DECENT: Distribution-aware Prototypical Calibration with Fine-grained Voting for VLM Misclassification Detection

Anonymous ACL submission

Abstract

This paper studies the problem of misclassification detection for vision-language models (VLMs), which aims to identify samples wrongly assigned by the pre-trained VLM. Previous works usually focus on exploring the latent space and measuring pairwise similarity scores for confidence estimation. However, these similarity scores can be biased for boundary points with different fine-grained features. To address this issue, this paper proposes a novel approach named Distribution-aware Prototypical Calibration with Fine-grained Voting (DECENT). The core idea of DECENT is to utilize probabilistic distributions to characterize both the semantics of the test image and its corresponding class prototype. Specifically, we first conduct clustering within each class in the latent space to construct a distribution of fine-grained prototypes. More importantly, multiple augmented views of the test image are considered as samples from its underlying semantic distribution. Then, we measure the distribution distance between the image's underlying semantic distribution and its neighboring fine-grained prototype distributions, which is aggregated via fine-grained voting into a novel distributional score. This new score can be fused with the VLM's semantic alignment score to significantly enhance misclassification detection sensitivity, or to improve overall classification robustness. Extensive experiments on benchmark datasets validate the dual effectiveness of the proposed DECENT.

1 Introduction

In recent years, Vision-Language Models (VLMs) have achieved remarkable success (Radford et al., 2021; Jia et al., 2021). By conducting contrastive learning on massive-scale image-text pairs, these models have effectively bridged the modality gap between vision and text, demonstrating a powerful cross-modal understanding. This capability has led to breakthrough advancements in a multitude of downstream tasks, such as image retrieval (Li et al., 2022), visual question answering (VQA) (Pramanick et al., 2024), and image captioning (Zhou et al., 2020).

However, despite their formidable capabilities, VLMs are not infallible. In practical applications, VLMs still produce misclassifications, yielding incorrect predictions. In many high-stakes domains, such as medical image diagnostics (Dadalto et al., 2023; Kakda et al., 2026), scene perception for autonomous driving (Boreiko et al., 2024; Zhang et al., 2025b; Zitkovich et al., 2023), or content moderation (AlDahoul et al., 2024), VLM misclassifications can lead to catastrophic consequences (Taioli et al., 2025; Vice et al., 2025; Guan et al., 2024). Therefore, the ability to accurately detect when a VLM is wrong (i.e., misclassification detection) has emerged as a critical and urgent research problem (Chen et al., 2021).

To address the challenge of VLM misclassification detection, previous work has largely focused on exploring the rich latent space of VLMs. These methods typically rely on computing pairwise similarity scores between a test sample and its predicted class, using this score as an estimate of confidence. The underlying assumption is that a correct prediction should exhibit a higher similarity score in the feature space than an incorrect one. However, this foundational strategy suffers from critical limitations that undermine its reliability in high-stakes scenarios. First, similarity-based approaches are notoriously unreliable for boundary points (i.e., hard samples near the decision boundary (Fawzi et al., 2018)). Their similarity scores in this region are often inherently biased and ambiguous. This boundary bias means that the most challenging cases, precisely where a confidence score is most needed, are the ones most likely to be misjudged, rendering the detector ineffective when it matters most. Second, these methods are fundamentally fine-grained insensitive (Peng et al.,

2024; Wu et al., 2025; Zhang et al., 2025a; Lee et al., 2024). By relying on a single holistic similarity score, they overlook subtle yet semantically critical intra-class features. Consequently, a VLM might confidently misclassify a *robin* as a *sparrow*, and existing detectors, being blind to these nuances, would fail to flag this obvious error. This insensitivity makes them unsuitable for real-world tasks requiring domain expertise, such as fine-grained species identification, defect detection, or medical anomaly analysis (Xiao et al., 2025).

To overcome the aforementioned limitations, we propose Distribution-aware Prototypical Calibration with Fine-grained Voting (DECENT), a novel framework for misclassification detection. Our core idea is a dual-branch architecture that leverages both a pre-trained VLM and an auxiliary image encoder. Our method generates two orthogonal scores: (1) Text-Image Semantic Score: Computed by the VLM, this score evaluates the degree of semantic alignment between the test image and the text prototype of its predicted class. (2) Image-Image Distributional Score: Computed by the auxiliary encoder. To specifically address the boundary point and fine-grained challenges, we first construct a database of fine-grained prototypes via clustering on a reference dataset. For a test sample, we generate a feature point cloud through data augmentation and then calculate the distribution distance between this point cloud and its nearest neighboring prototypes. This score is then generated via fine-grained soft-voting. Finally, we explore two strategies to fuse these scores via a weighted sum. One strategy (Fusion Prediction) uses the fused score for both prediction and confidence, while the other (Decoupled Prediction) uses the semantic score for prediction and a fused score for confidence. This fusion aims to produce a more robust and calibrated confidence score for misclassification detection. The main contributions of this paper are summarized as follows:

❶ ***Perspective Innovation.*** We propose a paradigm shift in VLM misclassification detection, moving beyond traditional reliance on single-point holistic similarity to explicitly model neighborhood distributional discrepancies. Building on this insight, we introduce DECENT, a dual-branch framework that synergizes the broad semantic alignment of VLMs with the fine-grained, distribution-aware perception of an auxiliary encoder.

❷ ***Methodological Design.*** We formulate an innovative distributional scoring mechanism that derives fine-grained class prototypes via feature clustering and leverages point-cloud-based distribution distances. By robustly evaluating sample reliability, this mechanism effectively mitigates the biased similarity scores inherently associated with challenging boundary samples.

❸ ***Empirical Superiority.*** Extensive evaluations across eight benchmark datasets demonstrate the good performance of DECENT. Our robust variant achieves an average accuracy of 77.59%, outperforming the state-of-the-art by a 6.69% margin. Concurrently, our sensitive variant establishes a new benchmark for detection capability, delivering an outstanding average AUROC of 93.02% and drastically reducing the average FPR95 to 33.28%.

2 Related Work

2.1 Vision-Language Models

In recent years, Vision-Language Models (VLMs) have achieved remarkable success. Pioneering work, exemplified by CLIP (Radford et al., 2021), ALIGN (Jia et al., 2021), and ViLBERT (Lu et al., 2019), has successfully aligned image and text features within a unified latent space through large-scale Contrastive Learning on massive datasets of image-text pairs. This potent cross-modal alignment capability enables VLMs to comprehend the intricate relationships between vision and language, thereby demonstrating exceptional generalization capabilities in a multitude of downstream tasks, such as image retrieval (Li et al., 2022), visual question answering (VQA) (Pramanick et al., 2024), and image captioning (Zhou et al., 2020). The success of these models has established a foundation for addressing complex cross-modal challenges. However, despite their formidable performance, issues concerning their robustness, interpretability, and predictive reliability remain significant areas of concern within the research community (Xiao et al., 2025; Xie et al., 2025; Groot and Valdenegro-Toro, 2024; Zhang et al., 2024).

2.2 Misclassification Detection

Detecting misclassifications for VLMs is a challenging task. Existing training-free methods, which operate without re-training the VLM, can be broadly categorized based on their approach: The most straightforward methods adapt classical techniques from classifier uncertainty. These

include using the maximum softmax probability (MSP) (Hendrycks and Gimpel, 2016), or the energy score (Liu et al., 2020b) from the VLM's output distribution. Other foundational approaches to uncertainty and OOD detection include MC Dropout (Gal and Ghahramani, 2016), Deep Ensembles (Lakshminarayanan et al., 2017), and ODIN (Liang et al., 2017; Lee et al., 2018; Maddox et al., 2019). While simple to implement, these methods primarily measure the ambiguity in the output space (Liu et al., 2020a; Fort et al., 2021; Qu et al., 2025), which often correlates poorly with true correctness and is highly susceptible to fine-grained insensitivity (e.g., *robin* vs. *sparrow*) (Mukhoti et al., 2020; Cheng and Vasconcelos, 2022; Peng et al., 2024; Zhang et al., 2025a). A more recent line of work leverages the VLM's rich latent space. Methods like DOCTOR (Granese et al., 2021) and TrustVLM (Dong et al., 2025) compute similarity scores (e.g., cosine similarity) between an image feature and the text prototype of its predicted class (Chopra et al., 2005; Jeong et al., 2023; Zhou et al., 2023). The core assumption is that misclassified samples will have lower similarity. However, this assumption breaks down for boundary points, where challenging-but-correct samples may also have low similarity, making them indistinguishable from true errors. Furthermore, as these methods typically compute a single, holistic similarity, they inherently fail to capture the intra-class feature variations needed to detect fine-grained misclassifications. The limitations of existing work highlight two critical gaps: (1) robustness to boundary points and (2) sensitivity to fine-grained features. Our work DECENT is explicitly designed to address both limitations. Unlike methods that rely on single-point similarity, we introduce a distribution distance (Székely and Rizzo, 2013; Chen and Ding, 2025) to robustly model the sample's ambiguity. Furthermore, by constructing fine-grained prototypes (Snell et al., 2017) via clustering (McQueen, 1967; Lloyd, 1982), we explicitly equip our model with the sensitivity needed to adjudicate subtle feature differences (Lu et al., 2024; Ke et al., 2024; Gong et al., 2025; Zou et al., 2025; Li et al., 2024).

3 The Proposed DECENT

3.1 Problem Definition

We study system-level misclassification detection for VLM-based inference. Let $M = (M_A, M_B)$ denote a dual-model system, where M_A is a VLM and M_B is an auxiliary image encoder, and let $D_{\text{ref}} = \{(x_i, y_i)\}_{i=1}^{N}$ be a labeled reference set. For a test image x with ground-truth label y_{true}, an inference strategy outputs

$$\begin{aligned} \hat{y} &= G_{\text{pred}}(x \mid M, D_{\text{ref}}), \\ s_{\text{conf}} &= F_{\text{conf}}(x, \hat{y} \mid M, D_{\text{ref}}), \\ r &= \mathbb{I}(\hat{y} = y_{\text{true}}). \end{aligned}$$

The objective is to make s_{conf} separate correct final predictions ($r = 1$) from misclassifications ($r = 0$). Different variants instantiate G_{pred} differently: DECENT-R uses the fused score for both prediction and confidence, whereas DECENT-S uses the semantic score for prediction and the fused score for reliability estimation.

3.2 Framework Overview

Our DECENT method evaluates the reliability of the VLM predictions. We hypothesize that an unreliable prediction (misclassification) will exhibit a misalignment with its true class prototype simultaneously in both the semantic space and the distributional manifold. Accordingly, we design a dual-branch framework that computes confidence scores from two orthogonal dimensions: Text-Image Semantic Alignment: Evaluates the similarity between the image $\mathbf{f}_A^I$ and the text-semantic prototype μ^T of its predicted class. Image-Image Distributional Discrepancy: Evaluates the discrepancy between the image's local feature distribution $\mathbf{P}_x$ and the fine-grained distributions $\mathcal{Z}$ of its predicted class. The final confidence s_{conf} is a fusion of these two branch scores. Model $\mathcal{M}_A$ (Text-Image): A VLM, providing $\mathcal{E}_A^I(\cdot)$ and $\mathcal{E}_A^T(\cdot)$. Model $\mathcal{M}_B$ (Image-Image): An image encoder, providing $\mathcal{E}_B^I(\cdot)$.

3.3 Text-Image Semantic Alignment

To establish a foundational semantic score, we leverage the pre-trained VLM to capture its primary, albeit sometimes unreliable, semantic judgment. This score is derived by comparing the test image's features against multi-template text prototypes for each class.

