MD-ProTector: Multiple Data-Driven Prototype-Based LLM-Generated Text Detection

Anonymous ACL submission

Abstract

As LLM-generated content becomes more sophisticated, detection systems for distinguishing those texts from human-written text must operate at scale while handling diverse writing styles, domains, languages, and generator models. Relying only on standard binary classification formulations can neglect diversity within each class. We propose MD-ProTector, an encoder-based detector that models internal diversity in both classes using multiple data-driven prototypes, which are representative vectors summarizing local regions within each class. MD-ProTector represents each class with multiple learnable prototypes that are initialized from the data distribution and optimized jointly with the encoder. Its training objective combines Prototype-to-Class, Sample-to-Prototype, and Prototype Positioning losses to preserve the human–machine decision while allowing prototypes to specialize to class-internal variation. Experiments across mixed, adversarial, language-shift, and MAGE leave-one-out settings show that MD-ProTector achieves competitive detection performance, and ablation studies validate the proposed prototype organization method.

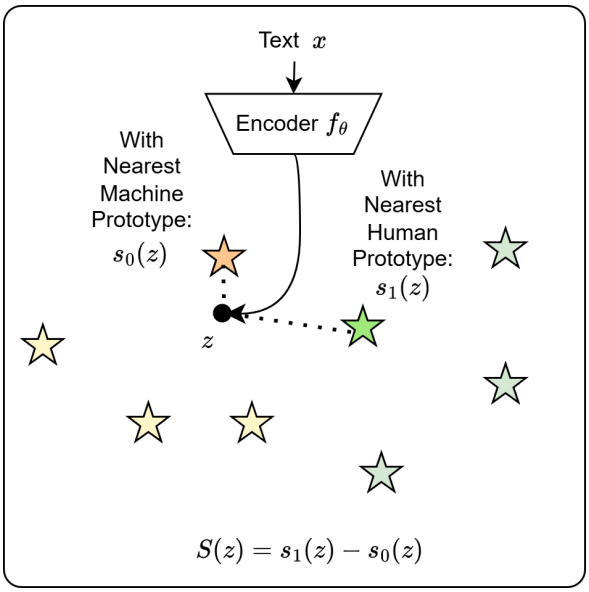


Figure 1: **Prototype-Based Inference.** Given an input text x, MD-ProTector encodes it into a normalized embedding z and compares it with the nearest machine and human prototypes. The detection score is computed as $S(z) = s_1(z) - s_0(z)$, where $s_0(z)$ and $s_1(z)$ denote the maximum similarities to machine and human prototype banks, respectively.

1 Introduction

As large language models have become increasingly sophisticated and widely accessible, detecting whether text is human-written or LLM-generated has become essential for ensuring credibility in digital communication (Kwon and Jang, 2025). This is particularly important for blocking automated fake news and phishing attacks, as well as maintaining academic integrity (Najjar et al., 2025). In large-scale deployment scenarios, detection systems must be applied to massive volumes of content that span diverse writing styles, domains, languages, and generator models (Li et al., 2024). Treating such diversity is important for practical detection systems operating in real-world platform services such as social media and news portals (Wu et al., 2024).

In these practical deployment settings, pre-embedded signals like watermarks or model internals such as log-likelihoods are not guaranteed to be available (Kandula et al., 2025; Christ et al., 2024; Mitchell et al., 2023). Under these conditions, detectors based on lightweight text encoders provide a practical solution. They operate directly on the input text without requiring access to the generation pipeline or model internals (Kuznetsov et al., 2024). This allows the same detection pipeline to be applied uniformly across diverse writing styles and domains, which is feasible for large-scale deployment (Rodriguez et al., 2022).

One of the simplest designs for encoder-based detectors is to attach a binary classification head and train it using a cross-entropy loss (Ippolito et al., 2020). Despite such intra-class diversity in writing style and domain, all texts are reduced to two categories during training under this design. The standard binary classification objective is a coarse supervision signal that does not explicitly consider such intra-class diversity (Cui et al., 2016). Several prior works have attempted to move beyond standard binary classification formulations by adding structural constraints, but still overlook intra-class diversity within at least one of the two classes, which can limit detection performance (Guo et al., 2024; Zeng et al., 2025). This observation motivates detectors that represent both human-written and LLM-generated texts with multiple local representatives rather than a single global class representation.

To this end, we propose **MD-ProTector**, a novel LLM-generated text detector with multiple data-driven prototypes. Our approach trains the encoder with multiple learnable prototypes, which are representative embedding vectors that summarize local regions within each class (Snell et al., 2017). Instead of collapsing each class into a single global cluster, MD-ProTector represents human-written and LLM-generated texts with class-specific prototype sets and classifies an input by comparing it with the nearest prototypes from the two sets. The encoder and prototypes are jointly trained with Prototype-to-Class, Sample-to-Prototype, and Prototype Positioning losses; the last encourages same-class prototypes to cover different local writing patterns rather than collapse into redundant anchors.

We evaluate MD-ProTector across a broad set of detection settings that cover mixed domain and generator conditions, domain and generator shifts, multilingual evaluation, and adversarially perturbed generations. The results show that MD-ProTector maintains competitive detection performance under diverse evaluation conditions. We further conduct extensive ablation studies to analyze the contribution of the proposed training objectives, data-driven prototype initialization, prototype count, temperature choice, and encoder backbone. The ablation results show that the full prototype organization is more effective than using a single prototype, random initialization, simple prototype diversity regularization, or prototype positioning without removing the class-level direction.

Our contributions can be summarized as follows:

- We propose MD-ProTector, a novel multiple-prototype detector for LLM-generated text detection that represents both human-written and LLM-generated texts with multiple data-driven prototypes.

