PROTECTING DNN FROM EVASION ATTACKS USING ENSEMBLE OF HIGH FOCAL DIVERSITY

Anonymous authors
Paper under double-blind review

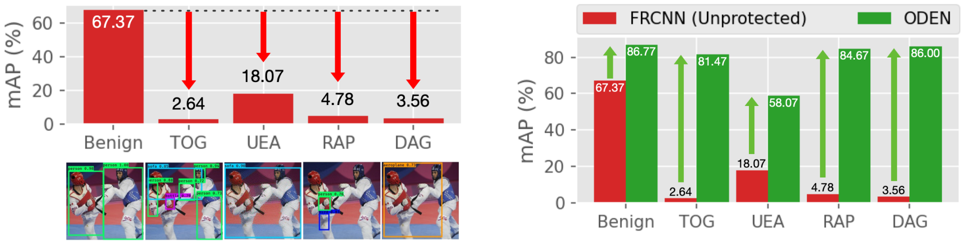
ABSTRACT

Edge AI continues to attract emerging applications that deploy well-trained DNN models on heterogeneous edge clients for real-time object detection. Recent studies have shown that evasion attacks on DNN object detection models at the test time are on the rise. Such evasion attacks generate deceptive queries using maliciously manipulated or out-of-distribution data, aiming to mislead high-quality object detectors during edge inference. This paper introduces ODEN, a novel approach to object detection ensemble, which combines a detection inconsistency solver with focal diversity-optimized ensemble pruning to defend against evasion attacks. The focal diversity ranking techniques enable ODEN to compose an ensemble from a pool of base object detectors with high failure independence, which strengthens the generalization performance of the ODEN ensemble in the presence of irregular query data and evasion attacks. The ODEN inconsistency solver can detect and resolve three types of inconsistency by combining detection results from multiple DNN object detectors: the inconsistency of the object existence, the size and location inconsistency of the bounding boxes of detected objects, and the classification inconsistency of detected objects and their confidence. Extensive experiments on three benchmark vision datasets (OpenImages, COCO, and VOC) show that under no attack, ODEN can outperform existing ensemble methods by up to 9.33% of mAP. Compared to the low mAP of 2.64~18.07% under four evasion attacks, ODEN can maintain a high mAP of 58.97~86.00%, achieving up to an 82.44% increase in AI safety.

1 INTRODUCTION

Evasion attacks allow an adversary to control the detection capability of well-trained DNN models by generating deceptive queries. Apart from irregular out-of-distribution data, a representative approach to creating such queries is to use the gradients of object detection models to find tiny perturbations to input (Chow et al., 2020a). The patterns can mislead the kernels in DNN object detectors to amplify the perturbation, which becomes large enough to interfere with the final decision in the output layer. Such attacks can drastically reduce the detection accuracy in mAP (mean average precision (Everingham et al., 2015)). Figure 1a shows that under four evasion attacks: TOG (Chow et al., 2020a), DAG (Xie et al., 2017), RAP (Li et al., 2018), and UEA (Wei et al., 2019), the mAP of a Faster RCNN (FRCNN) (Ren et al., 2015) model drastically drops from 67.37% to 2.64~18.07%. This deception-induced malfunctioning can lead to severe consequences in safety-critical edge AI applications such as autonomous vehicles (Feng et al., 2021) and intelligent surveillance (Teixidó et al., 2021).

This paper presents ODEN, a focal diversity-enhanced ensemble framework for real-time object detection with dual goals: (i) to improve the safety of edge AI under evasion attacks and (ii) to enhance the generalization performance of DNN models in benign scenarios for high-quality edge inference. Unlike reactive defense methods, ODEN is a proactive methodology with built-in auto-verification and auto-repairing capability through two novel and synergistic functional components: (i) **the inconsistency solver** for producing robust ensemble detection results by attesting and restoring inconsistent detection results from multiple member models of an ensemble, and (ii) **the focal diversity-optimized ensemble pruning** for producing the sub-ensemble of high focal diversity (high failure-independence) and small ensemble size, hence strengthening the effectiveness of our inconsistency solver at a low computational cost. *First*, unlike the ensemble of single-task learn-



(a) Evasion attack to FRCNN

(b) Protection by ODEN

Figure 1: Evasion attacks can cause well-trained object detectors (FRCNN in this example) to have severely reduced mAP. ODEN is a protection mechanism to enhance mAP under both benign and attack scenarios.

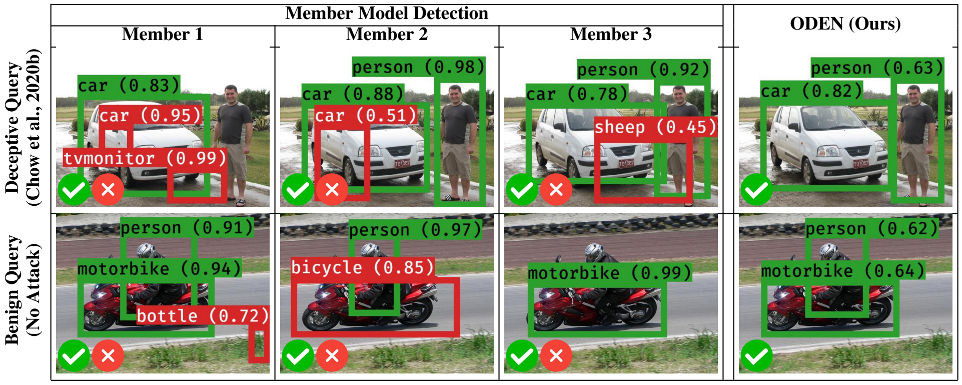


Table 1: A deceptive query (1st row) can mislead three object detectors to misbehave (1st-3rd columns). Since the ODEN focal diversity encourages models to make errors differently, the inconsistency solver can construct correct decision (4th column). The same can happen to the benign scenario (2nd row) with no attack.

ers such as DNN image classifiers (Wu et al., 2021b), DNN object detectors are multi-task learners (Redmon & Farhadi, 2018). ODEN has to deal with inconsistent detection results on all three learning tasks from each ensemble member model: inconsistency in object existence detection, inconsistency in bounding box locations of detected objects, and inconsistency in the classification of detected objects and their confidence scores. The inconsistency solver of ODEN distills those disagreeing predictions from the member models of an ensemble by jointly calibrating (i) the detection inconsistency in object existence and (ii) the perception inconsistency of detected objects, including their bounding boxes and their prediction confidence scores. *Second*, we introduce a focal diversity-optimized ensemble selection method, which can select the top-k ensembles from a pool of base DNN models using their focal detection diversity scores, ensuring that an ensemble with high focal diversity will result in high mAP performance under both evasion attacks and benign test scenarios. Figure 1b shows that ODEN can perform auto-verification and auto-repairing to boost the defensibility, achieving up to a 82.44% increase in edge AI safety under four evasion attacks and enhancing the benign mAP performance of FRCNN from 67.37% to 86.77%. Table 1 provides illustrative visualization examples. Consider the first three columns. The 1st row shows a deceptive query example generated by TOG (Chow et al., 2020b) at test time, which deceives all three DNN models pre-trained independently though the adverse effect of deception may vary. Model 1 fabricates two fake objects and cannot detect the person, while the other two models incorrectly detect one additional object. The 2nd row shows that even with a benign query at test time, these three DNN detectors may not succeed and interestingly fail independently. We show in the 4th column and Figure 1b that the ODEN-optimized detection ensemble can outperform the best-performing member detector in mAP under evasion attacks and in benign scenarios, thanks to both the ODEN focal diversity ensemble selection, which selects the detectors of high failure-independence to form an ensemble, and the ODEN inconsistency solver, which efficiently combines two types of detection calibrations to rectify three levels of inconsistency.

This paper makes two original contributions. First, we present a robust inconsistency solver to distill disagreeing predictions from member models of an ensemble. Second, we introduce the concept of focal detection diversity to measure the failure independence of member models of an ensemble and propose a focal diversity-optimized ensemble pruning method, which selects top sub-ensembles in terms of their high focal diversity scores, ensuring high mAP performance under benign scenarios and evasion attacks. We conduct extensive experiments with three popular vision benchmarks: MS COCO (Lin et al., 2014), Open Images (Kuznetsova et al., 2020), and PASCAL VOC (Everingham et al., 2015), under four recent evasion attacks to DNN object detection. Our evaluations show three important results: (1) Object detection ensembles from ODEN consistently offer high mAP over the best-performing member and improve the ensemble performance by about 9.33% in mAP compared to the existing representative detection ensemble methods. (2) ODEN can effectively select the top-k ensemble teams based solely on their high focal diversity scores, demonstrating the novelty and importance of our focal diversity-optimized ensemble pruning. (3) The combination of our inconsistency solver and focal diversity ensemble selection empowers the ODEN-enabled edge systems for object detection with high defensibility against state-of-the-art evasion attacks.

2 Related Work

Evasion Attacks. Evasion attacks include adversarial input perturbations, such as DAG (Wei et al., 2019), RAP (Xie et al., 2017), UEA (Li et al., 2018), and TOG (Chow et al., 2020a), and malicious physical world attacks using small color-dense patches (Song et al., 2018; Thys et al., 2019). They differ from one another in attack strategies. DAG and RAP attack the object existence prediction and bounding box regression components to cause the victim to misbehave randomly. UEA utilizes GAN to generate adversarial perturbations with attention regularization but tends to inject unnecessarily larger noise in comparison. TOG is a general-purpose evasion attack with both random attack and targeted attack formulations, such as TOG-vanishing, TOG-fabrication, and TOG-mislabeling. These attacks have shown transferability properties (Staff et al., 2021).

Evasion Defenses. Compared to the evasion attacks to well-trained DNN object detection algorithms, such as FRCNN (Ren et al., 2015), SSD (Liu et al., 2016), and YOLOv3 (Redmon & Farhadi, 2018), there has been limited study on defense methods for mitigating these threats. First, existing defenses against adversarial examples crafted to evade single-task learners like DNN classifiers are not applicable to object detection (Zhang et al., 2021), including adversarial training-based defenses (Bai et al., 2021). For example, the adversarial training of DNN object detectors (Zhang & Wang, 2019) suffers from significant performance reduction in benign mAP (dropped by 31.38%), while it can only improve the mAP of FRCNN under DAG attack to 35.58%.