We first construct a multi-template, L2-normalized text center μ_j^T for each class c_j:

$$\mu_j^T = \text{L2-Norm}\left(\frac{1}{M}\sum_{m=1}^{M} \text{L2-Norm}(\mathcal{E}_A^T(t_m(c_j)))\right). \quad (1)$$

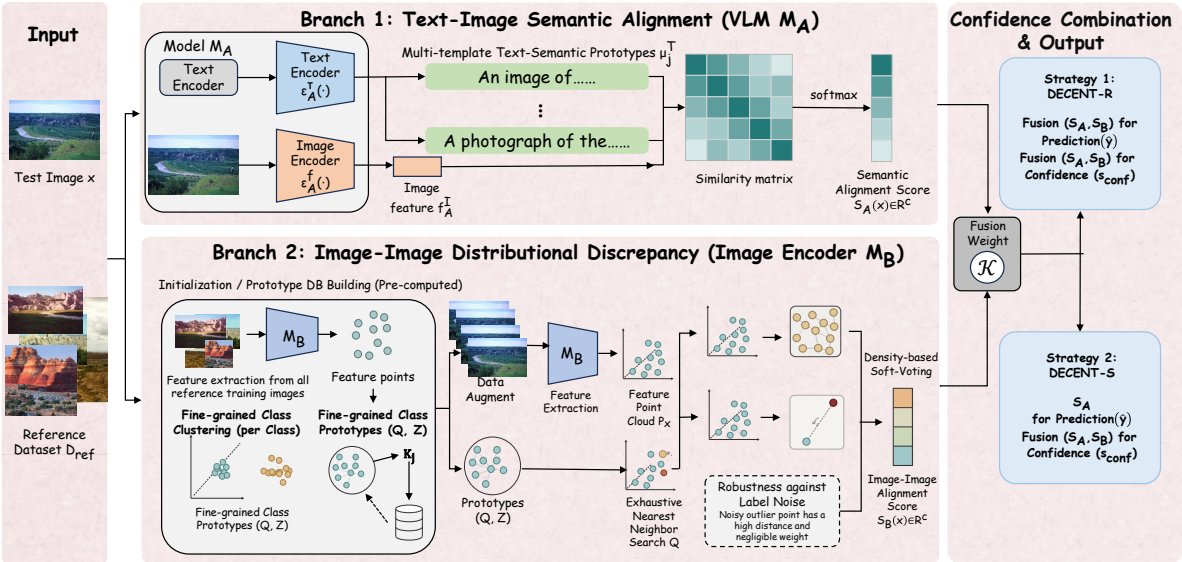


Figure 1: The overall framework of the proposed DECENT. DECENT detects misclassifications by fusing the VLM's semantic alignment score (S_A) with a novel distributional score (S_B). The distributional score (S_B) is computed by measuring the distribution distance between an image's feature point cloud and fine-grained class prototypes. The resulting fused score is used in two strategies: DECENT-S for high-sensitivity detection and DECENT-R for robust classification.

Given the extracted image feature $\mathbf{f}_A^I = \text{L2-Norm}(\mathcal{E}_A^I(x))$, we compute its cosine similarity with all C class centers. Applying a Softmax normalization with temperature τ_1 yields the semantic alignment score vector $\mathbf{S}_A(x) \in \mathbb{R}^C$:

$$\mathbf{S}_A(x)_j = \frac{\exp(\mathbf{f}_A^I \cdot (\mu_j^T)^T/\tau_1)}{\sum_{k=1}^{C} \exp(\mathbf{f}_A^I \cdot (\mu_k^T)^T/\tau_1)}. \quad (2)$$

3.4 Neighborhood Distributional Discrepancy

Because the semantic-only score is often unreliable for boundary points and insensitive to fine-grained features, we propose an orthogonal distributional score to correct this bias. This score measures the discrepancy between a test image's local feature distribution and its neighboring prototype clusters.

We begin by establishing a database of fine-grained prototypes. Using the auxiliary image encoder $\mathcal{M}_B$, we extract features $\mathbf{F}_B$ for all training samples. To avoid relying on an arbitrary hyperparameter, we dynamically determine the number of clusters K_j for each class via a density rule: $K_j = \lceil |\mathbf{F}_{B,j}|/\rho \rceil$, where the target density ρ ensures each fine-grained prototype covers an average of 10 to 20 reference images. Next, we apply K-Means (McQueen, 1967; Lloyd, 1982) clustering to the feature subset $\mathbf{F}_{B,j}$ of each class j to identify its K_j fine-grained sub-clusters. For the k-th cluster of class j, we store its normalized centroid $\mathbf{q}_{j,k}$ and the set of its member features $\mathbf{Z}_{j,k}$. This yields a comprehensive database $(\mathcal{Q}, \mathcal{Z})$ containing $\sum_{j=1}^{C} K_j$ prototypes and their corresponding members:

$$(\mathbf{q}_{j,k}, \mathbf{Z}_{j,k}) = \text{K-Means}_k(\mathbf{F}_{B,j}, K_j), \quad \forall j \in [1, C],\ \forall k \in [1, K_j]. \quad (3)$$

During inference, for a given test image x, we generate a feature point cloud $\mathbf{P}_x$ by applying V data augmentations $a_v(\cdot)$ (Kendall and Gal, 2017; Wang et al., 2019). This embeds the test sample into the same feature space defined by $\mathcal{M}_B$. We define this point cloud and compute its normalized centroid $\mathbf{p}_x$ as follows:

$$\mathbf{P}_x = \{\text{L2-Norm}(\mathcal{E}_B^I(a_v(x)))\}_{v=1}^{V}, \quad \mathbf{p}_x = \text{L2-Norm}\left(\frac{1}{V}\sum_{\mathbf{p}\in\mathbf{P}_x} \mathbf{p}\right). \quad (4)$$

With $\mathbf{P}_x$ and $\mathcal{Q}$ established, we identify the test sample's nearest neighbors via an exhaustive search within the feature space. First, we compute the Euclidean (L2) distance between the test centroid $\mathbf{p}_x$ and all prototype centroids $\mathbf{q} \in \mathcal{Q}$:

$$\text{Dist}(\mathbf{p}_x, \mathcal{Q}) = \{\|\mathbf{p}_x - \mathbf{q}\|_2 \mid \forall \mathbf{q} \in \mathcal{Q}\}. \quad (5)$$

Next, we select the N_{top} prototypes closest to $\mathbf{p}_x$ to form the nearest neighbor index set $\mathcal{I}_{top}$. To quantify the Neighborhood Distributional Discrepancy,

we calculate the Energy Distance (Székely and Rizzo, 2013) d_i^E between the test point cloud $\mathbf{P}_x$ and each selected neighboring cluster $\mathbf{Z}_i$ ($i \in \mathcal{I}_{top}$):

$$d_i^E = E(\mathbf{P}_x, \mathbf{Z}_i) = 2\mathbb{E}[\|\mathbf{p} - \mathbf{z}\|_2] - \mathbb{E}[\|\mathbf{p} - \mathbf{p}'\|_2] - \mathbb{E}[\|\mathbf{z} - \mathbf{z}'\|_2]. \quad (6)$$

We then calibrate discrepancy measures into voting weights w_i by applying a Softmax function with temperature τ_2 over the negative energy distances:

$$w_i = \frac{\exp(-d_i^E/\tau_2)}{\sum_{i' \in \mathcal{I}_{top}} \exp(-d_{i'}^E/\tau_2)}. \quad (7)$$

Crucially, this density-based voting mechanism naturally confers robustness against label noise in the reference set. Noisy samples typically manifest as low-density outliers in the feature space, leading to large energy distances from the test distribution. Consequently, Eq. 7 assigns them negligible voting weights, rendering our distributional branch highly resilient to mislabeled data. Finally, we obtain the image-image distributional score vector $\mathbf{S}_B(x) \in \mathbb{R}^C$ via a weighted soft-voting over the macro-class labels of the N_{top} neighbors (Goldberger et al., 2004):

$$\mathbf{S}_B(x)_j = \sum_{i \in \mathcal{I}_{top}} w_i \cdot \mathbb{I}(\text{class}(i) = j). \quad (8)$$

3.5 Confidence Combination

We combine the scores from both branches using a fusion weight κ. We implement two distinct strategies:

DECENT-R (Robust): Our first strategy, named for its focus on creating a more robust classifier, is DECENT-R (Fusion Prediction). The final score $\mathbf{S}_{final}$ is a weighted sum of the two score vectors. Both the prediction $\hat{y}$ and confidence s_{conf} are derived from this fused vector

$$\mathbf{S}_{final} = (1 - \kappa) \cdot \mathbf{S}_A + \kappa \cdot \mathbf{S}_B, \quad (9)$$

$$\hat{y} = \arg\max_j (\mathbf{S}_{final})_j, \; s_{conf} = \max_j (\mathbf{S}_{final})_j. \quad (10)$$

DECENT-S (Sensitive): Our second strategy, named for its high sensitivity in detection, is DECENT-S (Decoupled Prediction). The prediction $\hat{y}$ is determined solely by the semantic branch $\mathbf{S}_A$. The confidence s_{conf} is a fusion of the specific scores from both branches at the predicted class $\hat{y}$.

$$\hat{y} = \arg\max_j (\mathbf{S}_A)_j, \quad (11)$$

$$s_{conf} = (1 - \kappa) \cdot \mathbf{S}_A(\hat{y}) + \kappa \cdot \mathbf{S}_B(\hat{y}). \quad (12)$$

Algorithm 1 Inference and Scoring Algorithm of DECENT

Input: Test image x, VLM $\mathcal{M}_A$, Image Encoder $\mathcal{M}_B$, Pre-computed Prototype DB $(\mathcal{Q}, \mathcal{Z})$, Fusion parameter κ.
Output: DECENT-R outputs $(\hat{y}_R, s_{conf,R})$, DECENT-S outputs $(\hat{y}_S, s_{conf,S})$.

// Step 1: Text-Image Semantic Alignment
Extract VLM image feature f_A^I = L2-Norm($\mathcal{E}_A^I(x)$).
Compute semantic score $\mathbf{S}_A(x)$ using Eqn. 1 & 2.
//Step 2: Neighborhood Distributional Discrepancy
Generate feature point cloud $\mathbf{P}_x$ (Eqn. 4).
Compute centroid $\mathbf{p}_x$ from $\mathbf{P}_x$.
Find N_{top} nearest neighbors $\mathcal{I}_{top}$ from $\mathcal{Q}$ (Eqn. 5).
Initialize $\mathbf{S}_B(x) \in \mathbb{R}^C$ to zeros.
for each neighbor $i \in \mathcal{I}_{top}$ **do**
  Compute Energy Distance d_i^E (Eqn. 6).
  Compute voting weight w_i (Eqn. 7).
  Let $j = \text{class}(i)$.
  $\mathbf{S}_B(x)_j = \mathbf{S}_B(x)_j + w_i$ {Accumulate (Eqn. 8)}
end for
// Step 3: Confidence Combination
// Strategy 1: DECENT-R (Robust)
$\mathbf{S}_{final} = (1 - \kappa) \cdot \mathbf{S}_A + \kappa \cdot \mathbf{S}_B$ (Eqn. 9).
$\hat{y}_R = \arg\max_j (\mathbf{S}_{final})_j$ (Eqn. 10).
$s_{conf,R} = \max_j (\mathbf{S}_{final})_j$ (Eqn. 10).
//Strategy 2: DECENT-S (Sensitive)
$\hat{y}_S = \arg\max_j (\mathbf{S}_A)_j$ (Eqn. 11).
$s_{conf,S} = (1 - \kappa) \cdot \mathbf{S}_A(\hat{y}_S) + \kappa \cdot \mathbf{S}_B(\hat{y}_S)$ (Eqn. 12).

4 Experiments

4.1 Experimental Settings

Datasets. To comprehensively evaluate the DECENT model, we conduct experiments on several widely used and representative benchmark datasets. These datasets cover diverse visual tasks and are detailed as follows: (1) FGVCAircraft (Maji et al., 2013). (2) StanfordCars (Cars) (Krause et al., 2013). (3) OxfordPets (Parkhi et al., 2012). (4) Food101 (Bossard et al., 2014). (5) SUN397 (Xiao et al., 2010). (6) EuroSAT (Helber et al., 2019). (7) UCF101 (Soomro et al., 2012). (8) Describable Textures Dataset (DTD) (Cimpoi et al., 2014). Collectively, these datasets span a diverse range

Table 1: Misclassification detection performance comparison on eight benchmark datasets (Part 1: DTD, Aircraft, and Pets). All methods are evaluated using CLIP ViT-B/16 as the base VLM (Radford et al., 2021). For the baselines, TrustVLM-C and -D use CLIP-I and DINOv2 as auxiliary encoders, respectively (Dong et al., 2025). The best results are shown in **boldface** . AURC scores are multiplied by 10^3, following previous work (Zhu et al., 2023). ↑ means higher is better , ↓ means lower is better. Results for the remaining datasets are shown in Table 2.

