Adversarial Attack and Disturbance Detection by Hadamard-Coded Output Representations for Object Detection and Semantic Segmentation

Lucas Görnhardt*, Timo Bartels*, Niklas Schwarz, and Tim Fingscheidt

Technische Universität Braunschweig, Germany
{lucas.goernhardt, timo.bartels, n.schwarz, t.fingscheidt}@tu-bs.de

Abstract. Conventional one-hot encodings often yield poorly calibrated models, being overconfident under attack, and letting entropy-based detection algorithms fail. Previous image classification works have demonstrated that Hadamard-coded output representations can improve adversarial robustness. However, attempts to integrate Hadamard codes into semantic segmentation fall far behind state-of-the-art models in mean intersection-over-union performance. Regarding object detection, such output encodings have not yet been investigated at all. Further, no prior art addressed intrinsic codeword inconsistencies or actually exploited intrinsic codeword redundancy. Accordingly, we first derive a novel decoding procedure for Hadamard codewords towards optimal class-wise probabilities, solving the underlying optimization problem by using the projection onto the probability simplex. Second, our optimization delivers a measure of prediction inconsistency. Third, we are the first to show how to exploit these inconsistencies for adversarial attack and disturbance detection. Fourth, we introduce `HadamardNet`, a framework employing Hadamard codes as output representations for semantic segmentation and object detection models and tasks. We conduct a comprehensive evaluation both on disturbances and adversarial attacks, *achieving state-of-the-art perturbation detection performance for both tasks in only a single detection pass*, while delivering equivalent or close-by reference performance on clean data. Code is available at https://github.com/ifnspaml/HadamardPerturbationDetection.

Keywords: adversarial attack, attack detection, semantic segmentation, object detection, Hadamard code

1 Introduction

In safety-critical domains such as autonomous driving, perception models must remain reliable under image perturbations, whether caused by adversarial attacks or natural disturbances. One direction to address this challenge is the introduction of redundancy into neural networks, for example through Hadamard-coded output representations [15–17, 39, 40, 44]. Hsu et al. [17] introduced Hadamard codes for large-scale image classification and reported improved accuracy.

*Marks equal contributions

Verma et al. [39] demonstrated improved calibration and adversarial robustness, while Hoyos et al. [16] confirmed these benefits across multiple attack types. However, these results do not directly transfer to semantic segmentation or object detection. The only Hadamard-based semantic segmentation method by Hoyos et al. [15] indicates faster convergence during training but falls far behind state-of-the-art mean intersection-over-union (mIoU) performance. Furthermore, existing work does not exploit the intrinsic redundancy of Hadamard codewords and its possible usage for perturbation detection.

Perturbation detection [13, 19, 23, 25, 27, 35, 36, 43, 45] remains an open challenge for safety-critical perception systems. A widely used baseline by Hendrycks et al. [13] employs maximum softmax probability, and Smith et al. [36] propose predictive entropy as detection signal. Liu et al. [25] introduced the energy score which provides a unified metric for out-of-distribution and adversarial detection. Feature squeezing [43] compares predictions on original and quantized or smoothed inputs and Ryu et al. [35] evaluate entropy differences after bit depth reduction, however, the former requires multiple forward passes and the latter degrades under strong perturbations. For semantic segmentation, Maag et al. train a detector based on entropy maps of both benign and FGSM-attacked images [27] and achieve strong results on FGSM-related attacks but generalize poorly to structurally different perturbations such as Metzen attacks [29]. For object detection, perturbation detection research remains limited [31, 38]. Existing approaches either depend on model-specific assumptions as proposed by Li et al. [23] who detect perturbations through context inconsistency or they are specifically tailored to a `Faster R-CNN` [33]. Yin et al. [45] propose heavy additional language models exploiting multi-object relationships.

In this work, we leverage the redundancy of Hadamard-coded output representations for perturbation detection. *Existing work provides no rigorous derivation for mapping predicted Hadamard codewords back to valid class probabilities while exploiting redundancy.* This is why we first propose a novel and fully probabilistic decoding procedure. We construct a linear equation system (LES) linking the unknown class probabilities to the predicted codeword and *introduce an error vector to make the typically inconsistent equation system solvable under probabilistic constraints.* The resulting optimization problem searches for the minimal error vector satisfying these constraints. Second, we show that the solution is a projection onto the probability simplex, which yields a novel decoding procedure for Hadamard codewords, delivering the error vector as an inconsistency measure of the network output. Third, based on this error vector, we propose an adversarial attack and disturbance detector that requires no adversarial training data, achieving state-of-the-art detection performance with a single detection pass with negligible computational overhead and being model- and task-agnostic. Fourth, we are first to provide a comprehensive evaluation of Hadamard output encodings for state-of-the-art semantic segmentation baselines and to extend them to object detection.

2 Related Works

Output Representations: For semantic segmentation and object detection, one-hot encoding is standard due to its simplicity and performance. Multiple alternatives have been proposed [1, 8, 11, 15–17, 21, 34, 39–41], including expert knowledge-based embeddings [1, 11], learned embeddings [40, 41], and data-independent embeddings such as error-correcting codes (ECCs) [15, 17, 21, 34]. Dietterich et al. [8] introduce ECCs to improve multi-class performance, and Kong et al. [21] show that ECCs reduce bias and variance across training subsets. Within ECCs, Hadamard codes have drawn particular interest [14–17, 39, 40, 44] due to their constant Hamming distance. Our work extends this line by mathematically exploiting Hadamard code redundancy to obtain a perturbation-sensitive inconsistency measure, along with a rigorous mathematical derivation.

Hadamard-Coded Output Representations: Hsu et al. [17] first introduced Hadamard codes [32] for large-scale image classification, showing improved accuracy and label space efficiency. Yang et al. [44] reported improved class discriminability in the penultimate layer. Verma et al. [39] evaluated activation functions for Hadamard-coded outputs and found that tanh improves calibration and adversarial robustness. Hoyos et al. [16] confirmed improved robustness across FGSM [12], PGD [22], and other attacks. Wan et al. [40] further increased robustness by learning codewords starting from Hadamard codes. Hoyos et al. [15] applied Hadamard codes to semantic segmentation, reporting faster convergence, however, under very short training schedules, substantially lower mIoU than standard baselines, and with inconsistencies between paper and released code. Their decoding uses a matrix multiplication and softmax but does not leverage codeword redundancy. No prior work applies Hadamard codes to object detection. We address these gaps by, for the first time, deriving a redundancy-aware decoding method, producing both valid class probabilities and an inconsistency signal, and by extending Hadamard-coded outputs to object detection.

Adversarial Attack Detection: Adversarial attack and disturbance detection mostly relies on uncertainty in the model output. Hendrycks et al. [13] propose an indicator based on maximum softmax probability and Smith et al. [36] an entropy-based indicator. Liu et al. [25] introduce the energy score for attack detection. Transformation-based approaches include feature squeezing [43], which requires multiple forward passes, and entropy difference methods by Ryu et al. [35], which degrades under stronger attacks. For semantic segmentation, Maag et al. [27] trained a detector on FGSM [12] perturbations using entropy maps but generalize poorly to different attacks such as Metzen [29]. For object detection, existing methods are either architecture-specific, such as context inconsistency detection for `Faster R-CNN` by Li et al. [23] or they employ heavy language models for multi-object reasoning [45]. Our proposed detector does not depend on model-specific assumptions, or heavy modules, but is even task-agnostic by deriving a detection signal directly from the Hadamard-code redundancy.

3 Methods

Here, we describe how we employ Hadamard output encodings for perception models and utilize the redundancy for attack and disturbance detection.

