FADE: POST-HOC FAILURE DETECTION AND UNCERTAINTY ESTIMATION ACROSS CLASSICAL AND VISION FOUNDATION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision Foundation Models (VFMs) have set new standards in zero-shot and open-vocabulary segmentation and object detection. Yet reliably assessing their predictions remains a central challenge: model-reported confidence scores are often poorly calibrated. We introduce FADE, a generic post-hoc framework that trains meta-classifiers to discriminate correct from incorrect predictions based on a predefined IoU threshold, using a common representation built from model-reported confidence, geometric features, and semantic information from a frozen vision-language encoder (CLIP). Requiring neither modifications to the underlying model nor access to its internal representations, FADE is fully model-agnostic and applies to classical as well as modern VFMs, spanning both segmentation and object detection. We use FADE to systematically compare prediction reliability between classical and foundation models across multiple datasets and tasks. Our results demonstrate that FADE reliably estimates the probability of success for both model types using this model-agnostic representation. Moreover, in terms of Expected Calibration Error (ECE), estimating this probability of success is harder for foundation models than for classical models. This points to a concrete limitation in the reliability of current VFMs, which a trustworthy predictor should not exhibit. Our code is publicly available at TBA.

1 INTRODUCTION

Recent advances in Vision Foundation Models (VFMs) have fundamentally changed the field of computer vision (Bommasani et al., 2021; Wu et al., 2024). Models such as Segment Anything (SAM) (Kirillov et al., 2023), Grounded SAM (Ren et al., 2024), and Vision-Language Models (VLMs) such as CLIP (Radford et al., 2021) achieve strong performance across a wide range of tasks without requiring task-specific training or dataset-dependent fine-tuning. In particular, their open-vocabulary and zero-shot capabilities make them attractive for real-world applications, especially in environments with limited annotated data or changing visual domains where traditional supervised approaches often reach their limits. Despite their capabilities, however, it remains unclear how reliable their predictions actually are. In practice, VFMs can produce erroneous predictions for which their internal confidence does not reliably reflect the actual prediction quality: correct predictions can receive surprisingly low scores, and incorrect ones surprisingly high scores, presumably because models trained on large, heterogeneous data are not necessarily well calibrated for the specific domain in which they are deployed. In line with this, several studies report poorly calibrated confidence measures (Guo et al., 2017; Soon & Hsieh, 2026) that correlate only weakly with actual prediction quality (Küppers et al., 2022), alongside substantial failure cases of VFMs in practical applications (Ji et al., 2024; Ingrisch et al., 2025). This is particularly critical in safety-critical applications such as autonomous driving and robotics, where overconfident errors may lead to unpredictable and hazardous system behavior.

This leads to two central research questions. First, and foremost: *how many of a model's errors can be detected post-hoc, i.e., how reliably can correct predictions be separated from incorrect ones based on the model's self-reported confidence together with semantic and geometric information about the prediction?* Second: *how well calibrated, in a statistical sense, are the resulting confidence and uncertainty estimates?* We use both criteria to compare classical models, trained directly

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



(a) Ground truth (b) Prediction (c) Raw score (d) FADE h_i^{base}

Figure 1: A Pascal VOC segmentation example of Grounded SAM2, evaluated on its own raw self-reported score versus our post-hoc meta-classifier FADE. Green color corresponds to high probability of success values and red color to low ones. FADE correctly flags the prediction as unreliable.

on the target dataset, against VFMs, which generalize more broadly but are not specifically tailored to it: which model type is more uncertain, whose errors are easier to anticipate, and whose confidence estimates are better calibrated for downstream decision-making, e.g., for setting thresholds on when to trust a prediction. Figure 1 shows an example comparing raw model score and our method FADE. Classical uncertainty estimation offers no straightforward answer. Established techniques such as softmax-based confidence (Hendrycks & Gimpel, 2016), ensembles (Lakshminarayanan et al., 2017), or Bayesian approximations (Gal & Ghahramani, 2016) perform well in supervised closed-set settings, but their assumptions do not necessarily transfer to VFMs. Structured predictions carrying geometric and semantic information beyond a single confidence value have long posed challenges for such methods; in VFMs, this is compounded by their frequent zero-shot, out-of-distribution use (Hendrycks et al., 2019; Vaze et al., 2022) and by the heterogeneity of model families, where the very notion of internal confidence itself differs, e.g., softmax probabilities versus a predicted IoU, ruling it out as a common basis for comparison in the first place. This calls for an approach that is inherently model- and task-agnostic.

We address this by formulating uncertainty estimation as a separate, post-hoc learning task, building on meta-classification approaches, in the context of deep learning originally developed for classical segmentation and detection models (Rottmann et al., 2020; Schubert et al., 2021). Instead of relying on internal, model-specific confidence, we train a lightweight meta-classifier directly on the properties of a prediction and its input: the confidence already reported by the model, a semantic representation of the predicted region from a frozen vision-language encoder (CLIP), and geometric descriptors such as segment or bounding-box size and position within the image. Since these features require neither modifications to the underlying model nor access to its internal representations, the same formulation applies model-agnostically, to classical architectures and modern VFMs alike, and task-agnostically, across semantic, instance, and panoptic segmentation, and object detection.

We first consider this fully task- and model-agnostic feature set, providing a common basis for comparison without requiring a shared confidence scale or access to model internals. We then relax task-agnosticism by adding task-specific structural features in the spirit of MetaSeg (Rottmann et al., 2020) and MetaDetect (Schubert et al., 2021), while retaining model independence, to study the trade-off between generality and failure-detection performance.

Our contribution consists of three parts:

- We extend post-hoc uncertainty estimation to modern VFMs by augmenting the model's self-reported confidence with model-external semantic and geometric features, without requiring any modification of the underlying model or access to its internal representations.
- We show that this task- and model-agnostic feature set enables a fair comparison of classical and foundation model backbones.
- We show that, while classical models and VFMs achieve comparable discrimination (AUROC) under our model-agnostic representation, VFM confidence estimates are consistently worse calibrated (ECE), revealing a clear reliability gap between the two model families.

We validate these findings across multiple datasets and a range of model families, spanning from classical CNN- and transformer-based detectors and segmenters to modern VFMs. The improvements are reflected in the meta-classification of correct versus incorrect predictions. Overall, our results show that separating prediction generation from prediction evaluation provides a practical approach for estimating the reliability of predictions from modern VFMs.

2 Related Work

Uncertainty estimation in deep learning is a broad field, spanning Bayesian approaches such as Monte-Carlo Dropout (Gal & Ghahramani, 2016) and Deep Ensembles (Lakshminarayanan et al., 2017), model-reported confidence measures such as the Maximum Softmax Probability (Hendrycks & Gimpel, 2016), and post-hoc calibration methods such as Temperature Scaling (Guo et al., 2017), among many others (Kendall & Gal, 2017; Abdar et al., 2021; Gawlikowski et al., 2023; He et al., 2026). However, these estimates remain tied to the specific model's training and inference process and do not necessarily transfer to modern VFMs (Ovadia et al., 2019; Hendrycks et al., 2019; Gruber et al., 2023). Below, we review the fields most relevant to our work: post-hoc prediction quality estimation, uncertainty estimation for VFMs, and VLMs for reliability assessment.

Prediction Quality Estimation. Instead of estimating uncertainty directly from the prediction model, several works formulate prediction quality assessment as a separate post-hoc learning problem. MetaSeg (Rottmann et al., 2020) and MetaDetect (Schubert et al., 2021) introduced this approach for semantic segmentation and object detection, respectively: a meta-classifier is trained on handcrafted features derived from the model's output, such as dispersion measures of the predicted class distribution or bounding-box statistics, to discriminate correct from incorrect predictions. Both approaches are, however, tailored to their specific task and output representation; MetaDetect further relies on access to the detector's non-maximum suppression step, which is not available for all architectures. Extensions thereof include entropy-based meta-classification for out-of-distribution detection (Chan et al., 2021) and gradient-based uncertainty metrics for object detection (Riedlinger et al., 2023), among others (Colling et al., 2021; Corbière et al., 2019; Rahman et al., 2022). In contrast, we investigate uncertainty estimation across different model families and tasks by combining model-independent features, while allowing task-specific structural information to be used for prediction assessment.

Uncertainty in VFMs. The zero-shot and open-vocabulary capabilities of VFMs (Liu et al., 2024b; Cheng et al., 2024; Kirillov et al., 2023; Ravi et al., 2025; Carion et al., 2026; Ren et al., 2024) introduce challenging settings for assessing prediction reliability. Recent work reports limitations under distribution shift (Chhipa et al., 2024) and in open-world recognition (Vaze et al., 2022); for segmentation, Ji et al. (2024) identify failure cases of SAM on small or irregular objects and domain-specific data. For open-vocabulary detectors, confidence scores can be systematically biased by object scale and query specificity (Soon & Hsieh, 2026), and Ingrisch et al. (2025) show that YOLO-World is poorly calibrated even after temperature scaling. Recent work has therefore also investigated uncertainty estimation specifically for VFMs: several approaches estimate SAM uncertainty from multiple prompted predictions (Deng et al., 2023; Zhang et al., 2023b) or use it to guide fine-tuning (Liu et al., 2024a), while UncertainSAM (Kaiser et al., 2025) proposes a lightweight, Bayesian-entropy-based post-hoc estimator. Beyond vision-specific models, Dutta et al. (2023) apply web-based conformal prediction to multimodal foundation models, and Yang et al. (2025) survey reliability and uncertainty estimation across foundation model families. These approaches remain tied to a specific model or prediction framework. In contrast, we assess prediction reliability independently of the underlying model, combining signals from model outputs, prediction structure, and foundation models into a single success-probability estimate.

VLMs for Reliability Assessment. VLMs such as CLIP (Radford et al., 2021) provide semantic representations effective for open-vocabulary recognition, retrieval, and out-of-distribution detection, and have also been used for uncertainty estimation. Probabilistic approaches model uncertainty directly in the embedding space (Upadhyay et al., 2023; Chun et al., 2025), Cui et al. (2024) estimate aleatoric uncertainty from VFM features via a Gaussian mixture model, and ViLU (Lafon et al., 2025) uses vision-language representations for post-hoc failure prediction of VLMs. In contrast, we use semantic representations as descriptive features for a separate uncertainty estimation model.

3 Methodology

We propose FADE shown in Figure 2, a post-hoc evaluation framework that quantifies the reliability of predictions from an arbitrary, frozen vision model f. Rather than modifying f or its training procedure, we decouple prediction generation from prediction evaluation. Each prediction is reduced

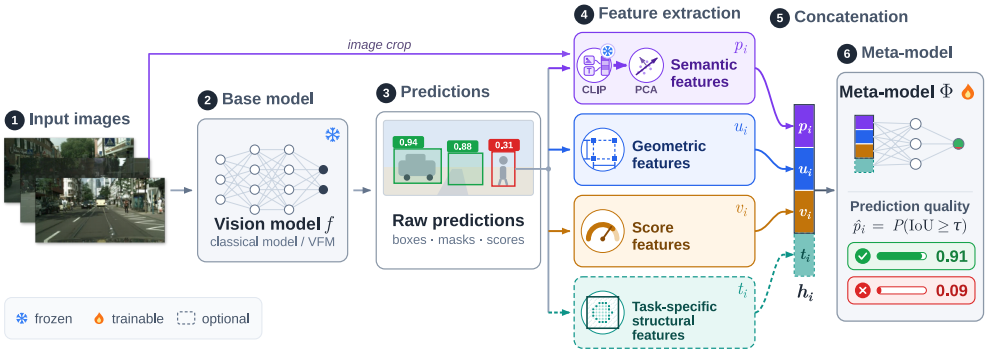


Figure 2: Overview of FADE, the proposed post-hoc uncertainty estimation framework. A frozen vision model f generates predictions (s_i, c_i) from the input images. For each individual prediction, semantic (p_i), geometric (u_i), and self-reported score (v_i) features are extracted. These features form the task- and model-agnostic representation h_i^{base}. Optionally, task-specific structural features t_i following MetaSeg and MetaDetect are added, yielding the model-agnostic (but no longer task-agnostic) representation h_i^{full}. The resulting feature representation is passed to a meta-classifier that estimates the probability that the prediction satisfies a predefined IoU threshold.

to a standardized image crop and described by geometric, self-reported confidence, semantic and optionally task-specific features. These features are provided to a lightweight meta-classifier that estimates the individual prediction's probability of success. Since the meta-classifier does not require access to the internal operations of f, the same framework can be applied to different model architectures, including classical detectors and segmenters as well as VFMs.