Datasets	DTD				Aircraft				Pets			
Method	AURC↓	AUROC↑	FPR95↓	Acc↑	AURC↓	AUROC↑	FPR95↓	Acc↑	AURC↓	AUROC↑	FPR95↓	Acc↑
MaxLogit	395.13	69.63	85.23	44.39	731.25	55.73	93.77	23.85	48.75	75.37	68.59	88.23
Energy	433.25	64.82	90.01	44.39	780.11	45.48	97.71	23.85	56.23	71.90	70.67	88.23
Entropy	319.02	79.03	77.05	44.39	575.71	73.12	83.96	23.85	21.91	89.31	53.35	88.23
MCM	333.43	78.16	78.45	44.39	583.28	72.58	81.65	23.85	44.95	77.03	64.90	88.23
DOCTOR	314.37	79.66	76.51	44.39	575.53	73.05	83.84	23.85	21.08	89.92	55.20	88.23
MSP	313.43	79.81	77.36	44.39	576.97	72.62	85.61	23.85	21.04	89.94	52.19	88.23
TrustVLM-C	302.18	82.52	67.27	44.39	563.77	75.20	81.51	23.85	20.93	89.89	51.73	88.23
TrustVLM-D	268.71	88.55	44.10	44.39	562.02	75.62	83.21	23.85	20.69	90.05	50.81	88.23
TrustVLM*-D	124.15	78.39	72.14	71.57	554.12	75.36	83.05	24.60	**20.05**	90.30	49.07	88.28
DECENT-R	**96.97**	83.91	65.04	**72.10**	**291.4**	81.10	72.22	**46.32**	21.01	87.43	64.95	**89.42**
DECENT-S	233.53	**94.52**	**27.10**	43.74	493.17	**88.01**	**52.98**	23.79	25.98	**93.69**	**33.50**	83.48

of visual tasks and represent the core benchmarks used by TrustVLM (Dong et al., 2025), thereby ensuring a fair and comprehensive comparison for our evaluation.

Baselines. We compare DECENT against nine training-free baseline methods to ensure a fair evaluation. Six classical methods: MSP (Hendrycks and Gimpel, 2016), MaxLogit (Hendrycks and Gimpel, 2016), Entropy (Hendrycks and Gimpel, 2016), Energy (Liu et al., 2020b), MCM (Ming et al., 2022), and DOCTOR (Granese et al., 2021). Three SOTA methods: TrustVLM-C (Dong et al., 2025), TrustVLM-D (Dong et al., 2025), and TrustVLM*-D (Dong et al., 2025).

Evaluation Metrics. We adopt four standard metrics to comprehensively evaluate model performance: (1) ACC (Accuracy, ↑): Measures classification accuracy. Higher is better. (2) AURC (Area Under the Risk-Coverage Curve, ↓) (Geifman and El-Yaniv, 2017): Measures the prediction risk (error rate) as confidence varies. Lower is better. (3) AUROC (Area Under the ROC Curve, ↑): Measures the detector's ability to distinguish between correct and incorrect predictions. Higher is better. (4) FPR95 (False Positive Rate at 95% TPR, ↓) (Hendrycks and Gimpel, 2016): Measures how many incorrect predictions are missed (false positives) when 95% of correct predictions (true positives) are recalled. Lower is better.

Implementation Details. Our method utilizes a dual-branch architecture. Our VLM ($\mathcal{M}_A$) utilizes CLIP ViT-B/16, consistent with the baseline papers. The auxiliary encoder ($\mathcal{M}_B$) is CLIP-I (i.e., CLIP's image-side vision encoder). All experiments were conducted on a single NVIDIA RTX 4090 GPU.

4.2 Main Results

We conduct a comprehensive comparison of our proposed DECENT framework against nine training-free baseline methods across eight benchmark datasets. All evaluations are based on the CLIP ViT-B/16 VLM. The detailed results, presented in Table 1 and Table 2, clearly demonstrate that both DECENT strategies establish a new state-of-the-art.

DECENT-R: SOTA Accuracy and Risk. Our DECENT-R strategy, which fuses both branches for prediction, functions as a superior and more robust classifier. As shown in Table 1 and Table 2, DECENT-R achieves SOTA Accuracy (Acc) across all eight datasets. Its average accuracy of 77.59% represents a substantial absolute improvement of 6.69% over the strongest baseline, TrustVLM*-D (70.90%). Furthermore, DECENT-R dominates the risk-aware metric, achieving SOTA AURC (lower is better) on seven of the eight datasets (all except Pets). Its average AURC of 82.68 is less than 60% of the score from TrustVLM*-D (139.22), confirming its ability to effectively mitigate prediction risk by correcting the VLM's original misclassifications.

DECENT-S: SOTA Detection Performance. For the task of pure misclassification detection (i.e., without altering the VLM's prediction), the DECENT-S strategy exhibits sensitivity and reliability. As evidenced in Table 1 and Table 2, DECENT-S achieves SOTA performance on all eight datasets for both AUROC (higher is better) and FPR95 (lower is better). It attains an average AUROC of 93.02% and an average FPR95 of 33.28%. This represents a massive leap in detection capability compared to the best SOTA baseline,

Table 2: Misclassification detection performance comparison (Part 2: Remaining datasets). Please refer to the caption of Table 1 for setup details and metric definitions.

Datasets	Cars				EuroSAT				UCF101			
Method	AURC↓	AUROC↑	FPR95↓	Acc↑	AURC↓	AUROC↑	FPR95↓	Acc↑	AURC↓	AUROC↑	FPR95↓	Acc↑
MaxLogit	259.81	61.94	89.58	65.61	522.56	59.42	87.79	42.10	258.47	61.85	91.41	65.21
Energy	304.90	54.85	93.89	65.61	568.57	53.44	88.30	42.10	299.84	55.38	94.83	65.21
Entropy	145.61	79.83	75.76	65.61	389.07	72.18	83.16	42.10	130.10	83.97	70.52	65.21
MCM	248.87	63.77	85.97	65.61	416.37	71.28	81.18	42.10	188.00	74.14	75.47	65.21
DOCTOR	138.00	81.50	73.66	65.61	368.77	74.43	81.69	42.10	123.41	85.67	65.81	65.21
MSP	136.27	81.95	72.25	65.61	355.66	76.39	80.65	42.10	122.44	85.98	64.89	65.21
TrustVLM-C	129.45	83.67	67.73	65.61	322.52	82.90	55.73	42.10	111.95	88.69	55.93	65.21
TrustVLM-D	137.54	82.05	70.62	65.61	303.79	85.48	53.52	42.10	107.13	90.21	50.68	65.21
TrustVLM*-D	137.97	81.73	71.25	65.83	72.96	74.87	73.50	83.56	65.26	86.23	62.93	77.11
DECENT-R	**82.11**	80.34	72.66	**77.30**	**12.72**	90.41	52.35	**92.12**	**45.21**	89.86	46.97	**80.81**
DECENT-S	100.68	**90.93**	**42.11**	66.01	193.41	**99.26**	**3.38**	45.25	77.59	**95.48**	**21.49**	67.64
Datasets	Food101				SUN397				Average			
Method	AURC↓	AUROC↑	FPR95↓	Acc↑	AURC↓	AUROC↑	FPR95↓	Acc↑	AURC↓	AUROC↑	FPR95↓	Acc↑
MaxLogit	67.81	75.81	78.13	83.66	278.17	62.77	90.46	62.57	320.24	65.31	85.62	59.45
Energy	76.05	72.33	84.69	83.66	304.50	58.34	93.47	62.57	352.93	59.57	89.2	59.45
Entropy	55.80	83.76	63.73	83.66	194.33	75.19	80.97	62.57	228.94	79.54	73.57	59.45
MCM	60.90	79.90	67.23	83.66	242.95	69.09	83.62	62.57	264.84	73.25	77.31	59.45
DOCTOR	54.99	84.39	61.11	83.66	184.41	77.35	77.37	62.57	222.57	80.74	71.90	59.45
MSP	54.80	84.51	59.73	83.66	182.39	77.90	76.42	62.57	220.38	81.14	71.14	59.45
TrustVLM-C	36.02	88.34	57.67	83.66	171.89	80.39	69.20	62.57	207.34	83.95	63.35	59.45
TrustVLM-D	36.18	88.52	55.96	83.66	156.16	83.80	58.44	62.57	199.02	85.53	58.40	59.45
TrustVLM*-D	35.01	88.49	57.47	84.10	104.25	82.31	72.82	72.18	139.22	82.21	67.78	70.90
DECENT-R	**22.47**	90.35	53.26	**87.39**	**88.13**	81.78	71.86	**75.22**	**82.51**	85.65	62.41	**77.59**
DECENT-S	26.27	**93.66**	**33.34**	83.63	112.8	**88.85**	**50.80**	65.34	157.98	**93.02**	**33.28**	59.86

TrustVLM-D (AUROC 85.53%, FPR95 58.40%). This superiority is particularly stark on datasets like EuroSAT, where DECENT-S achieves a near-perfect AUROC (99.26%) and a minimal FPR95 (3.38%).

4.3 Ablation Study

Table 3 validates the contribution of each core component, including text templates, the distributional branch, feature point clouds, fine-grained prototypes.

Ablation on DECENT-R. For robust classification, the semantic branch alone is limited: adding multi-template prototypes only improve Acc from 58.23% to 59.86% and reduce AURC from 253.48 to 238.38. In contrast, introducing the distributional branch substantially improves Acc to 77.59% and reduces AURC to 82.51, confirming its main contribution to classification robustness. Removing fine-grained prototypes or the feature point cloud degrades performance to 74.84%/96.86 and 77.35%/83.17 in Acc/AURC, respectively, showing that both components are beneficial.

Ablation on DECENT-S. For sensitive misclassification detection, the semantic branch achieves only 76.97% AUROC and 74.43% FPR95. Fusing the distributional score improves AUROC to 93.02% and reduces FPR95 to 33.28%, demonstrating that distributional evidence is the key source of detection sensitivity. Removing fine-grained prototypes or the feature point cloud consistently weakens detection, yielding 91.50%/39.49% and 92.68%/34.63% in AUROC/FPR95, respectively.

Table 3: Ablation studies on core components of DECENT-R and DECENT-S We report AURC↓, AUROC↑, FPR95↓, and Accuracy↑. All results are the averages of the eight datasets used in previous work. Red arrows indicate performance drop compared to the full model. Best results in **boldface**.

Method	Variant	AURC↓	AUROC↑	FPR95↓	Acc↑
DECENT-R	Model M_A only (No Template)	253.48 ↑ 170.97	76.34 ↓ 9.31	74.57 ↑ 12.16	58.23 ↓ 19.36
	Model M_A only	238.38 ↑ 155.87	76.97 ↓ 8.68	74.43 ↑ 12.02	59.86 ↓ 17.73
	w/o Feature Point Cloud (P_x)	83.17 ↑ 0.66	85.52 ↓ 0.13	62.52 ↑ 0.11	77.35 ↓ 0.24
	w/o Fine-grained Prototypes	96.86 ↑ 14.35	83.98 ↓ 1.67	66.98 ↑ 4.57	74.84 ↓ 2.75
	Full Model	**82.51**	**85.65**	**62.41**	**77.59**
DECENT-S	Model M_A only (No Template)	253.48 ↑ 95.5	76.34 ↓ 16.68	74.57 ↑ 41.29	58.23 ↓ 1.63
	Model M_A only	238.38 ↑ 80.4	76.97 ↓ 16.05	74.43 ↑ 41.15	59.86 = 0.00
	w/o Feature Point Cloud (P_x)	160.92 ↑ 2.94	92.68 ↓ 0.34	34.63 ↑ 1.35	59.86 = 0.00
	w/o Fine-grained Prototypes	164.65 ↑ 6.68	91.5 ↓ 1.52	39.49 ↑ 6.21	59.86 = 0.00
	Full Model	**157.98**	**93.02**	**33.28**	**59.86**

4.4 Sensitivity Analysis

We conducted a sensitivity analysis to evaluate the impact of key hyperparameters on DECENT-R (classification performance: Acc and AURC) and DECENT-S (detection performance: AUROC and FPR95). We analyzed the fusion parameter κ and the number of augmentations V, with results presented in Figure 2.