3.1 Hadamard Codes for Classification

We propose to utilize ECC-based output representations for perception models using Hadamard codes [32], the resulting network we denote as `HadamardNet`. Due to its favorable mathematical properties, the Hadamard code is particularly well suited for the output encoding of perception models for classification, such as semantic segmentation or object detection. Hadamard codes provide large, equal Hamming distance between codewords, while both encoding and decoding are computationally efficient. A Hadamard code is of length $L = 2^k$ with $k \in \mathbb{N}$. The Hamming distance between each pair of codewords is 2^{k-1}. Hadamard codes are generated using Sylvester's construction [37] of the bipolar Hadamard matrix $\check{\mathbf{H}}^{(L)} \in \{-1, 1\}^{L \times L}$, which is detailed with an example in Supplement A. We use S columns $\check{\mathbf{H}}_s^{(L)} = (\check{H}_{s,\ell}^{(L)}) \in \{-1, 1\}^L$ of the Hadamard matrix $\check{\mathbf{H}}^{(L)} = (\check{\mathbf{H}}_s^{(L)})$ to obtain the unipolar representations $H_{s,\ell}^{(L)} = \frac{1}{2}(\check{H}_{s,\ell}^{(L)} + 1)$ with columns

$$\mathbf{H}_s^{(L)} = (H_{s,\ell}^{(L)}) \in \mathbb{B}^L \tag{1}$$

and column index $s \in \mathcal{S} = \{1, \ldots, S\}$ representing one of S classes, $\ell \in \mathcal{L} = \{1, \ldots, L\}$ being the (bit) index within the codeword, and $\mathbb{B} = \{0, 1\}$. In this work, we will utilize these *column vectors (1) as target labels* for classification tasks such as semantic segmentation or object detection, which leads to our proposed `HadamardNet` architecture with a modified output representation. Since L is always a power of two, it is not guaranteed that the codeword length L and S are equal. The codeword length must only satisfy

$$L \geq S. \tag{2}$$

Using (1), our *unipolar* Hadamard matrix for *classification purposes* is given by

$$\mathbf{H} = \left(\mathbf{H}_1^{(L)} \; \mathbf{H}_2^{(L)} \cdots \mathbf{H}_S^{(L)} \right) \in \mathbb{B}^{L \times S}. \tag{3}$$

In analogy, the *bipolar* Hadamard matrix for classification results in

$$\check{\mathbf{H}} = \left(\check{\mathbf{H}}_1^{(L)} \; \check{\mathbf{H}}_2^{(L)} \cdots \check{\mathbf{H}}_S^{(L)} \right) \in \{-1, 1\}^{L \times S}. \tag{4}$$

Using $\mathbf{1}^{(L \times S)} = \{1\}^{L \times S}$, the two notations (3) and (4) are related by

$$\check{\mathbf{H}} = 2\mathbf{H} - \mathbf{1}^{(L \times S)}. \tag{5}$$

Since the matrix $\check{\mathbf{H}}(4)$ is orthogonal, the following holds:

$$\check{\mathbf{H}}^\mathsf{T} \cdot \check{\mathbf{H}} = L \cdot \mathbf{I}, \tag{6}$$

where $\mathbf{I} \in \mathbb{B}^{S \times S}$ denotes the identity matrix and $(\,)^\mathsf{T}$ is the transpose.*

*Note that $\check{\mathbf{H}} \cdot \check{\mathbf{H}}^\mathsf{T} \neq S \cdot \mathbf{I}$.

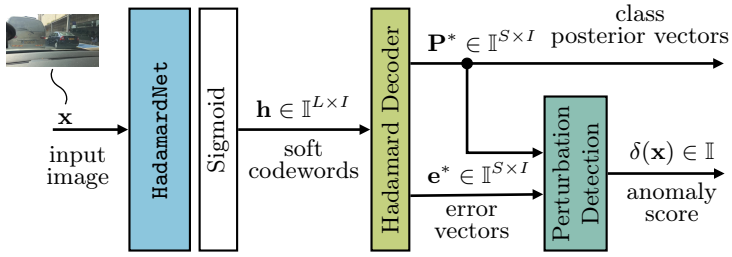

Fig. 1: Inference setup using the `HadamardNet`. The Hadamard decoder is performed according to Sec. 3.2. The perturbation detection is detailed in Sec. 3.3.

3.2 `HadamardNet` and Hadamard Decoder

Following Fig. 1, our proposed `HadamardNet` predicts a so-called *soft* codeword

$$\mathbf{h} = (\mathbf{h}_i) = (h_{i,\ell}) \in \mathbb{I}^{L \times I} \tag{7}$$

for each position, i.e., a pixel or an object, $i \in \mathcal{I} = \{1, \ldots, I\}$, where we have $\mathbb{I} = [0, 1]$ due to the sigmoid and $I \in \mathbb{N}$ is either the number of pixels or the number of objects. These predicted soft codewords $\mathbf{h}$ are then decoded (Fig. 1, green box) to deliver the optimal class posterior probabilities

$$\mathrm{P}^* = (\mathrm{P}_i^*) = (\mathrm{P}_{i,s}^*) \in \mathbb{I}^{S \times I}, \tag{8}$$

with $\mathrm{P}_{i,s}^* = \mathrm{P}_i^*(s|\mathbf{x})$, $i \in \mathcal{I}$, and $\mathbf{x} \in \mathbb{I}^{H \times W \times C}$ being the input image of height H, width W, and with $C = 3$ color channels. For each pixel or object $i \in \mathcal{I}$, the class posteriors have to fulfill the stochastic constraints

$$\sum_{s \in \mathcal{S}} \mathrm{P}_{i,s}^* = 1 \tag{9}$$

$$\text{and} \quad \mathrm{P}_{i,s}^* \geq 0 \qquad \forall s \in \mathcal{S}. \tag{10}$$

Optimization Problem: The estimation of the optimal class posterior probabilities $\mathrm{P}_i^*(s|\mathbf{x})$ given the predicted soft codewords $\mathbf{h} = (\mathbf{h}_i)$ is now step-by-step derived and outlined in the remainder of this section.

In the unlikely case that the network predicts an error-free soft codeword $\mathbf{h}_i^* = (h_{i,\ell}^*)$, which does not contain inconsistencies, the soft codeword fulfills $\mathbf{h}_i^* \in \mathcal{H}^* = \{\sum_{s \in S} \mathbf{H}_s^{(L)} \cdot P_{i,s} \mid \mathbf{P}_i = (P_{i,s}) \in \mathcal{P}\}$ and therefore

$$h_{i,\ell}^* = \sum_{s \in \mathcal{S}} H_{s,\ell}^{(L)} \cdot \mathrm{P}_{i,s}. \tag{11}$$

Here, $\mathcal{P} = \left\{\mathbf{P}_i = (P_{i,s}) \in \mathbb{I}^S \;\middle|\; \mathrm{P}_{i,s} \geq 0 \; \forall s \in \mathcal{S}, \; \sum_{s \in \mathcal{S}} \mathrm{P}_{i,s} = 1\right\}$ is the set of all valid class-wise probability distributions satisfying the constraints (9), (10). In

Supplement B we show a detailed derivation of (11) using our probabilistic approach. Given the unipolar Hadamard matrix for classification $\mathbf{H}$ (3), and the *error-free* soft codeword $\mathbf{h}_i^*$ for pixel or object i, we obtain a well-determined LES that we aim to solve for $\mathbf{P}_i \in \mathcal{P}$ for each pixel or object $i \in \mathcal{I}$:

$$\mathbf{H} \cdot \mathbf{P}_i = \mathbf{h}_i^*. \tag{12}$$

Since the prediction $\mathbf{h}_i = (h_{i,\ell})$ for each bit at position ℓ is intended to be mutually independent, contradictions can arise and potentially an output soft codeword $\mathbf{h}_i \notin \mathcal{H}^*$, making the equation system unsolvable under the probabilistic constraints (9), (10). To address this, we introduce an error vector $\check{\mathbf{e}}_i = (\check{e}_{i,\ell}) \in [-1,1]^L$ for each predicted soft codeword $\mathbf{h}_i$, with $\mathbf{h}_i^* = \mathbf{h}_i + \check{\mathbf{e}}_i$, and obtain

$$\boxed{\mathbf{H} \cdot \mathbf{P}_i = \mathbf{h}_i + \check{\mathbf{e}}_i.} \tag{13}$$

