}(\nabla_{\boldsymbol{x}}\mathcal{L}^i_{\mathrm{bbox}}, \nabla_{\boldsymbol{x}}\mathcal{L}^i_{\mathrm{bbox}})} \\ &\quad + \frac{\mathrm{Cov}(\nabla_{\boldsymbol{x}}\mathcal{L}^i_{\mathrm{cls}}, \nabla_{\boldsymbol{x}}\mathcal{L}^j_{\mathrm{obj}})}{\mathrm{Cov}(\nabla_{\boldsymbol{x}}\mathcal{L}^i_{\mathrm{cls}}, \nabla_{\boldsymbol{x}}\mathcal{L}^i_{\mathrm{cls}})} + \frac{\mathrm{Cov}(\nabla_{\boldsymbol{x}}\mathcal{L}^i_{\mathrm{cls}}, \nabla_{\boldsymbol{x}}\mathcal{L}^j_{\mathrm{bbox}})}{\mathrm{Cov}(\nabla_{\boldsymbol{x}}\mathcal{L}^i_{\mathrm{cls}}, \nabla_{\boldsymbol{x}}\mathcal{L}^i_{\mathrm{cls}})}\Bigg] \\ &\quad + \sum_{i=1}^{n}\Bigg[\frac{\mathrm{Cov}(\nabla_{\boldsymbol{x}}\mathcal{L}^i_{\mathrm{bbox}}, \nabla_{\boldsymbol{x}}\mathcal{L}^i_{\mathrm{obj}})}{\mathrm{Cov}(\nabla_{\boldsymbol{x}}\mathcal{L}^i_{\mathrm{bbox}}, \nabla_{\boldsymbol{x}}\mathcal{L}^i_{\mathrm{bbox}})} + \frac{\mathrm{Cov}(\nabla_{\boldsymbol{x}}\mathcal{L}^i_{\mathrm{cls}}, \nabla_{\boldsymbol{x}}\mathcal{L}^i_{\mathrm{obj}})}{\mathrm{Cov}(\nabla_{\boldsymbol{x}}\mathcal{L}^i_{\mathrm{cls}}, \nabla_{\boldsymbol{x}}\mathcal{L}^i_{\mathrm{cls}})} \\ &\quad + \frac{\mathrm{Cov}(\nabla_{\boldsymbol{x}}\mathcal{L}^i_{\mathrm{cls}}, \nabla_{\boldsymbol{x}}\mathcal{L}^i_{\mathrm{bbox}})}{\mathrm{Cov}(\nabla_{\boldsymbol{x}}\mathcal{L}^i_{\mathrm{cls}}, \nabla_{\boldsymbol{x}}\mathcal{L}^i_{\mathrm{cls}})}\Bigg],\end{aligned} \tag{21}$$

where $\mathcal{L}^i_{\mathrm{task}} = \mathcal{L}_{\mathrm{task}}(F_i(\boldsymbol{x}), \boldsymbol{O})$ for $\mathrm{task} \in \{\mathrm{obj}, \mathrm{bbox}, \mathrm{cls}\}$.

Theorem 6.2 offers three important insights on model fusion. First, the adversarial vulnerability of an object detection system reduces when multiple object detectors are jointly operated in the decision-making process. The improvement in adversarial robustness can be explained as the adversarial perturbations required to attack multiple models cancel each other out [28]. Second, the adversarial vulnerability also depends on the covariance of gradients between tasks and fusion members. The first term $\mathrm{Cov}_{\mathrm{intra}}(\boldsymbol{x}, \boldsymbol{O}; \Phi)$ indicates that to reduce adversarial vulnerability, the gradients of a task (e.g., objectness prediction) of one model should have a minimal covariance with another model in the fusion team. Interestingly, the second term $\mathrm{Cov}_{\mathrm{inter}}(\boldsymbol{x}, \boldsymbol{O}; \Phi)$ suggests that the gradients of one task should also be decorrelated with other tasks. This is because adversarial attacks on object detection systems typically manipulate multiple components in the standard detection loss (Equation 1). For example, RAP [13] attacks the objectness component and bounding box component such that the victim cannot correctly recognize the existence of objects, and even if objects are detected, their bounding boxes will be erroneous. By decorrelating the gradient directions of different tasks, the adversary cannot easily find a single perturbation that successfully deceives multiple tasks at the same time. Finally, the notion of covariance of gradients can be utilized for strategic teaming to cherrypick team members that deliver the strongest robustness (i.e., minimum overall covariance) by joint force. We provide formal analysis of FUSE from the perspective of adversarial perturbation lower bound in Appendix B.

Model Identifier	Detection Algorithm	Neural Configuration	mAP (Benign)
F_1	FRCNN	VGG16-Standard	67.37
F_2	YOLOv3-D	Darknet53-Standard	83.43
F_3	YOLOv3-M	MobileNetV1-Standard	71.84
F_4	SSD300	VGG16-Standard	76.11
F_5	SSD512	VGG16-Standard	79.83
F_6	FRCNN	VGG16-DivJT-DetLoss	64.77
F_7	FRCNN	VGG16-DivJT-DetLoss	65.58
F_8	FRCNN	VGG16-DivJT-KF	64.88
F_9	FRCNN	VGG16-DivJT-KF	64.82
F_{10}	FRCNN	VGG16-RandInit	67.51
F_{11}	FRCNN	VGG16-RandInit	67.32

Table 2: A summary of models in experimental evaluation.

7 EXPERIMENTS

We conduct extensive experiments to empirically analyze the effectiveness of FUSE against various deception attacks on two benchmark datasets: PASCAL VOC [7] and INRIA [6]. VOC consists of images containing 20 classes of objects (e.g., bicycle, chair, and dog), and INRIA focuses on pedestrian detection. All results reported are measured on their entire test set (4,952 images for VOC and 288 images for INRIA). We use the standard performance metric, called mean average precision (mAP) [7], to capture the detection quality of an object detection system. We consider three representative families of object detection algorithms: Faster R-CNN (FRCNN) [19], YOLOv3 [18], and SSD [14] with different neural architectures as backbones. Table 2 summarizes the models involved in our experiments, where F_1 to F_5 are trained independently with the standard detection loss (Equation 1). Using diversity joint training (DivJT), we further train two FRCNN models (F_6 and F_7) with our diversified detection loss (Equation 10) and two models (F_8 and F_9) with our diversified kernel filters (Equation 11). As a baseline training strategy, we provide two models (F_{10} and F_{11}) trained with random initialization of model weights without regularization promoting model diversity. Additional details are provided in Appendix C.