DNN Ensembles. Neural network ensembles are known to provide better generalization performance (Geman et al., 1992; Sharkey, 2012). Most of the existing attempts have been made to create DNN ensembles for image classifiers (Wu & Liu, 2021). In comparison, the DNN ensemble for object detection has received much less attention in both benign scenarios and under recent evasion attacks. Clearly, the consensus with majority voting popularly used for classification ensembles is not applicable. It fails miserably when dealing with detection inconsistency because different detectors may detect different sets of objects in terms of existence, the bounding box size and location of detected objects, and their classification prediction and confidence. NMS (Neubeck & Van Gool, 2006) and SoftNMS (Bodla et al., 2017) are popularly used to merge disagreeable bounding boxes in training a DNN object detector. Hence, they are used as the two baselines for comparison with ODEN. NMW (Zhou et al., 2017) and FUSE (Chow & Liu, 2021) are recent enhancements for combining detection results from multiple detectors. Both use a set of hand-picked models pre-trained using different NN backbones to compose an ensemble, where FUSE uses SoftNMS and NMW uses soft-weighting to recompute the confidence for each detection.

3 ODEN Methodology

Given an ensemble of N object detection models, denoted by $\boldsymbol{F} = \{F_1, ..., F_N\}$, a query image $\boldsymbol{x}$ to the ensemble $\boldsymbol{F}$ will be first sent to each of its N member models in parallel and obtain a set of predictions, denoted by $\{F_i(\boldsymbol{x})|F_i \in \boldsymbol{F}\}$. The problem of an object detection ensemble is to find a detection combination function $\boldsymbol{E}$ that maps the collection of candidate detection sets, one from each member model of the ensemble, to a carefully-constructed set of ensemble detected objects as close as possible to the ground-truth objects $\boldsymbol{\mathcal{G}}$ of the query $\boldsymbol{x}$ in a dataset $\boldsymbol{\mathcal{D}}$, i.e.,

$$\min_{(\boldsymbol{x},\boldsymbol{\mathcal{G}})\in\boldsymbol{\mathcal{D}}} ||\boldsymbol{E}(F_1(\boldsymbol{x}), ..., F_N(\boldsymbol{x})) - \boldsymbol{\mathcal{G}}||, \tag{1}$$

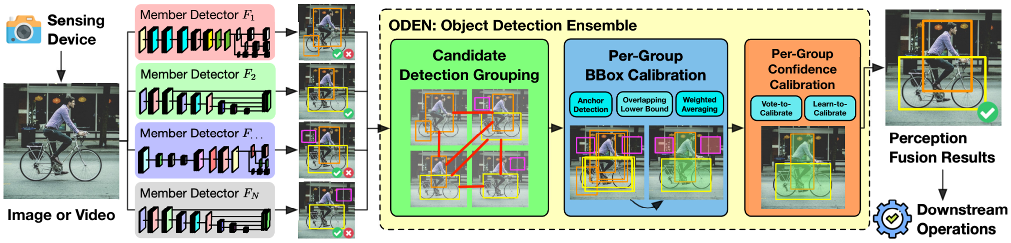


Figure 2: The framework for ODEN inconsistency solver.

where $||\cdot||$ denotes the difference between the ensemble detected objects and the ground truth. In particular, for a member F_i, each of its detected objects $\boldsymbol{o}_{i,j} \in F_i(\boldsymbol{x})$ has two perceptual attributes: (i) the estimated bounding box $\boldsymbol{b}_{i,j} = (b_{i,j}^{\text{xmin}}, b_{i,j}^{\text{ymin}}, b_{i,j}^{\text{xmax}}, b_{i,j}^{\text{ymax}})$, recorded by the top-left and bottom-right corners of the object in the input image and (ii) the predicted K-class probability vector $\boldsymbol{p}_{i,j} = (p_{i,j}^1, p_{i,j}^2, ..., p_{i,j}^K)$ indicating the object classification result. Hence, each detected object can be formally described by $\boldsymbol{o}_{i,j} = (\boldsymbol{b}_{i,j}, \ell_{i,j}, c_{i,j})$ where $\ell_{i,j} = \arg\max_{1\leq k\leq K} p_{i,j}^k$ is the class label and $c_{i,j} = \max_{1\leq k\leq K} p_{i,j}^k$ is the prediction confidence.

3.1 ODEN Inconsistency Solver

The ODEN inconsistency solver is a critical functional component for generating ensemble outputs by solving the three learning task-level inconsistency from multiple member detectors of an ensemble: (i) the inconsistency in object existence detection, (ii) the inconsistency in bounding box size and location of detected objects, and (iii) the inconsistency in classification prediction and confidence of detected objects. Figure 2 gives an architectural overview of our inconsistency solver with three phases of calibration and distillation.

Phase I: Candidate Detection Grouping. The goal of candidate detection grouping is to perform entity resolution: It determines whether two detected objects from different member models refer to the same entity and thus are associated based on (i) whether they are detected with the same class label and (ii) whether their bounding boxes (BBoxes) overlap significantly. Given a set of detection results from each of the N member models in an ensemble, we first partition all detected objects by their class label and sort the detected objects of each class in the descending order of their prediction confidence scores and produce a sorted list of detected objects for each class ℓ, denoted by $\boldsymbol{\mathcal{O}}_\ell$. Second, we further partition the sorted list $\boldsymbol{\mathcal{O}}_\ell$ into different groups. Each corresponds to the same entity. Concretely, we find the detected object with the highest confidence in $\boldsymbol{\mathcal{O}}_\ell$ and use it as the anchor prediction for the first group. Then, we choose the next detected object $\boldsymbol{o}_j \in \boldsymbol{\mathcal{O}}_\ell$ and assign it to a group $\boldsymbol{\gamma}$ if it satisfies the following conditions: (i) the model detecting the object $\boldsymbol{o}_j$ has not yet contributed any detected object to the group $\boldsymbol{\gamma}$, and (ii) there is a significant overlapping between the detected object $\boldsymbol{o}_j$ and those already in the group $\boldsymbol{\gamma}$. Otherwise, we will create a new group with $\boldsymbol{o}_j$. This process repeats until all detected objects in the partition $\boldsymbol{\mathcal{O}}_\ell$ are examined and added to a group. There are several options to make the overlapping comparison between the $\boldsymbol{o}_j$ and those already in the group $\boldsymbol{\gamma}$: The object with the highest detection confidence is used as the anchor, denoted by $\boldsymbol{o}_{anchor(\boldsymbol{\gamma})} = \arg\max_{\boldsymbol{o}_i\in\boldsymbol{\gamma}} c_i$ where c_i is the confidence of the detected object $\boldsymbol{o}_i$ in group $\boldsymbol{\gamma}$. Choosing the BBox of the anchor detection as the representative BBox of the group $\boldsymbol{\gamma}$, we can formally compare the BBox $\boldsymbol{b}_j$ of an object $\boldsymbol{o}_j$ as follows: $\beta_{anchor}(\boldsymbol{o}_j, \boldsymbol{\gamma}) = \text{IOU}(\boldsymbol{b}_j, \boldsymbol{b}_{anchor(\boldsymbol{\gamma})})$, which denotes the anchor detection-based overlapping. (2) An alternative is to examine the BBox of every detected object in the group $\boldsymbol{\gamma}$ and use the minimum overlapping to compare with a threshold $\mathcal{T}_{\text{IOU}}$ (i.e., 0.50). We call this the lower bound (LB) approach and define $\beta_{LB}(\boldsymbol{o}_j, \boldsymbol{\gamma})$ as follows: $\beta_{LB}(\boldsymbol{o}_j, \boldsymbol{\gamma}) = \min[\{\text{IOU}(\boldsymbol{b}_j, \boldsymbol{b}_r) \mid \boldsymbol{o}_r \in \boldsymbol{\gamma}\}]$. (3) Another technique to computing the overlapping of an object $\boldsymbol{o}_j$ with the group $\boldsymbol{\gamma}$ is to generate the representative BBox of the group $\boldsymbol{\gamma}$ by averaging all BBoxes of the detected objects in the group, weighted by their confidence scores, and measure the overlapping with it. We call this option the weighted averaging approach, denoted as $\beta_{WA}(\boldsymbol{o}_j, \boldsymbol{\gamma})$:

$$\beta_{WA}(\boldsymbol{o}_j, \boldsymbol{\gamma}) = \text{IOU}(\boldsymbol{b}_j, \sum_{\boldsymbol{o}_r\in\boldsymbol{\gamma}} \boldsymbol{b}_r c_r / \sum_{\boldsymbol{o}_i\in\boldsymbol{\gamma}} c_i). \tag{2}$$

The final result of Phase I is a list of groups, denoted by $\boldsymbol{\Gamma}$, where each group $\boldsymbol{\gamma} \in \boldsymbol{\Gamma}$ contains a set of detected objects of the same class label, each from a different member model, and all recognizing the same entity. By default, ODEN uses the weighted averaging (WA) approach to define the significant overlapping for grouping. We provide the pseudocode in Algorithm 1 in the appendix.