This allows (13) to become solvable for any $\mathbf{h}_i \in \mathbb{I}^L$ subject to the constraints (9), (10). Using (5), (6), and (9), we can rearrange (13) for $\mathbf{P}_i$, yielding

$$\boxed{\mathbf{P}_i = \tilde{\mathbf{P}}_i + \mathbf{e}_i,} \tag{14}$$

with $\tilde{\mathbf{P}}_i = \frac{1}{L}\check{\mathbf{H}}^\mathsf{T}\check{\mathbf{h}}_i \in [-1,1]^S$, $\mathbf{e}_i = \frac{2}{L}\check{\mathbf{H}}^\mathsf{T}\check{\mathbf{e}}_i = \frac{2}{L}(\check{\mathbf{e}}_i^\mathsf{T}\check{\mathbf{H}})^\mathsf{T} \in \mathbb{R}^S$ and $\check{\mathbf{h}}_i = 2\mathbf{h}_i - \mathbf{1}^{(L)} \in [-1,1]^L$. Note that $\tilde{\mathbf{P}}_i$ does not necessarily satisfy the constraints (9), (10). The rearrangement of (13) to obtain (14) is detailed in Supplement C. By introducing new unknowns $\check{\mathbf{e}}_i$, the LES (13) becomes under-determined, resulting in multiple feasible solutions for $\mathbf{P}_i$ and $\check{\mathbf{e}}_i$. Therefore, we employ (13) and search for the minimum squared L_2 error $\min_{\mathbf{P}_i \in \mathcal{P}} \|\check{\mathbf{e}}_i\|_2^2$, which is the optimization problem we aim to solve.

Solving the Optimization Problem: We solve this optimization problem by just minimizing $\|\mathbf{e}_i\|_2^2$ in (14), which we prove to be equivalent by showing that $\arg\min_{\mathbf{P}_i \in \mathcal{P}} \|\mathbf{e}_i\|_2^2 = \arg\min_{\mathbf{P}_i \in \mathcal{P}} \|\check{\mathbf{e}}_i\|_2^2$ in Supplement D.

To minimize $\|\mathbf{e}_i\|_2^2$ in (14), we utilize the projection $\boldsymbol{\phi}()$ onto the probability simplex by Michelot [30] and obtain the optimal class posterior probabilities delivered by the Hadamard decoder in Fig. 1:

$$\boxed{\mathbf{P}_i^* = \boldsymbol{\phi}(\tilde{\mathbf{P}}_i) = \arg\min_{\mathbf{P}_i \in \mathcal{P}} \|\mathbf{P}_i - \tilde{\mathbf{P}}_i\|_2^2 = \arg\min_{\mathbf{P}_i \in \mathcal{P}} \|\mathbf{e}_i\|_2^2.} \tag{15}$$

Please refer to Supplement E for a detailed description of the simplex projection. Substituting $\mathbf{P}_i$ in (14) by $\mathbf{P}_i^*$ delivered by the simplex projection (15) yields the smallest class-wise error vector

$$\mathbf{e}_i^* = \mathbf{P}_i^* - \tilde{\mathbf{P}}_i \in \mathbb{R}^S. \tag{16}$$

As depicted in Fig. 1, the class-wise probabilities $\mathbf{P}_i^*$ and error vector $\mathbf{e}_i^*$ are the two outputs of our Hadamard decoding. The error vector represents the inconsistencies of the individual binary predictions of the `HadamardNet`. Since all operations in the Hadamard decoding are differentiable, we can utilize the decoded probabilities $\mathbf{P}_i^*$, the error vector $\mathbf{e}_i^*$, as well as the predicted soft codeword $\mathbf{h}_i$ for the training.

3.3 Adversarial Attack and Disturbance Detection

As depicted in Fig. 1, our perturbation detection approach uses the output of our Hadamard decoding $\mathbf{P}^* = (\mathbf{P}_i^*)$ (15) and $\mathbf{e}^* = (\mathbf{e}_i^*)$ (16) as inputs and outputs an anomaly score $\delta(\mathbf{x}) \in \mathbb{I}$ for each image $\mathbf{x}$. During inference, an image is flagged as adversarial if $\delta(\mathbf{x}) \geq \theta$ for a given threshold $\theta \geq 0$.

Error Vector Anomaly Score: A simple first approach uses the L_1 norm of the error vector, averaged over all pixels or objects, as anomaly score

$$\delta(\mathbf{x}) = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \|\mathbf{e}_i^*\|_1. \tag{17}$$

Regression Anomaly Score: Clean images show a strong relationship between error magnitude $\|\mathbf{e}_i^*\|_1$ and class posterior probabilities $\mathbf{P}_i^*$. An analysis of this relationship is provided in Supplement F. Pixels or objects with low maximum posterior probability ($\mathrm{P}_{i,s^*}^* = \max_{s \in \mathcal{S}} \mathrm{P}_{i,s}^* \approx \frac{1}{S}$) typically exhibit larger inconsistencies $\|\mathbf{e}_i^*\|_1$. Thus, a detector based solely on (17) would tend to falsely alarm on images containing many low-confidence predictions. Under perturbations, the distribution of $\|\mathbf{e}_i^*\|_1$ changes from the clean-data distribution. We introduce a regression model $\mathcal{F}$ (details in Supplement G) that, for each pixel or object i, predicts the error magnitude $\widehat{\|\mathbf{e}_i^*\|}_1 = \mathcal{F}([\mathrm{P}_{i,s^*}^*, H(\mathbf{P}_i^*)]) \geq 0$ from the maximum posterior probability P_{i,s^*}^* and the entropy of the class posterior $H(\mathbf{P}_i^*) = -\sum_{s \in \mathcal{S}} \mathrm{P}_{i,s}^* \cdot \log(\mathrm{P}_{i,s}^*) \geq 0$. The per-image prediction error is then $r(\mathbf{x}) = \frac{1}{\sqrt{|\mathcal{I}|}} \cdot \sum_{i \in \mathcal{I}} L_2(\|\mathbf{e}_i^*\|_1, \widehat{\|\mathbf{e}_i^*\|}_1) \geq 0$. Using clean training data $\mathcal{D}_{\text{train}}$, we compute the mean prediction error $\mu^{\text{train}} = \frac{1}{|\mathcal{D}_{\text{train}}|} \cdot \sum_{\mathbf{x} \in \mathcal{D}_{\text{train}}} r(\mathbf{x}) \geq 0$. The anomaly score is defined as the absolute deviation from the training mean

$$\delta(\mathbf{x}) = |r(\mathbf{x}) - \mu^{\text{train}}|. \tag{18}$$

Quantile Anomaly Score: Using the regression anomaly score, the regression model $\mathcal{F}$ (details in Supplement G) predicts only the expected value of $\|\mathbf{e}_i^*\|_1$ for a pixel or object i. However, this formulation does not account for variability, and thus yields on average high anomaly scores in case of large standard deviations of $\|\mathbf{e}_i^*\|_1$. To explicitly incorporate standard deviation, our third approach uses a regression network $\mathcal{F}$ that predicts the conditional 85% quantile of $\|\mathbf{e}_i^*\|_1$ over all pixels or objects. This means that $\mathcal{F}([\mathrm{P}_{i,s^*}^*, H(\mathbf{P}_i^*)]) = E_i^{(0.85)} \geq 0$ provides, for each pixel or object $i \in \mathcal{I}$, the error vector L_1 norm $E_i^{(0.85)}$ for which 85% of $\|\mathbf{e}_i^*\|_1$, $i \in \mathcal{I}$, are lower. To obtain the anomaly score, we then count the fraction of pixels or objects whose L_1 norm of the error vector exceeds the predicted 85% threshold:

$$\delta(\mathbf{x}) = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I},\, \|\mathbf{e}_i^*\|_1 > E_i^{(0.85)}} 1. \tag{19}$$

3.4 `HadamardNet` and Attack/Disturbance Detector Training

`HadamardNet`: In our final `HadamardNet`, training is based on the encoded-space loss

$$J^{\text{cls}} = J^{\text{ENC}}(\mathbf{h}_i, \overline{\mathbf{H}}_i), \tag{20}$$

computed between the predicted soft codeword $\mathbf{h}_i$ and the target Hadamard codeword $\overline{\mathbf{H}}_i = \mathbf{H}^{(L)}_{s=\overline{s}_i}$ (1), with $\overline{s}_i$ denoting the class label of pixel or object i. Details on J^{cls} (51) and J^{ENC} (52) are given in Supplement H.