3.1 Problem Formulation & Learning Objective

In this section, we formalize the problem setting and, in particular, define the learning objective of the meta-classifier. Let f denote a vision model that maps an input image $x \in \mathbb{R}^{H \times W \times 3}$ to a set of $n \in \mathbb{N}$ predictions $\{(s_i, c_i)\}_{i=1}^{n}$, where c_i is the predicted class and s_i represents either a segmentation mask $m_i \in \{0,1\}^{H \times W}$ or a bounding box $b_i \in \mathbb{R}^4$. For a prediction (s_i, c_i), we determine whether it is a success from the ground-truth objects of the same class. Let $\mathcal{G}_i = \{g_j \mid a_j = c_i\}$ denote the set of ground-truth objects whose class a_j matches the predicted class c_i, and let $\tau \in (0,1)$ be a fixed IoU threshold. We define the response variable $r_i \in \{0,1\}$, indicating success of prediction i, as

$$r_i = \mathbb{1}\Big[\mathcal{G}_i \neq \emptyset \text{ and } \max_{g_j \in \mathcal{G}_i} \text{IoU}(s_i, g_j) > \tau\Big]. \tag{1}$$

Thus, $r_i = 1$ if prediction i overlaps a ground-truth object of the predicted class with an IoU exceeding τ, and $r_i = 0$ otherwise, including the case where no such ground-truth object exists. Importantly, each prediction is evaluated independently; a ground-truth object can therefore be the best match for multiple predictions. At deployment time, however, ground truth is unavailable, and the confidence reported by f alone does not necessarily provide a reliable estimate of success. We therefore estimate this probability, $P(r_i = 1)$, with a meta-classifier.

3.2 Meta Feature Extraction

This section describes the inputs to the meta-classifier. For both detection and segmentation b_i denotes the (tight) bounding box of the predicted region s_i.

Region standardization. To obtain a common representation for predictions from different vision tasks, we first extract for every prediction an image crop x_i of the image content of the box b_i. Additionally, for mask predictions all pixels outside the predicted mask m_i are set to black color. This provides one region representation per prediction independent of the output modality.

Geometric features u_i. We describe the spatial extent and position of each predicted region relative to image width and height. The normalized height $\tilde{h}_i$ and width $\tilde{w}_i$ are obtained from the tight

bounding box b_i of the predicted region s_i. The normalized centroid coordinates $\tilde{x}_i, \tilde{y}_i$ are the mean coordinates of s_i, again normalized to the image extent. The resulting feature vector is

$$u_i = [\tilde{h}_i, \tilde{w}_i, \tilde{x}_i, \tilde{y}_i] \in \mathbb{R}^4. \tag{2}$$

These features provide basic information about the size and location of the predicted region while remaining independent of the underlying model architecture and prediction task.

Self-reported score v_i. The self-reported score v_i is a vector of length equal to the number of classes over which f reports confidence, i.e., the evaluation label space for classical models or the dataset-specific inference-time vocabulary for VFMs. If f outputs a set of class scores, v_i is exactly this vector; if f outputs only a single score for the predicted class, v_i places this score at the predicted class's index, with all other entries set to zero (the case throughout our experiments, Sec. 4.1). Since the entries of v_i can carry a different meaning across model families, a shared vector format alone does not make the scores directly comparable; the meta-classifier is instead trained to interpret these model-specific scores jointly with the other feature groups, learning how the model-reported confidence relates to the probability of success rather than treating it as a calibrated success-probability estimate on its own.

Semantic Features p_i. Each crop x_i is encoded by a CLIP image encoder g_θ (ViT-L/14) into an ℓ_2-normalized embedding

$$z_i = \frac{g_\theta(x_i)}{\|g_\theta(x_i)\|_2} \in \mathbb{R}^{768}. \tag{3}$$

If not stated otherwise, the CLIP encoder g_θ is frozen. Since the dimensionality of the embedding is large relative to the number of labeled predictions available per model and dataset combination, we reduce the embedding dimensionality using principal component analysis (PCA) (Jolliffe, 2002):

$$p_i = P_k(z_i) \in \mathbb{R}^k, \tag{4}$$

where the number of retained components k is treated as a hyperparameter and tuned separately for each model-dataset combination based on validation dataset performance.

Task-specific structural features t_i. For mask predictions, we additionally compute structural features derived from the feature definitions used in MetaSeg (Rottmann et al., 2020). Dropping the subscript i for brevity, these include the total predicted size $|m|$, the number of interior pixels $|m_{\text{in}}|$, the number of boundary pixels $|m_{\text{bd}}| = |m| - |m_{\text{in}}|$, and the corresponding ratios

$$\tilde{S} = \frac{|m|}{|m_{\text{bd}}|}, \qquad \tilde{S}_{\text{in}} = \frac{|m_{\text{in}}|}{|m_{\text{bd}}|}. \tag{5}$$

For box predictions, we instead compute structural features following the feature definitions used in MetaDetect (Schubert et al., 2021), expressed via height h and width w of b_i: its area $d = h\,w$, perimeter $g = 2(h + w)$, and area-to-perimeter ratio d/g. These features are derived solely from the predicted region and therefore do not require access to the internal architecture of the prediction model. However, their definitions differ between masks and boxes, making the resulting representation task-specific.

3.3 Task- and Model-Agnostic vs. Model-Agnostic-Only Feature Sets

We study two feature configurations built from the blocks above. The *base* configuration uses only geometry, score, and semantics,

$$h_i^{\text{base}} = \left[u_i^\top,\, v_i^\top,\, p_i^\top\right]^\top, \tag{6}$$

none of which requires knowledge of the base architecture or of which vision task produced s_i: u_i and v_i are defined identically for a box or a mask, and x_i is a crop regardless of modality. As a result, h_i^{base} is both *model-agnostic* and *task-agnostic*, providing a common feature representation for comparing success-probability estimates across different model families and tasks.

The *full* configuration adds the structural block,

$$h_i^{\text{full}} = \left[u_i^\top,\, v_i^\top,\, p_i^\top,\, t_i^\top\right]^\top. \tag{7}$$

Since t_i is still computed only from f's output, h_i^{full} remains model-agnostic, but it is no longer task-agnostic, since t_i's definition switches between the mask- and box-based statistic above. We

use these two configurations to disentangle the contribution of task-agnostic features from task-specific structural information. Specifically, we evaluate whether a fully task- and model-agnostic representation (h_i^{base}) can estimate the probability of success, and quantify the performance gain achieved by relaxing task-agnosticism while maintaining model independence (h_i^{full}).

3.4 META-CLASSIFICATION

The representation $h_i \in \{h_i^{\text{base}}, h_i^{\text{full}}\}$ of prediction i is used to train a small and lightweight neural network meta-classifier Φ that estimates the probability of success,

$$\hat{p}_i = \Phi(h_i) \approx P(r_i = 1), \tag{8}$$

using the labels r_i. Once trained offline on predictions from the frozen vision model f labeled with their success, Φ is applied to predictions without ground-truth information at inference time.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

We evaluate FADE across multiple datasets, tasks, and model families, i.e., classical DNNs and VFMs for different object detection and segmentation tasks, applying the *same* meta-classifier architecture, feature pipeline, and evaluation protocol to every vision model. Since only the underlying vision model f changes between runs, differences in the resulting reliability estimates can be attributed to the vision model itself, enabling a direct, controlled comparison between classical, foundation, and domain-specific models. This is ensured by tuning the dimensionality reduction (PCA) separately for each model-dataset combination, thereby preserving the approximation capacity of the meta-classifier and bringing it close to the optimum achievable at the post-hoc level in terms of extracting false-positive detection cues.

Datasets. We use five datasets spanning different visual domains and tasks. For object detection and instance segmentation, we use the COCO (Lin et al., 2014) 2017 and Pascal VOC (Everingham et al., 2010) 2012 validation splits, together with the Cityscapes (Cordts et al., 2016) validation set for instance segmentation only. To probe FADE beyond common benchmarks, we additionally evaluate two non-public domain-specific object detection datasets: a fish detection dataset containing 46 images and one class, and a surgical instruments dataset containing 669 images and 23 classes.

Models. Table 2 in Appendix A.1 summarizes all evaluated models, spanning three groups. As *classical* models, we use Faster R-CNN (Ren et al., 2015), DINO (Zhang et al., 2023a), and YOLOX (Ge et al., 2021) for object detection, and Mask R-CNN (He et al., 2017), Mask2Former (Cheng et al., 2022), and RTMDet (Lyu et al., 2022) for instance segmentation. As *VFMs*, we evaluate Grounding DINO (Liu et al., 2024b) and YOLO-World (Cheng et al., 2024) for open-vocabulary detection, and Grounded SAM (Ren et al., 2024), Grounded SAM2, and SAM3 (Carion et al., 2026) for zero-shot instance segmentation. Finally, as *domain-specific* models, YOLO11x (Jocher & Qiu, 2024) and YOLO11m-Pose (Jocher & Qiu, 2024) are fine-tuned on the surgical instruments and fish datasets, respectively. All vision models are kept frozen. For each model-dataset pair, the complete set of predictions is generated once, independently per image, and reused across all feature configurations. Since only a scalar-valued predicted-class confidence is available, v_i is a one-hot vector for the predicted class scaled by its confidence in all experiments.

Meta-classifier training. The response variable r_i is determined from the ground-truth annotations using the success criterion of Sec. 3.1 with IoU threshold $\tau = 0.1$. Thus, all predictions produced by the respective model inference pipeline are retained for meta-classifier training, including both successful and failed predictions, without applying an additional confidence-based filtering step. For every vision model, dataset, and feature configuration, we train a separate meta-classifier, a feed-forward network with one hidden layer, using image-level train/validation/test splits and 10-fold cross-validation, with class imbalance induced by the success threshold τ addressed via random oversampling; full architecture, optimization, and splitting details are given in Appendix A.2. Together with the per-combination PCA tuning, this protocol is designed to make $\hat{p}_i$ as informative an estimate of $P(r_i = 1)$ as the available labeled predictions allow.

Feature preprocessing. All features are standardized per fold using training-fold statistics only; the 768-dimensional CLIP ViT-L/14 embedding is further reduced via PCA, with the number of retained components k tuned per model-dataset combination on the validation set. Full preprocessing details and the complete dimensionality study, including the selected k for every combination, are given in Appendix A.2 and A.10, respectively.

Feature configurations and baselines. Beyond the full representations h_i^{base} and h_i^{full} (Sec. 3.3), we evaluate four single-purpose configurations under the same protocol. These comprise the self-reported score v_i alone as input to a meta-classifier (a conventional confidence baseline), the CLIP representation p_i alone, geometry combined with the self-reported score $[u_i^T, v_i^T]^T$, and the task-specific structural features t_i alone, corresponding to the MetaSeg (Rottmann et al., 2020)/MetaDetect (Schubert et al., 2021) feature sets and thus the closest architecture-coupled baseline. We additionally report the raw model-reported score without any meta-classifier, isolating the gain from post-hoc meta-classification itself.

Evaluation metrics. We report AUROC, accuracy, expected calibration error (ECE), and the false positive rate at 95% true positive rate (FPR@95%TPR). AUROC measures discrimination between correct and incorrect predictions and FPR@95%TPR measures the false-positive cost of operating at a high, safety-relevant recall. ECE measures calibration of the resulting reliability estimates, defined explicitly in Appendix A.3. The low IoU threshold ($\tau = 0.1$) targets spatially meaningful localization of the predicted class rather than the stricter thresholds used in standard detection/segmentation benchmarks. Because datasets, features, and metrics are held fixed across all runs, these scores are directly comparable between classical and foundation models.

4.2 Numerical Results

Score Baseline versus Meta-Classification. Before comparing model families, we verify that FADE outperforms a raw score baseline without meta-classifier training, which directly uses each model's self-reported confidence as the success probability. Table 1 summarizes this comparison, aggregated by model families. Tables 3 and 4 in Appendix A.4 give per-model results. AUROC is substantially higher under h_i^{base} than under the raw score, across all model-dataset combinations of COCO and Pascal VOC. The trained meta-classifiers are also markedly better calibrated. Only for GroundingDINO on COCO is the score baseline about as well calibrated as the meta-classifier, where the ECE for both is around 0.033. Comparing VFMs to classical models, however, shows that the raw score already provides a stronger baseline for VFMs. Meta-training improves AUROC by 0.176 on average for classical models, compared with 0.082 for VFMs. Figure 1 illustrates this qualitatively on Pascal VOC. Grounded SAM2 predicts a *cow* as *dog* with a high raw score, whereas the meta-classifier assigns a substantially lower success probability, correctly flagging the error. Appendix A.5 extends this qualitative comparison to all evaluated models across all three datasets.

Table 1: Raw self-reported confidence score used directly as a probability of success, versus Φ under h_i^{base}, averaged over the $N = 22$ model-dataset combinations in total on Pascal VOC and COCO.