Phase II: Per-Group BBox Calibration. Different detectors often generate different bounding boxes and different confidence scores for their detection, and all detected objects in each group $\boldsymbol{\gamma} \in \boldsymbol{\Gamma}$ have the same class label and correspond to the same entity. To generate the ensemble detection results, each characterizes the *delegate* object representing a group, we need to compute the exact bounding box (location and size) by aggregating the BBoxes and the different confidence scores of the detected objects in each group. Based on how the group is composed, several approaches can be employed to calibrate the bounding boxes of each group. If we use the anchor detection for grouping (i.e., β_{anchor}), we can return the bounding box $\boldsymbol{b}_{anchor(\boldsymbol{\gamma})}$ of the anchor as the calibrated BBox. Alternatively, if we use the overlapping lower bound β_{LB} or the weighted averaging β_{WA}, we can compute the BBox of the delegate object by aggregating the bounding boxes of all detected objects in the group, each is weighted by the confidence of the corresponding detection. Formally, the bounding box $\hat{\boldsymbol{b}}$ of the delegate object is computed as:

$$\hat{\boldsymbol{b}} = \left(\frac{\sum_{\boldsymbol{o}_i \in \boldsymbol{\gamma}} b_i^{\text{xmin}} c_i}{\sum_{\boldsymbol{o}_j \in \boldsymbol{\gamma}} c_j}, \frac{\sum_{\boldsymbol{o}_i \in \boldsymbol{\gamma}} b_i^{\text{ymin}} c_i}{\sum_{\boldsymbol{o}_j \in \boldsymbol{\gamma}} c_j}, \frac{\sum_{\boldsymbol{o}_i \in \boldsymbol{\gamma}} b_i^{\text{xmax}} c_i}{\sum_{\boldsymbol{o}_j \in \boldsymbol{\gamma}} c_j}, \frac{\sum_{\boldsymbol{o}_i \in \boldsymbol{\gamma}} b_i^{\text{ymax}} c_i}{\sum_{\boldsymbol{o}_j \in \boldsymbol{\gamma}} c_j}\right). \tag{3}$$

The confidence-weighted calibration of the bounding boxes incorporates both the estimated location and size of each bounding box and how certain the estimation is from each corresponding member. We use this approach as the default in our prototype of ODEN.

Phase III: Per-Group Confidence Calibration. Upon completing the first two phases, we obtain the list $\boldsymbol{\Gamma}$ of groups, and for each group $\boldsymbol{\gamma} \in \boldsymbol{\Gamma}$, we have the class label $\hat{\boldsymbol{\ell}}$ and the bounding box $\hat{\boldsymbol{b}}$ for the delegate object representing the group. An intuitive approach to computing the confidence $\hat{\boldsymbol{c}}$ for the delegate object of each group is to take the average of the confidence scores of the detected objects in the group $\boldsymbol{\gamma}$: $\hat{c} = \sum_{\boldsymbol{o}_i \in \boldsymbol{\gamma}} c_i / |\gamma|$. However, this approach does not consider the votes from different member models of the ensemble and can work poorly when the member models generate fake detection, which is the weakness of existing techniques (Neubeck & Van Gool, 2006; Bodla et al., 2017; Zhou et al., 2017). A fake object produced by one model without significant overlapping with the others will form a single-object group with high confidence. One solution is to aggregate the confidence scores of all the detected objects normalized by the ensemble size N: $\hat{c} = \sum_{\boldsymbol{o}_i \in \boldsymbol{\gamma}} c_i / N$. If the group $\boldsymbol{\gamma}$ contains the detected objects from only a few member models, the ensemble detection should be assigned low confidence, reflecting that the delegate object representing the group is less likely to correspond to a real entity compared to another group supported by a larger number of member models. The third approach is *learn to calibrate*, which trains a model for confidence calibration using the validation data. It is motivated by the observation that a group having the detected objects of high confidence and high overlapping with their bounding boxes is more likely to correspond to a real entity, compared to a group having objects of low confidence and with marginally overlapping bounding boxes. Instead of manually examining these statistics for all the groups on each input image, the *learn-to-calibrate* approach will first perform feature extraction for each group $\boldsymbol{\gamma}$ to summarize useful perceptual properties of the group. Let V_c denote the confidence vector of N elements for group $\boldsymbol{\gamma}$, each element denotes the confidence of the detected object from a member model. Similarly, let V_{IOU} denote the IOU vector of the group with N elements, each denotes the overlapping between the BBox of each detected object in the group $\boldsymbol{\gamma}$ and that of the delegate object representing the group. Zero confidence and IOU are assigned if a member does not contribute any detected object to the group. We define the features extracted for the group $\boldsymbol{\gamma}$ as the concatenation of these two vectors: $\Theta(\boldsymbol{\gamma}, \boldsymbol{F}) = V_{\text{c}} || V_{\text{IOU}}$. To learn how to calibrate the confidence of the delegate object representing the group $\boldsymbol{\gamma}$, we train a model to estimate the probability of a given group corresponding to a real entity in the ground truth, i.e., $P(\textsc{Real} = \textsc{True} | \Theta(\boldsymbol{\gamma}, \boldsymbol{F}))$. We employ logistic regression to estimate such a distribution and compute the calibrated confidence $\hat{c}$:

$$\hat{c} = \frac{\sum_{\boldsymbol{o}_i \in \boldsymbol{\gamma}} c_i}{N(1 + \exp(-(\boldsymbol{W}\Theta(\boldsymbol{\gamma}, \boldsymbol{F}) + b)))}, \tag{4}$$

where $\boldsymbol{W}$ and b are learned using a validation set. The *learn to calibrate* is used by default in ODEN.

3.2 Diversity-based Ensemble Selection

Given a pool of N base models, we can formulate $2^N - (N + 1)$ teams with the size ranging from 2 to N. For instance, a 10-model pool leads to $1,013$ teams, and the number of choices jumps exponentially to $1,048,555$ when $N = 20$. In this section, we first introduce the focal detection ensemble diversity measure and then describe the ensemble selection algorithm. The selected sub-ensembles are of high focal diversity, can outperform the best-performing model in the respective team, and tend to have a smaller committee size yet more accurate than using all available N models to form a large ensemble.

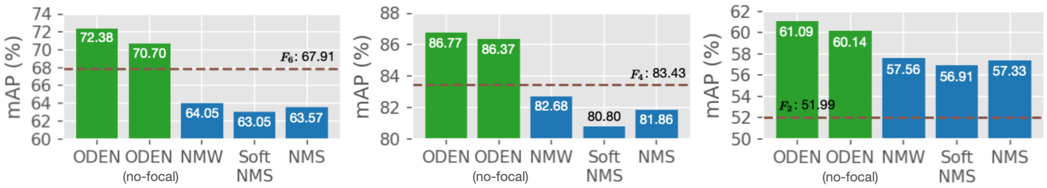
Focal Detection Ensemble Diversity. We adopt a focal model paradigm (Wu et al., 2021b) for diversity assessment. For each sub-ensemble of size M, we consider each member as a focal model to evaluate the diversity of the ensemble based on the negative samples of the focal model from a validation set. Finding negative samples of an object detection model is non-trivial because an object detector tends to detect far more objects than those in the ground truth set and it requires a confidence threshold to decide which ones to discard. An inadequate decision on the threshold may result in unnecessary false positives (too low) or false negatives (too high). In light of this, we implement a ranking-based approach for negative sample determination, which first sorts the detected objects of the focal model in the descending order of their confidence and finds a one-to-one mapping to the set of ground-truth objects. The approach requires the correctly detected objects to have higher confidence than other irrelevant detection (i.e., no false positives), and all ground-truth objects will be recognized (i.e., no false negatives). We provide the pseudocode in Algorithm 2 in the appendix.

Given an ensemble $\boldsymbol{F}$ of M models ($M \leq N$), we compute M focal detection diversity scores by considering each member to be the focal model. Given a focal model F_{focal}, we obtain a set of negative samples and measure the focal model-based disagreement among the other $M-1$ members. In our prototype of ODEN, we measure the focal ensemble diversity by leveraging the non-pairwise general disagreement (Partridge & Krzanowski, 1997). Let Y denote a random variable representing the proportion of models (i.e., i out of M) that fail to recognize a random input sample $\boldsymbol{x}$ defined in Algorithm 2. The probability of $Y = \frac{i}{M}$ is denoted as p_i. The focal diversity of an ensemble $\boldsymbol{F} = \{F_1, ..., F_{\text{focal}}, ..., F_M\}$ of size M w.r.t. the focal model F_{focal} is defined as follows:

$$div_{\text{focal}}(\boldsymbol{F}, F_{\text{focal}}) = 1 - \frac{\sum_{i=1}^{M} \frac{i}{M} p_i}{\sum_{i=1}^{M} \frac{i(i-1)}{M(M-1)} p_i}, \tag{5}$$

where div_{focal} is in the range of $[0, 1]$ with the maximum diversity score of 1 when the failure of one member model is accompanied by the correct recognition by the other.

Focal Diversity-based Ensemble Selection. Given a pool of N base models, say $N = 10$, by choosing F_1 as the focal model, we can compare all the sub-ensembles of size M containing F_1 as the focal model by their focal diversity scores. For $M = 5$, we have a total of 126 sub-ensembles containing the focal model F_1. We can utilize the focal diversity measure $div_{\text{focal}}(\boldsymbol{F}, F_1)$ to partition this set into those sub-ensembles of high focal diversity and those with low diversity and select the top-k sub-ensembles of highest focal diversity as our recommendation for the top-performing teams. For a given focal model F_{focal}, we denote $\boldsymbol{\Lambda}_{F_{\text{focal}}, M}$ to be the set of sub-ensembles of size M containing the focal model F_{focal}. Using Equation 5, we measure the focal ensemble diversity of each sub-ensemble and obtain the diversity-accuracy set, defined by $DA = \{div_{\text{focal}}(\boldsymbol{F}, F_{\text{focal}}), \text{ACC}(\boldsymbol{F}))|\boldsymbol{F} \in \boldsymbol{\Lambda}_{F_{\text{focal}}, M}\}$, where $\text{ACC}(\cdot)$ returns the mAP using ODEN's detection combination algorithm. To identify those ensembles with high focal diversity, we define two initial centroids: one for the cluster with high ensemble diversity using the maximum diversity and accuracy of ensembles in the DA set, and one for the low diversity using the minimum diversity and accuracy of ensembles in the DA set. Then, we partition the DA set using a binary clustering algorithm, such as K-Means, and use the largest diversity in the cluster with low diversity as the cut-off threshold. For each sub-ensemble of M member models, each of the M models will be used as a focal model once and thus it will have M focal diversity scores. For example, the ensemble $F_{1,2,3}$ (i.e., a team with F_1, F_2, and F_3 as members) has three focal diversity scores: one in $\boldsymbol{\Lambda}_{F_1,3}$ with F_1 as the focal model, one in $\boldsymbol{\Lambda}_{F_2,3}$ with F_2 as the focal model, and the third one in $\boldsymbol{\Lambda}_{F_3,3}$ with F_3 as the focal model. Let $\text{HDE}\textsc{ns}\text{S}\textsc{et}_{F_{\text{focal}}, M, \boldsymbol{F}}$ be the partition of the sub-ensembles of size M with high focal diversity for a given focal model F_{focal}. We can use affirmative vote or unanimous vote to determine if an ensemble should be recommended. Using the unanimous voting scheme (intersection), an ensemble $\mathcal{E}$ is selected if $\mathcal{E} \in \bigcap_{i=1}^{N} \text{HDE}\textsc{ns}\text{S}\textsc{et}_{F_i^{\text{focal}}, M, \boldsymbol{F}}$. Using affirmative voting (union), an ensemble $\mathcal{E}$ is selected if $\mathcal{E} \in \bigcup_{i=1}^{N} \text{HDE}\textsc{ns}\text{S}\textsc{et}_{F_i^{\text{focal}}, M, \boldsymbol{F}}$. The affirmative voting is used as the default in the prototype of ODEN.