Detector: After `HadamardNet` training, the regression model $\mathcal{F}$ *is trained exclusively on clean images* by minimizing $J^{\text{MSE}}(\|\mathbf{e}^*_i\|_1, \widehat{\|\mathbf{e}^*_i\|}_1)$ for the regression anomaly score (18), and by minimizing the 85% quantile loss [20]

$$J^{(0.85)} = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \max\big(0.85\,(\|\mathbf{e}^*_i\|_1 - E^{(0.85)}_i),\, 0.15\,(E^{(0.85)}_i - \|\mathbf{e}^*_i\|_1)\big) \tag{21}$$

for the quantile anomaly score (19).

4 Experimental Setup and Metrics

Semantic Segmentation: For semantic segmentation, `SegFormer MiT-B0` [42] is used in most of our studies, due to its fast training and strong performance. We additionally report results for `DeepLabv3+` [4]. We employ Cityscapes [7] training ($\mathcal{D}^{\text{CS}}_{\text{train}}$) and validation ($\mathcal{D}^{\text{CS}}_{\text{val}}$) sets, BDD100K [46] training ($\mathcal{D}^{\text{BDD,Seg}}_{\text{train}}$) and validation ($\mathcal{D}^{\text{BDD,Seg}}_{\text{val}}$) sets, and ADE20K [47] training ($\mathcal{D}^{\text{ADE20K}}_{\text{train}}$) and validation ($\mathcal{D}^{\text{ADE20K}}_{\text{val}}$) sets. Segmentation quality is measured using mean intersection over union (mIoU). The `mmsegmentation` framework [6] is used to train all models using default hyperparameters, except for the loss, where we use exclusively J^{cls} (20). Supplement I provides further training details.

Object Detection: For object detection, we consider `Faster R-CNN` [33] as a standard two-stage CNN-based detector widely used in the adversarial attack and detection literature [5, 23, 45]. We further extend our approach to `DETR` [2], an end-to-end single-stage transformer-based detector, demonstrating that our method is architecture-agnostic. We employ the combined Pascal VOC 2007 and 2012 trainval [9, 10] ($\mathcal{D}^{\text{VOC}}_{\text{train}}$) and Pascal VOC 2007 test ($\mathcal{D}^{\text{VOC}}_{\text{val}}$) sets, BDD100K [46] train ($\mathcal{D}^{\text{BDD,OD}}_{\text{train}}$) and test ($\mathcal{D}^{\text{BDD,OD}}_{\text{test}}$) sets, and COCO [24] train ($\mathcal{D}^{\text{COCO}}_{\text{train}}$) and validation ($\mathcal{D}^{\text{COCO}}_{\text{val}}$) sets. Object detection quality is measured using the COCO average precision (AP) metric [24]. Specifically, AP denotes the mean average precision averaged over multiple IoU thresholds from 0.50 to 0.95 in steps of 0.05. The models are trained by minimizing

$$J^{\text{OD}} = \lambda^{\text{reg}} J^{\text{reg}} + \lambda^{\text{cls}} J^{\text{cls}}, \tag{22}$$

where $\lambda^{\text{reg}}, \lambda^{\text{cls}} > 0$ weigh the bounding box regression loss J^{reg} and classification loss J^{cls} (20). Since Hadamard encodings are applied only to the classification branch, we use the standard J^{reg} from [2, 33] for `Faster R-CNN` and `DETR`. For classification, we employ `HadamardNet` as in the segmentation experiments, with J^{cls} in (22) as defined in (20). Unless noted otherwise, all hyperparameters follow the defaults of the `mmdetection` framework [3]. Supplement I provides further training details. For both, semantic segmentation and object detection, we report 95% confidence intervals obtained over three seeds.

Perturbation Detection and Attacks: The perturbation detection regression network $\mathcal{F}$ is trained in a second stage after the segmentation or object detection model have been trained. For this purpose, we use the same training set as in the first stage, which again only contains clean data. We train $\mathcal{F}$ for 20 epochs using the AdamW [26] optimizer, more details in Supplement I.

For semantic segmentation, we employ FGSM [12], PGD [28], and Metzen [29] attacks. For object detection, we employ TOG [5] untargeted, fabrication, vanishing and mislabeling attacks. Additionally, for both, we employ Gaussian noise and salt and pepper disturbances. For all experiments evaluating perturbation detection performance, the corresponding perturbation strength ϵ [18] is explicitly stated, which is the only hyperparameter used for Gaussian noise, salt and pepper noise, and the FGSM attack. The PGD, TOG, and Metzen attacks use additional hyperparameters. For PGD and TOG, we use 10 iterations, each with a step size of $\alpha_\epsilon = \frac{\epsilon}{4}$. For the Metzen attack, we use 60 iterations, a fixed step size of $\alpha_\epsilon^{\text{Metzen}} = \frac{1}{255}$, and match all remaining hyperparameters to those of Metzen et al. [29]. Details on our definition of ϵ and the generation of the attacks and disturbances can be found in Supplement J.

Perturbation detection performance is evaluated using the true positive rate (TPR), false positive rate (FPR), and the area under the receiver operating characteristic curve (AuROC). In addition, to assess generalization across different perturbations and strengths, we report *global* AuROC. To compute these global metrics, we aggregate detection outputs either across all perturbation strengths for a fixed perturbation, or across all perturbations and strengths, and evaluate AuROC on the resulting combined set. *This protocol enforces a single operating threshold across all perturbations and strengths, reflecting practical deployment conditions where the perturbation configuration is unknown.*

5 Experimental Evaluation and Discussion

Semantic Segmentation: Note that in Supplement K, we conduct an ablation study on the variants and combinations of the classification loss J^{cls} (20) components and adopt $J^{\text{cls}} = J^{\text{ENC}}$ with $J^{\text{ENC}} = J^{\text{MSE}}$ *for all following experiments both in the semantic segmentation and object detection task.*

In Tab. 1, we compare the perturbation detection performance of all three variants of our proposed method described in Sec. 3.3 with state-of-the-art detection approaches for semantic segmentation across multiple perturbation types

Table 1: Perturbation detection for semantic segmentation on $\mathcal{D}_{\text{val}}^{\text{CS}}$ using `SegFormer MiT-B0`. Global AuROC (best in bold) per perturbation across all strengths $\epsilon \in \{1, 2, 4, 8, 16\}$ and across all perturbations and strengths in %.