		AUROC (↑)		ECE (↓)	
Backbones	N	raw score	h_i^{base}	raw score	h_i^{base}
All	22	0.737	**0.870**	0.122	**0.025**
Classical	12	0.688	**0.864**	0.139	**0.017**
VFM	10	0.795	**0.877**	0.102	**0.034**

Prediction Reliability Across Model Families. We next investigate whether the ability to estimate prediction reliability differs systematically between classical models and VFMs, and whether it relates to the underlying model's predictive performance measured by Average Precision (AP). Aggregated over all 22 model-dataset combinations on COCO and Pascal VOC, classical models and VFMs turn out comparable in both AP and AUROC, with no consistent family effect, a picture corroborated by balanced accuracy and FPR@95%TPR (0.787 vs. 0.796 balanced accuracy, 0.591 vs. 0.551 FPR@95%TPR, classical vs. VFM, Appendix A.6). AP and AUROC also capture different properties of a model rather than tracking each other. On COCO (Figure 3 and Table 5 in Appendix A.6), DINO tops both rankings in object detection, yet Grounding DINO's second-highest AUROC pairs with the second-lowest AP, and in instance segmentation Grounded SAM achieves the highest AUROC despite being near the bottom of the AP ranking. Beyond this AP-AUROC mis-

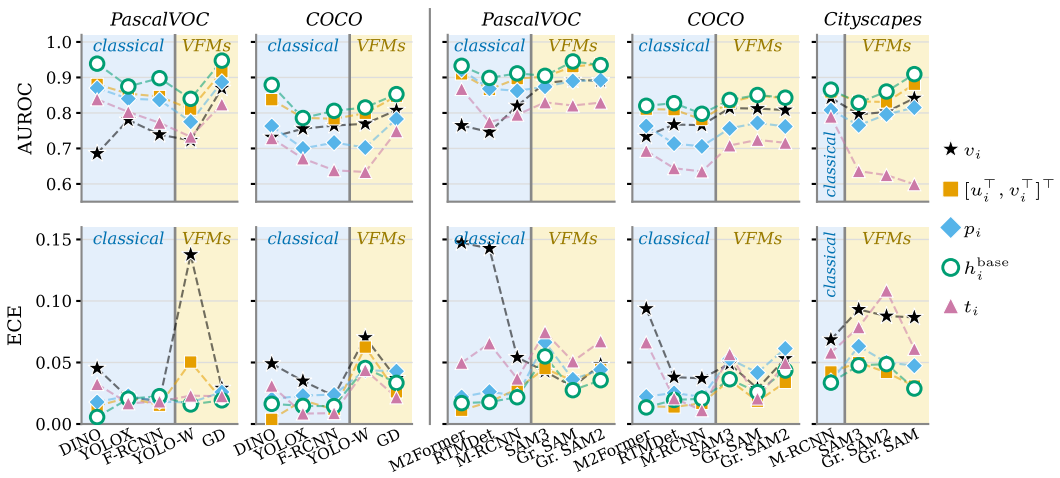


Figure 3: Meta-classification AUROC (top row) and ECE (bottom row) under five feature configurations, object detection (left) and instance segmentation (right). The configurations are the self-reported score v_i, CLIP representation p_i, geometry and score $[u_i^T, v_i^T]^T$, task-specific structural features t_i, and the full model-agnostic representation h_i^{base}. Within each block (classical, VFMs), models are sorted by decreasing Average Precision.

match, object detection also shows no consistent family-level trend. DINO's AUROC rises steadily as features are added (v_i=0.732 $\rightarrow$ h_i^{base}=0.879) while Grounding DINO's stays comparatively flat (v_i=0.808 $\rightarrow$ h_i^{base}=0.853), and neither behavior turns out to be family-typical. One narrower exception, however, concerns the task-specific structural cues t_i in instance segmentation: using t_i alone instead of the full h_i^{base} reduces AUROC by a similar amount for all three VFMs (between 0.127 and 0.129), whereas for classical models this reduction varies more widely, from 0.127 up to 0.184. The same picture holds on Pascal VOC (Figure 3 and Table 6), where Grounding DINO catches up to and slightly surpasses DINO (h_i^{base} = 0.947 vs. 0.939), reversing their COCO ranking. Across all three datasets, a few models have a self-reported score v_i that remains only weakly informative even when passed through the trained meta-classifier Φ, and trusting it alone can be misleading. DINO's v_i on Pascal VOC (0.686) is the lowest of all five object detectors there, yet its h_i^{base} reaches 0.939, the largest gap in the entire study at 0.253. This illustrates that raw confidence can be only weakly informative even though the same predictions contain substantially more extractable information for post-hoc reliability estimation. Taken together, these results are consistent across all three datasets and provide a common basis for comparing classical and VFM backbones: h_i^{base} achieves almost always the strongest AUROC while relying only on geometry, the self-reported score, and CLIP representations, without access to model-specific internals.

Calibration of Reliability Estimates. Although classical models and VFMs achieve comparable AUROC, VFMs are consistently less calibrated across all datasets. On COCO and Pascal VOC, h_i^{base} reaches a mean ECE of 0.017 for classical models versus 0.034 for VFMs, roughly twice as high (Table 1). The same direction holds on Cityscapes (Figure 3 (right)), where the sole classical model (Mask R-CNN, 0.034 ECE) again outperforms the mean of the three VFMs (0.042), though with only one classical model in this comparison, this does not constitute independent statistical confirmation. The clear exception is object detection on Pascal VOC, where both VFMs are better calibrated than most classical models. YOLO-World reaches 0.016 ECE, better than every classical model except DINO (0.006), and Grounding DINO reaches 0.019, still within the classical range (0.006-0.023) and ahead of half the classical models. The ECE panels of Figure 3 show this comparison against model AP for object detection and instance segmentation across all three datasets. They show no consistent relationship to AP. Even the self-reported score v_i alone, though already passed through the trained meta-classifier Φ, is again occasionally poorly calibrated in a way that adding the other features fixes outright. Mask2Former's v_i on Pascal VOC reaches 0.147 ECE, the highest score-only ECE in the entire study, yet its h_i^{base} drops to 0.017. This suggests that the model's self-reported confidence is not a reliable calibration signal on its own, even after recalibra-

tion by Φ, while combining it with the other features yields substantially better-calibrated reliability estimates. This gap has a concrete practical consequence. It determines how reliably a chosen confidence threshold corresponds to the observed frequency of successful predictions, for example to decide when a prediction is trustworthy enough to act on without human review. Thresholding on Φ's output therefore provides a more reliable estimate of the probability of success than thresholding on the raw score directly, consistent with the 0.122 vs. 0.025 mean ECE gap reported for the score baseline in Sec. 4.2. Because this gap is larger for VFMs, relying on their raw confidence without post-hoc calibration can be particularly misleading. Calibration nonetheless remains reasonable in absolute terms, staying below 0.03 ECE for 17 of the 26 model-dataset combinations. A natural question is whether this gap simply reflects VFMs being evaluated zero-shot, and would close once a VFM is finetuned on the target dataset. In Appendix A.7, this is tested directly for Grounding DINO on COCO, comparing the zero-shot checkpoint used above against a COCO-finetuned counterpart. Fine-tuning substantially improves detection performance but does not lead to an improvement in the calibration of the resulting reliability estimates. If anything, the calibration gap widens rather than closes, indicating that fine-tuning the underlying VFM for detection performance alone does not automatically lead to better-calibrated post-hoc reliability estimates.

Feature Ablation, PCA Dimensionality and Computational Costs. We study the individual feature groups introduced in Sec. 3.2, with full results in Appendix A.8. The model-reported score v_i, combined with geometric features u_i, already provides a strong basis for estimating prediction reliability. Adding CLIP representations, h_i^{base} achieves the strongest or near-strongest AUROC across almost all models and datasets. However, h_i^{base} is not always better calibrated than the combination of $[u_i^T, v_i^T]^T$. Task-specific structural features t_i provide no consistent additional benefit, with h_i^{full} remaining close to h_i^{base} across models. Practically, this means CLIP is worth its extra inference cost (Appendix A.9) in most, but not all, deployments. Appendix A.10 further analyses the CLIP embedding's PCA dimensionality k, finding a moderate sensitivity and confirming that our validation-based selection of k reliably tracks the test-optimal value.

Evaluation on Domain-Specific Datasets. We test whether FADE generalizes beyond known benchmarks by applying the identical protocol (Sec. 4.1) to two much smaller, real-world domain-specific datasets. Each dataset is evaluated with its own fine-tuned domain-specific detector as well as the two VFMs Grounding DINO and YOLO-World; full results are given in Table 18, Appendix A.11. The pattern from Sec. 4.2 largely replicates. In five of six model-dataset combinations, h_i^{base} improves AUROC over v_i alone, in one case substantially (YOLO-World on fish, v_i=0.695 $\rightarrow$ h_i^{base}=0.955 AUROC). ECE is reduced in all six model-dataset combinations. However, the near-perfect AUROC of the meta-classifier for Grounding DINO on surgical instruments should be interpreted with caution, as Grounding DINO itself reaches an AP of only 0.031. Its few successful predictions are concentrated in a single, visually distinct class, so the predicted class alone separates successes from failures (Appendix A.11). Near-perfect AUROC therefore does not imply reliable estimation across a fine-grained label space. We also test whether aligning CLIP with the surgical instruments domain further improves FADE's estimates, replacing the off-the-shelf CLIP encoder with one fine-tuned via a low-rank adaptation (Hu et al., 2022). Combined with the geometric and confidence-score features in h_i^{base}, the domain-adapted features yield no consistent improvement in AUROC or ECE (Appendix A.12).

5 Conclusion

We introduced FADE, a post-hoc, model- and task-agnostic framework that estimates the success probability of individual predictions from frozen vision models via a meta-model. Combining model confidence with geometric and CLIP-based semantic features, FADE raises mean AUROC across 22 model–dataset combinations from 0.737 (raw score) to 0.870, with the largest gains for classical models. Comparing classical models and VFMs, firstly doable due to this common representation, across five datasets and two tasks, we find comparable discrimination but consistently worse calibration for VFMs (mean ECE 0.034 vs. 0.017), a gap that persists across datasets, architectures, and fine-tuning. The reliability gap thus lies not in finding errors, but in how trustworthy the resulting estimates are. Deployment decisions, particularly for VFMs, should therefore rely on a calibrated post-hoc estimate such as that provided by FADE rather than raw model confidence, and calibration, not discrimination, remains the open problem for making foundation models trustworthy.

AI USE STATEMENT

In preparing this manuscript, we used a generative AI tool (Claude, Anthropic) to assist with editing and restructuring the manuscript for clarity and conciseness, to identify and correct LaTeX formatting issues, and to locate and format bibliographic entries for related work. The generative AI tool was also used to support the technical implementation of the experiments, including the setup and configuration of environments and existing models, as well as code troubleshooting and debugging. In addition, the generative AI tool supported the creation and refinement of selected figures and visualizations. All AI-assisted setups, text, citations, figures, and visual elements were reviewed and verified by the authors. We have not used generative AI tools to design experiments, generate results, or produce the core methodological or scientific claims of this work. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

For review, the current state of the code for inference, feature extraction, and meta-classifier training, together with example configurations, is provided in the supplementary material. Upon acceptance, the code will be released on GitHub with further improved usability, and the link will be added to the abstract. All evaluated models and their checkpoints are listed in Appendix A.1. The meta-classifier architecture, optimization, data splits, and preprocessing are detailed in Appendix A.2, and the PCA selection procedure in Appendix A.10. COCO, Pascal VOC, and Cityscapes are publicly available. The two domain-specific datasets are non-public.

REFERENCES

Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U Rajendra Acharya, et al. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information fusion*, 76:243–297, 2021. doi: 10.1016/j.inffus.2021.05.008.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris Coll-Vinent, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, RISHI HAZRA, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollar, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. Sam 3: Segment anything with concepts. In *International Conference on Learning Representations*, volume 2026, pp. 138846–138923, 2026. doi: 10.48550/arXiv.2511.16719.

Robin Chan, Matthias Rottmann, and Hanno Gottschalk. Entropy maximization and meta classification for out-of-distribution detection in semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5128–5137, October 2021.

Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention Mask Transformer for Universal Image Segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. doi: 10.48550/arXiv.2112.01527.

Tianheng Cheng, Lin Song, Yixiao Ge, Wenyu Liu, Xinggang Wang, and Ying Shan. Yolo-world: Real-time open-vocabulary object detection. In *2024 IEEE/CVF Conference on Computer Vision*

and Pattern Recognition (CVPR), pp. 16901–16911. IEEE, 2024. doi: 10.1109/CVPR52733.2024.01599.

Prakash Chandra Chhipa, Kanjar De, Meenakshi Subhash Chippa, Rajkumar Saini, and Marcus Liwicki. Open-vocabulary object detectors: Robustness challenges under distribution shifts. In *European Conference on Computer Vision*, pp. 62–79. Springer, 2024. doi: 10.48550/arXiv.2405.14874.

Sanghyuk Chun, Wonjae Kim, Song Park, and Sangdoo Yun. Probabilistic language-image pre-training. In *International Conference on Learning Representations*, volume 2025, pp. 57190–57221, 2025. doi: 10.48550/arXiv.2410.18857.

Pascal Colling, Matthias Rottmann, Lutz Roese-Koerner, and Hanno Gottschalk. False Positive Detection and Prediction Quality Estimation for LiDAR Point Cloud Segmentation. In *2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI)*, 2021. doi: 10.48550/arXiv.2110.15681.

Charles Corbière, Nicolas Thome, Avner Bar-Hen, Matthieu Cord, and Patrick Pérez. Addressing failure prediction by learning model confidence. *Advances in neural information processing systems*, 32, 2019. doi: 10.48550/arXiv.1910.04851.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. doi: 10.48550/arXiv.1604.01685.