(a) MS COCO

(b) PASCAL VOC

(c) Open Images

Figure 3: ODEN outperforms three representative detection ensemble methods in benign mAP and the best-performing base model in the respective pool marked by the horizontal line. Compared to ODEN (no-focal), ODEN achieves better mAP by using the ensemble of the highest focal diversity.

4 Experimental Evaluation

We conduct experiments on three object detection benchmarks: (i) MS COCO (Lin et al., 2014) with 80 classes of objects, (ii) Open Images (Kuznetsova et al., 2020) with 500 classes of objects, and (iii) PASCAL VOC (Everingham et al., 2015) with 20 classes of objects with the standard accuracy metric, mean average precision (mAP) (Everingham et al., 2015). **Table 2** summarizes the seventeen base models used in our experiments, including their mAP, the best-performing model, and the average mAP of each pool. For edge inference, we tested ODEN on an edge client connected to multiple Intel Neural Compute Stick 2 (Intel). The source code of ODEN is available at **[anonymized]**.

	MS COCO		Open Images		PASCAL VOC	
	Model	**mAP**	**Model**	**mAP**	**Model**	**mAP**
F_1	SSD300-R	52.47	CRCNN	50.60	FRCNN	67.37
F_2	SSD300-V	46.70	RetinaNet	51.99	SSD300	76.11
F_3	SSD512-R	57.67	CRCNN-FPN	50.55	SSD512	79.83
F_4	SSD512-V	55.81	MRCNN	49.14	YOLOv3-D	83.43
F_5	SSD512-M	42.70	FRCNN	45.28	YOLOv3-M	71.84
F_6	YOLOv3-D	67.91	-	-	-	-
F_7	YOLOv3-M	60.20	-	-	-	-
Best	YOLOv3-D	67.91	RetinaNet	51.99	YOLOv3-D	83.43
Avg.	-	54.78	-	49.51	-	75.72

Table 2: A summary of base models for three benchmark datasets in experiments.

4.1 Benign Detection Performance Analysis

We first evaluate ODEN under no attack because maintaining high benign mAP is a prerequisite for any defense mechanism to be usable in practice. **Figure 3** compares ODEN with non-maximum weighted (NMW), soft non-maximum suppression (SoftNMS), and non-maximum suppression (NMS) in terms of benign mAP on three vision benchmarks. ODEN refers to our ensemble with both inconsistency solver and focal diversity ensemble pruning turned on. The team with the highest focal diversity is $F_{1,3,4,6,7}$ for MS COCO, $F_{1,2,3,4}$ for PASCAL VOC, and $F_{1,2,3,5}$ for Open Images. To provide a zoom-in comparison of ODEN with NMW, SoftNMS, and NMS, which use the entire base model pool as the ensemble, we also include ODEN (no-focal), which is the version of ODEN that has the inconsistency solver but does not use focal diversity-optimized ensemble pruning. Instead, the entire pool of the base models is used as the ensemble team. We make two observations. First, both ODEN and ODEN (no-focal) significantly outperform existing approaches for all benchmark datasets, and both provide better generalization performance than the best-performing base model in the pool. Second, compared to ODEN (no-focal), we show that the generalization performance of ODEN can be further strengthened by combining the detection inconsistency solver with the focal diversity ensemble pruning. **Table 3** provides two visual examples to compare ODEN (the 4th column) with three existing baselines: NMW, SoftNMS, and NMS (the 5th to 7th columns). We use the same ensemble team of $F_{2,3,4}$ on PASCAL VOC for a fair comparison. The comparison shows their effectiveness in resolving detection inconsistency when combining partially correct decisions from individual member models (the 1st to 3rd columns). Consider the first example, F_2 and F_3 correctly recognize the car, but they either estimate a wrong bounding box for the pedestrian or misdetect a fake bird. While F_4 can detect the pedestrian, the bounding box of the car is imprecise. ODEN produces the correct detection of the pedestrian by combining the two correct bounding boxes from F_3 and F_4. The per-group confidence calibration in ODEN reduces the confidence of the incorrect person bounding box by F_2 from 0.99 to 0.03, which is negligible and thus not considered a false positive. In contrast, the incorrect person bounding box from F_2 is preserved by the other three methods with high confidence, even though it is only detected by one out of three member models. Similar observations can also be made on the fake bird by F_3, where ODEN successfully corrects the confidence to remove it from the detection results of the ensemble, while the other three approaches still include it with a confidence of 0.93. These results demonstrate that the per-group BBox and

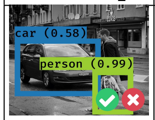
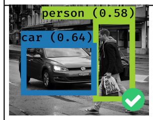
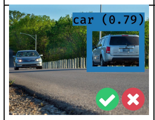
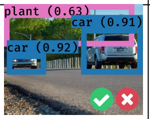
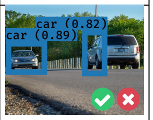
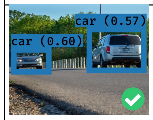
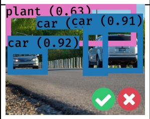
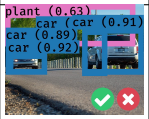
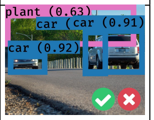
Member Model Detection			**Ensemble Detection $F_{2,3,4}$**			
Member F_2	**Member F_3**	**Member F_4**	**ODEN**	**NMW**	**SoftNMS**	**NMS**
						
						

Table 3: Detection results on two test images by three member models and four ensemble methods using the same ensemble team $F_{2,3,4}$. ODEN inconsistency solver successfully removes false positives.

Dataset	**MS COCO**					**Open Images**			**PASCAL VOC**		
Ensemble	$F_{1,2,3,4,5,6,7}$	$F_{1,2,3,4,6,7}$	$F_{1,3,4,6,7}$	$F_{1,3,6,7}$	$F_{1,4,6}$	$F_{1,2,3,4,5}$	$F_{1,2,3,5}$	$F_{1,2,3}$	$F_{1,2,3,4,5}$	$F_{1,2,3,4}$	$F_{2,3,4}$
mAP	70.70%	71.32%	72.38%	72.19%	71.69%	60.14%	61.09%	60.33%	86.37%	86.77%	86.62%
mAP Gain	0%	+0.62%	+1.68%	+1.49%	+0.99%	0%	+0.95%	+0.19%	0%	+0.40%	+0.25%
Best Mem. (mAP)	$F_6(67.91\%)$	$F_6(67.91\%)$	$F_6(67.91\%)$	$F_6(67.91\%)$	$F_6(67.91\%)$	$F_2(51.99\%)$	$F_2(51.99\%)$	$F_2(51.99\%)$	$F_4(83.43\%)$	$F_4(83.43\%)$	$F_4(83.43\%)$
Best Mem. mAP Gain	+2.79%	+3.41%	+4.47%	+4.28%	+3.78%	+8.15%	+9.10%	+8.34%	+2.94%	+3.34%	+3.19%
Team Size	7	6	5	4	3	5	4	3	5	4	3
Cost	100%	86%	71%	57%	43%	100%	80%	60%	100%	80%	60%

Table 4: An illustration of diversity-based ensemble selection by examples in ODEN. The 4th and 6th rows compare the mAP gains of using the ensembles selected by high diversity compared to the ensemble composed of all base models and the best mAP member model. The last two rows show that the higher mAP of sub-ensembles can be achieved with smaller ensemble team size and lower execution cost.

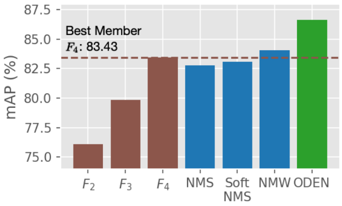


Figure 4: ODEN improves mAP over the best-performing member.

confidence calibration modules in ODEN play significant roles in boosting the ensemble robustness when member models generate fake detection with high confidence. **Figure 4** shows a quantitative comparison with the same team, where NMS and SoftNMS perform worse than the best member (F_5) with an mAP of 83.43%, and ODEN reaches an ensemble mAP of 86.62%, having a 3.19% improvement. **Table 4** gives the top-k sub-ensembles with the highest diversity scores identified by ODEN on MS COCO (top-4), Open Images (top-2), and PASCAL VOC (top-2). The 2nd, 7th, and 10th columns show the teams using all available models in the respective pool (i.e., the ODEN (no-focal) in Figure 3). In such cases, the detection mAP reaches 70.70% on MS COCO, 60.14% on Open Images, and 86.37% on PASCAL VOC. Ensembles with a smaller size can lead to a higher mAP than the ensemble composed of all base models. For example, the 5-member ensemble $F_{1,3,4,6,7}$ on MS COCO achieves an mAP of 72.38%, which is $+4.47\%$ higher than the best member model and $+1.68\%$ higher than the ensemble using all seven models, while the cost of ensemble execution is only 71% compared with the ensemble using all base models. Similar observations can be made in the other two datasets.