7.1 Robustness Analysis

Our empirical analysis begins with examining the robustness of FUSE protecting FRCNN (i.e., F_1 in Table 2 as the victim). With strategic teaming on models summarized in Table 2, we construct ten fusion teams by pairing up the victim with two additional models as our experimental studies show a fusion of three diverse detectors is already sufficient for strong robustness, even though we proved in Section 6 that a larger team can lead to an even stronger defense. Four state-of-the-art attack techniques, including TOG [5], UEA [24], RAP [13], and DAG [26], are exploited to launch both untargeted random attacks and targeted specificity attacks with object vanishing, fabrication, and mislabeling effects on the victim. Table 3 summarizes the results on VOC with benign mAP (3rd column) and mAP under attacks (4th to 10th columns) where we compare FUSE (3rd and 4th rows) with the victim F_1 with no protection (1st row). Since defenses proposed for single-task learners (i.e., image classifiers) are not applicable to object detection systems with multi-task objectives, we include the adversarial detection

Object Detector		mAP (Benign)	Untargeted Random Attacks				Targeted Specificity Attacks		
			TOG	UEA	RAP	DAG	TOG-v	TOG-f	TOG-m
F_1: FRCNN (Victim)		67.37	2.64	18.07	4.78	3.56	0.14	1.24	2.14
Adversarial Detection Training [29]		35.99 (0.53×)	34.07 (12.91×)	17.67 (0.98×)	35.60 (7.45×)	35.58 (9.99×)	34.03 (243.07×)	34.18 (27.56×)	34.81 (16.27×)
FUSE	Most Diverse $\Phi = \{F_1, F_2, F_5\}$	**85.95*** **(1.28×)**	**81.46 (30.86×)**	**53.52 (2.96×)**	**81.96 (17.15×)**	**84.60 (23.76×)**	**81.23 (580.21×)**	**81.91 (66.06×)**	**82.29 (38.45×)**
	Average (Std)	**80.82 (3.62) (1.20×)**	76.14 (4.36) (28.84×)	48.15 (3.53) (2.66×)	77.39 (3.49) (16.19×)	80.21 (3.05) (22.53×)	77.23 (2.51) (551.64×)	77.20 (3.17) (62.26×)	77.75 (2.93) (36.33×)

*The best-performing member, F_2, achieves an mAP of 83.43%.

Table 3: Evaluation of FUSE with FRCNN as the victim under seven attacks compared with adversarial detection training.

training approach (2nd row) [29], the only recent proposal to the best of our knowledge, for baseline comparisons.

We make four observations. First, all seven attacks significantly bring down the mAP of the victim detector with no protection from 67.37% to 0.14~18.07% (see the 1st row). Second, FUSE maintains high mAP on benign inputs (boldfaced in the 3rd column), and for the most diverse fusion team with F_1: FRCNN (victim), F_2: YOLOv3-D, and F_5: SSD512 (3rd row), FUSE achieves an mAP of 85.95% on benign inputs, which is 2.5% higher than the best-performing member (i.e., F_2: YOLOv3-D with an mAP of 83.43%). Third, FUSE significantly improves the robustness of object detection by boosting the mAP under all seven attacks, delivering high mAPs of 53.52~84.60% (i.e., 2.96× to 580.21× improvement over the mAP of the victim detector: 0.14~18.07%). The 4th row shows that all ten diverse fusion teams can empower the object detection system with strong robustness against all seven attacks, achieving a high average mAP of 48.15~80.21% with a small standard deviation. This is because even if an adversarial input may succeed in partially deceiving both the victim and some other members, FUSE can still produce correct detection by analyzing objectness, bounding boxes, and class labels of objects. In comparison, the adversarial detection training approach recently proposed in [29] has a low mAP of 17.67~35.60% under seven deception attacks. One of the reasons for its low resilience is due to its low benign mAP of 35.99%, compared to the mAP of 67.37% of the victim detector F_1 (0.53× performance degradation). Finally, it is worth noting that the adversarial detection training approach [29] fails to defend the UEA attack with an mAP of 17.67%, which is worse than the mAP of 18.07% by the victim detector F_1 without protection. In comparison, the victim detector under FUSE protection can achieve the high mAP of 53.52%, delivering a 2.96× mAP improvement (18.07% → 53.52%). One reason that UEA is relatively hard to mitigate compared with the other six attacks can be explained by its excessive amount of adversarial distortion. Figure 4 reports the imperceptibility of adversarial perturbations generated by all seven attacks in terms of the structural similarity (SSIM) [23] between the benign input and its adversarial counterpart. RAP [13] and DAG [26] both have an SSIM close to 1.00, meaning that they are almost completely imperceptible. In contrast, UEA generates adversarial examples with significantly more perceptible perturbations with a low SSIM of 0.65.

Robustness: YOLOv3 as the Victim. We next consider the best-performing model from Table 2, i.e., F_2: YOLOv3-D, to be the victim under TOG attacks [5] since other attacks are not applicable to one-shot detectors. Table 4 reports the results where the victim with an mAP of 83.43% given benign inputs suffers severely from

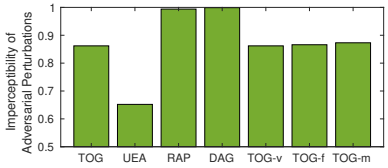

Figure 4: Imperceptibility of adversarial perturbations (1.00 means completely imperceptible).

Object Detector	mAP (Benign)	mAP (Under Attacks)			
		TOG	TOG-v	TOG-f	TOG-m
F_2: YOLOv3-D (Victim)	83.43	0.56	1.24	0.25	3.15
FUSE	82.28 (0.99×)	**73.02 (130.39×)**	**75.88 (61.19×)**	**76.83 (307.32×)**	**76.15 (24.17×)**

Table 4: Evaluation of FUSE with the best-performing model, YOLOv3-D, as the victim with $\Phi = \{F_1, F_2, F_5\}$.