Peng Cui, Guande He, Dan Zhang, Zhijie Deng, Yinpeng Dong, and Jun Zhu. Exploring aleatoric uncertainty in object detection via vision foundation models. *arXiv preprint arXiv:2411.17767*, 2024. doi: 10.48550/arXiv.2411.17767.

Guoyao Deng, Ke Zou, Kai Ren, Meng Wang, Xuedong Yuan, Sancong Ying, and Huazhu Fu. Sam-u: Multi-box prompts triggered uncertainty estimation for reliable sam in medical image. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 368–377. Springer, 2023. doi: 10.48550/arXiv.2307.04973.

Shiladitya Dutta, Hongbo Wei, Lars van der Laan, and Ahmed M Alaa. Estimating uncertainty in multimodal foundation models using public internet data. *arXiv preprint arXiv:2310.09926*, 2023. doi: 10.48550/arXiv.2310.09926.

Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010. doi: 10.1007/s11263-009-0275-4.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059. PMLR, 2016.

Jakob Gawlikowski, Cedrique Rovile Njieutcheu Tassi, Mohsin Ali, Jongseok Lee, Matthias Humt, Jianxiang Feng, Anna Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, et al. A survey of uncertainty in deep neural networks. *Artificial intelligence review*, 56(Suppl 1):1513–1589, 2023. doi: 10.48550/arXiv.2107.03342.

Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021.

Cornelia Gruber, Patrick Oliver Schenk, Malte Schierholz, Frauke Kreuter, and Göran Kauermann. Sources of uncertainty in supervised machine learning–a statisticians' view. *arXiv preprint arXiv:2305.16703*, 2023. doi: 10.48550/arXiv.2305.16703.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. *International conference on machine learning*, pp. 1321–1330, 2017.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. doi: 10.48550/arXiv.1703.06870.

Wenchong He, Zhe Jiang, Tingsong Xiao, Zelin Xu, and Yukun Li. A survey on uncertainty quantification methods for deep learning. *ACM Computing Surveys*, 58(7):1–35, 2026. doi: 10.48550/arXiv.2302.13425.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016. doi: 10.48550/arXiv.1610.02136.

Dan Hendrycks, Steven Basart, Mantas Mazeika, Andy Zou, Joe Kwon, Mohammadreza Mostajabi, Jacob Steinhardt, and Dawn Song. Scaling out-of-distribution detection for real-world settings. *arXiv preprint arXiv:1911.11132*, 2019. doi: 10.48550/arXiv.1911.11132.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Max Andreas Ingrisch, Subashkumar Rajanayagam, Ingo Chmielewski, and Stefan Twieg. Calibration of the open-vocabulary model yolo-world by using temperature scaling. *International Conference on Applied Innovations in IT*, 2025. doi: http://dx.doi.org/10.25673/119228.

Wei Ji, Jingjing Li, Qi Bi, Tingwei Liu, Wenbo Li, and Li Cheng. Segment anything is not always perfect: An investigation of sam on different real-world applications, 2024.

Glenn Jocher and Jing Qiu. Ultralytics YOLO11, 2024. URL https://github.com/ultralytics/ultralytics.

Ian T Jolliffe. *Principal component analysis*. Springer, 2002.

Timo Kaiser, Thomas Norrenbrock, and Bodo Rosenhahn. UncertainSAM: Fast and efficient uncertainty quantification of the segment anything model. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=G3j3kq7rSC.

Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30, 2017.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *2023 IEEE/CVF international conference on computer vision (ICCV)*, pp. 3992–4003. IEEE, 2023.

Fabian Küppers, Anselm Haselhoff, Jan Kronenberger, and Jonas Schneider. Confidence calibration for object detection and segmentation. In *Deep neural networks and data for automated driving: Robustness, uncertainty quantification, and insights towards safety*, pp. 225–250. Springer International Publishing Cham, 2022.

Marc Lafon, Yannis Karmim, Julio Silva-Rodríguez, Paul Couairon, Clément Rambour, Raphaël Fournier-Sniehotta, Ismail Ben Ayed, Jose Dolz, and Nicolas Thome. Vilu: Learning vision-language uncertainties for failure prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17807–17817, 2025.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

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

Kangning Liu, Brian Price, Jason Kuen, Yifei Fan, Zijun Wei, Luis Figueroa, Krzysztof J Geras, and Carlos Fernandez-Granda. Uncertainty-aware fine-tuning of segmentation foundation models. *Advances in Neural Information Processing Systems*, 37:53317–53389, 2024a.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European conference on computer vision*, pp. 38–55. Springer, 2024b.

Chengqi Lyu, Wenwei Zhang, Haian Huang, Yue Zhou, Yudong Wang, Yanyi Liu, Shilong Zhang, and Kai Chen. Rtmdet: An empirical study of designing real-time object detectors. *arXiv preprint arXiv:2212.07784*, 2022.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in neural information processing systems*, 32, 2019.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Quazi Marufur Rahman, Niko Sünderhauf, Peter Corke, and Feras Dayoub. Fsnet: A failure detection framework for semantic segmentation. *IEEE Robotics and Automation Letters*, 7(2):3030–3037, 2022.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. In *International Conference on Learning Representations*, volume 2025, pp. 28085–28128, 2025.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015. doi: 10.48550/arXiv.1506.01497.

Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, et al. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024.

Tobias Riedlinger, Matthias Rottmann, Marius Schubert, and Hanno Gottschalk. Gradient-based quantification of epistemic uncertainty for deep object detectors. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 3921–3931, 2023.

Matthias Rottmann, Pascal Colling, Thomas Paul Hack, Robin Chan, Fabian Hüger, Peter Schlicht, and Hanno Gottschalk. Prediction error meta classification in semantic segmentation: Detection via aggregated dispersion measures of softmax probabilities. In *2020 international joint conference on neural networks (IJCNN)*, pp. 1–9. IEEE, 2020.

Marius Schubert, Karsten Kahl, and Matthias Rottmann. Metadetect: Uncertainty quantification and prediction quality estimates for object detection. In *2021 international joint conference on neural networks (IJCNN)*, pp. 1–10. IEEE, 2021.

Yi Tang Soon and Jun-Wei Hsieh. Confidence scores in open-vocabulary detection are a biased mixture of scale and semantics. *arXiv preprint arXiv:2607.10993*, 2026.

Uddeshya Upadhyay, Shyamgopal Karthik, Massimiliano Mancini, and Zeynep Akata. Probvlm: Probabilistic adapter for frozen vison-language models. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1899–1910. IEEE, 2023.

Sagar Vaze, Kai Hant, Andrea Vedaldi, and Andrew Zisserman. Generalized Category Discovery. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7482–7491, New Orleans, LA, USA, June 2022. IEEE. ISBN 978-1-6654-6946-3. doi: 10.1109/CVPR52688.2022.00734.

Jianzong Wu, Xiangtai Li, Shilin Xu, Haobo Yuan, Henghui Ding, Yibo Yang, Xia Li, Jiangning Zhang, Yunhai Tong, Xudong Jiang, et al. Towards open vocabulary learning: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 46(7):5092–5113, 2024.

Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019.

Xinyu Yang, Junlin Han, Rishi Bommasani, Jinqi Luo, Wenjie Qu, Wangchunshu Zhou, Adel Bibi, Xiyao Wang, Jaehong Yoon, Elias Stengel-Eskin, et al. Reliable and responsible foundation models. *Transactions on Machine Learning Research*, 2025.

Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M. Ni, and Heung-Yeung Shum. DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection. In *International Conference on Learning Representations (ICLR)*, 2023a. doi: 10.48550/arXiv.2203.03605.

Yichi Zhang, Shiyao Hu, Sijie Ren, Chen Jiang, Yuan Cheng, and Yuan Qi. Enhancing the reliability of segment anything model for auto-prompting medical image segmentation with uncertainty rectification. *arXiv preprint arXiv:2311.10529*, 2023b.

A Appendix

A.1 Model Checkpoints

Table 2 lists, for every classical and VFM architecture evaluated in Sec. 4, the exact backbone and source used.

A.2 Training and Preprocessing Details

For every vision model, dataset, and feature configuration, we train a separate meta-classifier: a feed-forward network with one hidden layer (128 units, ReLU), optimized with Adam (lr $= 10^{-3}$, weight decay 10^{-4}) and a validation-loss plateau scheduler (factor 0.5, patience 10 epochs, minimum lr 10^{-6}) for up to 120 epochs with batch size 64, retaining the checkpoint with the lowest validation loss. Each dataset is split at the image level into 70%/15%/15% train/validation/test partitions, so that predictions from the same image cannot appear across splits. We perform 10-fold cross-validation over the training and validation partitions and average all test metrics over folds.

All features are standardized using training-fold statistics only, then re-standardized after concatenation. The 768-dimensional CLIP ViT-L/14 embedding is further reduced via PCA fitted on the training predictions with the number of retained components k carefully tuned per model-dataset combination to maximize failure-detection performance on the validation set. This procedure is repeated independently for each fold, so no validation or test information enters the feature transformation.

A.3 Expected Calibration Error

ECE is defined here explicitly, following Guo et al. (2017). For a set of n predictions with meta-classifier estimate $\hat{p}_i = \Phi(h_i)$ and binary success label r_i (Sec. 3.1), ECE is computed by partitioning the unit interval $[0, 1]$ into $M = 15$ equal-width bins $B_1, \ldots, B_M$. Each prediction i is assigned to the bin containing $\hat{p}_i$. Within a bin B_m, the accuracy and mean confidence are defined as

$$\mathrm{acc}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} r_i, \qquad \mathrm{conf}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \hat{p}_i. \tag{9}$$

ECE is the weighted average, over all non-empty bins, of the absolute gap between these two quantities,

$$\mathrm{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{n} \left| \mathrm{acc}(B_m) - \mathrm{conf}(B_m) \right|. \tag{10}$$

Intuitively, $\mathrm{acc}(B_m)$ is the observed fraction of correct predictions among those assigned a predicted probability near $\mathrm{conf}(B_m)$. A well-calibrated Φ satisfies $\mathrm{acc}(B_m) \approx \mathrm{conf}(B_m)$ for every bin, yielding $\mathrm{ECE} = 0$ in the ideal case. Larger ECE values indicate larger discrepancies between predicted probabilities and the observed fraction of correct predictions. ECE is computed on the same held-out test predictions as AUROC and the other metrics (Sec. 4.1), and, like them, is averaged over the 10 cross-validation folds.

Table 2: Vision models evaluated in our experiments. All models are used frozen. Domain-specific models are fine-tuned on the respective target datasets.

Model	Abbrev.	Task	Backbone / Variant	Framework
Classical				
Faster R-CNN (Ren et al., 2015)	F-RCNN	Detection	ResNet-50-FPN	Detectron2
YOLOX (Ge et al., 2021)	YOLOX	Detection	YOLOX-L	MMDetection
DINO (Zhang et al., 2023a)	DINO	Detection	Swin-L, 5-scale	MMDetection
Mask R-CNN (He et al., 2017)	M-RCNN	Instance segmentation	ResNet-50-FPN	Detectron2
Mask2Former (Cheng et al., 2022)	M2Former	Instance segmentation	Swin-T	MMDetection
RTMDet (Lyu et al., 2022)	RTMDet	Instance segmentation	RTMDet-Ins-L	MMDetection
VFM				
Grounding DINO (Liu et al., 2024b)	GD	Detection	Swin-T	Official repo
YOLO-World (Cheng et al., 2024)	YOLO-W	Detection	YOLO-World-v2-L	Official repo
Grounded SAM (Ren et al., 2024)	Gr. SAM	Instance segmentation	GD Swin-T + SAM ViT-H	Official repo
Grounded SAM2	Gr. SAM2	Instance segmentation	GD Swin-T + SAM2 Hiera-L	Official repo
SAM3 (Carion et al., 2026)	SAM3	Instance segmentation		Official repo
Domain-specific				
YOLO11m-Pose (Jocher & Qiu, 2024)	YOLO11m	Detection	YOLO11m-Pose	Ultralytics
YOLO11x (Jocher & Qiu, 2024)	YOLO11x	Detection	YOLO11x	Ultralytics

Detectron2 (Wu et al., 2019); MMDetection (Chen et al., 2019); "Official repo" denotes the model's own official codebase (IDEA-Research Grounding DINO, YOLO-World, Segment Anything / SAM2 / SAM3) rather than a general-purpose detection framework. Ultralytics denotes the official Ultralytics YOLO11 implementation.

A.4 Score Baseline – Full Results

Tables 3 and 4 give the full per-model breakdown underlying the aggregated comparison in Table 1, for COCO and Pascal VOC respectively, sorted within each task block by decreasing model AP. The raw score does not depend on training and therefore has zero fold-to-fold variance by construction, unlike h_i^{base}, whose values are averaged over the 10 cross-validation folds (Sec. 4.1).

Table 3: Raw self-reported score versus Φ under h_i^{base} on COCO, grouped by classical versus VFM and sorted within each group by decreasing model AP.