4.2 Defensibility Under Evasion Attacks

This section evaluates how effective ODEN can protect object detection against evasion attacks. We conduct experiments on PASCAL VOC using four state-of-the-art evasion attacks: TOG (Chow et al., 2020b), UEA (Wei et al., 2019), RAP (Li et al., 2018), and DAG (Wei et al., 2018). We compare ODEN and ODEN (no-focal) with three ensemble defense methods (NMW, SoftNMS, and NMS) and adversarial training (AdvDetTrain) (Zhang & Wang, 2019). We report the comparison results in **Table 5**. F_1 (i.e., FRCNN) is the victim model. We make three observations. First, ODEN outperforms ODEN (no-focal) and the other three ensemble approaches as well as the representative adversarial training defense under all four evasion attacks and in benign scenarios (2nd column). Second, all five ensemble methods significantly outperform the adversarial training defense under all four evasion attacks and in benign scenarios. Third, the ensemble methods NMW, SoftNMS, and NMS suffer severely under TOG evasion attack with a low mAP of 13.41~17.56%, showing its poor defensibility. In comparison, AdvDetTrain offers slightly better defensibility under TOG attack (from 2.64% to 34.07%), but the benign mAP drops significantly from 67.37% to 35.99%.

Table 6 provides the visualization of the defensibility of ODEN under TOG targeted attacks: TOG-vanishing (row 1), TOG-fabrication (row 2), and TOG-mislabeling (row 3) (Chow et al., 2020b). It

	Benign mAP (%)	Attack mAP (%)			
		TOG	UEA	RAP	DAG
No Protection					
F_1: FRCNN	67.37	2.64	18.07	4.78	3.56
Protected					
ODEN (Inconsistency solver + focal pruning)	**86.77**	**81.47**	**58.97**	**84.67**	**86.00**
ODEN (no-focal) (Inconsistency solver only)	86.37	80.34	57.98	84.17	84.68
NMW (Zhou et al., 2017)	82.98	17.56	54.64	75.65	76.29
SoftNMS (Bodla et al., 2017)	82.23	13.41	53.29	76.67	76.11
NMS (Neubeck & Van Gool, 2006)	82.15	16.86	54.08	75.02	76.01
AdvDetTrain (Zhang & Wang, 2019)	35.99	34.07	17.67	35.60	35.58

Table 5: Defensibility comparison under four evasion attacks on PASCAL VOC.

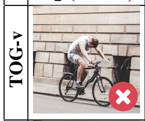
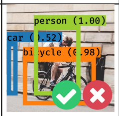
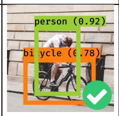
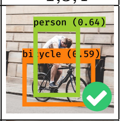
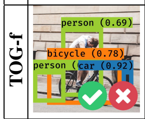
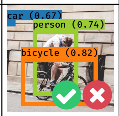
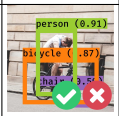
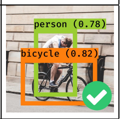
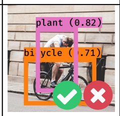
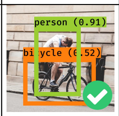
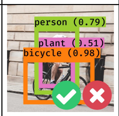
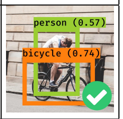
	Member Model Detection			ODEN
	F_1 (Victim)	F_3	F_4	$F_{1,3,4}$
TOG-v				
TOG-f				
TOG-m				

Table 6: Detection results by three member models and ODEN under three targeted TOG attacks.

shows the detection results by three member models (the 1st to 3rd columns) on PASCAL VOC and the ensemble team $F_{1,3,4}$ (the 4th column) under three different TOG attacks, and F_1 is the victim. TOG-v fools the victim model F_1 to detect no object, TOG-f deceives the victim model F_1 to detect the extra person and the car, and TOG-m makes the victim model mislabel the person as a plant. As shown in the 4th column, ODEN successfully attests and restores the correct decision, showing that ODEN can defend against both untargeted and targeted evasion attacks. Additional experiments and visualization are provided in the appendix.

4.3 Computation Time Analysis

This section reports the computation time comparison in **Figure 5** on PASCAL VOC. We compare the average time spent to detect one query image using ODEN, ODEN (no-focal), NMW, SoftNMS, and NMS. This includes the model inference and detection combination time in milliseconds. Even though ODEN uses the focal diversity-optimized ensemble, which is $F_{1,2,3,4}$, instead of the ensemble of all five detectors in the base model pool like the other approaches, the computation time is comparable. This is because all ensemble methods run on an edge node with Intel Neural Compute Stick 2 (Intel), enabling parallel execution of all five member models (Wu et al., 2021a). Other alternative Edge AI accelerators include NVIDIA Jetson (NVIDIA) and Google Coral (Coral). Hence, the computation time is dominated by the slowest model (i.e., FRCNN), which takes 55.56 milliseconds to compute. Comparatively, the time spent on ensemble detection inconsistency solver is negligible: 3.60 milliseconds by ODEN, 3.65 milliseconds by ODEN (no-focal), 2.15 milliseconds by NMW, 2.16 milliseconds by SoftNMS, and 0.92 milliseconds by NMS.

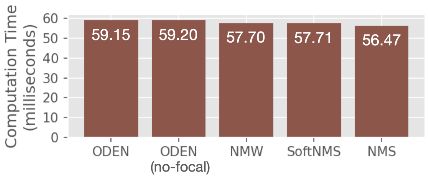


Figure 5: Computation time analysis for detecting objects on an image.

5 Conclusions

We have presented ODEN, a novel object detection ensemble approach to protecting DNNs from evasion attacks. ODEN consists of two synergistic functional components: a robust inconsistency solver to combine object detection results from multiple detectors and a focal diversity-optimized ensemble selection algorithm. Validated by extensive experiments on three benchmark datasets covering seventeen object detection models, we show that (1) ODEN enhances AI safety by maintaining high mAP of an object detection system under evasion attacks; (2) ODEN outperforms existing representative ensemble methods and adversarial training defense over all three vision benchmarks and also consistently outperforms the best-performing member detector in the base model pool; and (3) ODEN focal diversity ensemble pruning algorithm can find the top-k best performing sub-ensembles with high mAP and smaller ensemble size.

REFERENCES

Tao Bai, Jinqi Luo, Jun Zhao, Bihan Wen, and Qian Wang. Recent advances in adversarial training for adversarial robustness. In *IJCAI*, 2021.

Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-nms–improving object detection with one line of code. In *ICCV*, 2017.

Ka-Ho Chow and Ling Liu. Robust object detection fusion against deception. In *ACM SIGKDD*, 2021.

Ka-Ho Chow, Ling Liu, Mehmet Emre Gursoy, Stacey Truex, Wenqi Wei, and Yanzhao Wu. Understanding object detection through an adversarial lens. In *Springer ESORICS*, 2020a.

Ka-Ho Chow, Ling Liu, Margaret Loper, Juhyun Bae, Mehmet Emre Gursoy, Stacey Truex, Wenqi Wei, and Yanzhao Wu. Adversarial objectness gradient attacks in real-time object detection systems. In *IEEE TPS-ISA*, 2020b.

Coral. Coral. `https://coral.ai/`. [Online; Accessed 2022/02/10].

Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *IJCV*, 2015.

Di Feng, Ali Harakeh, Steven L Waslander, and Klaus Dietmayer. A review and comparative study on probabilistic object detection in autonomous driving. *IEEE TITS*, 2021.

Stuart Geman, Elie Bienenstock, and René Doursat. Neural networks and the bias/variance dilemma. *Neural computation*, 4(1):1–58, 1992.

Intel. Intel neural compute stick 2. `https://ark.intel.com/content/www/us/en/ark/products/140109/intel-neural-compute-stick-2.html`. [Online; Accessed 2022/02/10].

Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4. *IJCV*, 2020.

Yuezun Li, Daniel Tian, Xiao Bian, Siwei Lyu, et al. Robust adversarial perturbation on deep proposal-based models. In *BMVC*, 2018.

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 *ECCV*, 2014.

Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single shot multibox detector. In *ECCV*, 2016.

Alexander Neubeck and Luc Van Gool. Efficient non-maximum suppression. In *IEEE ICPR*, 2006.

NVIDIA. Autonomous machines: The future of ai. `https://www.nvidia.com/en-us/autonomous-machines/`. [Online; Accessed 2022/02/10].

Derek Partridge and Wojtek Krzanowski. Software diversity: practical statistics for its measurement and exploitation. *Elsevier IST*, 1997.

Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015.

Amanda JC Sharkey. *Combining artificial neural nets: ensemble and modular multi-net systems*. Springer Science & Business Media, 2012.

Dawn Song, Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Florian Tramer, Atul Prakash, and Tadayoshi Kohno. Physical adversarial examples for object detectors. In *WOOT*, 2018.

Alexander Michael Staff, Jin Zhang, Jingyue Li, Jing Xie, Elizabeth Ann Traiger, Jon Arne Glomsrud, and Kristian Bertheussen Karolius. An empirical study on cross-data transferability of adversarial attacks on object detectors. In *AI-Cybersec@ SGAI*, pp. 38–52, 2021.

Pedro Teixidó, Juan Antonio Gómez-Galán, Rafael Caballero, Francisco J Pérez-Grau, José M Hinojo-Montero, Fernando Muñoz-Chavero, and Juan Aponte. Secured perimeter with electromagnetic detection and tracking with drone embedded and static cameras. *Sensors*, 21(21):7379, 2021.

Simen Thys, Wiebe Van Ranst, and Toon Goedemé. Fooling automated surveillance cameras: adversarial patches to attack person detection. In *CVPRW*, 2019.