Object Detector	mAP (Benign)	mAP (Under Attacks)			
		TOG	UEA	RAP	DAG
F_1: FRCNN (Victim)	67.37	2.64	18.07	4.78	3.56
(a) Objectness Fusion	77.43 (1.15×)	2.68 (1.02×)	45.92 (2.54×)	75.85 (15.87×)	78.59 (22.08×)
(b) Objectness & Bounding Box Fusion	81.31 (1.21×)	76.93 (29.14×)	46.21 (2.56×)	77.01 (16.11×)	81.01 (22.76×)
(c) Objectness & Bounding Box & Classification Fusion	**85.95 (1.28×)**	**81.46 (30.86×)**	**53.52 (2.96×)**	**81.96 (17.15×)**	**84.60 (23.76×)**

Table 5: Ablation studies of FUSE.

those white-box attacks with low mAPs ranging from 0.25% to 3.15%. Even though the best-performing member in the fusion team of $\Phi = \{F_1, F_2, F_5\}$ is under attack, FUSE can still deliver highly competitive mAP in benign scenarios and effectively mitigate adversarial attacks, boosting the mAP under attacks to 73.02~76.83%, which is 24.17× to 307.32× improvement over the victim. Hence, FUSE does not rely on a single member in the team that performs comparatively well to mitigate adversarial attacks but leverages multiple diverse models by joint force against deception.

Ablation Studies. We conduct ablation studies in Table 5 to understand the contributions of the key components to the robustness of FUSE by comparing three versions: (a) *objectness fusion* combining detection results from all member models with a redundancy filter implemented by non-maximum suppression; (b) *objectness & bounding box fusion* combining detection results from all members using our graph partitioning approach to find robust resolution; and (c) *objectness & bounding box & classification fusion* incorporating Equation 12 for classification fusion (i.e., the complete FUSE).

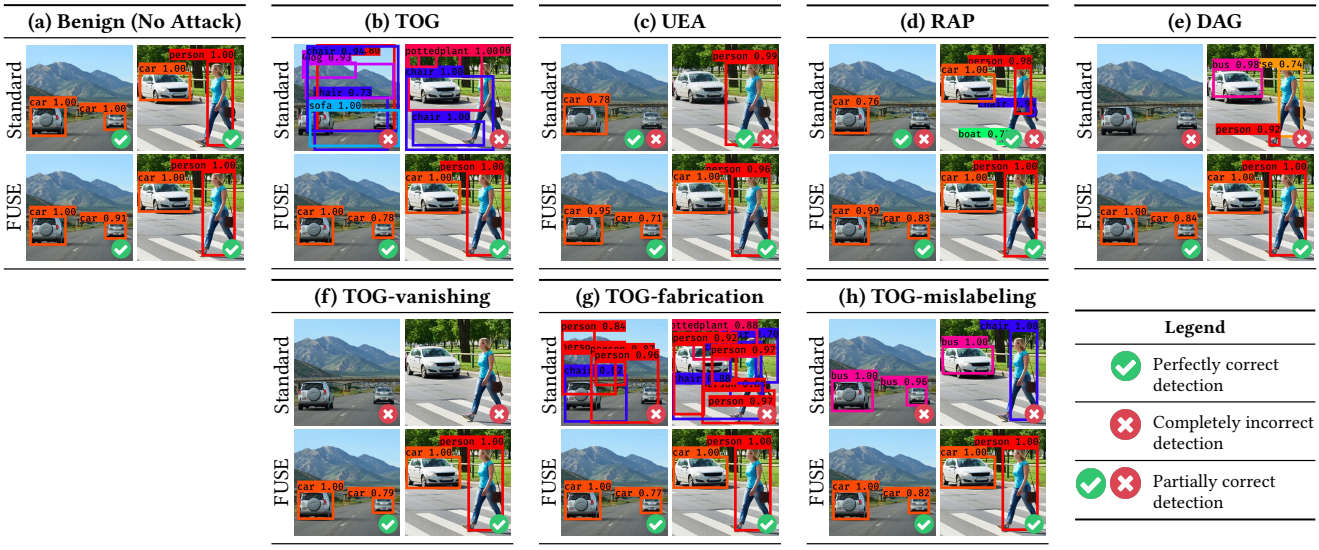

Table 6: Detection visualization with the 1st row using the standard FRCNN without protection and the 2nd row having FUSE protection.

Table 5 reports the results on benign scenarios and four attacks (TOG, UEA, RAP, and DAG) with FRCNN as the victim.

We make three observations. First, the objectness fusion is only effective for three out of four attacks. This is because combining detection results simply through a redundancy filter cannot remove false positives where TOG attack tends to generate (it can merely improve the mAP of the victim from 2.64% to 2.68% under TOG). Second, finding robust resolution of objects detected by different members allows the objectness & bounding box fusion approach to locate highly suspicious objects to be pruned. Finally, by further improving the fusion capability with classification fusion, FUSE shows an additional boost in robustness against all attacks (2.96~30.86× improvement over the victim) and in benign scenarios (1.28× improvement over the victim).

Visualization. To demonstrate the mitigation effectiveness of FUSE, Table 6 gives visualizations of two examples under benign (a) and adversarial settings (b-h). With the standard FRCNN perfectly detecting all objects with 100% confidence (see the 1st row in Table 6(a)), we can observe that all seven attacks successfully deceive the victim with different adverse effects. For instance, TOG in Table 6(b) misleads the victim to detect multiple false objects, and UEA in Table 6(c) fools the victim to misdetect one car in each example. Even with such highly diverse malicious effects incurred by different attack methods, FUSE (2nd row) can consistently repair all adversarial examples, restore the operation of the object detection system, and maintain high performance on benign inputs.