		AUROC (↑)		ECE (↓)	
Model	Type	raw	h_i^{base}	raw	h_i^{base}
Object Detection					
DINO	classical	0.607	**0.879**	0.203	**0.016**
YOLOX	classical	0.721	**0.786**	0.182	**0.015**
Faster R-CNN	classical	0.710	**0.806**	0.142	**0.015**
YOLO-World	VFM	0.688	**0.815**	0.122	**0.046**
Grounding DINO	VFM	0.772	**0.853**	0.033	0.033
Instance Segmentation					
Mask2Former	classical	0.642	**0.820**	0.157	**0.013**
RTMDet	classical	0.705	**0.828**	0.080	**0.020**
Mask R-CNN	classical	0.729	**0.798**	0.106	**0.021**
SAM 3	VFM	0.799	**0.837**	0.115	**0.036**
Grounded SAM	VFM	0.778	**0.851**	0.045	**0.026**
Grounded SAM 2	VFM	0.785	**0.843**	0.062	**0.043**

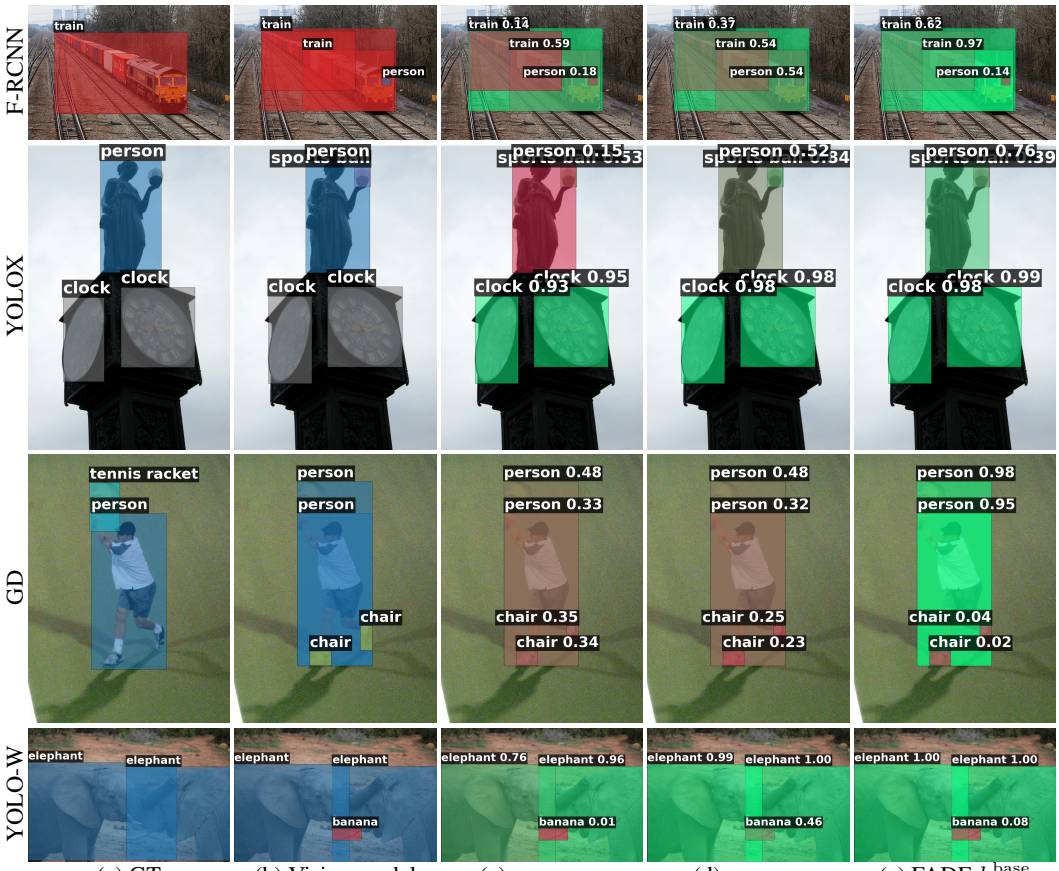
Table 4: Raw self-reported score versus Φ under h_i^{base} on Pascal VOC, grouped by classical versus VFM and sorted within each group by decreasing model AP.

		AUROC (↑)		ECE (↓)	
Model	Type	raw	h_i^{base}	raw	h_i^{base}
Object Detection					
DINO	classical	0.536	**0.939**	0.238	**0.006**
YOLOX	classical	0.765	**0.875**	0.150	**0.021**
Faster R-CNN	classical	0.722	**0.898**	0.159	**0.023**
YOLO-World	VFM	0.662	**0.840**	0.129	**0.016**
Grounding DINO	VFM	0.841	**0.947**	0.095	**0.019**
Instance Segmentation					
RTMDet	classical	0.714	**0.899**	0.043	**0.018**
Mask2Former	classical	0.597	**0.933**	0.127	**0.017**
Mask R-CNN	classical	0.808	**0.912**	0.078	**0.022**
SAM 3	VFM	0.877	**0.905**	0.125	**0.055**
Grounded SAM	VFM	0.872	**0.945**	0.143	**0.027**
Grounded SAM 2	VFM	0.873	**0.935**	0.146	**0.036**

A.5 Qualitative Results Across All Models

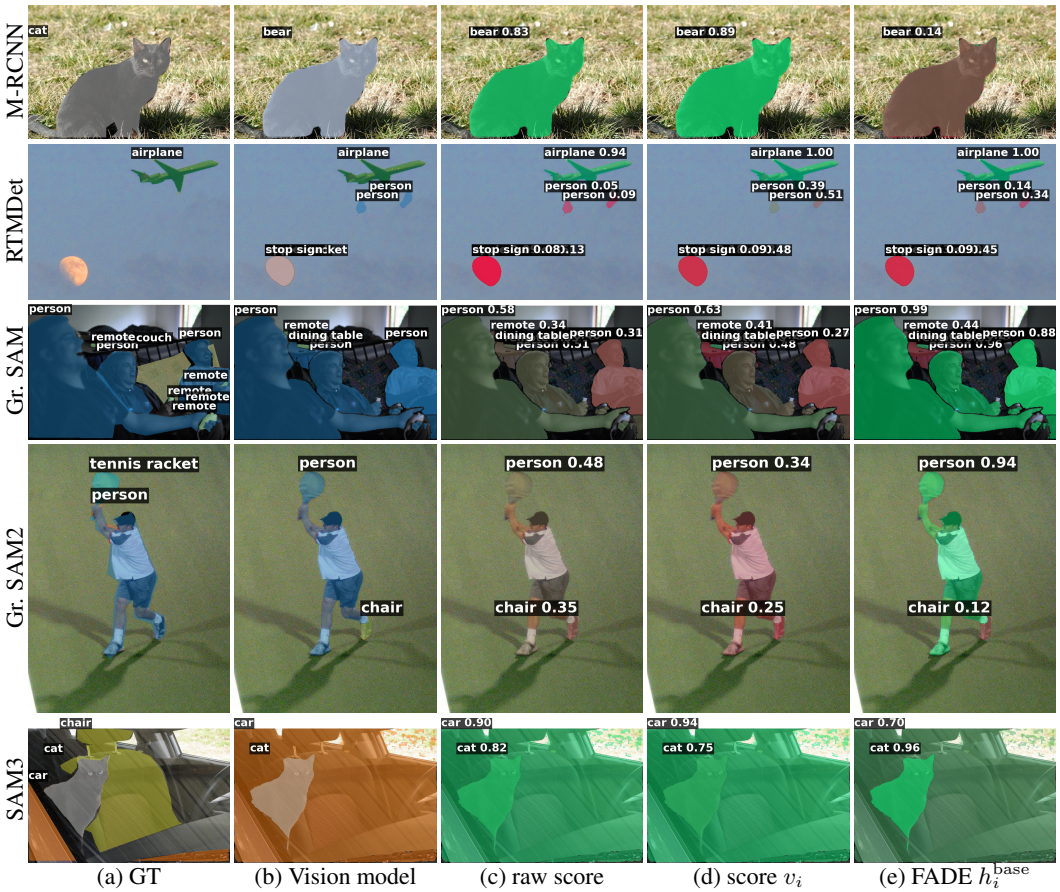
Figs. 4–8 extend Fig. 1 to the evaluated models on all three datasets, with one row per model showing the ground truth, the vision model's prediction, the raw-score baseline, the score-only meta-classifier (v_i), and the FADE meta-classifier h_i^{base}. In the visualizations, green predictions correspond to high estimated probabilities of success, whereas red predictions indicate low values, with ideally correct ones shown in green and incorrect ones in dark red. The examples show cases where the raw score is misleading. In comparison, the meta-classifiers can assign higher reliability to correct predictions with only moderate raw scores. The full h_i^{base} representation generally provides the clearest distinction.

Fig. 9 complements these per-model views with a direct comparison across model families. It shows one Pascal VOC image, evaluated under h_i^{base} for all six instance segmentation models. This example was selected to illustrate how differently the two families can behave on the same input. The three VFMs segment the cow almost identically, as a single instance with a high estimated probability of success. The classical models, in contrast, diverge considerably. Mask R-CNN labels the cow as *horse*, which h_i^{base} assigns a low probability of success. Mask2Former splits it into many overlapping fragments with widely varying estimates. RTMDet segments it correctly but adds a spurious *horse* prediction, which is again assigned a low probability of success. Despite these differences in the underlying predictions, all estimates are produced from the same model-agnostic feature representation and are therefore directly comparable across model families.



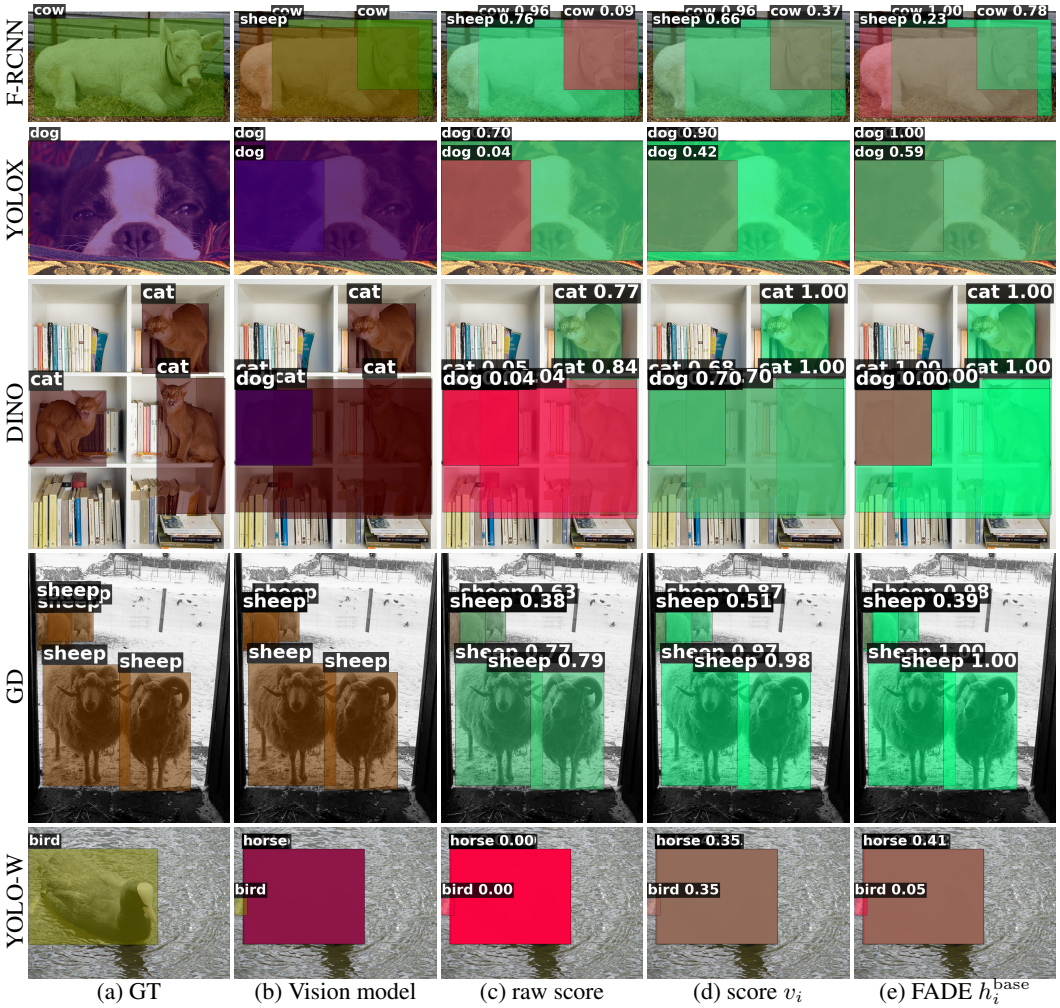
(a) GT (b) Vision model (c) raw score (d) score v_i (e) FADE h_i^{base}

Figure 4: Qualitative comparison on COCO object detection, showing ground truth, vision model prediction, raw score, score-only meta-classifier v_i, and FADE h_i^{base} for a held-out test image per model. DINO is omitted because it produces too many objects per image for a clear visualization.