Wenqi Wei, Ling Liu, Margaret Loper, Stacey Truex, Lei Yu, Mehmet Emre Gursoy, and Yanzhao Wu. Adversarial examples in deep learning: Characterization and divergence. *arXiv preprint arXiv:1807.00051*, 2018.

Xingxing Wei, Siyuan Liang, Ning Chen, and Xiaochun Cao. Transferable adversarial attacks for image and video object detection. In *IJCAI*, 2019.

Yanzhao Wu and Ling Liu. Boosting deep ensemble performance with hierarchical pruning. In *IEEE ICDM*, 2021.

Yanzhao Wu, Ling Liu, and Ramana Kompella. Parallel detection for efficient video analytics at the edge. In *IEEE CogMI*, 2021a.

Yanzhao Wu, Ling Liu, Zhongwei Xie, Ka-Ho Chow, and Wenqi Wei. Boosting ensemble accuracy by revisiting ensemble diversity metrics. In *CVPR*, 2021b.

Cihang Xie, Jianyu Wang, Zhishuai Zhang, Yuyin Zhou, Lingxi Xie, and Alan Yuille. Adversarial examples for semantic segmentation and object detection. In *ICCV*, 2017.

Haichao Zhang and Jianyu Wang. Towards adversarially robust object detection. In *ICCV*, 2019.

Xingwei Zhang, Xiaolong Zheng, and Wenji Mao. Adversarial perturbation defense on deep neural networks. *ACM Computing Surveys (CSUR)*, 54(8):1–36, 2021.

Huajun Zhou, Zechao Li, Chengcheng Ning, and Jinhui Tang. Cad: Scale invariant framework for real-time object detection. In *ICCV Workshops*, 2017.

A Supplementary Materials - Outline

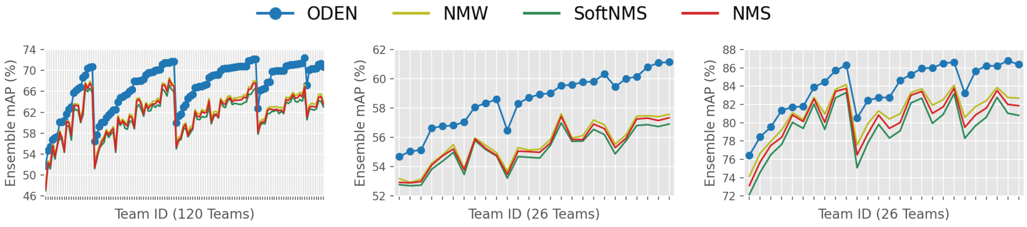
In this supplementary material, we provide additional experiments for ODEN, including the quantitative studies of ODEN under targeted evasion attacks (Section B), the consistent mAP improvement over existing combination approaches (Section C), an ablation study of ODEN's inconsistency solver (Section D), additional visual examples under benign scenarios with no attack (Section E) and under seven evasion attacks (Section F to Section L), the detailed experiment setup (Section M), and the pseudocode of ODEN's key components (Section N).

B Robustness under Targeted Evasion Attacks

Targeted attacks are considered stealthy in applied scenarios because the adversary can control the malicious behavior of the victim system to maximize the fatality (Chow et al., 2020a). We conduct targeted attacks using TOG (Chow et al., 2020b), the state-of-the-art attacks on object detection systems, with object-vanishing, object-fabrication, and object-mislabeling effects. Table 7 summarizes the quantitative results. The three targeted attacks successfully reduce the mAP of the victim detector F_1 on PASCAL VOC from 67.37% to 0.14%, 1.24%, and 2.14% respectively, making the victim system with virtually no utility. ODEN offers a high mAP of 83.68% under TOG-v, 81.67% under TOG-f, and 80.25% under TOG-m. In contrast, NMW, SoftNMS, NMS, and AdvDetTrain can lead to an mAP higher than the victim but are comparatively less effective than ODEN with TOG-v having 34.03~80.64%, TOG-f having 34.18~72.14%, and TOG-m having 34.81~44.28%.

	Benign mAP (%)	**Attack mAP (%)**		
		TOG-v	**TOG-f**	**TOG-m**
No Protection				
F_1: FRCNN (Victim)	67.37	0.14	1.24	2.14
Protected				
ODEN	**86.77**	**83.68**	**81.67**	**80.25**
NMW (Zhou et al., 2017)	82.98	80.64	72.14	44.28
SoftNMS (Bodla et al., 2017)	82.23	79.57	71.01	43.01
NMS (Neubeck & Van Gool, 2006)	82.15	80.50	72.01	44.02
AdvDetTrain (Zhang & Wang, 2019)	35.99	34.03	34.18	34.81

Table 7: Detection ensemble robustness under evasion attacks on PASCAL VOC.



(a) MS COCO (b) Open Images (c) PASCAL VOC

Figure 6: Ensemble mAP comparisons for all possible ensemble teaming with at least two members.

C Detection Ensemble Combination Analysis

For each dataset and its corresponding base model pool, we evaluate all ensemble teams with at least two members, resulting in 120 ensembles for MS COCO, 26 ensembles for Open Images, and 26 ensembles for PASCAL VOC. Figure 6 reports the ensemble mAP of all teams by comparing ODEN with three existing representative detection combination methods. Among the 172 teams across three datasets, ODEN consistently outperforms the three existing schemes by a large margin. The improvement can be as large as 9.14% on MS COCO, 4.58% on Open Images, and 6.05% on PASCAL VOC. Also, the three existing representative methods for combining multiple detections (i.e., NMW, SoftNMS, and NMS) behave similarly in terms of the ensemble mAP performance for different teams, with NMW performing slightly better than NMS and SoftNMS is the worst among the three with a marginally lower mAP for all three datasets.

D Confidence Calibration: Impact of Design Choices

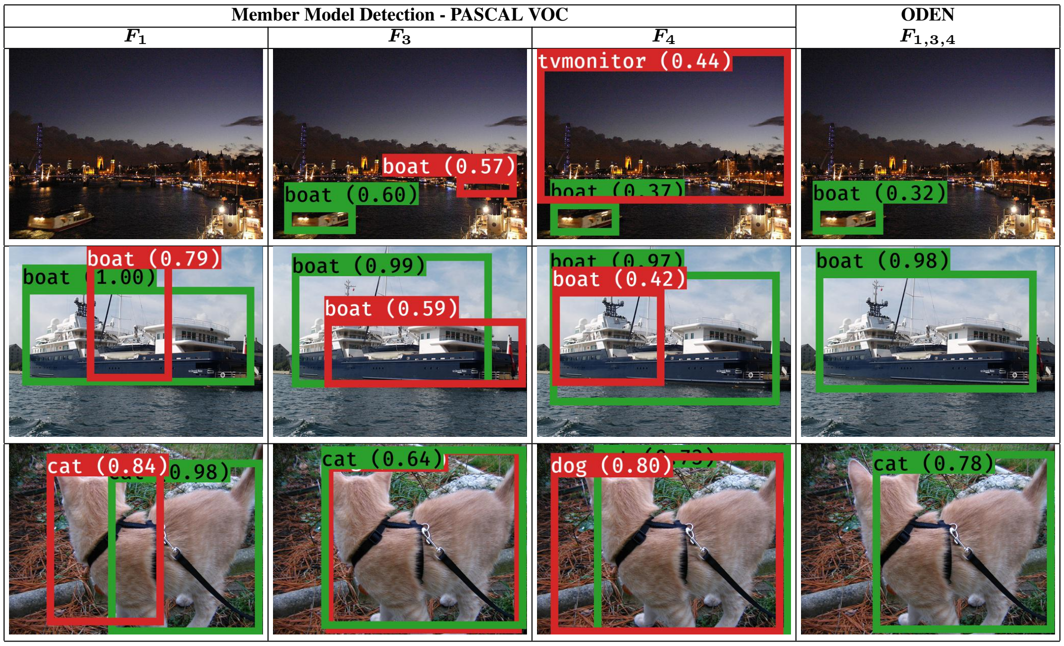
We compare different design choices for the inconsistency solver in Table 8 to understand the contribution of different decisions to the mAP performance of ensembles in ODEN. We use the best ensemble team identified by our diversity-based ensemble selection method for each dataset: $F_{1,3,4,6,7}$ for MS COCO, $F_{1,2,3,5}$ for Open Images, and $F_{1,2,3,4}$ for PASCAL VOC. We make the following observations. First, conducting confidence averaging (a) on each group alone is insufficient to offer satisfactory detection performance on all datasets. In particular, this approach reaches an ensemble mAP of 62.56%, 55.58%, and 81.59% on three datasets but two of them are even worse than the best-performing member in their respective team: 67.91% by F_6 on MS COCO and 83.43% by F_4 on PASCAL VOC. Second, empowering the detection ensemble combination with (b) vote-to-calibrate or (c) learn-to-calibrate immediately improves the ensemble mAP on MS COCO from 62.56% to 69.70% and 70.02% respectively. Improvements of a similar scale can be found in the

	Ensemble mAP (%)		
	MS COCO	**Open Images**	**PASCAL VOC**
(a) Confidence Averaging	62.56	55.58	81.59
(b) Vote-to-Calibrate	69.70	60.38	85.55
(c) Learn-to-Calibrate	70.02	60.32	85.72
(a) + (b) + (c) = ODEN	**72.38**	**61.09**	**86.77**