Method	Optim. on	Global AuROC (%) FGSM	PGD	Metzen	Gauss.	S&P	Global AuROC	FLOPs (G)
Multi-pass detector								
Xu et al. [43]	-	$97.1^{\pm 0.3}$	$98.2^{\pm 0.4}$	$78.0^{\pm 1.0}$	$64.2^{\pm 1.3}$	$65.4^{\pm 0.8}$	$80.7^{\pm 0.6}$	366
Single-pass detector								
Entropy [36]	-	$95.7^{\pm 0.5}$	$92.2^{\pm 1.1}$	$45.0^{\pm 1.6}$	$62.9^{\pm 0.6}$	$62.1^{\pm 0.5}$	$71.6^{\pm 0.2}$	122
Max posterior [13]	-	$95.5^{\pm 0.4}$	$91.7^{\pm 1.2}$	$45.2^{\pm 1.8}$	$62.8^{\pm 0.1}$	$61.6^{\pm 0.5}$	$71.4^{\pm 0.2}$	122
Maag et al. [27]	FGSM	$\mathbf{98.2}^{\pm 0.0}$	$94.9^{\pm 2.7}$	$48.0^{\pm 2.9}$	$67.6^{\pm 0.6}$	$64.3^{\pm 0.2}$	$74.8^{\pm 0.9}$	122
Ours: Error (17)	-	$95.7^{\pm 0.8}$	$95.2^{\pm 0.8}$	$68.4^{\pm 1.9}$	$65.0^{\pm 0.8}$	$60.6^{\pm 0.9}$	$77.1^{\pm 0.7}$	123
Ours: Reg. (18)	-	$89.6^{\pm 2.8}$	$94.1^{\pm 0.9}$	$72.7^{\pm 3.3}$	$59.9^{\pm 1.4}$	$55.9^{\pm 1.0}$	$74.6^{\pm 1.7}$	124
Ours: Quantile (19)	-	$95.9^{\pm 0.7}$	$\mathbf{96.1}^{\pm 0.5}$	$\mathbf{75.9}^{\pm 2.1}$	$\mathbf{68.5}^{\pm 1.1}$	$\mathbf{64.4}^{\pm 1.0}$	$\mathbf{80.2}^{\pm 0.9}$	124

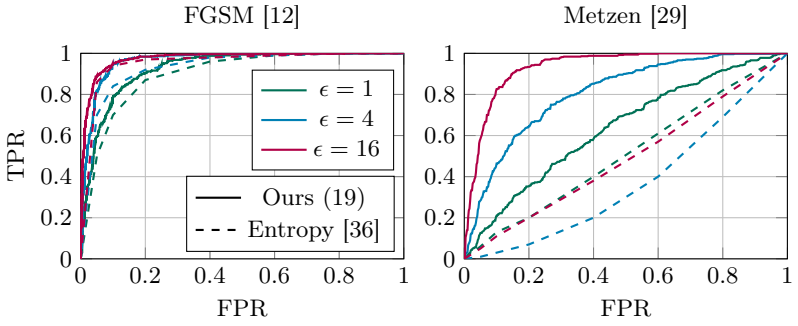

Fig. 2: ROC for our proposed quantile detection (19) for semantic segmentation, left for the FGSM [12] attack, right for the Metzen [29] attack at perturbation strengths $\epsilon \in \{1, 4, 16\}$. All results are obtained using a `SegFormer MiT-B0` model [42] trained on $\mathcal{D}_{\text{train}}^{\text{CS}}$ and evaluated on perturbed images from $\mathcal{D}_{\text{val}}^{\text{CS}}$.

and strengths ϵ. All methods employ a `SegFormer MiT-B0` [42] trained on $\mathcal{D}_{\text{train}}^{\text{CS}}$ and evaluated on perturbed samples from $\mathcal{D}_{\text{val}}^{\text{CS}}$. We report the perturbation detection accuracy measured by global AuROC (%) per perturbation across all strengths and across all perturbations and strengths. Perturbed mIoU and detection performance per strength ϵ are provided in Supplement L. We observe that the three-pass detector by Xu et al. [43] perform best in three of five perturbations, however, naturally requiring three times the computational complexity of the single-pass detectors. In the single-pass regime, Maag et al. [27] excel in FGSM (where they optimized for). The entropy [36] and the max posterior [13] detectors are lacking behind, so do Maag et al. for attacks other than FGSM. Among our methods, our quantile detector is ahead, delivering also the best global AuROC (80.2%), which is statistically equivalent to the multi-pass per-

Table 2: One-hot vs. Hadamard output encodings on **semantic segmentation** models on $\mathcal{D}_{\text{val}}^{\text{CS}}$ and $\mathcal{D}_{\text{val}}^{\text{BDD,Seg}}$. Global AuROC (best in bold, second-best underlined) across all perturbations and strengths $\epsilon \in \{1, 2, 4, 8, 16\}$ and mIoU in %.

Model	Output representation / detector	Cityscapes (CS) mIoU clean	Cityscapes (CS) AuROC attack	BDD mIoU clean	BDD AuROC attack	Size (M)	FLOPs (G)
`DeepLabv3+` [4]	One-hot/entropy [36]	$80.2^{\pm 0.2}$	$72.5^{\pm 0.1}$	$59.8^{\pm 1.5}$	$59.2^{\pm 3.2}$	43.6	1413
	One-hot/max post. [13]	$80.2^{\pm 0.2}$	$72.2^{\pm 1.4}$	$59.8^{\pm 1.5}$	$60.3^{\pm 2.9}$	43.6	1413
	Hadamard/(17) (ours)	$80.1^{\pm 0.1}$	$80.4^{\pm 1.1}$	$57.2^{\pm 0.5}$	$\mathbf{70.9}^{\pm 2.8}$	43.6	1414
	Hadamard/(18) (ours)	$80.1^{\pm 0.1}$	$\mathbf{81.0}^{\pm 0.4}$	$57.2^{\pm 0.5}$	$63.4^{\pm 6.6}$	43.6	1415
	Hadamard/(19) (ours)	$80.1^{\pm 0.1}$	$\underline{80.7}^{\pm 0.2}$	$57.2^{\pm 0.5}$	$\underline{64.8}^{\pm 4.6}$	43.6	1415
`SegFormer-B0` [42]	One-hot/entropy [36]	$76.4^{\pm 0.2}$	$72.1^{\pm 1.4}$	$59.5^{\pm 0.3}$	$62.0^{\pm 3.0}$	3.7	122
	One-hot/max post. [13]	$76.4^{\pm 0.2}$	$72.1^{\pm 1.4}$	$59.5^{\pm 0.3}$	$62.8^{\pm 2.7}$	3.7	122
	Hadamard/(17) (ours)	$76.5^{\pm 0.3}$	$\underline{77.1}^{\pm 0.7}$	$57.6^{\pm 1.5}$	$\mathbf{70.3}^{\pm 0.6}$	3.7	123
	Hadamard/(18) (ours)	$76.5^{\pm 0.3}$	$74.6^{\pm 1.7}$	$57.6^{\pm 1.5}$	$64.5^{\pm 1.5}$	3.7	124
	Hadamard/(19) (ours)	$76.5^{\pm 0.3}$	$\mathbf{80.2}^{\pm 0.9}$	$57.6^{\pm 1.5}$	$\underline{67.2}^{\pm 0.9}$	3.7	124

formance (80.7%) of complex Xu et al. [43]. The second-best of our approaches is the quite balanced error anomaly score (17) method (77.1% global AuROC) which for each single perturbation delivers a global AuROC of more than 60%.

In Fig. 2, we display receiver operating characteristic (ROC) curves for FGSM [12] (left) and Metzen [29] (right) attacks on semantic segmentation models for various attack strengths ϵ for our quantile regression approach (19) and the entropy baseline [36]. The results are obtained using the `SegFormer MiT-B0` model [42], which is trained on $\mathcal{D}_{\text{train}}^{\text{CS}}$ and evaluated on attacked images from $\mathcal{D}_{\text{val}}^{\text{CS}}$. The ROC curves are obtained by varying the decision threshold θ applied to the anomaly score $\delta(\mathbf{x})$ (19). Expectedly, we observe that ROC curves get better the stronger the attacks are. Our quantile method excels the entropy approach for both attacks, for Metzen even with high significance. Further ROC curves are provided in Supplement M.