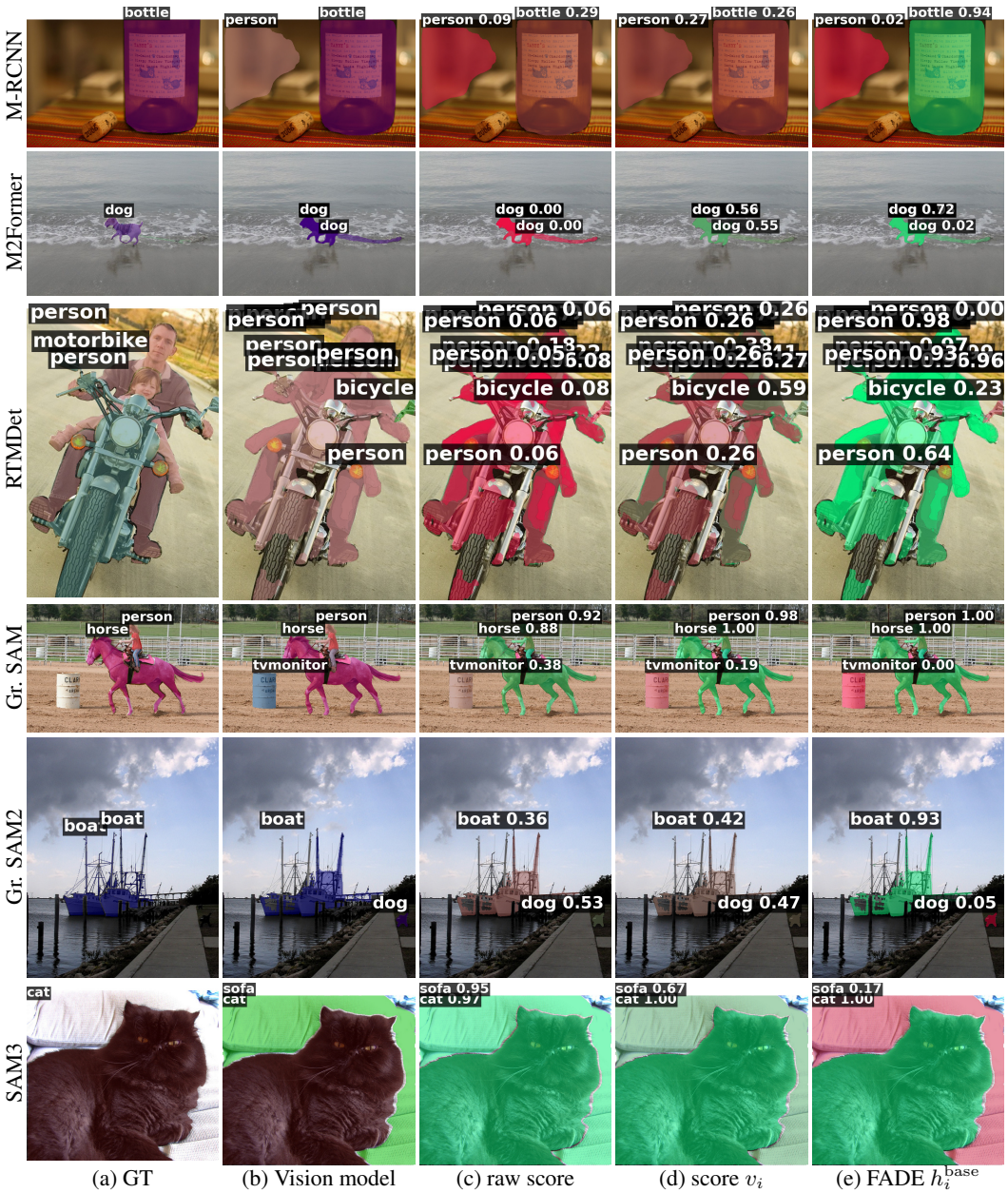
(a) GT (b) Vision model (c) raw score (d) score v_i (e) FADE h_i^{base}

Figure 5: Qualitative comparison on COCO instance segmentation, showing ground truth, vision model prediction, raw score, score-only meta-classifier v_i, and FADE h_i^{base} for a held-out test image per model. Mask2Former omitted because it produces too many objects per image for a clear visualization. Model abbreviations are M-RCNN for Mask R-CNN, Gr. SAM for Grounded SAM, and Gr. SAM2 for Grounded SAM2.



(a) GT (b) Vision model (c) raw score (d) score v_i (e) FADE h_i^{base}

Figure 6: Qualitative comparison on Pascal VOC object detection, showing ground truth, vision model prediction, raw score, score-only meta-classifier v_i, and FADE h_i^{base} for a held-out test image per model.



(a) GT (b) Vision model (c) raw score (d) score v_i (e) FADE h_i^{base}

Figure 7: Qualitative comparison on Pascal VOC instance segmentation, showing ground truth, vision model prediction, raw score, score-only meta-classifier v_i, and FADE h_i^{base} for a held-out test image per model. Model abbreviations are M-RCNN for Mask R-CNN, M2Former for Mask2Former, Gr. SAM for Grounded SAM, and Gr. SAM2 for Grounded SAM2.

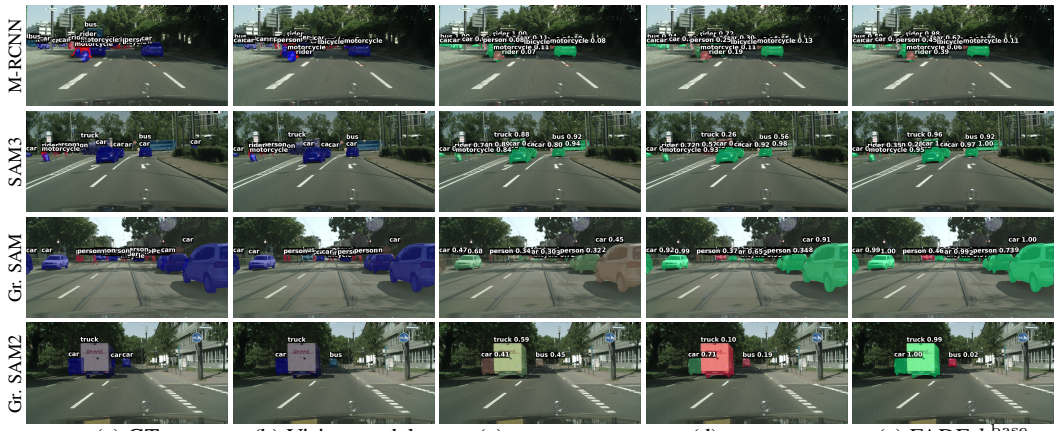


Figure 8: Qualitative comparison on Cityscapes instance segmentation, showing ground truth, vision model prediction, raw score, score-only meta-classifier v_i, and FADE h_i^{base} for a held-out test image per model. Model abbreviations are M-RCNN for Mask R-CNN, Gr. SAM for Grounded SAM, and Gr. SAM2 for Grounded SAM2.

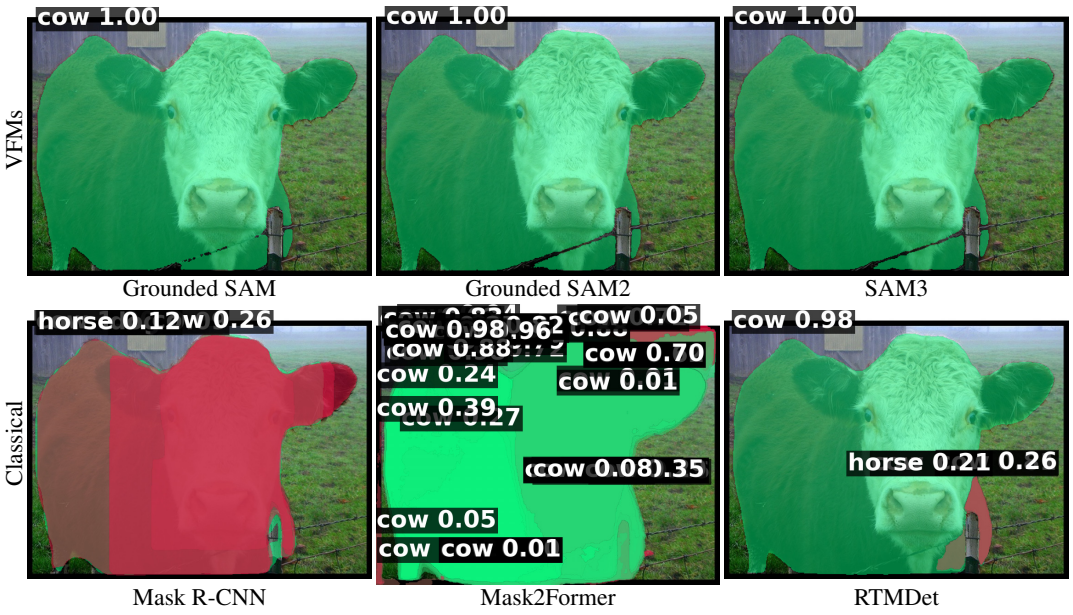


Figure 9: FADE h_i^{base} estimated probability of success for the same Pascal VOC image across all six instance segmentation models, VFMs (top) and classical models (bottom). Because h_i^{base} is built only from model-agnostic features, the estimates are directly comparable across model families, even though the underlying predictions differ considerably.

A.6 Full Evaluation Metrics under h_i^{base}

Tables 5 and 6 give the complete metric breakdown underlying the calibration discussion of Sec. 4.2: AUROC, ECE, balanced accuracy, and FPR@95%TPR for all 22 model-dataset combinations under h_i^{base}.

Table 5: Meta-classification results under h_i^{base} for all 11 COCO model combinations, sorted within each block by decreasing model AP. All metrics are test-set values, mean over 10 folds; per-fold standard errors are below 0.012 throughout and omitted for readability.

Model	Type	AP (↑)	AUROC (↑)	ECE (↓)	Bal. Acc. (↑)	FPR@95%TPR (↓)
Object Detection						
DINO	classical	0.573	0.879	0.016	0.795	0.538
YOLOX	classical	0.491	0.786	0.015	0.715	0.787
YOLO-World	VFM	0.460	0.815	0.046	0.738	0.708
Grounding DINO	VFM	0.418	0.853	0.033	0.770	0.642
Faster R-CNN	classical	0.402	0.806	0.015	0.735	0.761
Instance Segmentation						
Mask2Former	classical	0.448	0.820	0.013	0.732	0.640
SAM 3	VFM	0.441	0.837	0.036	0.755	0.667
RTMDet	classical	0.437	0.828	0.020	0.745	0.675
Mask R-CNN	classical	0.372	0.798	0.021	0.729	0.784
Grounded SAM	VFM	0.340	0.851	0.026	0.769	0.647
Grounded SAM 2	VFM	0.331	0.843	0.043	0.767	0.666

Table 6: Meta-classification results under h_i^{base} for all 11 Pascal VOC model combinations, sorted within each block by decreasing model AP. All metrics are test-set values, mean over 10 folds; per-fold standard errors are below 0.012 throughout and omitted for readability.

Model	Type	AP	AUROC (↑)	ECE (↓)	Bal. Acc. (↑)	FPR@95%TPR (↓)
Object Detection						
YOLO-World	VFM	0.631	0.840	0.016	0.757	0.663
Grounding DINO	VFM	0.567	0.947	0.019	0.878	0.292
DINO	classical	0.525	0.939	0.006	0.864	0.340
YOLOX	classical	0.491	0.875	0.021	0.796	0.629
Faster R-CNN	classical	0.394	0.898	0.023	0.822	0.545
Instance Segmentation						
SAM 3	VFM	0.559	0.905	0.055	0.809	0.499
Grounded SAM	VFM	0.448	0.945	0.027	0.869	0.330
Grounded SAM 2	VFM	0.432	0.935	0.036	0.845	0.396
RTMDet	classical	0.369	0.899	0.018	0.818	0.526
Mask2Former	classical	0.369	0.933	0.017	0.856	0.363
Mask R-CNN	classical	0.316	0.912	0.022	0.833	0.499

A.7 Effect of VFM Fine-tuning on Calibration

A natural question raised by Sec. 4.2's classical/VFM calibration gap is whether it is simply an artifact of evaluating VFMs zero-shot, and would close once a VFM is finetuned on the target dataset. We test this directly for Grounding DINO on COCO object detection, comparing the zero-shot checkpoint used throughout the paper against a COCO-finetuned Grounding DINO (Swin-T) checkpoint from MMDetection's model zoo (Chen et al., 2019), evaluated under the exact same meta-classification protocol (Sec. 4.1) and all five feature configurations.

Fine-tuning substantially improves detection performance, raising AP from 0.418 to 0.525, but does not close the calibration gap. Under h_i^{base}, AUROC drops from 0.853 to 0.806 and ECE nearly doubles, from 0.033 to 0.067 (Table 7); the same direction holds for every one of the five feature configurations, both for discrimination and for calibration. This suggests that the classical/VFM calibration gap described in Sec. 4.2 is not merely a byproduct of zero-shot evaluation of VFMs, and that fine-tuning for detection performance does not automatically lead to better-calibrated reliability estimates. This is consistent with our more general finding that AP and meta-classification quality capture different, only loosely related properties of a model (Sec. 4.2), and reinforces the conclusion that a post-hoc meta-classifier such as FADE remains necessary even for finetuned VFMs.