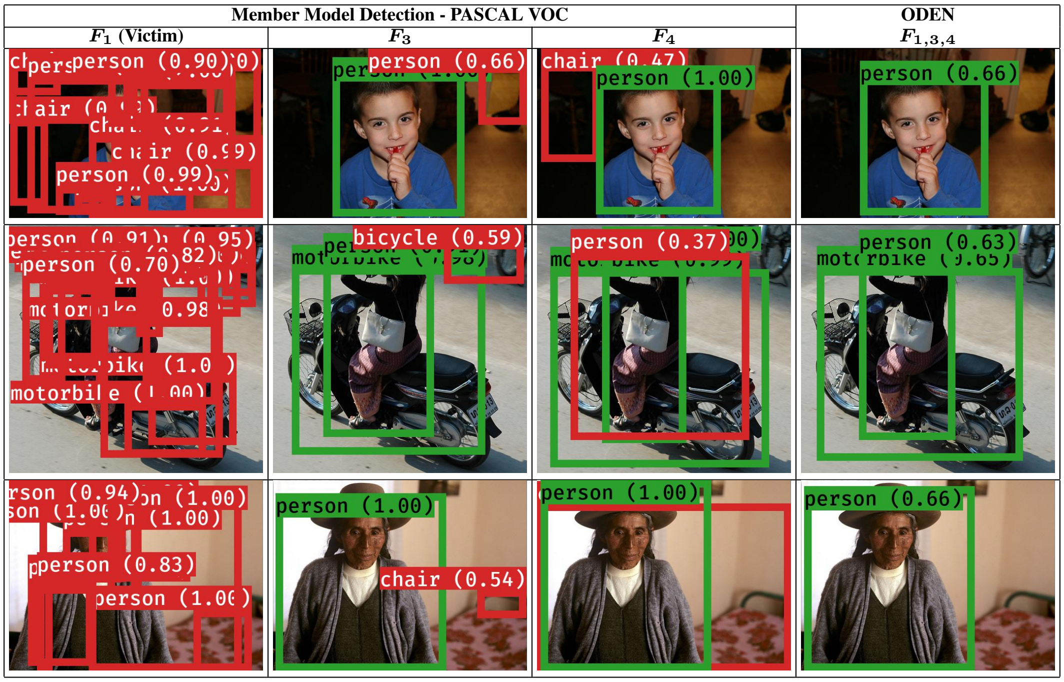
Table 8: Analysis on the design choice in ODEN inconsistency solver.

other two datasets, showing the importance of having dedicated modules to handle the existence of objects through confidence calibration. Third, the vote-to-calibrate and learn-to-calibrate components together further strengthen the decision on the object existence and result in the best ensemble performance.

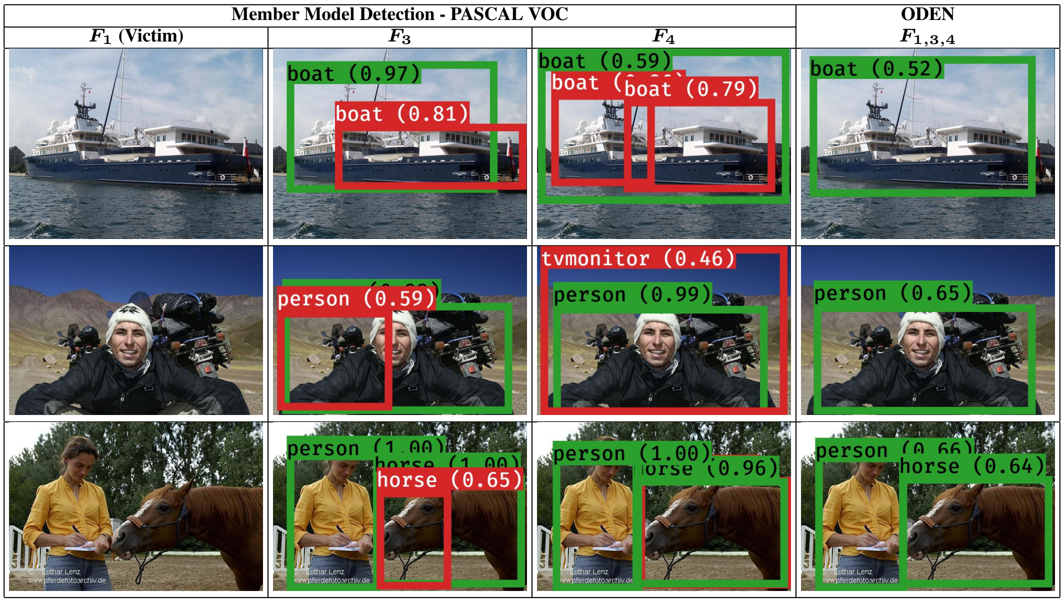
E Visual Examples: Benign (No Attack)



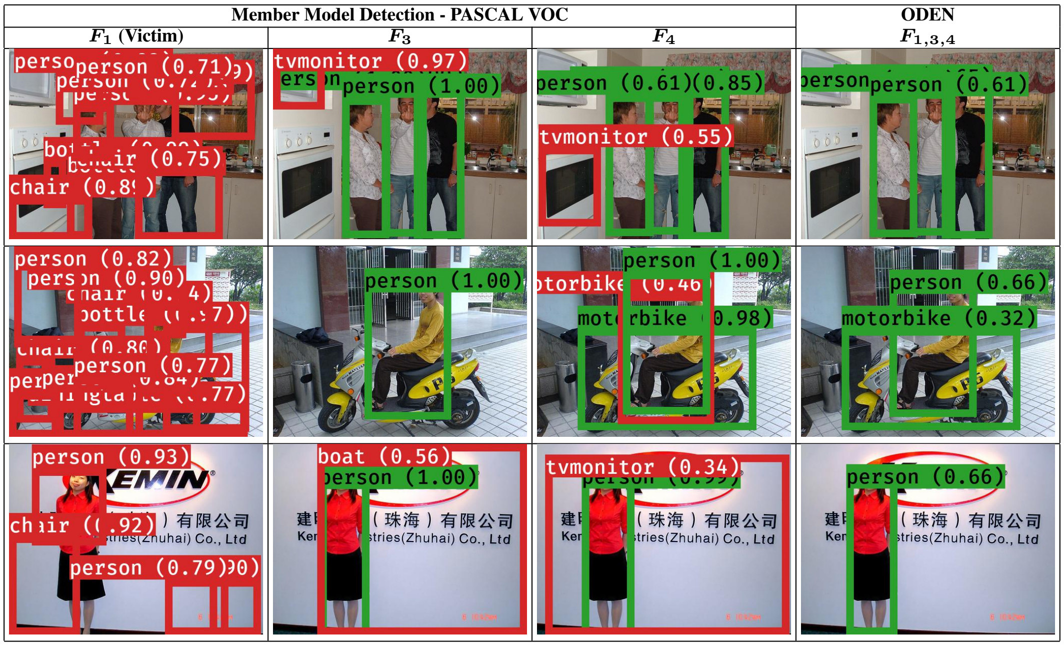
F Visual Examples: Evasion Attack - TOG (Untargeted)



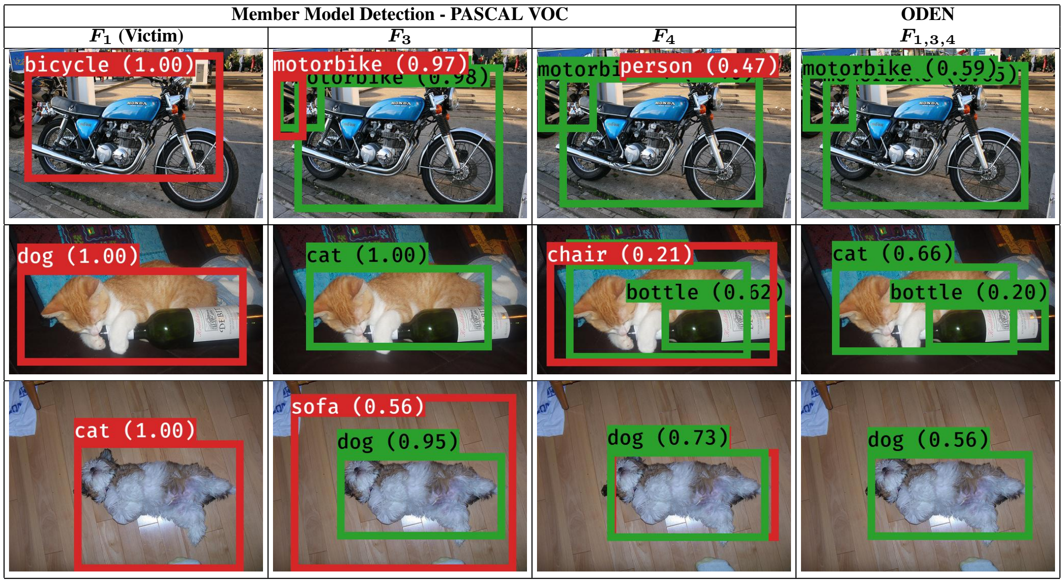
G Visual Examples: Evasion Attack - TOG (Vanishing)



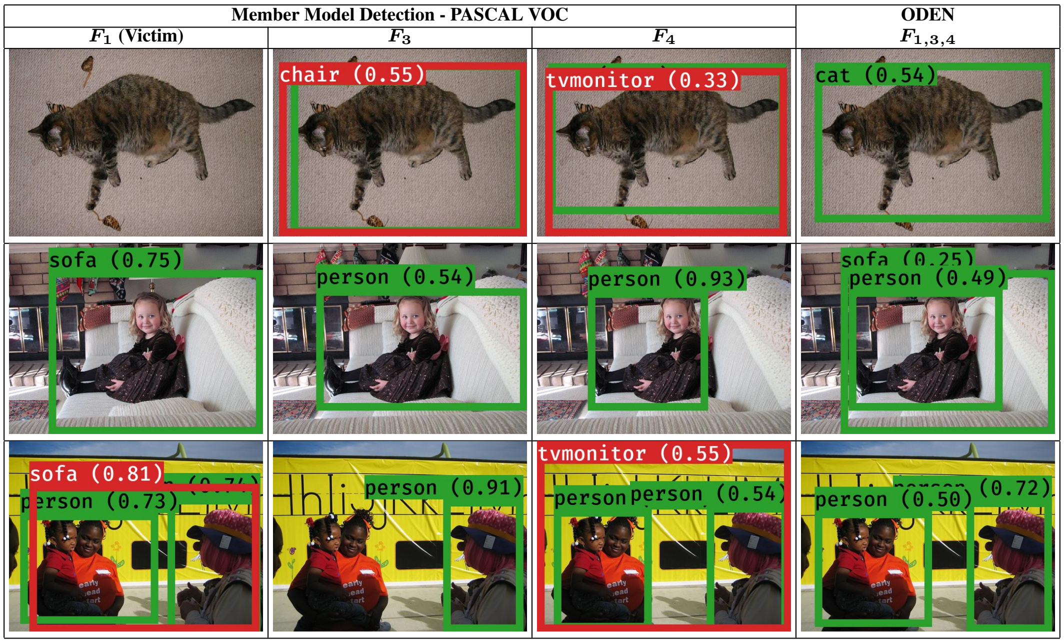
H Visual Examples: Evasion Attack - TOG (Fabrication)