Fusion Parameter (κ). This parameter balances the semantic (S_A) and distributional (S_B) branches. As shown in Figure 2, both strategies demonstrate strong performance across the evaluated κ values, reaching a near-optimal balance at $\kappa = 0.5$ (DECENT-R AURC 82.68; DECENT-S AUROC

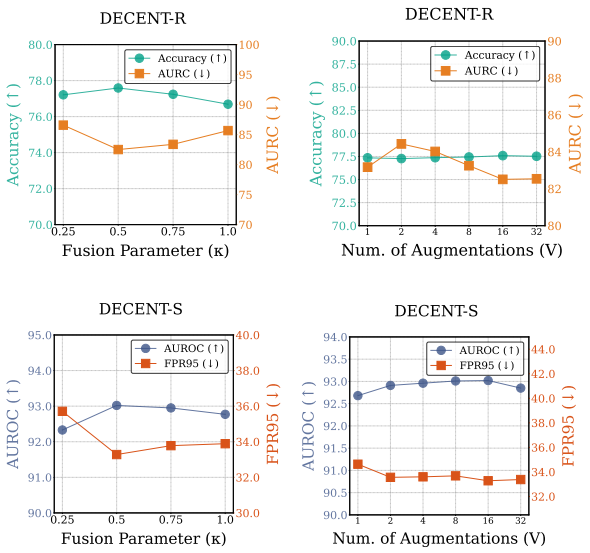


Figure 2: Sensitivity analysis for DECENT. All results are the averages of the eight datasets. We evaluate Accuracy (Acc ↑, green) and AURC (↓, orange) for DECENT-R, and AUROC (↑, blue) and FPR95 (↓, red) for DECENT-S, by varying the fusion parameter (κ) and number of augmentations (V).

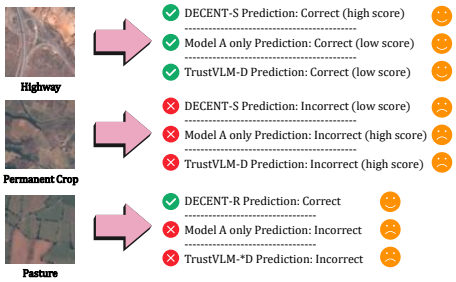
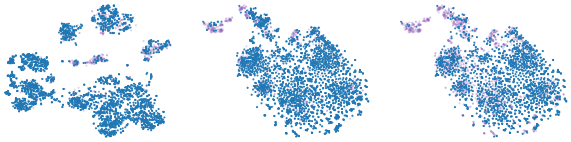


Figure 3: Qualitative comparison of DECENT against baselines on the EuroSAT dataset.



(a) TrustVLM-*D (b) Model M_A only (c) DECENT-R

Figure 4: Blue crosses represent incorrect classifications, and purple dots represent correct classifications.

93.02%, FPR95 33.28%). This confirms that an equal fusion ($\kappa = 0.5$) is a robust default choice.

Number of Augmentations (V). This parameter controls the feature cloud size P_x. We evaluated $V \in \{1, 2, 4, 8, 16, 32\}$. As shown in Figure 2, a single sample ($V = 1$) yields suboptimal performance, indicating that a multi-sample point cloud ($V > 1$) is critical. For both DECENT-R and DECENT-S, performance steadily improves as V increases, peaking at $V = 16$ (highest Acc and lowest AURC for DECENT-R; highest AUROC and lowest FPR95 for DECENT-S).

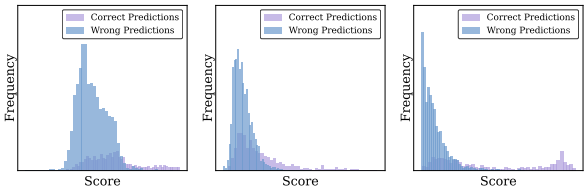
4.5 Case Study

Figure 3 presents representative EuroSAT cases comparing DECENT with Model $\mathcal{M}_A$ and TrustVLM-D/*D. For correctly classified samples, DECENT-S assigns higher confidence than the baselines; for misclassified samples, it produces low confidence and successfully flags the error. In harder cases where both baselines fail, DECENT-R can revise the prediction by incorporating distributional evidence, demonstrating its classification robustness.

4.6 Visualization

Feature Space Separability. Figure 4 shows that correct and incorrect samples are entangled for TrustVLM-*D and Model $\mathcal{M}_A$, while DECENT-R produces clearer feature-space separation, explaining its robust classification gains in Tables 1 and 2.

Confidence Score Distribution. Figure 5 shows severe overlap between correct and incorrect samples for the baselines. In contrast, DECENT-S produces more separable confidence distributions, consistent with its stronger AUROC performance.



(a) TrustVLM-D (b) Model M_A only (c) DECENT-S

Figure 5: The x-axis represents the score, the y-axis represents the frequency, blue represents misclassification, and purple represents correct classification.

5 Conclusion

This paper proposes DECENT, a framework for VLM misclassification detection that integrates semantic alignment with a distributional score. Extensive experiments validate that DECENT achieves new state-of-the-art results for both classification robustness (DECENT-R) and detection sensitivity (DECENT-S).

Limitations

DECENT has two main limitations. First, it incurs additional inference cost because it uses multi-view feature extraction, nearest-prototype search, and distribution distance computation. While this cost is manageable in our experiments, future work can further reduce it through approximate search, prototype pruning, or adaptive view selection. Second, DECENT requires a labeled reference set to build fine-grained class prototypes. This makes the method less suitable for scenarios where labeled reference data are unavailable or significantly mismatched with the test distribution. Extending DECENT to weakly supervised, unlabeled, and open-vocabulary settings is a promising direction.

References

Nouar AlDahoul, Myles Joshua Toledo Tan, Harishwar Reddy Kasireddy, and Yasir Zaki. 2024. Advancing content moderation: Evaluating large language models for detecting sensitive content across text, images, and videos. *arXiv preprint arXiv:2411.17123*.

Valentyn Boreiko, Matthias Hein, and Jan Hendrik Metzen. 2024. Identification of fine-grained systematic errors via controlled scene generation. *arXiv preprint arXiv:2404.07045*.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. 2014. Food-101–mining discriminative components with random forests. In *European conference on computer vision*, pages 446–461. Springer.

Jiefeng Chen, Frederick Liu, Besim Avci, Xi Wu, Yingyu Liang, and Somesh Jha. 2021. Detecting errors and estimating accuracy on unlabeled data with self-training ensembles. *Advances in Neural Information Processing Systems*, 34:14980–14992.

Qi Chen and Hu Ding. 2025. Dual energy-based model with open-world uncertainty estimation for out-of-distribution detection. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 25728–25737.

Jiacheng Cheng and Nuno Vasconcelos. 2022. Calibrating deep neural networks by pairwise constraints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13709–13718.

Sumit Chopra, Raia Hadsell, and Yann LeCun. 2005. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 539–546. IEEE.

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. 2014. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3606–3613.

Eduardo Dadalto, Marco Romanelli, Georg Pichler, and Pablo Piantanida. 2023. A data-driven measure of relative uncertainty for misclassification detection. *arXiv preprint arXiv:2306.01710*.

Hao Dong, Moru Liu, Jian Liang, Eleni Chatzi, and Olga Fink. 2025. To trust or not to trust your vision-language model's prediction. *arXiv preprint arXiv:2505.23745*.

Alhussein Fawzi, Seyed-Mohsen Moosavi-Dezfooli, Pascal Frossard, and Stefano Soatto. 2018. Empirical study of the topology and geometry of deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3762–3770.

Stanislav Fort, Jie Ren, and Balaji Lakshminarayanan. 2021. Exploring the limits of out-of-distribution detection. *Advances in neural information processing systems*, 34:7068–7081.

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

Yonatan Geifman and Ran El-Yaniv. 2017. Selective classification for deep neural networks. *Advances in neural information processing systems*, 30.

Jacob Goldberger, Geoffrey E Hinton, Sam Roweis, and Russ R Salakhutdinov. 2004. Neighbourhood components analysis. In *Advances in Neural Information Processing Systems*, volume 17. MIT Press.

Mingrong Gong, Chaoqi Chen, Qingqiang Sun, Yue Wang, and Hui Huang. 2025. Out-of-distribution detection with prototypical outlier proxy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 16835–16843.

Federica Granese, Marco Romanelli, Daniele Gorla, Catuscia Palamidessi, and Pablo Piantanida. 2021. Doctor: A simple method for detecting misclassification errors. *Advances in Neural Information Processing Systems*, 34:5669–5681.

Tobias Groot and Matias Valdenegro-Toro. 2024. Overconfidence is key: Verbalized uncertainty evaluation in large language and vision-language models. In *Proceedings of the 4th Workshop on Trustworthy Natural Language Processing (TrustNLP 2024)*, pages 145–171.

Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, and 1 others. 2024. Hallusionbench: an advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *Proceedings of*

the IEEE/CVF conference on computer vision and pattern recognition, pages 14375–14385.

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. 2019. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226.

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

Jongheon Jeong, Yang Zou, Taewan Kim, Dongqing Zhang, Avinash Ravichandran, and Onkar Dabeer. 2023. Winclip: Zero-/few-shot anomaly classification and segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19606–19616.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. 2021. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pages 4904–4916. PMLR.

Mohit Kakda, Mirudula Shri Muthukumaran, Uttapreksha Patel, and Lawrence Swaminathan Xavier Prince. 2026. Analyzing vlm-based approaches for anomaly classification and segmentation. *arXiv preprint arXiv:2601.13440*.

Ao Ke, Wenlong Chen, Chuanwen Feng, Yukun Cao, Xike Xie, S Kevin Zhou, and Lei Feng. 2024. Prototype-based optimal transport for out-of-distribution detection. *arXiv preprint arXiv:2410.07617*.

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

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 2013. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561.

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

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. 2018. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in neural information processing systems*, 31.

Shih-Po Lee, Zijia Lu, Zekun Zhang, Minh Hoai, and Ehsan Elhamifar. 2024. Error detection in egocentric procedural task videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18655–18666.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pages 12888–12900. PMLR.

Xiaofan Li, Zhizhong Zhang, Xin Tan, Chengwei Chen, Yanyun Qu, Yuan Xie, and Lizhuang Ma. 2024. Promptad: Learning prompts with only normal samples for few-shot anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16838–16848.

Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. 2017. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*.

Jeremiah Liu, Zi Lin, Shreyas Padhy, Dustin Tran, Tania Bedrax Weiss, and Balaji Lakshminarayanan. 2020a. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. *Advances in neural information processing systems*, 33:7498–7512.

Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. 2020b. Energy-based out-of-distribution detection. *Advances in neural information processing systems*, 33:21464–21475.

Stuart Lloyd. 1982. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137.

Haodong Lu, Dong Gong, Shuo Wang, Jason Xue, Lina Yao, and Kristen Moore. 2024. Learning with mixture of prototypes for out-of-distribution detection. *arXiv preprint arXiv:2402.02653*.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, 32.

Wesley J Maddox, Pavel Izmailov, Timur Garipov, Dmitry P Vetrov, and Andrew Gordon Wilson. 2019. A simple baseline for bayesian uncertainty in deep learning. *Advances in neural information processing systems*, 32.

Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. 2013. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*.

James B McQueen. 1967. Some methods of classification and analysis of multivariate observations. In *Proc. of 5th Berkeley Symposium on Math. Stat. and Prob.*, pages 281–297.

Yifei Ming, Ziyang Cai, Jiuxiang Gu, Yiyou Sun, Wei Li, and Yixuan Li. 2022. Delving into out-of-distribution detection with vision-language representations. *Advances in neural information processing systems*, 35:35087–35102.

Jishnu Mukhoti, Viveka Kulharia, Amartya Sanyal, Stuart Golodetz, Philip Torr, and Puneet Dokania. 2020. Calibrating deep neural networks using focal loss. *Advances in neural information processing systems*, 33:15288–15299.

Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. 2012. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3498–3505. IEEE.