7.2 FUSE Against Adversarial Patches

To study the effectiveness of FUSE against adversarial patches, we exploit TOG [5] to train patches of size (64×64) with object vanishing and mislabeling effects. Following [21], the INRIA dataset for pedestrian detection [6] is exploited with YOLOv3-D as the victim. The patch is placed at the center of the person detected on the benign test image sent to the victim to generate its adversarial example. Table 7 visualizes an example test image under no attack (1st column), the TOG-vanishing patch attack (2nd column), and the TOG-mislabeling patch attack (3rd column). For the TOG-vanishing

Table 7: An image attacked by adversarial patches using TOG with object vanishing (2nd column) and mislabeling (3rd column) effects.

patch, FUSE successfully protects the victim by drastically reducing the percentage of vanished pedestrians in the entire INRIA test set from 97.35% to 15.30%. For TOG-mislabeling patches, we train 19 adversarial patches, aiming at deceiving the victim to mislabel the person with the patch as one of the 19 different target classes on VOC (e.g., the one in the 3rd column in Table 7 fools the victim to mislabel any person attached with the patch to be a chair). Even under such highly aggressive adversarial patches for fooling the standard victim without protection, led to 80% (on average) of detected pedestrians being purposefully mislabeled, FUSE can mitigate such adversarial patch attacks with strong robustness and significantly reduce the percentage of mislabeled pedestrians to zero in most cases. The adversarial patches are by design to simulate the physical attack digitally for systematic evaluation [21]. These experiments show that FUSE holds great potential to provide strong robustness against physical attacks and visual distortions due to other real-world factors, e.g., the transmission distortion, the angle or condition of taking the image or video.

7.3 Execution Efficiency

We further investigate the execution efficiency of FUSE. Table 8 summarizes the detection time and frame-per-second (FPS) measurements on both the standard FRCNN detector without protection and with FUSE protection. With the highly parallelizable design,

Object Detector	Detection Time (s)	FPS
Standard	0.1399	7.15
FUSE	0.1455	6.87

Table 8: The detection time and FPS comparisons (FRCNN).

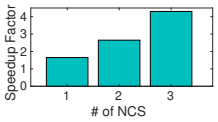

Figure 5: FPS speedup using AI accelerators.

Group 1: Tiny Objects	Group 2: Poor Lighting

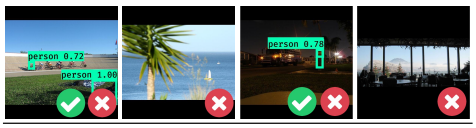

Table 9: Two groups of challenging scenarios.

FUSE incurs a low overhead to the detector being protected because the graph constructed by FUSE for each query image is tiny.

Adversarial attacks are particularly detrimental in applications running on the edge as the data captured by sensors will be immediately processed locally to trigger downstream operations. It is of utmost importance to safeguard those systems perceiving malicious inputs and propagating erroneous decisions along the application pipeline. In light of this, we extend FUSE for edge deployments with AI accelerators such as the Intel Neural Compute Stick (NCS) [11]. Multiple detection models can be run efficiently within each device with low power consumption and low cost. The FPS speedup reported in Figure 5 demonstrates the capability of FUSE in efficiently enhancing the robustness of such Edge-AI environments.

7.4 Challenging Cases

Although FUSE offers robust performance, it has limitations. Table 9 shows two common scenarios where FUSE faces challenges to correctly detect objects: (1) tiny objects, e.g., the image with people and bicycles in the 1st column under Group 1, and (2) images taken with a poor lighting condition, e.g., the bus cannot be detected in the 1st image under Group 2. One way to address those scenarios could be to incorporate scale-aware neural architectures through trident blocks [12] or tiling [16] in our generation process of diverse models, forming fusion teams with detectors specialized in different settings. We could also conduct adaptive image preprocessing and attempt to repair the poor lighting conditions with automatic color enhancement [17]. It would also be interesting to explore the feasibility of selectively involving human-in-the-loop.

8 CONCLUSION

We have presented FUSE, a robust object detection fusion approach for strong robustness and survivability against deception to object detection systems by a growing number of digital or physical adversarial attacks. The paper makes three original contributions. First, we present diversity-enhanced fusion teaming algorithms for producing diverse fusion detectors. Second, we introduce a three-tier detection fusion framework to safeguard the victim detector through three mutually reinforcing components: objectness fusion, bounding box fusion, and classification fusion. Third, we formally analyze the robustness enhancement of our object detection fusion in terms of adversarial vulnerability. Extensive experiments on eleven detectors and two benchmark datasets validate the strong robustness of FUSE against various deception attacks while maintaining high performance on benign inputs.

ACKNOWLEDGMENTS

This research is partially sponsored by the National Science Foundation CISE grants 2038029, 2026945, 1564097, an IBM faculty award, and a Cisco grant on Edge Computing. The first author acknowledges the Croucher Scholarship for Doctoral Study from the Croucher Foundation. The source code of FUSE is available at https://github.com/git-disl/FUSE.