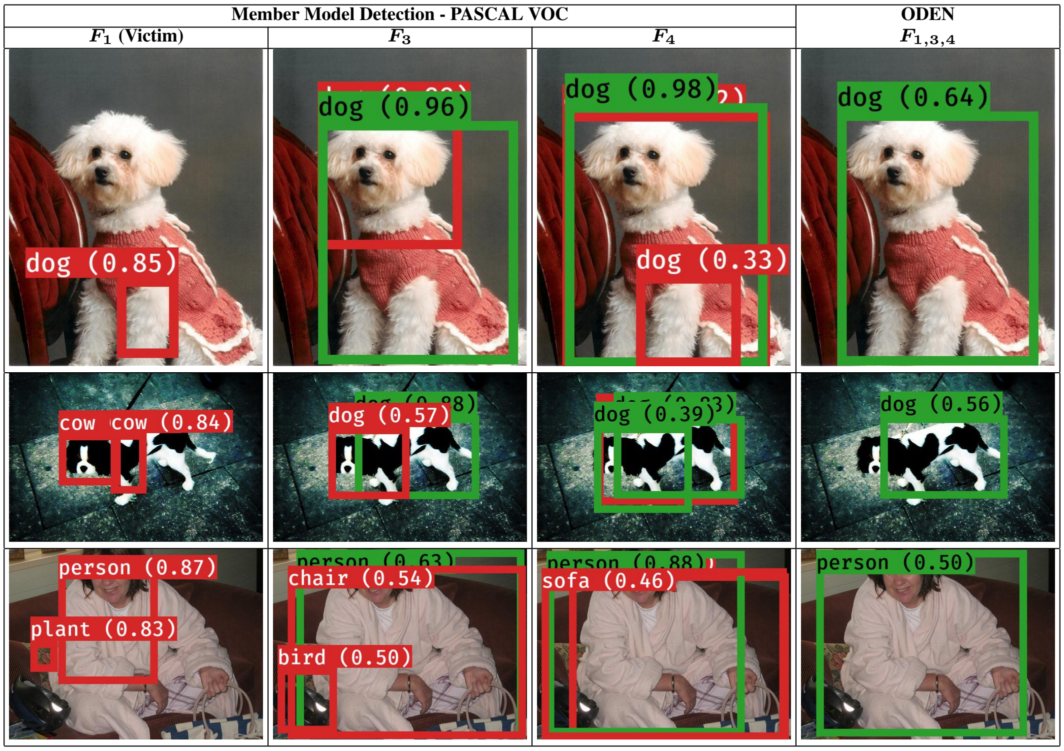
I VISUAL EXAMPLES: EVASION ATTACK - TOG (MISLABELING)



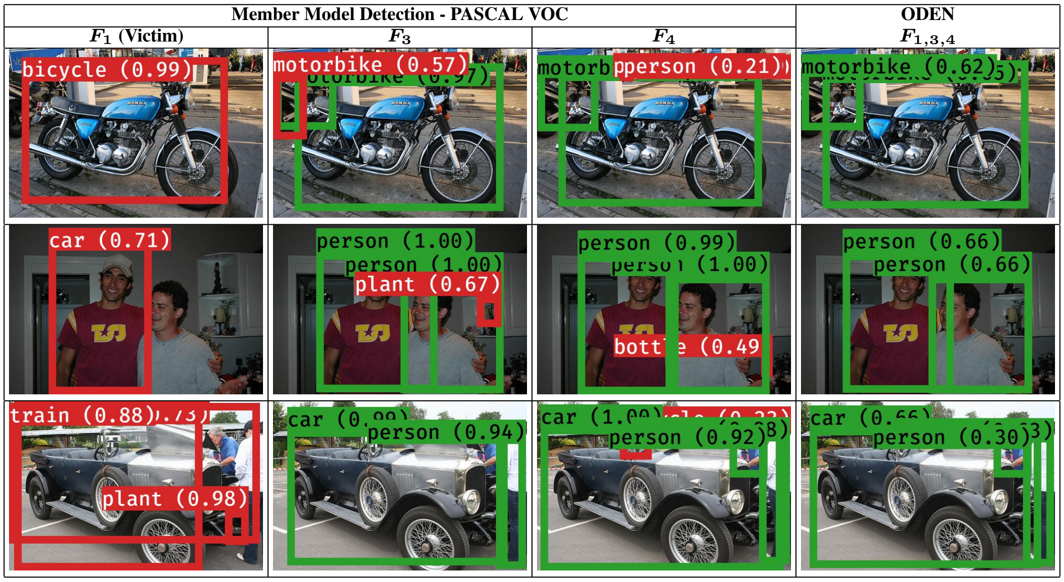
J VISUAL EXAMPLES: EVASION ATTACK - UEA



K Visual Examples: Evasion Attack - RAP



L Visual Examples: Evasion Attack - DAG



M Experiment Setup

The default IOU threshold is set to 0.5. We conduct grid search to find the hyperparameters for all detection combination schemes. The original test set for each dataset is randomly split with a ratio of 80:20. The smaller partition is used for hyperparameter tuning and training the learn-to-calibrate model for confidence calibration in ODEN. We repeat each experiment five times and report the mean performance to account for the randomness in partitioning datasets.

N Pseudocode

Algorithm 1 Candidate Detection Grouping

```
Input: $\boldsymbol{x}$: prediction query; $\boldsymbol{F}$: ensemble team
Output: $\boldsymbol{\Gamma}$: object groups
$\boldsymbol{\mathcal{M}} \leftarrow \textsc{HashMap}()$
for member $F_i \in \boldsymbol{F}$ do
    for candidate object $\boldsymbol{o}_j \in F_i(\boldsymbol{x})$ do
        $\boldsymbol{\mathcal{M}}[\ell_j].\textsc{Append}(\boldsymbol{o}_j)$
    end for
end for
$\boldsymbol{\Gamma} \leftarrow \textsc{List}()$
for class $\ell$ in $\boldsymbol{\mathcal{M}}$ do
    $\boldsymbol{\mathcal{O}}_\ell \leftarrow \textsc{SortByConfidenceDescending}(\boldsymbol{\mathcal{M}}[\ell])$
    $\boldsymbol{\Gamma}_\ell \leftarrow \textsc{List}(\textsc{List}(\boldsymbol{\mathcal{O}}_\ell.\textsc{PopFirst}()))$
    for candidate object $\boldsymbol{o}_j \in \boldsymbol{\mathcal{O}}_\ell$ do
        $\textsc{MaxGroupIndex} \leftarrow -1$
        $\textsc{MaxIOU} \leftarrow \mathcal{T}_{\text{IOU}}$
        for group $\boldsymbol{\gamma}_k \in \boldsymbol{\Gamma}_\ell$ do
            $\beta \leftarrow \beta_{WA}(\boldsymbol{o}_j, \boldsymbol{\gamma}_k)$
            $d_j \leftarrow$ the member detector producing $\boldsymbol{o}_j$
            if member $d_j$ not in $\boldsymbol{\gamma}$ and $\beta \geq \textsc{MaxIOU}$ then
                $\textsc{MaxGroupIndex} \leftarrow k$
                $\textsc{MaxIOU} \leftarrow \beta$
            end if
        end for
        if $\textsc{MaxGroupIndex} \geq 0$ then
            $\boldsymbol{\gamma}_{\textsc{MaxGroupIndex}}.\textsc{Append}(\boldsymbol{o}_j)$
        else
            $\boldsymbol{\Gamma}_\ell.\textsc{Append}(\textsc{List}(\boldsymbol{o}_j))$
        end if
    end for
    $\boldsymbol{\Gamma}.\textsc{Concatenate}(\boldsymbol{\Gamma}_\ell)$
end for
return $\boldsymbol{\Gamma}$
```

Algorithm 2 Negative Sample Determination

```
Input: $F_{\text{focal}}$: focal model; $\boldsymbol{x}$: validation sample; $\boldsymbol{\mathcal{G}}$: ground-truth objects; $\mathcal{T}_{\text{IOU}}$: an IOU threshold
Output: A boolean indicating whether $\boldsymbol{x}$ is a negative sample of focal model $F_{\text{focal}}$
$\boldsymbol{\mathcal{O}} \leftarrow$ SortByConfidenceDescending$(F_{\text{focal}}(\boldsymbol{x}))$
for detected object $\boldsymbol{o}_i \in \boldsymbol{\mathcal{O}}$ do
    MaxGTIndex $\leftarrow -1$
    MaxIOU $\leftarrow \mathcal{T}_{\text{IOU}}$
    for ground-truth object $\boldsymbol{g}_j \in \boldsymbol{\mathcal{G}}$ do
        $\eta \leftarrow$ IOU$(\boldsymbol{o}_i, \boldsymbol{g}_j)$
        if SameClass$(\boldsymbol{o}_i, \boldsymbol{g}_j)$ and $\eta \geq$ MaxIOU then
            MaxGTIndex $\leftarrow j$
            MaxIOU $\leftarrow \eta$
        end if
    end for
    if MaxGTIndex$\geq 0$ then
        $\boldsymbol{\mathcal{G}}$.Remove$(\boldsymbol{g}_j)$
        if $\boldsymbol{\mathcal{G}}$.Length() $== 0$ then
            return False
        end if
    else
        return True
    end if
end for
return not$(\boldsymbol{\mathcal{G}}$.Length() $== 0)$
```