Wujian Peng, Sicheng Xie, Zuyao You, Shiyi Lan, and Zuxuan Wu. 2024. Synthesize diagnose and optimize: Towards fine-grained vision-language understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13279–13288.

Shraman Pramanick, Rama Chellappa, and Subhashini Venugopalan. 2024. Spiqa: A dataset for multimodal question answering on scientific papers. *Advances in Neural Information Processing Systems*, 37:118807–118833.

Zhen Qu, Xian Tao, Xinyi Gong, Shichen Qu, Qiyu Chen, Zhengtao Zhang, Xingang Wang, and Guiguang Ding. 2025. Bayesian prompt flow learning for zero-shot anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 30398–30408.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, and 1 others. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30.

Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. 2012. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*.

Gábor J Székely and Maria L Rizzo. 2013. Energy statistics: A class of statistics based on distances. *Journal of statistical planning and inference*, 143(8):1249–1272.

Francesco Taioli, Edoardo Zorzi, Gianni Franchi, Alberto Castellini, Alessandro Farinelli, Marco Cristani, and Yiming Wang. 2025. Collaborative instance object navigation: Leveraging uncertainty-awareness to minimize human-agent dialogues. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18781–18792.

Jordan Vice, Naveed Akhtar, Yansong Gao, Richard Hartley, and Ajmal Mian. 2025. On the reliability of vision-language models under adversarial frequency-domain perturbations. *arXiv preprint arXiv:2507.22398*.

Guotai Wang, Wenqi Li, Michael Aertsen, Jan Deprest, Sébastien Ourselin, and Tom Vercauteren. 2019. Aleatoric uncertainty estimation with test-time augmentation for medical image segmentation with convolutional neural networks. *Neurocomputing*, 338:34–45.

Xueqing Wu, Yuheng Ding, Bingxuan Li, Pan Lu, Da Yin, Kai-Wei Chang, and Nanyun Peng. 2025. Visco: Benchmarking fine-grained critique and correction towards self-improvement in visual reasoning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 9527–9537.

Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. 2010. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 3485–3492. IEEE.

Rui Xiao, Sanghwan Kim, Mariana-Iuliana Georgescu, Zeynep Akata, and Stephan Alaniz. 2025. Flair: Vlm with fine-grained language-informed image representations. In *Proceedings of the computer vision and pattern recognition conference*, pages 24884–24894.

Shaoyuan Xie, Lingdong Kong, Yuhao Dong, Chonghao Sima, Wenwei Zhang, Qi Alfred Chen, Ziwei Liu, and Liang Pan. 2025. Are vlms ready for autonomous driving? an empirical study from the reliability, data and metric perspectives. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6585–6597.

Lintong Zhang, Kang Yin, and Seong-Whan Lee. 2025a. Towards fine-grained interpretability: Counterfactual explanations for misclassification with saliency partition. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 30053–30062.

Ruifei Zhang, Junlin Xie, Wei Zhang, Weikai Chen, Xiao Tan, Xiang Wan, and Guanbin Li. 2025b. Adadrive: Self-adaptive slow-fast system for language-grounded autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5112–5121.

Ruiyang Zhang, Hu Zhang, and Zhedong Zheng. 2024. Vl-uncertainty: Detecting hallucination in large vision-language model via uncertainty estimation. *arXiv preprint arXiv:2411.11919*.

Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason Corso, and Jianfeng Gao. 2020. Unified vision-language pre-training for image captioning and vqa. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 13041–13049.

Qihang Zhou, Guansong Pang, Yu Tian, Shibo He, and Jiming Chen. 2023. Anomalyclip: Object-agnostic prompt learning for zero-shot anomaly detection. *arXiv preprint arXiv:2310.18961*.

Fei Zhu, Zhen Cheng, Xu-Yao Zhang, and Cheng-Lin Liu. 2023. Openmix: Exploring outlier samples for

Table A1: Statistics of the Reference Dataset ($\mathcal{D}_{ref}$, derived from official Training splits) used for prototype construction and the Test Dataset used for evaluation.

Dataset	Reference ($\mathcal{D}_{ref}$ / Train)	Test
Aircraft	3,334	3,333
Cars	6,509	8,041
DTD	2,820	1,692
EuroSAT	13,500	8,100
Food101	50,500	30,300
Pets	2,944	3,669
SUN397	15,880	19,850
UCF101	7,639	3,783

misclassification detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12074–12083.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, and 1 others. 2023. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pages 2165–2183. PMLR.

Zhipeng Zou, Sheng Wan, Guangyu Li, Bo Han, Tongliang Liu, Lin Zhao, and Chen Gong. 2025. Provable discriminative hyperspherical embedding for out-of-distribution detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 13483–13491.

A Implementation Details

This section provides a comprehensive and exhaustive account of the technical specifications required to fully reproduce the findings presented in our main paper. We meticulously detail our model choices, the complete set of hyperparameters, and the precise methodologies used for constructing both the semantic and distributional features that form the core of our DECENT framework.

A.1 Reference Dataset Statistics.

The prototype database $(\mathcal{Q}, \mathcal{Z})$ is explicitly constructed using the official *training split* of each benchmark as the reference dataset $\mathcal{D}_{ref}$. Consequently, the size of the prototype source (N_{ref}) corresponds exactly to the training set size. The specific sample counts for the reference dataset (used for construction) and the test dataset (used for evaluation) across all eight benchmarks are detailed in Table A1.

A.2 Models and Experimental Setup

Our DECENT framework operates using a dual-branch architecture, with specific models assigned to each branch. Model $\mathcal{M}_A$ (Text-Image Branch): For the primary branch, which computes the Text-Image Semantic Alignment score (S_A), we utilize the official pre-trained CLIP ViT-B/16 model. This model's joint embedding space provides the foundational semantic similarity score. Model $\mathcal{M}_B$ (Image-Image Branch): For the auxiliary branch, which computes the Neighborhood Distributional Discrepancy score (S_B), we also employ the official pre-trained CLIP ViT-B/16 model. This model serves as the fixed feature extractor $\mathcal{E}_B^I(\cdot)$ used to generate the image feature point clouds (P_x) and the fine-grained prototype database $(\mathcal{Q}, \mathcal{Z})$. Reference Dataset ($\mathcal{D}_{ref}$): The prototype database $(\mathcal{Q}, \mathcal{Z})$ is constructed using a reference dataset, which we define as the official training split for each respective benchmark (refer to Table A1 for detailed statistics). For example, when evaluating on the Food101 test set, $\mathcal{D}_{ref}$ consists of the 50,500 images from the Food101 official training set. All features for $\mathcal{D}_{ref}$ are pre-computed and extracted using Model $\mathcal{M}_B$. Crucially, our entire DECENT framework operates in a training-free manner. Both CLIP models are used "as-is" from their original pre-training. No gradient updates, fine-tuning, or backpropagation are performed on either the reference datasets ($\mathcal{D}_{ref}$) or the test datasets. All input images are resized and center-cropped to 224×224 resolution and normalized using the standard CLIP normalization parameters (mean: *[0.481, 0.458, 0.408]*, std: *[0.269, 0.261, 0.276]*).

A.3 Hyperparameter Configuration

The behavior of the DECENT framework is governed by a set of key hyperparameters. We provide the exact values used to generate the main results presented in our paper.

Fusion Parameter (κ): This critical parameter balances the contribution of the semantic branch (S_A) and the distributional branch (S_B). Based on the sensitivity analysis presented in our main paper, this value was fixed at $\kappa = 0.5$ for all experiments, representing an equal weighting of the two scores.

τ_1: This temperature parameter regulates the sharpness of the semantic score distribution S_A. Although our initial empirical sensitivity analysis suggested that dataset-specific tuning could potentially refine calibration, we found that a consistent value of $\tau_1 = 0.05$ provided a robust and near-optimal balance across all eight benchmarks (e.g., SUN397, DTD, Food101). Consequently, to maintain framework simplicity and prevent overfitting

to specific datasets, we fixed $\tau_1 = 0.05$ for all primary experiments.

τ_2: This parameter calibrates the voting weights w_i derived from the negative Energy Distance during the fine-grained voting process. A consistent value was found to be effective across all datasets. We set $\tau_2 = 0.05$ for all experiments.

Number of Clusters (K_{avg}): This hyperparameter defines the granularity of our prototype database $(\mathcal{Q}, \mathcal{Z})$. Instead of applying a uniform K value, we determined the optimal setting by analyzing the average number of reference samples assigned to each cluster. Specifically, we evaluated the performance across four distinct ranges of cluster density: (1) no clustering (utilizing the full class distribution); (2) 20–50 samples per cluster; (3) 10–20 samples per cluster; and (4) < 10 samples per cluster. It is observed that for datasets with limited class sizes (e.g., fewer than 50 samples), the setting with 20–50 samples is effectively equivalent to the no-clustering scenario. Through empirical validation, we found that maintaining an average of 10–20 samples per cluster consistently yielded superior performance, striking an optimal balance between capturing intra-class diversity and ensuring statistical stability. Consequently, we adopted dataset-specific K values to align with this optimal range. The empirically selected K value for each of the eight benchmarks is detailed in Figure A1 and A2. The selected optimal K values for DECENT-R and DECENT-S are as follows: SUN397: $K = 3$, Food101: $K = 32$, DTD: $K = 4$, UCF101: $K = 4$, Pets: $K = 8$, Aircraft: $K = 3$, Cars: $K = 3$, Eurosat: $K = 64$.

Number of Neighbors (N_{top}): This parameter controls the scope of the local neighborhood search during the voting process. For all main results, we set $N_{top} = 16$. This means that for a given test image's feature centroid p_x, we identify the 16 nearest sub-prototypes from the entire database $\mathcal{Q}$ based on Euclidean distance. These 16 neighbors are then used in the weighted soft-voting. A detailed sensitivity analysis for N_{top} (e.g., $N_{top} \in \{8, 16, 32\}$) is also provided in table A2 and A3. For DECENT-R and DECENT-S, $N_{top} = 16$ is optimal on average, although it may not be the best setting for every specific metric or dataset.

Number of Views (V):This parameter determines the size of the feature point cloud P_x generated for each test image. We conducted a sensitivity analysis on V, with the results detailed in Table A4 and A5. While the $V = 1$ setting (using only a single center-crop view without augmentation) achieves competitive or superior results on some individual datasets, the average performance across all eight benchmarks indicates that using augmented views provides a more robust and superior result overall. Specifically, the $V = 16$ setting consistently delivers the best, or near-best, average performance for both DECENT-R and DECENT-S.

A.4 Semantic Prototype Construction (Text Templates)

To construct robust text-semantic prototypes μ_j^T and mitigate potential prompt sensitivity, we employed a multi-template enhancement strategy. For each class name (e.g., "highway" or "permanent crop"), we generated 11 distinct text prompts by inserting the class name into the templates listed below. The CLIP text encoder was used to extract features for all 11 prompts, which were then L2-normalized, averaged, and L2-normalized again to produce the final, stable class-semantic center μ_j^T.