REFERENCES

[1] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. 2017. Soft-NMS–improving object detection with one line of code. In *ICCV*.
[2] Gavin Brown. 2004. *Diversity in neural network ensembles*. Ph.D. Dissertation.
[3] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal Sentence Encoder. In *EMNLP*.
[4] Ka-Ho Chow, Ling Liu, Mehmet Emre Gursoy, Stacey Truex, Wenqi Wei, and Yanzhao Wu. 2020. Understanding Object Detection Through an Adversarial Lens. In *ESORICS*.
[5] Ka-Ho Chow, Ling Liu, Margaret Loper, Juhyun Bae, Mehmet Emre Gursoy, Stacey Truex, Wenqi Wei, and Yanzhao Wu. 2020. Adversarial Objectness Gradient Attacks on Real-time Object Detection Systems. In *TPS*. IEEE.
[6] Navneet Dalal and Bill Triggs. 2005. Histograms of oriented gradients for human detection. In *CVPR*.
[7] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. 2015. The pascal visual object classes challenge: A retrospective. *IJCV* (2015).
[8] Vandit Gajjar, Ayesha Gurnani, and Yash Khandhediya. 2017. Human detection and tracking for video surveillance: A cognitive science approach. In *ICCV*.
[9] Ross Girshick. 2015. Fast r-cnn. In *CVPR*.
[10] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *ICLR*.
[11] Intel. 2019. Intel Neural Compute Stick 2. https://software.intel.com/content/www/us/en/develop/hardware/neural-compute-stick.html
[12] Yanghao Li, Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. 2019. Scale-aware trident networks for object detection. In *ICCV*.
[13] Yuezun Li, Daniel Tian, Xiao Bian, Siwei Lyu, et al. 2018. Robust adversarial perturbation on deep proposal-based models. In *BMVC*.
[14] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. 2016. SSD: Single shot multibox detector. In *ECCV*.
[15] Xin Liu, Huanrui Yang, Ziwei Liu, Linghao Song, Hai Li, and Yiran Chen. 2019. DPatch: An Adversarial Patch Attack on Object Detectors. In *SafeAI*.
[16] F Ozge Unel, Burak O Ozkalayci, and Cevahir Cigla. 2019. The power of tiling for small object detection. In *CVPRW*.
[17] Jongchan Park, Joon-Young Lee, Donggeun Yoo, and In So Kweon. 2018. Distort-and-recover: Color enhancement using deep reinforcement learning. In *CVPR*.
[18] Joseph Redmon and Ali Farhadi. 2018. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767* (2018).
[19] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*.
[20] Aman Sinha, Hongseok Namkoong, Riccardo Volpi, and John Duchi. 2018. Certifying some distributional robustness with principled adversarial training. In *ICLR*.
[21] Simen Thys, Wiebe Van Ranst, and Toon Goedemé. 2019. Fooling automated surveillance cameras: adversarial patches to attack person detection. In *CVPRW*.
[22] Paul Viola and Michael Jones. 2001. Rapid object detection using a boosted cascade of simple features. In *CVPR*.
[23] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *TIP* (2004).
[24] Xingxing Wei, Siyuan Liang, Ning Chen, and Xiaochun Cao. 2019. Transferable adversarial attacks for image and video object detection. In *IJCAI*.
[25] Tsui-Wei Weng, Huan Zhang, Pin-Yu Chen, Jinfeng Yi, Dong Su, Yupeng Gao, Cho-Jui Hsieh, and Luca Daniel. 2018. Evaluating the robustness of neural networks: An extreme value theory approach. In *ICLR*.
[26] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Yuyin Zhou, Lingxi Xie, and Alan Yuille. 2017. Adversarial examples for semantic segmentation and object detection. In *ICCV*.
[27] Pengtao Xie, Yuntian Deng, and Eric Xing. 2015. Diversifying restricted boltzmann machine for document modeling. In *SIGKDD*.
[28] Junfeng Yang and Carl Vondrick. 2020. Multitask learning strengthens adversarial robustness. In *ECCV*.
[29] Haichao Zhang and Jianyu Wang. 2019. Towards adversarially robust object detection. In *ICCV*.

A PROOF OF THEOREM 6.2

We define the overall output loss of an n-model fusion team $\Phi = \{F_1, \ldots, F_n\}$ given an input $\boldsymbol{x}$ to be

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{O}; \Phi) = \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(F_i(\boldsymbol{x}), \boldsymbol{O}). \tag{22}$$

Then, we can quantify adversarial vulnerability to be the maximum change caused by any perturbation $\boldsymbol{\delta}$ to $\boldsymbol{x}$ with an upper bound of ϵ in ℓ_p-norm, i.e.,

$$\Delta\mathcal{L}(\boldsymbol{x}, \boldsymbol{O}; \Phi, \epsilon) = \max_{\|\boldsymbol{\delta}\|_p < \epsilon} |\mathcal{L}(\boldsymbol{x}, \boldsymbol{O}; \Phi) - \mathcal{L}(\boldsymbol{x} + \boldsymbol{\delta}, \boldsymbol{O}; \Phi)|. \tag{23}$$

Since the adversarial perturbation $\boldsymbol{\delta}$ is often constrained to be human-imperceptible (i.e., $\epsilon \to 0$), a first-order Taylor expansion on Equation 23 in ϵ gives

$$\Delta\mathcal{L}(\boldsymbol{x}, \boldsymbol{O}; \Phi, \epsilon) \approx \max_{||\boldsymbol{\delta}||_p < \epsilon} |\nabla_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{O}; \Phi) \cdot \boldsymbol{\delta}|, \tag{24}$$

where $\nabla_{\boldsymbol{x}}$ is a partial derivative operator with respect to $\boldsymbol{x}$. The FUSE-protected object detection system is vulnerable (or robust) if, on average over $\boldsymbol{x}$, $\Delta\mathcal{L}(\boldsymbol{x}, \boldsymbol{O}; \Phi, \epsilon)$ is large (or small) [10, 20]. Using the definition of dual norm, the adversarial vulnerability over the dataset $\mathcal{D}$ is

$$\begin{aligned}
&\mathop{\mathbb{E}}_{(\boldsymbol{x},\boldsymbol{O})\sim\mathcal{D}} [\Delta\mathcal{L}(\boldsymbol{x}, \boldsymbol{O}; \Phi, \epsilon)] \\
&\approx \mathop{\mathbb{E}}_{(\boldsymbol{x},\boldsymbol{O})\sim\mathcal{D}} [\max_{||\boldsymbol{\delta}||_p < \epsilon} |\nabla_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{O}; \Phi) \cdot \boldsymbol{\delta}|] \\
&\approx \mathop{\mathbb{E}}_{(\boldsymbol{x},\boldsymbol{O})\sim\mathcal{D}} [||\nabla_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{O}; \Phi)||_q] \cdot ||\boldsymbol{\delta}||_p \\
&\propto \mathop{\mathbb{E}}_{(\boldsymbol{x},\boldsymbol{O})\sim\mathcal{D}} [||\nabla_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{O}; \Phi)||_q],
\end{aligned} \tag{25}$$

where q is the dual norm of p, which satisfies $\frac{1}{p} + \frac{1}{q} = 1$ and $1 \le p \le \infty$.