In Tab. 2, we compare the clean mIoU and global AuROC across all perturbations, model size, and computational complexity in terms of GFLOPs of one-hot and Hadamard output encodings for two model architectures and datasets. For perturbation detection with one-hot models, we use the entropy [36] and max posterior [13] baselines, while for our `HadamardNet` we report our three perturbation detectors (17), (18), (19). On clean data, we reach among all methods statistically equivalent semantic segmentation mIoU on $\mathcal{D}_{\text{val}}^{\text{CS}}$, and close-by performance on $\mathcal{D}_{\text{val}}^{\text{BDD,Seg}}$. *Concerning global AuROC of the detectors, the* ***`HadamardNet`*** *with our detector (17) is significantly better than one-hot encodings with any of the baseline detectors [36], [13] for both models and datasets.* This holds also for our quantile detection (19) on $\mathcal{D}_{\text{val}}^{\text{CS}}$. A comparison to the (very poor-performing) the so-far only Hadamard semantic segmentation method by Hoyos et al. [15] is provided in Supplement N. More results on $\mathcal{D}_{\text{val}}^{\text{BDD,Seg}}$ are reported in Supplement O and on $\mathcal{D}_{\text{val}}^{\text{ADE20K}}$ in Supplement P.

Table 3: Global AuROC (%) for semantic segmentation on $\mathcal{D}_{\text{val}}^{\text{CS}}$, comparing entropy-based one-hot detection [36], inconsistency-based detection in the one-hot output space, and inconsistency-based detection in the Hadamard output space. All models use `DeepLabv3+`. Best results in bold.

One-hot+entropy [36]	One-hot+(17)	Hadamard+(17)
$72.5^{\pm 0.1}$	$76.9^{\pm 0.3}$	$\mathbf{80.4}^{\pm 1.1}$

In Tab. 3, we further investigate whether the performance gain stems only from allowing prediction inconsistencies or whether the Hadamard output encoding itself contributes to perturbation detection. To this end, we train a one-hot model to directly predict $\tilde{\mathbf{P}}_i$ using sigmoid instead of softmax activation, while keeping the remaining setup unchanged, therefore allowing the model to have prediction inconsistencies. We then apply the same simplex projection (15), error vector computation (16), and perturbation detector (17) as for the `HadamardNet`. We denote this setup as one-hot+(17). Results are reported on $\mathcal{D}_{\text{val}}^{\text{CS}}$ using a `DeepLabv3+`. Compared to the one-hot entropy baseline [36], using the prediction inconsistency in the one-hot space already improves the global AuROC by 4.4% absolute. This shows that the general principle of exploiting inconsistencies between predictions and their projection is already beneficial. However, replacing the one-hot output encoding with Hadamard output encoding further improves the global AuROC by another 3.5% absolute, reaching 80.4%. *This shows that both, allowing prediction inconsistencies and structuring them through Hadamard encodings, improve perturbation detection.*

Object Detection: In Supplement Q, we conduct an ablation study on $\mathcal{D}_{\text{val}}^{\text{VOC}}$ for weight λ^{cls} of the classification loss J^{cls} (20) as part of J^{OD} (22), and fix the optimum for all further object detection experiments on all datasets.

In Tab. 4, we compare the perturbation detection performance of all three variants of our proposed method in Sec. 3.3 with state-of-the-art detection approaches for object detection across multiple perturbation types and strengths ϵ. All methods employ a `Faster R-CNN` [33] model trained on $\mathcal{D}_{\text{train}}^{\text{VOC}}$ and evaluated on perturbed samples from $\mathcal{D}_{\text{val}}^{\text{VOC}}$. We report the perturbation detection accuracy measured by global AuROC (%) per perturbation across all strengths and across all perturbations and strengths. We observe that Xu et al.'s multi-pass approach [43] shows imbalanced performance across perturbations, being strong under the vanishing attack, but particularly poor under Gaussian noise. The max posterior baseline [13] is the overall best among the single-pass baselines. Expectedly, our quantile method (19) performs poorly due to the small number of objects compared to the high number of pixels in semantic segmentation. *Our regression detector (18), however, is best among all single-pass methods in each perturbation. With a global AuROC of 74.4% it is overall best, significantly excelling even Xu et al.'s multi-pass approach (69.8%).* The AP and detection performance detailed for each strength ϵ is provided in Supplement R.

Table 4: Perturbation detection for object detection on $\mathcal{D}_{\text{val}}^{\text{VOC}}$ using `Faster R-CNN`. Global AuROC (best in bold) per perturbation (Untargeted, Fabrication, Vanishing, Mislabeling, Gaussian, S&P) across all perturbation strengths $\epsilon \in \{0.25, 0.5, 1, 2, 4, 8, 16\}$ and across all perturbations and strengths in %.

Method	Global AuROC (%)						Global	FLOPs
	Untar.	Fabric.	Vanish.	Mislab.	Gauss	S&P	AuROC	(G)
Multi-pass detector								
Xu et al. [43]	$81.0^{\pm 3.9}$	$82.1^{\pm 3.9}$	$83.5^{\pm 2.6}$	$80.9^{\pm 3.9}$	$34.6^{\pm 0.7}$	$63.4^{\pm 2.4}$	$69.8^{\pm 2.6}$	645
Single-pass detector								
Entropy [36]	$84.8^{\pm 0.9}$	$87.5^{\pm 0.9}$	$8.2^{\pm 0.3}$	$84.8^{\pm 1.0}$	$49.8^{\pm 0.2}$	$46.1^{\pm 2.0}$	$59.6^{\pm 0.1}$	215
Max posterior [13]	$92.1^{\pm 0.5}$	$94.0^{\pm 0.4}$	$5.4^{\pm 0.1}$	$92.2^{\pm 0.6}$	$48.4^{\pm 0.2}$	$40.8^{\pm 1.5}$	$61.2^{\pm 0.1}$	215
Ours: Error (17)	$92.3^{\pm 1.3}$	$93.5^{\pm 1.3}$	$9.7^{\pm 0.7}$	$92.4^{\pm 1.4}$	$49.7^{\pm 0.8}$	$47.0^{\pm 0.5}$	$63.3^{\pm 0.5}$	215
Ours: Reg. (18)	$\mathbf{93.9}^{\pm 0.9}$	$\mathbf{94.7}^{\pm 0.8}$	$\mathbf{70.6}^{\pm 1.1}$	$\mathbf{93.8}^{\pm 0.9}$	$\mathbf{50.2}^{\pm 0.5}$	$\mathbf{51.2}^{\pm 2.1}$	$\mathbf{74.4}^{\pm 0.4}$	215
Ours: Quantile (19)	$64.1^{\pm 19}$	$64.3^{\pm 18}$	$38.5^{\pm 17}$	$64.5^{\pm 18}$	$47.5^{\pm 0.2}$	$49.1^{\pm 4.8}$	$54.3^{\pm 17}$	215

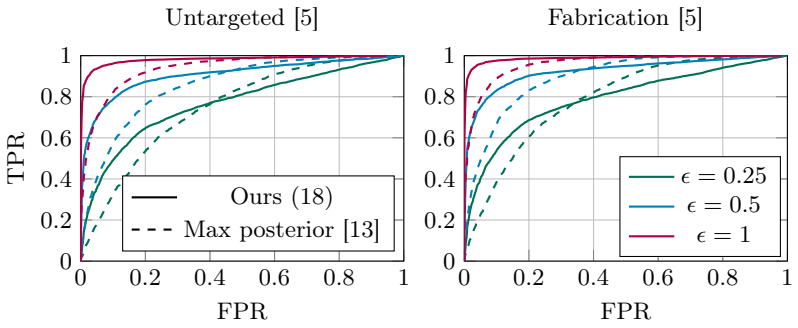

Fig. 3: ROC of our proposed regression perturbance detection (18) for object detection, untargeted [5] attack (left), fabrication [5] attack (right) at specific perturbation strengths $\epsilon \in \{0.25, 0.5, 1\}$. All results are obtained using a `Faster R-CNN` [33] model trained on $\mathcal{D}_{\text{train}}^{\text{VOC}}$ and evaluated on perturbed images from $\mathcal{D}_{\text{val}}^{\text{VOC}}$.