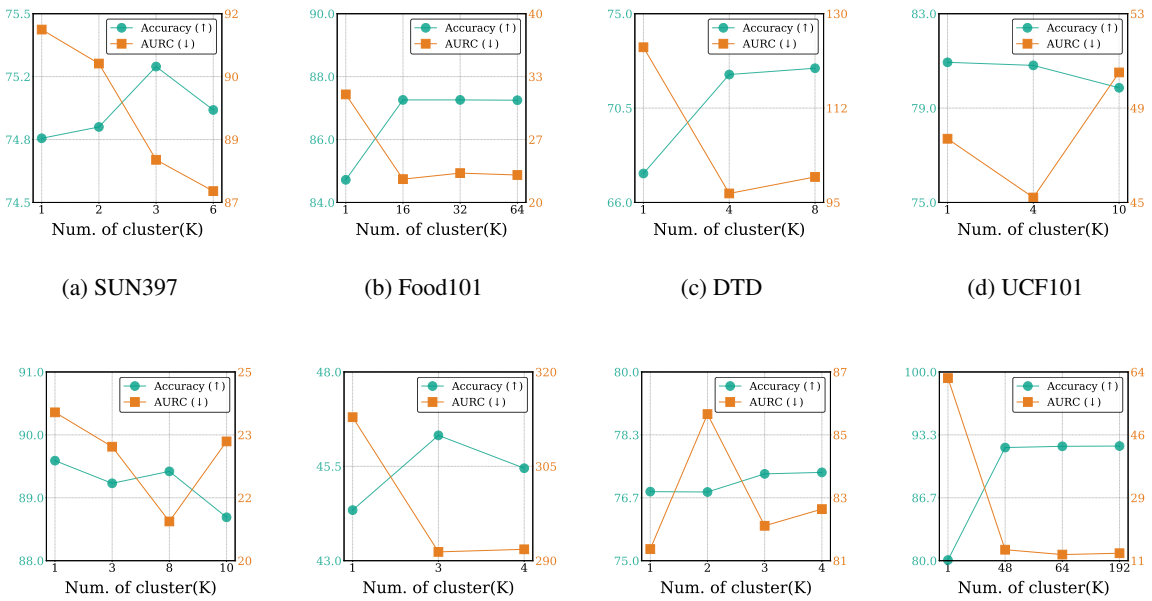
The 11 templates used are:

```
'a photo of {class name}.'
'a photo of the {class name}.'
'an image of {class name}.'
'an image of the {class name}.'
'a picture of {class name}.'
'a picture of the {class name}.'
'a photograph of {class name}.'
'a photograph of the {class name}.'
'This is a photo of {class name}.'
'A photo showing {class name}.'
'An image depicting {class name}.'
```

B Extended Model Performance

To demonstrate the universality and scalability of our framework, we extend our evaluation beyond the standard settings used in the main paper. While our primary experiments utilize ViT-B/16 as the auxiliary visual encoder $\mathcal{M}_B$, we investigate how the DECENT framework adapts to different backbone capacities by testing ResNet-50 and ViT-L/14 (detailed in Table A6 and Table A7). Furthermore, we thoroughly evaluate the impact of different vision foundation models (Table A8) and distance measurement algorithms (Table A9). Through these extensive ablations, we observe the following consistent properties:

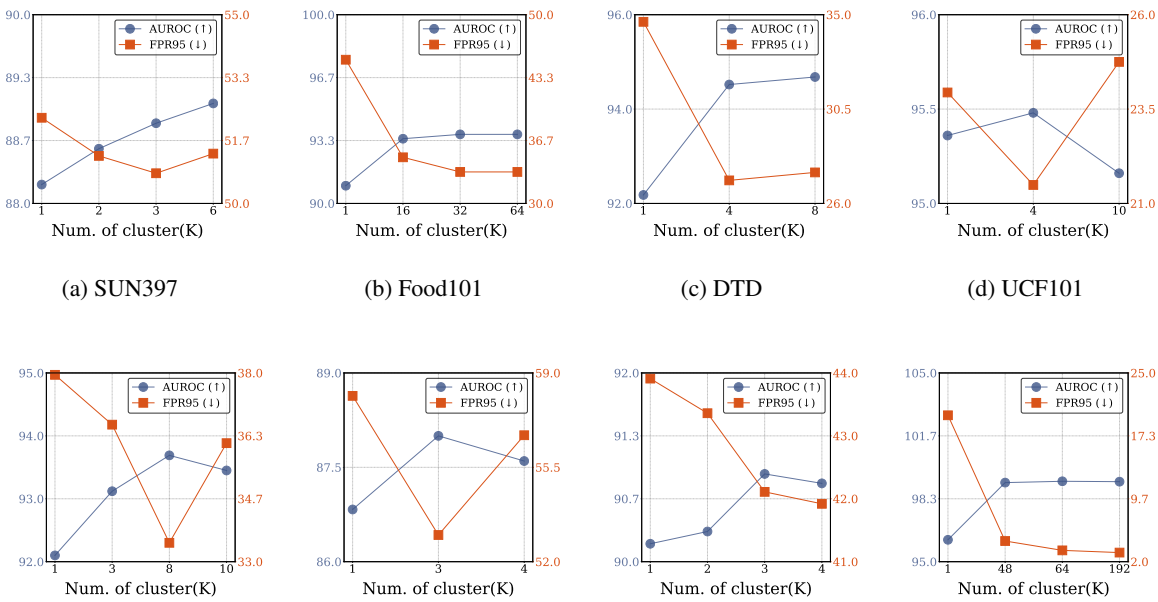
Scalability: There is a positive correlation between the visual encoder's capacity and the final



(a) SUN397 (b) Food101 (c) DTD (d) UCF101

(e) Pets (f) Aircraft (g) Cars (h) Eurosat

Figure A1: Sensitivity analysis for the number of clusters (K) on the DECENT-R method. The plots show the changes in Accuracy (ACC) and AURC as K varies across the eight benchmark datasets.



(a) SUN397 (b) Food101 (c) DTD (d) UCF101

(e) Pets (f) Aircraft (g) Cars (h) Eurosat

Figure A2: Sensitivity analysis for the number of clusters (K) on the DECENT-S method. The plots show the changes in AUROC and FPR95 as K varies across the eight benchmark datasets.

Table A2: Sensitivity analysis of the number of neighbors (N_{top}) on DECENT-R. Best results are in **bold**.

Datasets	DTD				Aircraft				Pets			
DECENT-R	AURC↓	AUROC↑	FPR95↓	Acc↑	AURC↓	AUROC↑	FPR95↓	Acc↑	AURC↓	AUROC↑	FPR95↓	Acc↑
$N_{top} = 8$	97.24	82.17	67.02	71.87	295.84	80.60	**71.80**	45.42	21.42	**88.29**	65.04	88.77
$N_{top} = 16$	**96.97**	**83.91**	**65.04**	72.10	291.40	**81.10**	72.22	**46.32**	**21.01**	87.43	**64.95**	89.42
$N_{top} = 32$	99.42	83.10	65.96	**72.22**	**291.31**	80.17	72.54	46.30	22.42	85.90	68.44	**89.72**
Datasets	Cars				EuroSAT				UCF101			
DECENT-R	AURC↓	AUROC↑	FPR95↓	Acc↑	AURC↓	AUROC↑	FPR95↓	Acc↑	AURC↓	AUROC↑	FPR95↓	Acc↑
$N_{top} = 8$	**78.22**	**81.81**	**70.75**	77.08	13.24	90.19	**52.12**	92.02	**44.95**	89.46	49.04	80.70
$N_{top} = 16$	82.11	80.34	72.66	77.30	**12.72**	90.41	52.35	**92.12**	45.21	**89.86**	46.97	**80.81**
$N_{top} = 32$	86.11	78.96	76.16	**77.42**	13.34	**92.15**	52.40	91.78	45.77	89.67	**46.34**	80.04
Datasets	Food101				SUN397				Average			
DECENT-R	AURC↓	AUROC↑	FPR95↓	Acc↑	AURC↓	AUROC↑	FPR95↓	Acc↑	AURC↓	AUROC↑	FPR95↓	Acc↑
$N_{top} = 8$	23.16	90.32	**51.78**	87.10	89.47	81.52	71.95	**75.37**	82.94	85.54	62.44	77.29
$N_{top} = 16$	**22.47**	**90.35**	53.26	**87.39**	**88.13**	**81.78**	**71.86**	75.22	**82.51**	**85.65**	**62.41**	**77.59**
$N_{top} = 32$	23.67	89.74	56.30	87.26	89.25	81.55	72.07	75.15	83.91	85.15	63.78	77.49

Table A3: Sensitivity analysis of the number of neighbors (N_{top}) on DECENT-S. Best results are in **bold**.

Datasets	DTD				Aircraft				Pets			
DECENT-S	AURC↓	AUROC↑	FPR95↓	Acc↑	AURC↓	AUROC↑	FPR95↓	Acc↑	AURC↓	AUROC↑	FPR95↓	Acc↑
$N_{top} = 8$	233.94	93.78	27.84	43.74	493.66	87.55	60.20	23.79	**25.15**	93.27	**31.68**	83.48
$N_{top} = 16$	**233.53**	**94.52**	**27.10**	43.74	**493.17**	**88.01**	**52.98**	23.79	25.98	**93.69**	33.50	83.48
$N_{top} = 32$	234.33	94.45	27.73	43.74	494.21	87.96	52.91	23.79	27.21	93.12	35.15	83.48
Datasets	Cars				EuroSAT				UCF101			
DECENT-S	AURC↓	AUROC↑	FPR95↓	Acc↑	AURC↓	AUROC↑	FPR95↓	Acc↑	AURC↓	AUROC↑	FPR95↓	Acc↑
$N_{top} = 8$	102.11	**91.46**	**41.02**	66.01	193.92	98.95	**3.16**	45.25	77.66	95.32	22.03	67.64
$N_{top} = 16$	**100.68**	90.93	42.11	66.01	**193.41**	**99.26**	3.38	45.25	**77.59**	**95.48**	21.49	67.64
$N_{top} = 32$	103.59	90.32	42.82	66.01	193.67	99.18	3.74	45.25	77.90	95.45	**21.00**	67.64
Datasets	Food101				SUN397				Average			
DECENT-S	AURC↓	AUROC↑	FPR95↓	Acc↑	AURC↓	AUROC↑	FPR95↓	Acc↑	AURC↓	AUROC↑	FPR95↓	Acc↑
$N_{top} = 8$	26.60	93.49	34.43	83.63	113.54	88.68	**49.56**	65.34	158.32	92.81	33.74	59.86
$N_{top} = 16$	**26.27**	**93.66**	**33.34**	83.63	**112.80**	**88.85**	50.80	65.34	**157.98**	**93.02**	**33.28**	59.86
$N_{top} = 32$	27.07	93.22	36.46	83.63	113.96	88.56	52.18	65.34	158.99	92.78	33.99	59.86

Table A4: Sensitivity analysis of the number of views (V) on DECENT-R. Best results are in **bold**.

Datasets	DTD				Aircraft				Pets			
DECENT-R	AURC↓	AUROC↑	FPR95↓	Acc↑	AURC↓	AUROC↑	FPR95↓	Acc↑	AURC↓	AUROC↑	FPR95↓	Acc↑
$V=1$	**96.71**	83.83	64.39	**72.28**	291.42	80.50	**69.43**	46.11	25.11	86.72	65.98	88.14
$V=2$	100.26	83.80	66.32	71.57	293.60	79.94	74.99	46.14	23.75	86.13	70.62	88.96
$V=4$	98.33	83.94	**62.03**	71.99	294.89	79.54	74.61	46.23	22.76	86.93	67.99	89.02
$V=8$	98.39	83.75	66.60	72.04	293.61	80.40	73.78	45.87	22.02	86.85	68.45	89.29
$V=16$	96.97	83.91	65.04	72.10	**291.40**	**81.10**	72.22	**46.32**	**21.01**	**87.43**	64.95	89.42
$V=32$	98.35	**84.03**	63.06	71.87	290.63	80.08	73.95	**46.32**	21.48	86.93	**64.16**	**89.51**
Datasets	Cars				EuroSAT				UCF101			
DECENT-R	AURC↓	AUROC↑	FPR95↓	Acc↑	AURC↓	AUROC↑	FPR95↓	Acc↑	AURC↓	AUROC↑	FPR95↓	Acc↑
$V=1$	83.43	**80.75**	73.54	76.83	**12.09**	**90.60**	52.40	**92.30**	45.60	89.93	46.87	80.60
$V=2$	87.54	80.16	73.79	76.54	12.64	90.34	52.44	92.16	45.97	89.97	45.87	80.47
$V=4$	86.65	80.67	73.02	76.40	12.88	90.58	**49.32**	91.93	45.08	89.67	48.40	80.94
$V=8$	83.20	80.56	**72.13**	77.02	12.45	90.56	51.33	92.11	45.13	90.05	**46.92**	80.68
$V=16$	82.11	80.34	72.66	**77.30**	12.72	90.41	52.35	92.12	45.21	89.86	46.97	80.81
$V=32$	**81.45**	80.45	72.49	76.98	12.55	90.43	53.25	91.87	**44.56**	**89.58**	49.72	**81.18**
Datasets	Food101				SUN397				Average			
DECENT-R	AURC↓	AUROC↑	FPR95↓	Acc↑	AURC↓	AUROC↑	FPR95↓	Acc↑	AURC↓	AUROC↑	FPR95↓	Acc↑
$V=1$	23.05	90.11	54.98	87.27	**87.94**	81.75	72.61	75.28	83.17	85.52	62.52	77.35
$V=2$	23.38	90.17	53.92	87.11	88.31	81.72	72.22	75.22	84.43	85.27	63.77	77.27
$V=4$	23.19	90.21	54.19	87.16	88.36	81.59	72.14	75.28	84.02	85.39	62.71	77.37
$V=8$	23.20	90.17	54.20	87.18	88.02	81.62	72.18	**75.34**	83.25	85.50	63.20	77.44
$V=16$	**22.47**	**90.35**	**53.26**	87.39	88.13	**81.78**	71.86	75.22	**82.51**	**85.65**	**62.41**	**77.59**
$V=32$	23.12	90.07	55.25	87.19	88.21	81.68	**71.59**	75.25	82.54	85.41	62.93	77.52

Table A5: Sensitivity analysis of the number of views (V) on DECENT-S. Best results are in **bold**.