Following [28], we consider ℓ_2-norm and set $p = q = 2$ in this derivation. To construct the adversarial vulnerability, we first obtain the expected value of the square of ℓ_2-norm of gradients using Equation 22, i.e.,

$$\begin{aligned}
&\mathop{\mathbb{E}}_{(\boldsymbol{x},\boldsymbol{O})\sim\mathcal{D}} [||\nabla_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{O}; \Phi)||_2^2] \\
&= \mathop{\mathbb{E}}_{(\boldsymbol{x},\boldsymbol{O})\sim\mathcal{D}} \left[||\frac{1}{n} \sum_{i=1}^{n} \nabla_{\boldsymbol{x}} \mathcal{L}(F_i(\boldsymbol{x}), \boldsymbol{O})||_2^2 \right].
\end{aligned} \tag{26}$$

Since models were trained until convergence before deployment, the gradients of each task in object detection has a zero mean. Taking the objectness component as an example, the zero mean property (i.e., $\mathbb{E}_{(\boldsymbol{x},\boldsymbol{O})\sim\mathcal{D}}[\nabla_{\boldsymbol{x}} \mathcal{L}_{\text{obj}}(F(\boldsymbol{x}), \boldsymbol{O})] = 0$) yields

$$\begin{aligned}
&\text{Cov}(\nabla_{\boldsymbol{x}} \mathcal{L}_{\text{obj}}(F_i(\boldsymbol{x}), \boldsymbol{O}), \nabla_{\boldsymbol{x}} \mathcal{L}_{\text{obj}}(F_j(\boldsymbol{x}), \boldsymbol{O})) \\
&= \mathop{\mathbb{E}}_{(\boldsymbol{x},\boldsymbol{O})\sim\mathcal{D}} [\nabla_{\boldsymbol{x}} \mathcal{L}_{\text{obj}}(F_i(\boldsymbol{x}), \boldsymbol{O}) \nabla_{\boldsymbol{x}} \mathcal{L}_{\text{obj}}(F_j(\boldsymbol{x}), \boldsymbol{O})] \\
&- \mathop{\mathbb{E}}_{(\boldsymbol{x},\boldsymbol{O})\sim\mathcal{D}} [\nabla_{\boldsymbol{x}} \mathcal{L}_{\text{obj}}(F_i(\boldsymbol{x}), \boldsymbol{O})] \mathop{\mathbb{E}}_{(\boldsymbol{x},\boldsymbol{O})\sim\mathcal{D}} [\nabla_{\boldsymbol{x}} \mathcal{L}_{\text{obj}}(F_j(\boldsymbol{x}), \boldsymbol{O})] \\
&= \mathop{\mathbb{E}}_{(\boldsymbol{x},\boldsymbol{O})\sim\mathcal{D}} [\nabla_{\boldsymbol{x}} \mathcal{L}_{\text{obj}}(F_i(\boldsymbol{x}), \boldsymbol{O}) \nabla_{\boldsymbol{x}} \mathcal{L}_{\text{obj}}(F_j(\boldsymbol{x}), \boldsymbol{O})].
\end{aligned} \tag{27}$$

The covariance of the other two tasks (i.e., bounding box estimation and class label prediction) can be similarly derived:

$$\begin{aligned}
&\text{Cov}(\nabla_{\boldsymbol{x}} \mathcal{L}_{\text{bbox}}(F_i(\boldsymbol{x}), \boldsymbol{O}), \nabla_{\boldsymbol{x}} \mathcal{L}_{\text{bbox}}(F_j(\boldsymbol{x}), \boldsymbol{O})) \\
&= \mathop{\mathbb{E}}_{(\boldsymbol{x},\boldsymbol{O})\sim\mathcal{D}} [\nabla_{\boldsymbol{x}} \mathcal{L}_{\text{bbox}}(F_i(\boldsymbol{x}), \boldsymbol{O}) \nabla_{\boldsymbol{x}} \mathcal{L}_{\text{bbox}}(F_j(\boldsymbol{x}), \boldsymbol{O})].
\end{aligned} \tag{28}$$

$$\begin{aligned}
&\text{Cov}(\nabla_{\boldsymbol{x}} \mathcal{L}_{\text{cls}}(F_i(\boldsymbol{x}), \boldsymbol{O}), \nabla_{\boldsymbol{x}} \mathcal{L}_{\text{cls}}(F_j(\boldsymbol{x}), \boldsymbol{O})) \\
&= \mathop{\mathbb{E}}_{(\boldsymbol{x},\boldsymbol{O})\sim\mathcal{D}} [\nabla_{\boldsymbol{x}} \mathcal{L}_{\text{cls}}(F_i(\boldsymbol{x}), \boldsymbol{O}) \nabla_{\boldsymbol{x}} \mathcal{L}_{\text{cls}}(F_j(\boldsymbol{x}), \boldsymbol{O})].
\end{aligned} \tag{29}$$

Expanding Equation 26 with Equation 1 and the above property, we obtain

$$\begin{aligned}
&\mathop{\mathbb{E}}_{(\boldsymbol{x},\boldsymbol{O})\sim\mathcal{D}} [\Delta\mathcal{L}(\boldsymbol{x}, \boldsymbol{O}; \Phi, \epsilon)] \\
&\propto \sqrt{\frac{1}{3n} + \frac{2}{9n^2} \left[\text{Cov}_{\text{intra}}(\boldsymbol{x}, \boldsymbol{O}; \Phi) + \text{Cov}_{\text{inter}}(\boldsymbol{x}, \boldsymbol{O}; \Phi) \right]}.
\end{aligned} \tag{30}$$