In Fig. 3, we display ROC curves for the untargeted [5] (left) and fabrication [5] (right) adversarial attacks for different attack strengths ϵ employing our regression approach (18) and the (strong) max posterior baseline [13]. The results are obtained using the `Faster R-CNN` [33] model, which is trained on $\mathcal{D}_{\text{train}}^{\text{VOC}}$ and evaluated on attacked images from $\mathcal{D}_{\text{val}}^{\text{VOC}}$. The ROC curves are obtained by varying the decision threshold θ applied to the anomaly score $\delta(\mathbf{x})$ (18). Again, we observe that ROC curves improve with increasing attack strength. For reasonably small FPRs, our regression approach (18) delivers better ROC curves than the strongest baseline max posterior [13]. Note that further ROC curves are provided in Supplement S.

In Tab. 5, we report the clean AP and global AuROC across all perturbations, model size, and computational complexity in terms of GFLOPs of one-hot and our proposed Hadamard output encodings across two object detection archi-

Table 5: One-hot vs. Hadamard output encodings on **object detection** models on $\mathcal{D}_{\text{val}}^{\text{VOC}}$ and $\mathcal{D}_{\text{test}}^{\text{BDD,OD}}$. Global AuROC (best in bold, second-best underlined) across all perturbations and strengths $\epsilon \in \{0.25, 0.5, 1, 2, 4, 8, 16\}$ and AP in (%).

Model	Output representation / detector	VOC AP clean	VOC AuROC attack	BDD AP clean	BDD AuROC attack	Size (M)	FLOPs (G)
`Faster R-CNN` [33]	One-hot/entropy [36]	$41.0^{\pm 0.4}$	$59.6^{\pm 0.1}$	$32.8^{\pm 0.2}$	$53.7^{\pm 2.3}$	41.4	215
	One-hot/max post. [13]	$41.0^{\pm 0.4}$	$61.2^{\pm 0.1}$	$32.8^{\pm 0.2}$	$60.5^{\pm 0.4}$	41.4	215
	Hadamard/(17) (ours)	$40.4^{\pm 0.5}$	$\underline{63.3}^{\pm 0.5}$	$32.3^{\pm 0.3}$	$\underline{63.2}^{\pm 0.4}$	41.4	215
	Hadamard/(18) (ours)	$40.4^{\pm 0.5}$	$\mathbf{74.4}^{\pm 0.4}$	$32.3^{\pm 0.3}$	$\mathbf{65.0}^{\pm 2.9}$	41.4	215
	Hadamard/(19) (ours)	$40.4^{\pm 0.5}$	$39.2^{\pm 17}$	$32.3^{\pm 0.3}$	$57.5^{\pm 23}$	41.4	215
`DETR` [2]	One-hot/entropy [36]	$47.5^{\pm 0.5}$	$66.0^{\pm 3.6}$	$26.7^{\pm 0.7}$	$57.2^{\pm 2.8}$	41.6	102
	One-hot/max post. [13]	$47.5^{\pm 0.5}$	$60.2^{\pm 4.0}$	$26.7^{\pm 0.7}$	$55.2^{\pm 2.4}$	41.6	102
	Hadamard/(17) (ours)	$46.9^{\pm 0.5}$	$\mathbf{68.2}^{\pm 3.6}$	$25.8^{\pm 0.9}$	$58.3^{\pm 4.5}$	41.6	102
	Hadamard/(18) (ours)	$46.9^{\pm 0.5}$	$\underline{68.1}^{\pm 1.9}$	$25.8^{\pm 0.9}$	$\underline{58.4}^{\pm 1.3}$	41.6	102
	Hadamard/(19) (ours)	$46.9^{\pm 0.5}$	$39.2^{\pm 6.9}$	$25.8^{\pm 0.9}$	$\mathbf{63.6}^{\pm 5.8}$	41.6	102

tectures and datasets. For perturbation detection with one-hot models, we use the entropy [36] and max posterior [13] baselines, while for the `HadamardNet` we report our three detectors (17), (18), (19). Models are trained on either $\mathcal{D}_{\text{train}}^{\text{VOC}}$ or $\mathcal{D}_{\text{train}}^{\text{BDD,OD}}$ and evaluated on $\mathcal{D}_{\text{val}}^{\text{VOC}}$ or $\mathcal{D}_{\text{test}}^{\text{BDD,OD}}$, respectively. On clean data, we observe close-by AP performance on $\mathcal{D}_{\text{test}}^{\text{BDD,OD}}$ for `Faster R-CNN`, while for any other configuration all methods reach a statistically equivalent AP. Among the baselines, for `Faster R-CNN` the max posterior perturbation detector [13] delivers the better global AuROC than the entropy one [36], while this is vice versa for `DETR`. *Overall, our regression perturbation detector (18) in conjunction with the `HadamardNet` shows the best global AuROC results for both network topologies and on both datasets.* Again, the second-best perturbation detector (17) shows good and balanced performance and confirms its well generalizing applicability to both semantic segmentation and object detection on various datasets along with various network topologies. Detailed object detection results on $\mathcal{D}_{\text{test}}^{\text{BDD,OD}}$ are provided in Supplement T and on $\mathcal{D}_{\text{val}}^{\text{COCO}}$ in Supplement P.

6 Conclusions

Hadamard codes are advantageous output representations for semantic segmentation and object detection tasks, as we have shown in this work that *their redundancy allows for a single-pass state-of-the-art error detection.* This is achieved while we preserve an overall equal or close-by task performance w.r.t. mIoU and AP metrics on clean data. We reported results on four network topologies and five datasets, and support our claim by extensive ablations.

Acknowledgments

The research leading to these results is funded by the German Federal Ministry for Economic Affairs and Energy within the project "Safe AI Engineering – Sicherheitsargumentation befähigendes AI Engineering über den gesamten Lebenszyklus einer KI-Funktion". The authors would like to thank the consortium for the successful cooperation.

References

1. Akata, Z., Perronnin, F., Harchaoui, Z., Schmid, C.: Label-Embedding for Attribute-Based Classification. In: Proc. of CVPR. pp. 819–826. Portland, OR, USA (Jun 2013)
2. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-End Object Detection with Transformers. In: Proc. of ECCV. pp. 213–229. Glasgow, United Kingdom (Nov 2020)
3. Chen, K., Wang, J., Pang, J., Cao, Y., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Xu, J., Zhang, Z., Cheng, D., Zhu, C., Cheng, T., Zhao, Q., Li, B., Lu, X., Zhu, R., Wu, Y., Dai, J., Wang, J., Shi, J., Ouyang, W., Loy, C.C., Lin, D.: MMDetection: Open MMLab Detection Toolbox and Benchmark. arXiv preprint arXiv:1906.07155 (2019)
4. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-Decoder With Atrous Separable Convolution for Semantic Image Segmentation. In: Proc. of ECCV. pp. 801–818. Munich, Germany (Sep 2018)
5. Chow, K.H., Liu, L., Loper, M., Bae, J., Gursoy, M.E., Truex, S., Wei, W., Wu, Y.: Adversarial Objectness Gradient Attacks in Real-time Object Detection Systems. In: Proc. of TPS-ISA. pp. 263–272. Atlanta, GA, USA (Oct 2020)
6. Contributors, M.: MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. `https://github.com/open-mmlab/mmsegmentation` (2020), accessed: 2026-06-25
7. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The Cityscapes Dataset for Semantic Urban Scene Understanding. In: Proc. of CVPR. pp. 3213–3223. Las Vegas, NV, USA (Jun 2016)
8. Dietterich, T.G., Bakiri, G.: Solving Multiclass Learning Problems via Error-Correcting Output Codes. J. Artif. Intell. Res. **2**(1), 263–286 (Jan 1995)
9. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes (VOC) Challenge. International Journal of Computer Vision **88**(2), 303–338 (Sep 2010)
10. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The Pascal Visual Object Classes Challenge: A Retrospective. International Journal of Computer Vision (IJCV) **111**(1), 98–136 (Jan 2015)
11. Frome, A., Corrado, G.S., Shlens, J., Bengio, S., Dean, J., Ranzato, M., Mikolov, T.: DeViSE: A Deep Visual-Semantic Embedding Model. In: Proc. of NeurIPS. pp. 2121–2129. Lake Tahoe, NV, USA (Dec 2013)
12. Goodfellow, I., Shlens, J., Szegedy, C.: Explaining and Harnessing Adversarial Examples. In: Proc. of ICLR. pp. 1–10. San Diego, CA, USA (May 2015)