Datasets	DTD				Aircraft				Pets			
DECENT-S	AURC↓	AUROC↑	FPR95↓	Acc↑	AURC↓	AUROC↑	FPR95↓	Acc↑	AURC↓	AUROC↑	FPR95↓	Acc↑
$V=1$	259.06	91.59	38.87	43.74	**492.34**	**88.19**	52.60	23.79	27.08	93.15	33.83	83.48
$V=2$	233.42	94.41	28.47	43.74	493.08	87.96	52.68	23.79	26.56	93.39	**32.54**	83.48
$V=4$	233.80	94.50	29.10	43.74	493.55	87.97	52.44	23.79	26.26	93.54	33.33	83.48
$V=8$	233.92	94.48	29.73	43.74	493.08	88.03	52.28	23.79	26.01	93.65	34.52	83.48
$V=16$	233.53	**94.52**	**27.10**	43.74	493.17	88.01	52.98	23.79	25.98	93.69	33.50	83.48
$V=32$	**233.49**	**94.52**	29.62	43.74	492.64	88.12	**51.42**	23.79	**25.94**	**93.71**	33.50	83.48
Datasets	Cars				EuroSAT				UCF101			
DECENT-S	AURC↓	AUROC↑	FPR95↓	Acc↑	AURC↓	AUROC↑	FPR95↓	Acc↑	AURC↓	AUROC↑	FPR95↓	Acc↑
$V=1$	**99.30**	**91.31**	**40.76**	66.01	**193.02**	**99.30**	3.22	45.25	**77.34**	95.53	21.24	67.64
$V=2$	102.51	90.60	42.85	66.01	193.32	99.25	3.52	45.25	77.63	95.45	21.57	67.64
$V=4$	101.45	90.73	43.29	66.01	193.44	99.23	3.34	45.25	77.41	95.51	20.83	67.64
$V=8$	101.12	90.89	42.89	66.01	193.23	99.26	**3.20**	45.25	77.45	95.51	**20.26**	67.64
$V=16$	100.68	90.93	42.11	66.01	193.41	99.26	3.38	45.25	77.59	95.48	21.49	67.64
$V=32$	100.40	91.02	41.94	66.01	193.28	99.26	3.40	45.25	77.49	**94.56**	21.16	67.64
Datasets	Food101				SUN397				Average			
DECENT-S	AURC↓	AUROC↑	FPR95↓	Acc↑	AURC↓	AUROC↑	FPR95↓	Acc↑	AURC↓	AUROC↑	FPR95↓	Acc↑
$V=1$	26.55	93.47	34.83	83.63	**112.67**	**88.88**	51.67	65.34	160.92	92.68	34.63	59.86
$V=2$	26.77	93.37	35.13	83.63	112.89	88.83	51.69	65.34	158.40	92.91	33.56	59.86
$V=4$	26.66	93.41	35.23	83.63	113.03	88.78	51.26	65.34	158.21	92.96	33.60	59.86
$V=8$	26.71	93.40	35.44	83.63	112.75	88.85	51.12	65.34	158.03	93.01	33.68	59.86
$V=16$	**26.27**	**93.66**	**33.34**	83.63	112.80	88.85	50.80	65.34	157.98	**93.02**	**33.28**	59.86
$V=32$	26.64	92.75	35.34	83.63	112.83	88.85	**50.62**	65.34	**157.84**	92.85	33.38	59.86

Table A6: Sensitivity analysis of different visual backbones on DECENT-R. Best results are in **bold**.

Datasets	DTD				Aircraft				Pets			
DECENT-R	AURC↓	AUROC↑	FPR95↓	Acc↑	AURC↓	AUROC↑	FPR95↓	Acc↑	AURC↓	AUROC↑	FPR95↓	Acc↑
RN50	123.93	82.99	68.56	67.27	487.28	72.01	91.23	32.25	76.66	90.46	56.26	70.84
ViT-B/16	96.97	83.91	65.04	72.10	**291.40**	**81.10**	**72.22**	46.32	21.01	87.43	64.95	**89.42**
ViT-L/14	**79.24**	**83.86**	**64.63**	**75.77**	323.07	63.00	95.46	**54.37**	**17.91**	**91.50**	**55.06**	88.42
Datasets	Cars				EuroSAT				UCF101			
DECENT-R	AURC↓	AUROC↑	FPR95↓	Acc↑	AURC↓	AUROC↑	FPR95↓	Acc↑	AURC↓	AUROC↑	FPR95↓	Acc↑
RN50	168.88	**84.06**	**50.95**	61.19	65.75	71.86	76.55	87.63	77.36	86.98	61.99	73.65
ViT-B/16	82.11	80.34	72.66	77.30	**12.72**	**90.41**	**52.35**	92.12	45.21	89.86	**46.97**	80.81
ViT-L/14	**55.32**	80.57	73.50	**83.20**	39.23	59.87	88.26	**94.95**	**27.98**	**89.97**	51.26	**86.39**
Datasets	Food101				SUN397				Average			
DECENT-R	AURC↓	AUROC↑	FPR95↓	Acc↑	AURC↓	AUROC↑	FPR95↓	Acc↑	AURC↓	AUROC↑	FPR95↓	Acc↑
RN50	56.00	87.22	61.54	78.55	96.41	81.41	72.38	74.06	143.66	82.12	67.43	68.23
ViT-B/16	22.47	90.35	53.26	87.39	88.13	81.78	71.86	75.22	82.51	**85.65**	**62.41**	77.59
ViT-L/14	**10.49**	**92.40**	**44.03**	**91.93**	**73.44**	**82.08**	**70.95**	**78.24**	**78.34**	80.41	67.89	**81.66**

Table A7: Sensitivity analysis of different visual backbones on DECENT-S. Best results are in **bold**.

Datasets	DTD				Aircraft				Pets			
DECENT-S	AURC↓	AUROC↑	FPR95↓	Acc↑	AURC↓	AUROC↑	FPR95↓	Acc↑	AURC↓	AUROC↑	FPR95↓	Acc↑
RN50	241.99	92.99	32.88	43.74	540.96	79.22	79.53	23.79	37.14	88.67	48.02	83.48
ViT-B/16	233.53	94.52	27.10	43.74	**493.17**	**88.01**	**52.98**	23.79	**25.98**	93.69	33.50	83.48
ViT-L/14	**226.88**	**95.63**	**21.95**	43.74	505.22	84.84	74.57	23.79	26.41	**93.97**	**27.56**	83.48
Datasets	Cars				EuroSAT				UCF101			
DECENT-S	AURC↓	AUROC↑	FPR95↓	Acc↑	AURC↓	AUROC↑	FPR95↓	Acc↑	AURC↓	AUROC↑	FPR95↓	Acc↑
RN50	138.68	82.43	72.48	66.01	216.85	95.63	24.98	45.25	81.95	93.84	29.76	67.64
ViT-B/16	**100.68**	**90.93**	**42.11**	66.01	**193.41**	**99.26**	3.38	45.25	77.59	**95.48**	21.49	67.64
ViT-L/14	108.78	90.51	46.07	66.01	206.25	97.89	**2.60**	45.25	**70.04**	**97.44**	**10.87**	67.64
Datasets	Food101				SUN397				Average			
DECENT-S	AURC↓	AUROC↑	FPR95↓	Acc↑	AURC↓	AUROC↑	FPR95↓	Acc↑	AURC↓	AUROC↑	FPR95↓	Acc↑
RN50	35.99	88.74	52.27	83.63	128.99	85.60	62.15	65.34	177.81	88.39	50.26	59.86
ViT-B/16	26.27	93.66	33.34	83.63	**112.80**	88.85	**50.80**	65.34	**157.98**	93.02	33.28	59.86
ViT-L/14	**19.41**	**97.26**	**13.58**	83.63	113.98	**89.18**	51.42	65.34	159.62	**93.34**	**31.08**	59.86

Table A8: Performance of different vision foundation models on DECENT-R and DECENT-S.

Models	DECENT-R				DECENT-S			
	AURC↓	AUROC↑	FPR95↓	Acc↑	AURC↓	AUROC↑	FPR95↓	Acc↑
SigLip	52.74	88.14	57.88	83.01	146.96	95.86	19.98	59.86
DinoV2	672.96	62.35	86.80	26.09	244.01	76.89	68.07	59.86
Clip	82.51	85.65	62.41	77.59	157.98	93.02	33.28	59.86

Table A9: Performance after replacing energy distance with other algorithms on DECENT-R and DECENT-S.

Methods	DECENT-R				DECENT-S			
	AURC↓	AUROC↑	FPR95↓	Acc↑	AURC↓	AUROC↑	FPR95↓	Acc↑
KNN	103.61	84.05	66.39	74.56	166.29	90.94	41.27	59.86
TTA	87.23	84.56	64.50	77.17	160.21	92.44	34.81	59.86
Energy Distance	82.51	85.65	62.41	77.59	157.98	93.02	33.28	59.86

performance. The larger ViT-L/14 consistently outperforms the baseline ViT-B/16, while ResNet-50 yields lower but still effective results.

Impact of Vision-Language Alignment: To evaluate the universality of DECENT, we replaced the default CLIP encoder with other state-of-the-art vision foundation models, namely SigLIP and DINOv2. As shown in Table A8, SigLIP achieves performance levels comparable to CLIP. This success is attributed to its robust vision-language alignment, which ensures a rich, semantically structured embedding space where text and image features are naturally co-located—a prerequisite for our semantic branch. Conversely, DINOv2 exhibits significantly sub-optimal performance. While DINOv2 provides powerful discriminative visual features, it lacks the cross-modal alignment necessary to link image distributions to text-based category priors. In the DECENT framework, the distributional score S_B functions as a calibrator for the semantic score S_A; without the foundational semantic anchor provided by multi-modal pre-training, the structural information captured by DINOv2 fails to translate into effective misclassification detection.

Superiority of Distribution-Level Measurement: We further benchmarked our proposed Energy Distance against widely used point-to-point algorithms, specifically KNN and TTA (Table A9). The results clearly demonstrate that Energy Distance yields superior results. The performance gap stems from the fact that KNN and TTA rely on instance-level (point-to-point) similarities, which are highly sensitive to local noise and individual sample outliers. Conversely, Energy Distance operates on a distribution-to-distribution level. By measuring the divergence between the augmented test point cloud P_x and the prototype cluster distributions, it comprehensively captures macroscopic distribution shifts, providing a more stable and robust signal for identifying OOD samples and misclassifications.

This analysis confirms that DECENT is a flexible and highly effective framework. It clearly demonstrates that the combination of vision-language aligned models (like CLIP) and distribution-aware metrics (like Energy Distance) forms the optimal synergy for our task.

C Additional Visualizations and Case Study

In this section, we provide supplementary qualitative results to intuitively demonstrate the internal mechanisms and performance of the DECENT framework.

Feature Space Separability (DECENT-R) To investigate why DECENT-R improves classification robustness, we visualize the feature space using t-SNE. As shown in FigureA4, DECENT-R significantly enhances feature separability by pushing correct (purple dots) and incorrect (blue crosses) samples into more distinct clusters. This intuitively explains its ability to correct the VLM's erroneous judgments.