Table 7: Zero-shot versus COCO-finetuned Grounding DINO on COCO object detection, under all five feature configurations. Fine-tuning raises AP from 0.418 to 0.525 but does not improve meta-classification AUROC or ECE under any configuration.

	AUROC (↑)		ECE (↓)	
Feature config	zero-shot	finetuned	zero-shot	finetuned
v_i	**0.808**	0.786	**0.037**	0.098
$[u_i^\top, v_i^\top]^\top$	**0.847**	0.803	**0.026**	0.073
p_i	**0.783**	0.725	**0.043**	0.079
t_i	**0.748**	0.689	**0.022**	0.067
h_i^{base}	**0.853**	0.806	**0.033**	0.067

Table 8: Meta-classification AUROC (↑) on COCO under all seven completed feature configurations, sorted within each block by decreasing model AP.

Model	Type	v_i	u_i	p_i	$[u_i^\top, v_i^\top]^\top$	t_i	h_i^{base}	h_i^{full}
Object Detection								
DINO	classical	0.732	0.733	0.763	0.837	0.729	**0.879**	0.877
YOLOX	classical	0.756	0.678	0.700	**0.786**	0.672	**0.786**	0.780
YOLO-World	VFM	0.769	0.643	0.703	0.799	0.634	**0.815**	0.813
Grounding DINO	VFM	0.808	0.749	0.783	0.847	0.748	**0.853**	**0.853**
Faster R-CNN	classical	0.764	0.639	0.716	0.784	0.638	**0.806**	0.805
Instance Segmentation								
Mask2Former	classical	0.734	0.738	0.763	0.811	0.693	0.820	**0.822**
SAM 3	VFM	0.814	0.728	0.756	0.830	0.708	**0.837**	0.836
RTMDet	classical	0.767	0.665	0.714	0.809	0.644	0.828	**0.829**
Mask R-CNN	classical	0.766	0.638	0.706	0.782	0.635	0.798	**0.800**
Grounded SAM	VFM	0.812	0.740	0.772	0.849	0.724	0.851	**0.852**
Grounded SAM 2	VFM	0.808	0.738	0.762	0.841	0.716	**0.843**	**0.843**

A.8 Feature Ablation – Full Results

This section gives the full per-model breakdown underlying the feature ablation summarized in Sec. 4.2, reporting exact values for all seven feature configurations. Fig. 3 already shows five of them (v_i, p_i, $[u_i^\top, v_i^\top]^\top$, t_i, and h_i^{base}). Here we give exact numbers for these, together with the two configurations omitted there, geometry alone (u_i) and the full representation h_i^{full}. Tables 8 and 10 report AUROC for all seven configurations; corresponding ECE values are given in Tables 9 and 11. Tables 12 and 13 give the same breakdown for Cityscapes instance segmentation, covering one classical model (Mask R-CNN) and three VFMs (SAM3, Grounded SAM, Grounded SAM2).

The model-reported score alone (v_i) already provides a useful signal after meta-training, but combining it with geometric information consistently improves AUROC. For example, on COCO, the mean AUROC of v_i is substantially lower than that of $[u_i^\top, v_i^\top]^\top$ for both classical models and VFMs. Geometry alone (u_i), in contrast, is generally insufficient to provide a strong reliability estimate. This indicates that geometric properties are most useful when interpreted jointly with the confidence reported by the underlying model.

CLIP (p_i) provides a complementary source of information. Although p_i alone is generally weaker than $[u_i^\top, v_i^\top]^\top$, adding semantic information yields a further improvement for most models. The resulting h_i^{base} configuration achieves the strongest AUROC for all COCO models. On Pascal VOC, it is also strongest or tied for most models, while h_i^{full} marginally exceeds it for three models. An exception is YOLOX on COCO, for which adding p_i to $[u_i^\top, v_i^\top]^\top$ does not improve AUROC. Overall, the results support the use of semantic information as a complementary signal rather than as a replacement for model-reported confidence and geometric descriptors.

Task-specific structural features (t_i) do not provide a competitive standalone alternative. t_i consistently achieves lower AUROC than h_i^{base} and is also weaker than $[u_i^\top, v_i^\top]^\top$ for nearly all evaluated models. Adding these features to the model-agnostic representation therefore produces only marginal changes: on COCO, h_i^{full} is very close to h_i^{base} for all 11 models, with small improvements for some models and small decreases for others. The same behavior is observed on Pascal VOC.

Table 9: Meta-classification ECE (↓) on COCO under all seven completed feature configurations (lower is better), sorted within each block by decreasing model AP.

Model	Type	v_i	u_i	p_i	$[u_i^\top, v_i^\top]^\top$	t_i	h_i^{base}	h_i^{full}
Object Detection								
DINO	classical	0.049	0.026	0.020	**0.004**	0.031	0.016	0.018
YOLOX	classical	0.035	0.009	0.023	0.019	**0.008**	0.015	0.019
YOLO-World	VFM	0.070	**0.038**	0.044	0.063	0.044	0.046	0.045
Grounding DINO	VFM	0.037	0.024	0.043	0.026	**0.022**	0.033	0.037
Faster R-CNN	classical	0.023	0.012	0.024	0.012	**0.009**	0.015	0.016
Instance Segmentation								
Mask2Former	classical	0.094	0.031	0.022	0.014	0.066	0.013	**0.012**
SAM 3	VFM	0.049	0.039	0.053	0.036	0.057	0.036	**0.035**
RTMDet	classical	0.038	0.022	0.025	**0.014**	0.021	0.020	0.021
Mask R-CNN	classical	0.037	**0.011**	0.022	0.017	**0.011**	0.021	0.019
Grounded SAM	VFM	0.029	**0.018**	0.042	0.019	0.021	0.026	0.025
Grounded SAM 2	VFM	0.053	0.038	0.061	**0.034**	0.050	0.043	0.041

Table 10: Meta-classification AUROC (↑) on Pascal VOC under all seven completed feature configurations, sorted within each block by decreasing model AP.

Model	Type	v_i	u_i	p_i	$[u_i^\top, v_i^\top]^\top$	t_i	h_i^{base}	h_i^{full}
Object Detection								
YOLO-World	VFM	0.723	0.737	0.775	0.811	0.733	**0.840**	0.836
Grounding DINO	VFM	0.870	0.829	0.887	0.916	0.824	**0.947**	**0.947**
DINO	classical	0.686	0.841	0.870	0.880	0.838	**0.939**	**0.939**
YOLOX	classical	0.780	0.808	0.840	0.854	0.803	**0.875**	0.873
Faster R-CNN	classical	0.739	0.777	0.837	0.846	0.772	**0.898**	0.896
Instance Segmentation								
SAM 3	VFM	0.884	0.849	0.874	0.902	0.830	0.905	**0.908**
Grounded SAM	VFM	0.893	0.840	0.890	0.931	0.821	**0.945**	0.944
Grounded SAM 2	VFM	0.891	0.851	0.892	**0.938**	0.829	0.935	0.934
RTMDet	classical	0.746	0.812	0.868	0.866	0.774	**0.899**	**0.899**
Mask2Former	classical	0.765	0.891	0.922	0.910	0.868	0.933	**0.934**
Mask R-CNN	classical	0.820	0.809	0.862	0.897	0.795	0.912	**0.919**

This suggests that the additional task-specific structure can provide complementary information in individual cases, but is not required to obtain strong post-hoc reliability estimates.

The calibration results show a similar pattern. The h_i^{base} probabilities generally have substantially lower ECE than v_i alone and remain well calibrated across both datasets and model families. The lowest ECE is not always obtained by the configuration with the highest AUROC, illustrating that discrimination and calibration capture different properties of the reliability estimates (Sec. 4.2). For example, on COCO, DINO's $[u_i^\top, v_i^\top]^\top$ configuration reaches an ECE of 0.004, lower than h_i^{base}'s 0.016, despite a lower AUROC (0.837 vs. 0.879). We therefore consider AUROC and ECE jointly when assessing the quality of the meta-classifier.

A.9 Inference-Time Overhead of the Meta-Classifier

We measure the inference-time overhead of the meta-classifier directly on 100 real images per dataset/model combination, after a short unmeasured warm-up. The measurements are pooled over the five VFMs evaluated in this work, namely Grounding DINO, YOLO-World, SAM 3, Grounded SAM, and Grounded SAM 2. We distinguish between the inference cost of the underlying vision model and the additional cost introduced by the meta-classifier. The VFM cost corresponds to the forward pass of the vision model f alone and is measured on the GPU used throughout our experiments, a Quadro RTX 8000. For h_i^{base}, we show the cost of extracting the geometric features u_i, the model-reported score v_i, and the CLIP representation p_i, followed by the fitted PCA projection and the meta-classifier Φ. These operations are measured as a single end-to-end block on the CPU. We additionally measure a reduced CLIP-free configuration $[u_i^\top, v_i^\top, t_i^\top]^\top$, which uses the geometry, the model-reported score, and the task-specific structural features instead of the CLIP representation.

Table 11: Meta-classification ECE (↓) on Pascal VOC under all seven completed feature configurations (lower is better), sorted within each block by decreasing model AP.

Model	Type	v_i	u_i	p_i	$[u_i^\top, v_i^\top]^\top$	t_i	h_i^{base}	h_i^{full}
Object Detection								
YOLO-World	VFM	0.138	0.020	0.017	0.050	0.023	0.016	**0.014**
Grounding DINO	VFM	0.029	0.020	0.026	0.020	0.023	**0.019**	0.023
DINO	classical	0.045	0.032	0.018	0.015	0.032	**0.006**	**0.006**
YOLOX	classical	0.021	**0.017**	0.023	0.021	**0.017**	0.021	0.024
Faster R-CNN	classical	0.019	0.016	0.018	**0.015**	0.018	0.023	0.024
Instance Segmentation								
SAM 3	VFM	**0.043**	0.044	0.066	0.045	0.075	0.055	0.052
Grounded SAM	VFM	0.030	0.029	0.036	0.035	0.051	**0.027**	0.031
Grounded SAM 2	VFM	0.048	0.052	0.044	0.044	0.067	0.036	**0.033**
RTMDet	classical	0.143	0.031	0.026	**0.018**	0.065	**0.018**	0.020
Mask2Former	classical	0.147	0.016	0.022	**0.011**	0.050	0.017	0.018
Mask R-CNN	classical	0.054	0.029	0.023	0.029	0.037	**0.022**	0.023

Table 12: Meta-classification AUROC (↑) on Cityscapes instance segmentation under all seven feature configurations, sorted within each block by decreasing model AP.

Model	Type	v_i	u_i	p_i	$[u_i^\top, v_i^\top]^\top$	t_i	h_i^{base}	h_i^{full}
SAM 3	VFM	0.796	0.716	0.765	**0.832**	0.636	0.829	0.831
Mask R-CNN	classical	0.842	0.773	0.808	0.859	0.789	0.866	**0.868**
Grounded SAM 2	VFM	0.803	0.679	0.796	0.831	0.625	0.860	**0.862**
Grounded SAM	VFM	0.842	0.728	0.815	0.881	0.599	**0.909**	0.905

The results in Table 14 show that the additional cost of the full h_i^{base} configuration is comparable to the inference cost of the underlying VFMs themselves, with 9.54 s/image compared to 7.69 s/image. This overhead is dominated by CLIP feature extraction, which costs approximately 200–310 ms per object and therefore accumulates with the number of predictions in an image. In contrast, using $[u_i^\top, v_i^\top, t_i^\top]^\top$ avoids CLIP extraction and requires only 0.0099–0.032 s/image on COCO and Pascal VOC. On Cityscapes, its cost increases to 0.4346 s/image, primarily due to the processing of the high-resolution masks required for the task-specific structural features. The forward pass of the meta-classifier Φ itself contributes only marginally to the total overhead and remains below 0.05 s/image in all measured configurations. Thus, the computational cost of the proposed post-hoc assessment is primarily determined by feature extraction rather than by the meta-classifier itself, with the CLIP-free variant providing a substantially lower-cost alternative.

Table 13: Meta-classification ECE (↓) on Cityscapes instance segmentation under all seven feature configurations (lower is better), sorted within each block by decreasing model AP.

Model	Type	v_i	u_i	p_i	$[u_i^\top, v_i^\top]^\top$	t_i	h_i^{base}	h_i^{full}
SAM 3	VFM	0.093	0.057	0.063	0.050	0.079	0.048	**0.045**
Mask R-CNN	classical	0.069	0.033	0.035	0.042	0.058	0.034	**0.028**
Grounded SAM 2	VFM	0.088	0.069	0.050	**0.042**	0.108	0.049	0.043
Grounded SAM	VFM	0.087	0.068	0.047	0.031	0.061	**0.029**	0.030

Table 14: Average per-image inference cost (s/image) over the five VFMs (Sec. 4.1), with 100 real images per dataset and model combination. VFM denotes the forward pass of the underlying model f. FADE's h_i^{base} and additional cost of meta-classifier inference, including feature extraction.