$\text{Cov}_{\text{intra}}(\boldsymbol{x}, \boldsymbol{O}; \Phi)$ and $\text{Cov}_{\text{inter}}(\boldsymbol{x}, \boldsymbol{O}; \Phi)$ are two terms capturing the covariance of gradients across tasks and models:

$$\begin{aligned}
\text{Cov}_{\text{intra}}(\boldsymbol{x}, \boldsymbol{O}; \Phi) = \sum_{i=2}^{n} \sum_{j=1}^{i-1} \Bigg[&\frac{\text{Cov}(\nabla_{\boldsymbol{x}} \mathcal{L}^i_{\text{obj}}, \nabla_{\boldsymbol{x}} \mathcal{L}^j_{\text{obj}})}{\text{Cov}(\nabla_{\boldsymbol{x}} \mathcal{L}^i_{\text{obj}}, \nabla_{\boldsymbol{x}} \mathcal{L}^i_{\text{obj}})} \\
&+ \frac{\text{Cov}(\nabla_{\boldsymbol{x}} \mathcal{L}^i_{\text{bbox}}, \nabla_{\boldsymbol{x}} \mathcal{L}^j_{\text{bbox}})}{\text{Cov}(\nabla_{\boldsymbol{x}} \mathcal{L}^i_{\text{bbox}}, \nabla_{\boldsymbol{x}} \mathcal{L}^i_{\text{bbox}})} + \frac{\text{Cov}(\nabla_{\boldsymbol{x}} \mathcal{L}^i_{\text{cls}}, \nabla_{\boldsymbol{x}} \mathcal{L}^j_{\text{cls}})}{\text{Cov}(\nabla_{\boldsymbol{x}} \mathcal{L}^i_{\text{cls}}, \nabla_{\boldsymbol{x}} \mathcal{L}^i_{\text{cls}})} \Bigg],
\end{aligned} \tag{31}$$

$$\begin{aligned}
&\text{Cov}_{\text{inter}}(\boldsymbol{x}, \boldsymbol{O}; \Phi) \\
&= \sum_{i=2}^{n} \sum_{j=1}^{i-1} \Bigg[\frac{\text{Cov}(\nabla_{\boldsymbol{x}} \mathcal{L}^i_{\text{obj}}, \nabla_{\boldsymbol{x}} \mathcal{L}^j_{\text{bbox}})}{\text{Cov}(\nabla_{\boldsymbol{x}} \mathcal{L}^i_{\text{obj}}, \nabla_{\boldsymbol{x}} \mathcal{L}^i_{\text{obj}})} + \frac{\text{Cov}(\nabla_{\boldsymbol{x}} \mathcal{L}^i_{\text{obj}}, \nabla_{\boldsymbol{x}} \mathcal{L}^j_{\text{cls}})}{\text{Cov}(\nabla_{\boldsymbol{x}} \mathcal{L}^i_{\text{obj}}, \nabla_{\boldsymbol{x}} \mathcal{L}^i_{\text{obj}})} \\
&\quad + \frac{\text{Cov}(\nabla_{\boldsymbol{x}} \mathcal{L}^i_{\text{bbox}}, \nabla_{\boldsymbol{x}} \mathcal{L}^j_{\text{obj}})}{\text{Cov}(\nabla_{\boldsymbol{x}} \mathcal{L}^i_{\text{bbox}}, \nabla_{\boldsymbol{x}} \mathcal{L}^i_{\text{bbox}})} + \frac{\text{Cov}(\nabla_{\boldsymbol{x}} \mathcal{L}^i_{\text{bbox}}, \nabla_{\boldsymbol{x}} \mathcal{L}^j_{\text{cls}})}{\text{Cov}(\nabla_{\boldsymbol{x}} \mathcal{L}^i_{\text{bbox}}, \nabla_{\boldsymbol{x}} \mathcal{L}^i_{\text{bbox}})} \\
&\quad + \frac{\text{Cov}(\nabla_{\boldsymbol{x}} \mathcal{L}^i_{\text{cls}}, \nabla_{\boldsymbol{x}} \mathcal{L}^j_{\text{obj}})}{\text{Cov}(\nabla_{\boldsymbol{x}} \mathcal{L}^i_{\text{cls}}, \nabla_{\boldsymbol{x}} \mathcal{L}^i_{\text{cls}})} + \frac{\text{Cov}(\nabla_{\boldsymbol{x}} \mathcal{L}^i_{\text{cls}}, \nabla_{\boldsymbol{x}} \mathcal{L}^j_{\text{bbox}})}{\text{Cov}(\nabla_{\boldsymbol{x}} \mathcal{L}^i_{\text{cls}}, \nabla_{\boldsymbol{x}} \mathcal{L}^i_{\text{cls}})} \Bigg] \\
&+ \sum_{i=1}^{n} \Bigg[\frac{\text{Cov}(\nabla_{\boldsymbol{x}} \mathcal{L}^i_{\text{bbox}}, \nabla_{\boldsymbol{x}} \mathcal{L}^i_{\text{obj}})}{\text{Cov}(\nabla_{\boldsymbol{x}} \mathcal{L}^i_{\text{bbox}}, \nabla_{\boldsymbol{x}} \mathcal{L}^i_{\text{bbox}})} + \frac{\text{Cov}(\nabla_{\boldsymbol{x}} \mathcal{L}^i_{\text{cls}}, \nabla_{\boldsymbol{x}} \mathcal{L}^i_{\text{obj}})}{\text{Cov}(\nabla_{\boldsymbol{x}} \mathcal{L}^i_{\text{cls}}, \nabla_{\boldsymbol{x}} \mathcal{L}^i_{\text{cls}})} \\
&\quad + \frac{\text{Cov}(\nabla_{\boldsymbol{x}} \mathcal{L}^i_{\text{cls}}, \nabla_{\boldsymbol{x}} \mathcal{L}^i_{\text{bbox}})}{\text{Cov}(\nabla_{\boldsymbol{x}} \mathcal{L}^i_{\text{cls}}, \nabla_{\boldsymbol{x}} \mathcal{L}^i_{\text{cls}})} \Bigg],
\end{aligned} \tag{32}$$

where $\mathcal{L}^i_{\text{task}} = \mathcal{L}_{\text{task}}(F_i(\boldsymbol{x}), \boldsymbol{O})$ for task $\in \{\text{obj}, \text{bbox}, \text{cls}\}$. The detailed mathematical steps of a similar proof can be referred to [28].

B FORMAL ANALYSIS: ADVERSARIAL PERTURBATION LOWER BOUND

We further present a formal analysis on the robustness offered by FUSE from the perspective of adversarial perturbation lower bound [25]. By considering a special case where input images contain only one object, we show that the adversarial attack is already hardened by FUSE, and then attacking an object detection system with multi-object recognition and localization is even more challenging.