13. Hendrycks, D., Gimpel, K.: A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks. In: Proc. of ICLR. pp. 1–12. Toulon, France (Apr 2017)
14. Hoffer, E., Hubara, I., Soudry, D.: Fix Your Classifier: The Marginal Value of Training the Last Weight Layer. In: Proc. of ICLR - Workshops. pp. 1–11. Vancouver, Canada (Apr 2018)
15. Hoyos, A., Rivera, M.: Hadamard Layer for Improve Semantic Segmentation. IEEE Access **12**, 194367–194377 (2024)
16. Hoyos, A., Ruiz, U., Chavez, E.: Hadamard's Defense Against Adversarial Examples. IEEE Access **9**, 118324–118333 (2021)
17. Hsu, D.J., Kakade, S.M., Langford, J., Zhang, T.: Multi-Label Prediction via Compressed Sensing. In: Proc. of NeurIPS. pp. 772–780. Vancouver, BC, Canada (Dec 2009)
18. Klingner, M., Bär, A., Fingscheidt, T.: Improved Noise and Attack Robustness for Semantic Segmentation by Using Multi-Task Training with Self-Supervised Depth Estimation. In: Proc. of CVPR - Workshops. pp. 1299–1309. Seattle, WA, USA (Jun 2020)
19. Klingner, M., Kumar, V.R., Yogamani, S., Bär, A., Fingscheidt, T.: Detecting Adversarial Perturbations in Multi-Task Perception. In: Proc. of IROS. pp. 13050–13057. Kyoto, Japan (Oct 2022)
20. Koenker, R., Bassett, G.: Regression Quantiles. Econometrica **46**(1), 33–50 (1978)
21. Kong, E.B., Dietterich, T.G.: Error-Correcting Output Coding Corrects Bias and Variance. In: Proc. of ICML. pp. 313–321. Tahoe City, CA, USA (Jul 1995)
22. Kurakin, A., Goodfellow, I., Bengio, S.: Adversarial Examples in the Physical World. In: Proc. of ICLR - Workshops. pp. 1–14. Toulon, France (Apr 2017)
23. Li, S., Zhu, S., Paul, S., Roy-Chowdhury, A., Song, C., Krishnamurthy, S., Swami, A., Chan, K.S.: Connecting the Dots: Detecting Adversarial Perturbations Using Context Inconsistency. In: Proc. of ECCV. pp. 396–413. Glasgow, United Kingdom (Nov 2020)
24. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: Common Objects in Context. In: Proc. of ECCV. pp. 740–755. Zurich, Switzerland (Sep 2014)
25. Liu, W., Wang, X., Owens, J., Li, Y.: Energy-based Out-of-Distribution and Adversarial Detection. In: Proc. of NeurIPS. pp. 21464–21475. virtual (Dec 2020)
26. Loshchilov, I., Hutter, F.: Decoupled Weight Decay Regularization. In: Proc. of ICLR. pp. 1–18. New Orleans, LA, USA (May 2019)
27. Maag, K., Resner, R., Fischer, A.: Detecting Adversarial Attacks in Semantic Segmentation via Uncertainty Estimation: A Deep Analysis. arXiv:2408.10021 (Aug 2024)
28. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards Deep Learning Models Resistant to Adversarial Attacks. In: Proc. of ICLR. pp. 1–28. Vancouver, BC, Canada (Apr 2018)
29. Metzen, J.H., Kumar, M.C., Brox, T., Fischer, V.: Universal Adversarial Perturbations Against Semantic Image Segmentation. In: Proc. of ICCV. pp. 2774–2783. Venice, Italy (Oct 2017)
30. Michelot, C.: A Finite Algorithm for Finding the Projection of a Point onto the Canonical Simplex of α^n. Journal of Optimization Theory and Applications **50**(1), 195–200 (1986)
31. Nguyen, K.N.T., Zhang, W., Lu, K., Wu, Y.H., Zheng, X., Li Tan, H., Zhen, L.: A Survey and Evaluation of Adversarial Attacks in Object Detection. IEEE Transactions on Neural Networks and Learning Systems **36**(9), 15706–15722 (2025)

32. Proakis, J.G.: Digital Communications. McGraw-Hill Book Company (1989)
33. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards Real-Time Object Detection With Region Proposal Networks. In: Proc. of NIPS. pp. 91–99. Montréal, QC, Canada (Dec 2015)
34. Rodríguez, P., Bautista, M.A., Gonzàlez, J., Escalera, S.: Beyond One-Hot Encoding: Lower Dimensional Target Embedding. Image and Vision Computing **75**, 21–31 (2018)
35. Ryu, G., Choi, D.: Detection of Adversarial Attacks Based on Differences in Image Entropy. International Journal of Information Security **23**(1), 299–314 (Feb 2024)
36. Smith, L., Gal, Y.: Understanding Measures of Uncertainty for Adversarial Example Detection. arXiv preprint:1803.08533 (Mar 2018)
37. Sylvester, J.J.: LX. Thoughts on Inverse Orthogonal Matrices, Simultaneous Sign Successions, and Tessellated Pavements in Two or More Colours, With Applications to Newton's Rule, Ornamental Tile-Work, and the Theory of Numbers. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science **34**(232), 461–475 (1867)
38. Thunuguntla, A., Tadepalli, P., Raffa, G.: Defenses Against Adversarial Attacks on Object Detection: Methods and Future Directions. Information **16**(11) (2025)
39. Verma, G., Swami, A.: Error Correcting Output Codes Improve Probability Estimation and Adversarial Robustness of Deep Neural Networks. In: Proc. of NeurIPS. pp. 8646–8656. Vancouver, BC, Canada (Dec 2019)
40. Wan, L., Alpcan, T., Viterbo, E., Kuijper, M.: Efficient Error-Correcting Output Codes for Adversarial Learning Robustness. In: Proc. of ICC. pp. 2345–2350. Seoul, South Korea (May 2022)
41. Weston, J., Bengio, S., Usunier, N.: Large Scale Image Annotation: Learning to Rank with Joint Word-Image Embeddings. Machine learning **81**(1), 21–35 (2010)
42. Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P.: SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers. In: Proc. of NeurIPS. pp. 12077–12090. virtual (Dec 2021)
43. Xu, W., Evans, D., Qi, Y.: Feature Squeezing: Detecting Adversarial Examples in Deep Neural Networks. In: Proc. of Network and Distributed Systems Security Symposium. San Diego, CA, USA (Feb 2018)
44. Yang, S., Luo, P., Loy, C.C., Shum, K.W., Tang, X.: Deep Representation Learning with Target Coding. In: Proc. of AAAI. pp. 3848–3854. Austin, TX, USA (Jan 2015)
45. Yin, M., Li, S., Cai, Z., Song, C., Asif, M.S., Roy-Chowdhury, A.K., Krishnamurthy, S.V.: Exploiting Multi-Object Relationships for Detecting Adversarial Attacks in Complex Scenes. In: Proc. of ICCV. pp. 7858–7867. Montréal, QC, Canada (Oct 2021)
46. Yu, F., Chen, H., Wang, X., Xian, W., Chen, Y., Liu, F., Madhavan, V., Darrell, T.: BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning. In: Proc. of CVPR. pp. 1–14. Seattle, WA, USA (Jun 2020)
47. Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Scene Parsing through ADE20K Dataset. In: Proc. of CVPR. pp. 633–641. Honolulu, HI, USA (Jul 2017)