To demonstrate the superior detection sensitivity of DECENT-S, we plot the frequency distribution histograms of the confidence scores. As shown in Figure A5, DECENT-S successfully separates the two distributions, exhibiting a clear bimodal morphology: misclassifications (blue) are pushed towards the low-score region, while correct predictions (purple) are maintained in the high-score region.

Failure Case Analysis. A primary failure mode arises when overly aggressive augmentations compromise the semantic integrity of the input image. This leads to a compromised P_x extracted by $\mathcal{M}_B$ and a misleading distributional score S_B. These localized failures, illustrated in Figure A3, delineate the method's robustness limits.

DECENT-S Failure Cases (Detection Failure). As illustrated in the top-left case (e.g., "Diner"), the VLM ($\mathcal{M}_A$) correctly classifies the sample with high confidence. However, excessive augmentation artifacts distort the feature point cloud P_x, resulting in a low distributional score S_B. This anomalously low S_B unduly penalizes the correct semantic score S_A, resulting in a low final confidence s_{conf}. Conversely, the left case (e.g., "Iceberg") demonstrates a missed detection where the VLM prediction is incorrect. Ideally, this should yield a low S_B. However, due to stochastic augmentation noise, the distorted P_x coincidentally aligns with the prototype of an incorrect class, generating an artificially high S_B. This misleading signal masks the VLM's error, resulting in a high s_{conf} and failing to flag the misclassification.

DECENT-R Failure Cases (Robustness Failure). The bottom-left sample illustrates a failure case where the VLM ($\mathcal{M}_A$) originally classified the sam-

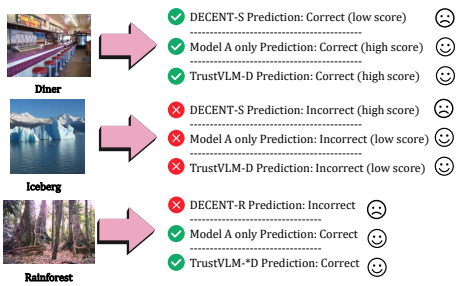


Figure A3: Qualitative examples of DECENT's failure modes.

ple correctly. However, extreme augmentation distortion caused the feature point cloud P_x (extracted by $\mathcal{M}_B$) to generate an S_B vector that strongly favored an incorrect class. In the fusion process $S_{final} = (1-\kappa)S_A + \kappa S_B$, this erroneous S_B signal overrode the correct S_A signal, resulting in an incorrect final prediction.

D Computational Cost

Our DECENT framework is a training-free method; thus, its computational cost does not involve gradient-based optimization. The cost is divided into a one-time offline setup cost and a per-sample online inference cost. All benchmarks were conducted on a single NVIDIA RTX 4090 GPU.

Offline Preparation Cost. Before inference, a prototype database $(\mathcal{Q}, \mathcal{Z})$ must be constructed for each benchmark dataset. This is a one-time cost. Feature Extraction: The primary cost is extracting features for the entire reference dataset $\mathcal{D}_{ref}$ (i.e., the training split). This requires N_{ref} (number of reference samples) forward passes through the auxiliary encoder $\mathcal{M}_B$ (CLIP ViT-B/16). Clustering: We then perform C (number of classes) independent K-Means clustering operations on the extracted features. This entire process is highly efficient. On our 4090 GPU, we can process approximately 150,000 images (including both feature extraction and clustering) in roughly one hour. This offline cost is only incurred once per dataset.

Online Inference Cost. The online cost is the latency required to process a single test image x. This involves three steps: S_A (Semantic Score) Computation: This requires one forward pass of the VLM $\mathcal{M}_A$ (image encoder) and a negligible similarity calculation. S_B (Distributional Score) Computation: This is the main computational bottleneck. It requires forward passes of $\mathcal{M}_B$ (e.g., $V = 16$) to generate the feature point cloud P_x, one N_{top} nearest neighbor search (e.g., $N_{top} = 16$) against the $\sum_{j=1}^{C} K_j$ prototype centroids, and N_{top} Energy Distance computations between P_x and the N_{top} neighboring cluster member sets Z_i. Fusion has a negligible cost for a single weighted sum of S_A and S_B.

In Summary, the total inference cost is dominated by the $V+1$ forward passes and the N_{top} Energy Distance calculations. On our single NVIDIA RTX 4090, our implementation achieves an online inference throughput of approximately 300 images per minute (i.e., ~200 ms per image). While this is more expensive than baselines like MSP (1 pass) or TrustVLM-D (2 passes).

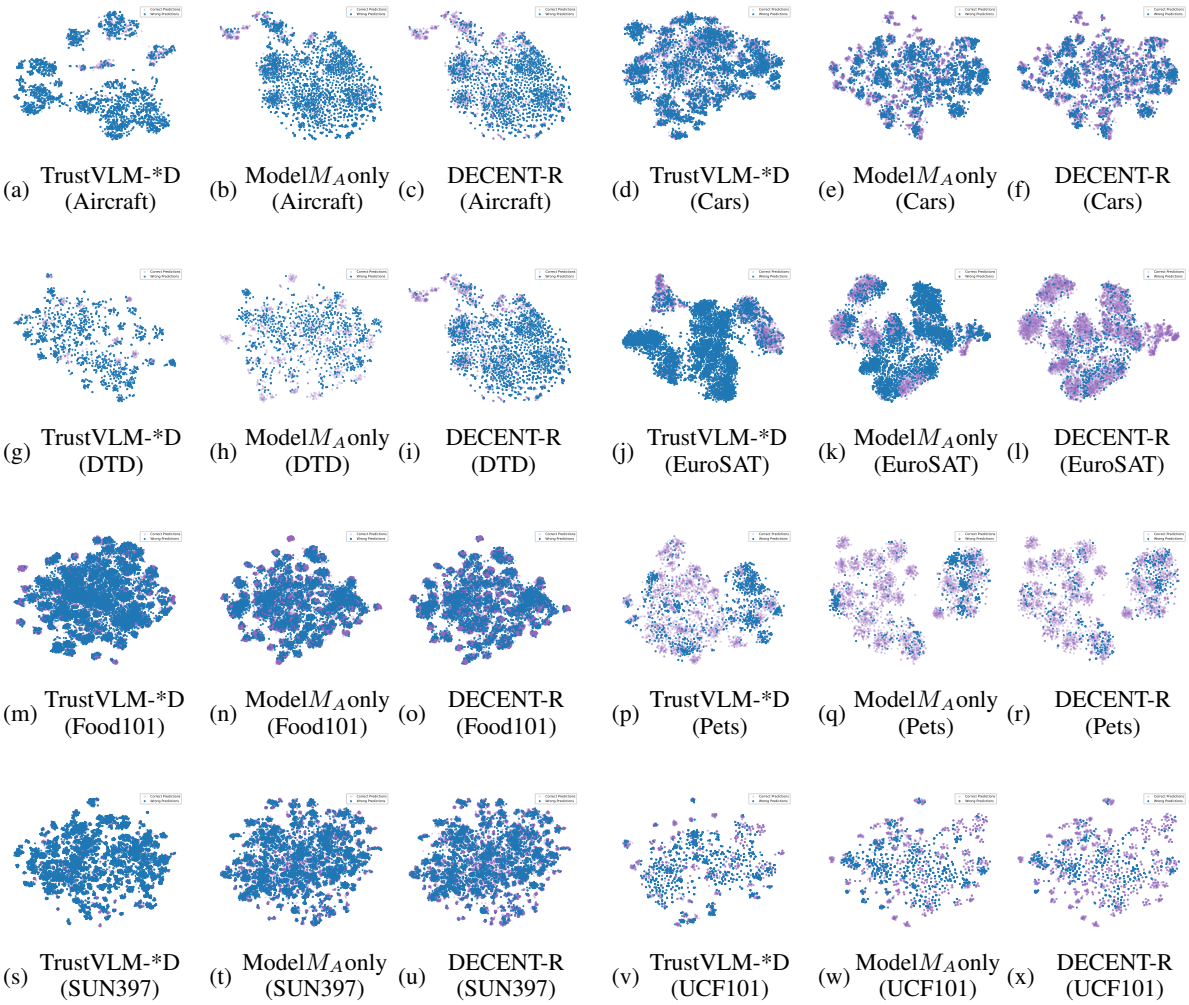


Figure A4: t-SNE visualization of feature space separability . Purple dots represent correct predictions, and blue crosses represent wrong predictions.

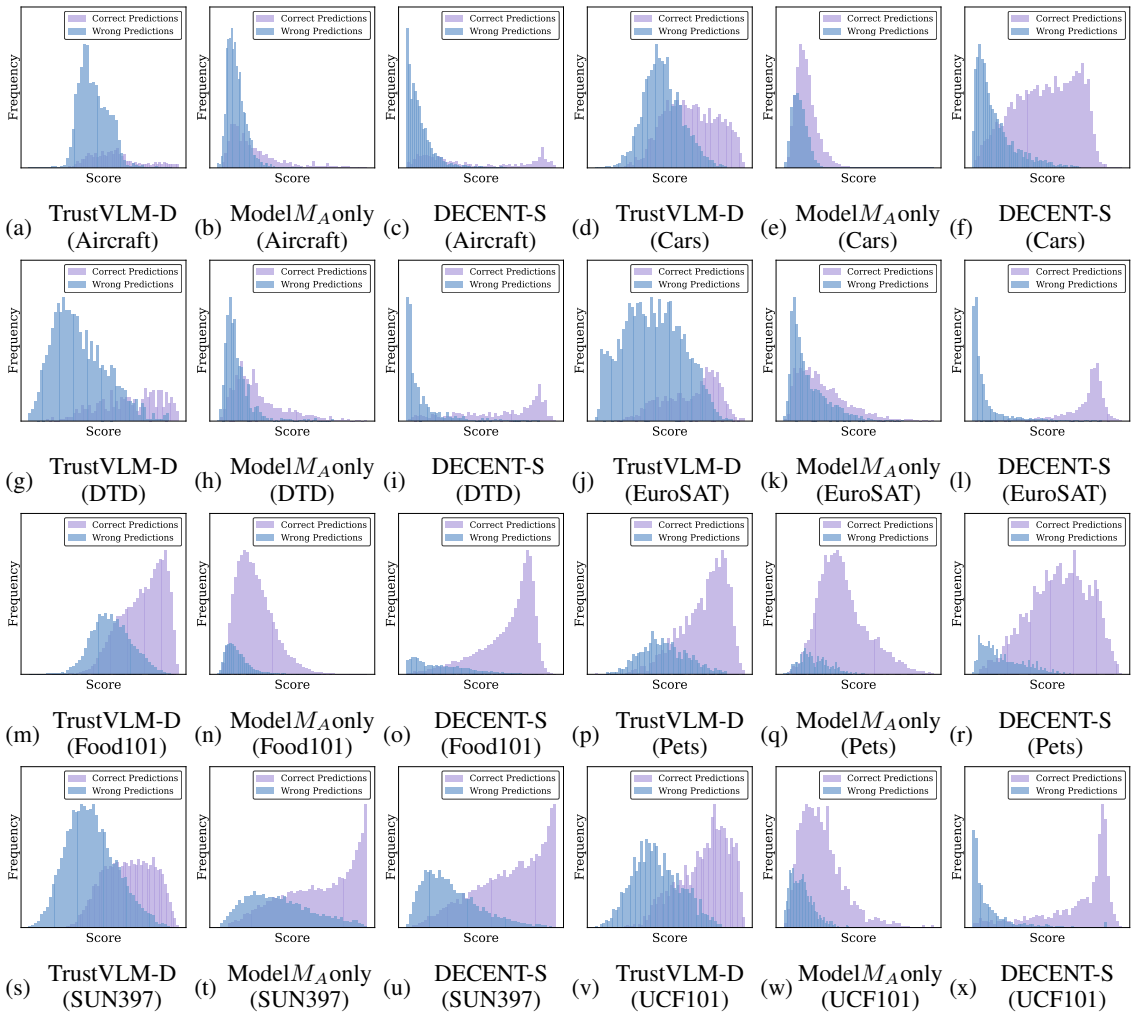


Figure A5: Confidence score distribution histograms for DECENT-R and baselines, successfully separating the two distributions by pushing misclassifications (blue) to the low-score region and correct predictions (purple) to the high-score region.