Let $p_j(\boldsymbol{x})$ be the pre-softmax confidence of the object contained in input $\boldsymbol{x}$ being a class j instance. With the Lipschitz assumption that p_j is a continuously differentiable function on the set S containing $\boldsymbol{x}$, $g(\boldsymbol{x}) = p_{\text{GT}}(\boldsymbol{x}) - p_{y^*}(\boldsymbol{x})$ is also a Lipschitz function, where GT is the true class label of $\boldsymbol{x}$ and y^* is an incorrect class label. Then, we obtain:

$$|g(\boldsymbol{y}) - g(\boldsymbol{x})| \le L_q \|\boldsymbol{y} - \boldsymbol{x}\|_p. \tag{33}$$

where $L_q = \max \{\|\nabla g(\boldsymbol{x})\|_q : \boldsymbol{x} \in S\}$ is the local Lipschitz constant with $\nabla g(\boldsymbol{x}) = (\frac{\partial g(\boldsymbol{x})}{\partial x_1}, \cdots, \frac{\partial g(\boldsymbol{x})}{\partial x_d})^\top$, $\boldsymbol{x}, \boldsymbol{y} \in S$, $\frac{1}{p} + \frac{1}{q} = 1$, $p \ge 1$, and $q \le \infty$. As introduced in Section 3, adversarial example $\boldsymbol{x}' \in S$ is generated by adding malicious perturbation $\boldsymbol{\delta}$ to the benign input $\boldsymbol{x}$ (i.e., $\boldsymbol{x}' = \boldsymbol{x} + \boldsymbol{\delta}$). By setting $\boldsymbol{y}$ in Equation 33 to be $\boldsymbol{x}'$ and rearranging, we have

$$g(\boldsymbol{x}) - L_q \|\boldsymbol{\delta}\|_p \le g(\boldsymbol{x} + \boldsymbol{\delta}) \le g(\boldsymbol{x}) + L_q \|\boldsymbol{\delta}\|_p. \tag{34}$$

When $g(\boldsymbol{x}+\boldsymbol{\delta}) < 0, p_{\text{GT}}(\boldsymbol{x}') < p_{y^*}(\boldsymbol{x}')$ is hold for an adversarial example $\boldsymbol{x}'$. Hence, if $\|\boldsymbol{\delta}\|_p$ is small enough such that $g(\boldsymbol{x}) - L_q \|\boldsymbol{\delta}\|_p \geq 0$, then no adversarial examples can be found. That is, if

$$g(\boldsymbol{x}) - L_q \|\boldsymbol{\delta}\|_p \geq 0 \Rightarrow \|\boldsymbol{\delta}\|_p \leq \frac{p_{\text{GT}}(\boldsymbol{x}) - p_{y^*}(\boldsymbol{x})}{L_q}, \quad (35)$$

then the detector's decision can never be changed and the attack will never succeed.

The above analysis states that there exists a guarded region (hypersphere) with a radius of $(p_{\text{GT}}(\boldsymbol{x}) - p_{y^*}(\boldsymbol{x}))/L_q$ centered at the benign input $\boldsymbol{x}$ where no adversarial examples can be found. If we can enlarge the guarded region, then the victim detector under protection will be more resilient to even stronger attacks at the increased cost in terms of adversarial distortion. Indeed, under the protection by a team of diverse models, simply generating adversarial examples with the minimum distortion (Equation 35) is insufficient as they must fool multiple members to misbehave identically, such as fabricating objects with the exact same location, dimension, and class label, which is extremely difficult, if not impossible. Hence, the guarded region under FUSE is larger than the original lower bound, and the increased cost for adversarial distortion ensures stronger adversarial robustness.

C EXPERIMENT SETUP

Datasets and Machine. We conduct experiments on two datasets: (1) PASCAL VOC (2007+2012) [7] and (2) INRIA [6]. All measurements are recorded on NVIDIA RTX 2080 SUPER GPU, Intel i7-9700K (3.60GHz) CPU, and 32 GB RAM on Ubuntu 18.04.

Object Detection Models. We include three dominant detection algorithms: Faster R-CNN (FRCNN) [19], YOLOv3 [18] and SSD [14] with heterogeneous neural architectures (F_1 to F_5 in Table 2). Those pre-trained models are available on public repositories. For YOLOv3 including the Darknet53 (i.e., YOLOv3-D) and MobileNetV1 (i.e., YOLOv3-M) backbones, we follow the repository[1]. For SSD, the code is exported from the repository[2]. For Faster R-CNN (FRCNN), we exploit the source code from the repository[3]. All diversity joint training in our experiments follow the same hyperparameter setting, including $\lambda = 10$ for diversity regularization optimizations. The learning rate schedule and all other hyperparameter configuration follow the original repository of Faster R-CNN without additional fine-tuning. The diversity joint training on kernel filters decorrelates the first two convolutional layers.

Attacks. Four state-of-the-art untargeted attacks are included: TOG [5], UEA [24], RAP [13], and DAG [26]. We also evaluate the robustness against targeted attacks [5] with object vanishing (TOG-v), fabrication (TOG-f), and mislabeling (TOG-m) effects. We generate adversarial examples by using the source code released by TOG[4], UEA[5], RAP[6], and DAG[7] attacks with their default hyperparameters. For the TOG patches, they are generated with size (64×64), trained with 30 epochs, a batch size of 8, and a learning rate of 0.1.

Defenses. For FUSE, we always form teams of three models (including the victim) as we found this team size is sufficient to offer strong robustness. The IOU threshold τ_{IOU} is always set to be 0.50. The mitigation baseline using the adversarial detection training method [29] is implemented by following the description in the paper as the original source code is unavailable.

[1] https://github.com/Adamdad/keras-YOLOv3-mobilenet
[2] https://github.com/pierluigiferrari/ssd_keras
[3] https://github.com/chenyuntc/simple-faster-rcnn-pytorch
[4] https://github.com/git-disl/TOG
[5] https://github.com/LiangSiyuan21/Adversarial-Attacks-for-Image-and-Video-Object-Detection
[6] https://github.com/yuezunli/BMVC2018R-AP
[7] https://github.com/cihangxie/DAG