Configuration	COCO	Pascal VOC	Cityscapes	Overall
VFM	8.84	4.46	9.77	7.69
FADE h_i^{base}	13.60	5.24	9.77	9.54
FADE $[u_i^\top, v_i^\top, t_i^\top]^\top$	**0.032**	**0.0099**	**0.4346**	**0.1588**

A.10 PCA Dimensionality Study

Sec. 3.2 treats the number of retained CLIP principal components as a hyperparameter. We consider $k \in \{2, 4, 8, 16, 32, 64, 128, 256\}$ and select k separately for each model-dataset combination based on validation AUROC. This section examines this selection for h_i^{base} across COCO, Pascal VOC, and Cityscapes, focusing first on the behavior of validation AUROC across k and then on the resulting test performance.

Across all 26 model-dataset combinations, validation AUROC varies only moderately with the number of retained principal components k. The gap between the best and worst k in the sweep is 0.024 AUROC on average (min 0.007, max 0.050). As shown in Figure 10, the curves typically increase from $k = 2$, peak between $k = 16$ and $k = 64$, and decline again toward $k = 256$. This behavior is consistent with the CLIP embedding's dimensionality being large relative to the number of labeled predictions available for training (Sec. 3.2). No single k is universally best. While $k = 32$ is selected most frequently (11 of 26 combinations), the selected value ranges from 4 to 128 across models and datasets, supporting the selection of k separately for each combination rather than fixing it globally.

Tables 15–17 report the resulting selected k for every combination, together with the corresponding test AUROC. These are identical to the h_i^{base} values already reported in Tables 8, 10, and 12, since this is the same validation-selected model.

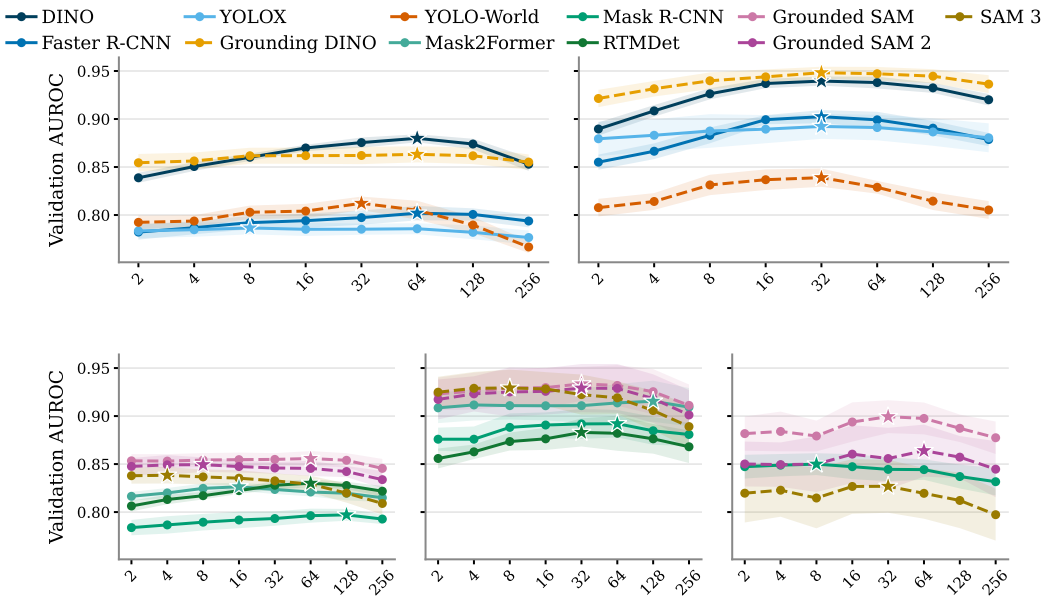


Figure 10: Validation AUROC (↑) of h_i^{base} as a function of the number of retained CLIP principal components k (log scale). AUROC is averaged over 10 cross-validation folds, with shaded bands indicating ± 1 standard deviation. From left to right, the top row shows object detection on COCO and Pascal VOC, while the bottom row shows instance segmentation on COCO, Pascal VOC, and Cityscapes. The star marks the k selected based on validation AUROC for each model-dataset combination.

Table 15: Number of retained CLIP principal components k_{sel} selected by validation AUROC for h_i^{base} on COCO, and the resulting test AUROC (↑), sorted within each block by decreasing model AP.

Model	Type	k_{sel}	AUROC(k_{sel})
Object Detection			
DINO	classical	64	0.879
YOLOX	classical	8	0.786
YOLO-World	VFM	32	0.815
Grounding DINO	VFM	64	0.853
Faster R-CNN	classical	64	0.806
Instance Segmentation			
Mask2Former	classical	16	0.820
SAM 3	VFM	4	0.837
RTMDet	classical	64	0.828
Mask R-CNN	classical	128	0.798
Grounded SAM	VFM	64	0.851
Grounded SAM 2	VFM	8	0.843

Table 16: Number of retained CLIP principal components k_{sel} selected by validation AUROC for h_i^{base} on Pascal VOC, and the resulting test AUROC (↑), sorted within each block by decreasing model AP.

Model	Type	k_{sel}	AUROC(k_{sel})
Object Detection			
YOLO-World	VFM	32	0.840
Grounding DINO	VFM	32	0.947
DINO	classical	32	0.939
YOLOX	classical	32	0.875
Faster R-CNN	classical	32	0.898
Instance Segmentation			
SAM 3	VFM	8	0.905
Grounded SAM	VFM	32	0.945
Grounded SAM 2	VFM	32	0.935
Mask2Former	classical	128	0.933
RTMDet	classical	32	0.899
Mask R-CNN	classical	64	0.912

Table 17: Number of retained CLIP principal components k_{sel} selected by validation AUROC for h_i^{base} on Cityscapes, and the resulting test AUROC, sorted by decreasing model AP.

Model	Type	k_{sel}	AUROC(k_{sel}) (↑)
SAM 3	VFM	32	0.829
Mask R-CNN	classical	8	0.866
Grounded SAM 2	VFM	64	0.860
Grounded SAM	VFM	32	0.909

A.11 External Datasets – Full Results

For the domain-specific datasets, we use the same random split as in the rest of the paper for the fish dataset, and a split grouped by recording session for the surgical instruments dataset, since images within a session are likely correlated in viewpoint and visible instruments. Table 18 gives the full per-model results on the two domain-specific datasets summarized in Sec. 4.2, fish detection and surgical instruments. Figs. 12 and 11 give a qualitative example per model on each dataset. The examples illustrate that the meta-classifier can identify incorrect predictions as unreliable and correct predictions as reliable even when the model's own confidence score does not reflect the prediction uncertainty. This is particularly visible in cases where incorrect predictions receive high model-reported scores, while FADE assigns them a low probability of success.

Regarding the surgical-instrument caveat discussed in Sec. 4.2: 93% of Grounding DINO's 1274 successful detections are correctly identified gauze balls, a class it identifies correctly 98.6% of the time, unlike the other 22 metal instrument classes. v_i (which encodes the predicted class) can therefore exploit the distinction between gauze balls and metal instruments to achieve near-perfect AUROC, without reliably distinguishing among the other 22 classes. AP, which aggregates detection performance across all 23 classes, is correspondingly very low.

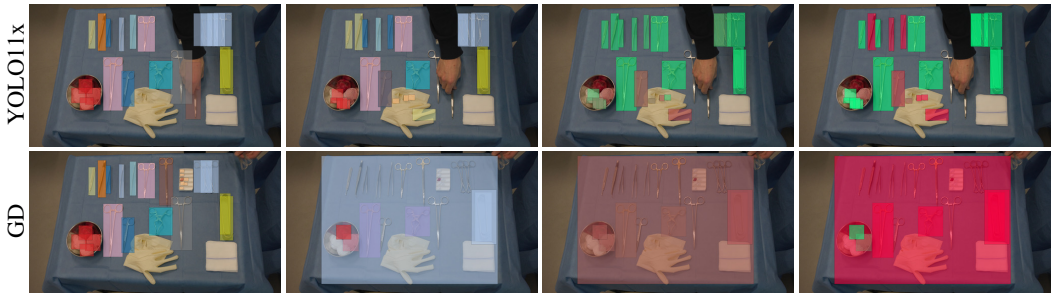
A.12 Domain-Adapted (PEFT) CLIP Features

Beyond the frozen, off-the-shelf CLIP encoder used throughout this paper for the semantic feature p_i, we additionally prepared a domain-adapted variant for the surgical-instrument dataset. We apply a parameter-efficient fine-tuning via a low rank adapter (LoRA) with rank 8 to the visual transformer of a CLIP ViT-B/32 model and fine-tune it on surgical-instrument imagery. The adapter is trained for 5 epochs using AdamW with a learning rate of 10^{-3} and a batch size of 8. Under this setup, the classification accuracy of CLIP increases from 20% to 60%. Semantic features from this LoRA-adapted CLIP encoder are extracted for every detection produced by the domain-specific YOLO11x.

Table 19 compares the off-the-shelf and domain-adapted CLIP features, both alone (p_i) and as part of h_i^{base}. The domain-adapted encoder does not provide a consistent benefit for YOLO11x. When used alone, it slightly improves AUROC ($0.954 \rightarrow 0.958$), although ECE increases from 0.026 to 0.033. Once combined with geometry and score, it instead reduces AUROC from 0.988 to 0.979 and increases ECE from 0.019 to 0.024. Thus, domain adaptation does not improve either discrimination or calibration when the semantic features are combined with the other model-agnostic cues. In particular, the off-the-shelf CLIP representation already achieves high AUROC and low ECE within h_i^{base}, leaving little room for improvement through this form of domain adaptation.

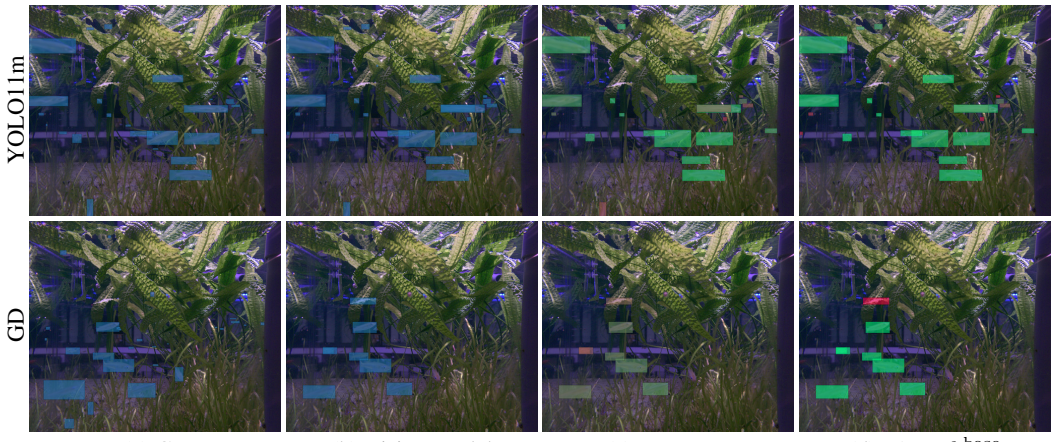
Table 18: Meta-classification results on the two external, domain-specific object-detection datasets, fish and surgical instruments, for v_i and h_i^{base}. With n, we denote the total number of predicted objects and with %pos the fraction of these meeting the success criterion r_i.

					AUROC (↑)		ECE (↓)	
Model	Type	n	%pos	AP	v_i	h_i^{base}	v_i	h_i^{base}
Surgical Instruments								
YOLO11x	domain-specific	16095	70.5%	0.516	0.902	**0.988**	0.062	**0.019**
Grounding DINO	VFM	474	26.9%	0.031	1.000	1.000	0.004	**0.002**
YOLO-World	VFM	195738	5.7%	0.098	0.839	**0.951**	0.113	**0.033**
Fish								
YOLO11m-Pose	domain-specific	2072	86.4%	0.345	0.892	**0.923**	0.100	**0.055**
Grounding DINO	VFM	1332	86.3%	0.259	0.860	**0.996**	0.093	**0.045**
YOLO-World	VFM	8624	39.4%	0.235	0.695	**0.955**	0.059	**0.028**



(a) GT (b) Vision model (c) raw score (d) FADE h_i^{base}

Figure 11: Qualitative comparison on the surgical-instrument dataset for YOLO11x and Grounding DINO (GD), illustrating how FADE assesses prediction reliability compared with the model's raw score.



(a) GT (b) Vision model (c) raw score (d) FADE h_i^{base}

Figure 12: Qualitative comparison on the fish detection dataset for YOLO11m-Pose (YOLO11m) and Grounding DINO (GD), illustrating how FADE assesses prediction reliability compared with the model's raw score.

Table 19: Off-the-shelf vs. LoRA domain-adapted CLIP encoder on surgical instruments (YOLO11x), p_i alone and as part of h_i^{base}. Domain adaptation shows no consistent benefit once combined with the other features.

	AUROC (↑)		ECE (↓)	
CLIP encoder	p_i	h_i^{base}	p_i	h_i^{base}
off-the-shelf	0.954	**0.988**	**0.026**	**0.019**
LoRA	**0.958**	0.979	0.033	0.024