- We develop a prototype-aware training strategy composed of Prototype-to-Class, Sample-to-Prototype, and Prototype Positioning losses, where Prototype Positioning encourages same-class prototypes to become distinct local representatives rather than redundant anchors.

- We evaluate MD-ProTector across diverse detection settings, including domain, generator, language, and adversarial shifts, and verify the effectiveness of its design choices through extensive ablation studies.

2 Related Works

2.1 LLM-Generated Text Detection

Prior work on LLM-generated text detection includes watermarking, zero-shot statistical methods, and supervised classifiers, among others, depending on model access and the types of detection signals used (Wu et al., 2025). Watermarking injects identifiable patterns during generation, enabling content source verification by model providers (Kirchenbauer et al., 2023). Zero-shot statistical detectors, such as DetectGPT and GLTR, exploit differences in probability distributions using metrics like perplexity or curvature without requiring labeled data (Mitchell et al., 2023; Gehrmann et al., 2019). However, watermarking assumes prior cooperation from generators, and statistical methods often require repeated model inference, incurring high computational cost (Christ et al., 2024; Bao et al., 2024). As a practical alternative in large-scale deployment scenarios, supervised detectors based on lightweight text encoders operate solely on input text and are suitable for in-the-wild deployment (Bakhtin et al., 2019; Uchendu et al., 2020; Wang et al., 2023). These detectors typically rely on encoder architectures such as BERT (Devlin et al., 2019) or RoBERTa (Liu et al., 2019), whose parameter counts are on the order of hundreds of millions—substantially smaller than the multi-billion-parameter scale of modern LLMs.

Despite their efficiency, these classifiers exhibit significant performance degradation on unseen domains or emerging model families (Bakhtin et al., 2019). Recent work attempts to improve robustness by imposing stronger structural constraints on the representation space. For example, DeTeCtive organizes text instances from multiple generators through a hierarchical contrastive objective, while implicitly treating human-written text as a single homogeneous class (Guo et al., 2024). In contrast, DeepSVDD-based one-class detectors enforce compactness by minimizing the variance of machine-generated samples and modeling human-written text as outliers (Zeng et al., 2025). These formulations collapse one of the classes into a single cluster, which is a restrictive assumption considering that both human-written and machine-generated text exhibit substantial heterogeneity across domains, prompts, and model families. MD-ProTector instead represents both human-written and LLM-generated texts with prototype banks and uses these banks directly for prototype-based human–machine decision making.

2.2 Prototype-Based Representation Learning

Prototype-based methods represent each class by one or more representative points in an embedding space and classify inputs based on their similarity to these prototypes. A canonical example is Prototypical Networks, which learn an encoder and compute class prototypes as the mean embeddings of support examples, classifying queries via distances to these prototypes in a metric-learning framework (Snell et al., 2017). This view treats prototypes as data-driven summaries of class structure: training encourages instances of the same class to concentrate around their prototype while separating prototypes of different classes. Subsequent work has extended this idea beyond the original few-shot setting, exploring multiple prototypes per class, part- or attribute-level prototypes, and more flexible similarity functions in settings ranging from metric and meta-learning to interpretable models in vision and NLP (Chen et al., 2019; Hong et al., 2023).

Prototype-based representations have also been applied beyond standard classification, including OOD detection via prototype feature similarity (Chen et al., 2024) and contrastive representation learning where prototypes serve as cluster-level latent variables (Li et al., 2021). Recent multi-prototype variants further model heterogeneous feature distributions: fine-grained prototypes and prototype mixtures have been used for OOD detection, including sample-to-prototype assignment and prototype-level objectives (Jia et al., 2025; Lu et al., 2024); multiple normal prototypes have been combined with clustering and contrastive learning for weakly supervised anomaly detection (Dong et al., 2024); and multi-prototype modeling has been used to capture intra-class structure in open-set noisy label learning (Zhang et al., 2025). Prior multi-prototype methods primarily organize heterogeneous embeddings for objectives such as OOD scoring, anomaly detection, or noisy-label/open-set learning. MD-ProTector instead uses prototypes as trainable components of black-box text-source detection: they are aligned with the human–machine decision, attached to same-class samples, and positioned to become distinct local representatives.

3 Proposed Method

We propose MD-ProTector, a prototype-based detector that represents both human-written and LLM-generated texts with multiple class-specific prototypes. The prototypes are initialized from the data distribution and then optimized jointly with the encoder. The training objective consists of three complementary losses. First, Prototype-to-Class loss aligns each prototype with the class-level direction. Second, Sample-to-Prototype loss attaches individual samples to a soft mixture of prototypes from the ground-truth class. Finally, Prototype Positioning loss uses the same assignment structure to separate prototype slots in the residual space after removing the class hub direction. Together, these objectives encourage prototypes to preserve class identity while capturing class-internal diversity.

3.1 Problem Formulation and Encoder

Given an input text x_i, each sample is associated with a binary label $y_i \in \{0, 1\}$, where $y_i = 0$ denotes LLM-generated text and $y_i = 1$ denotes human-written text. A lightweight encoder f_θ maps the input text into token-level representations, which are mean-pooled to obtain a sentence embedding. The output embedding is then projected onto the unit hypersphere:

$$z_i = \mathrm{norm}(f_\theta(x_i)), \tag{1}$$

For each class $c \in \{0, 1\}$, we maintain R learnable prototypes:

$$\mathcal{P}_c = \{p_{c,1}, \ldots, p_{c,R}\}, \qquad \|p_{c,r}\|_2 = 1. \tag{2}$$

The full prototype bank is denoted by $\mathcal{P} = \mathcal{P}_0 \cup \mathcal{P}_1$.

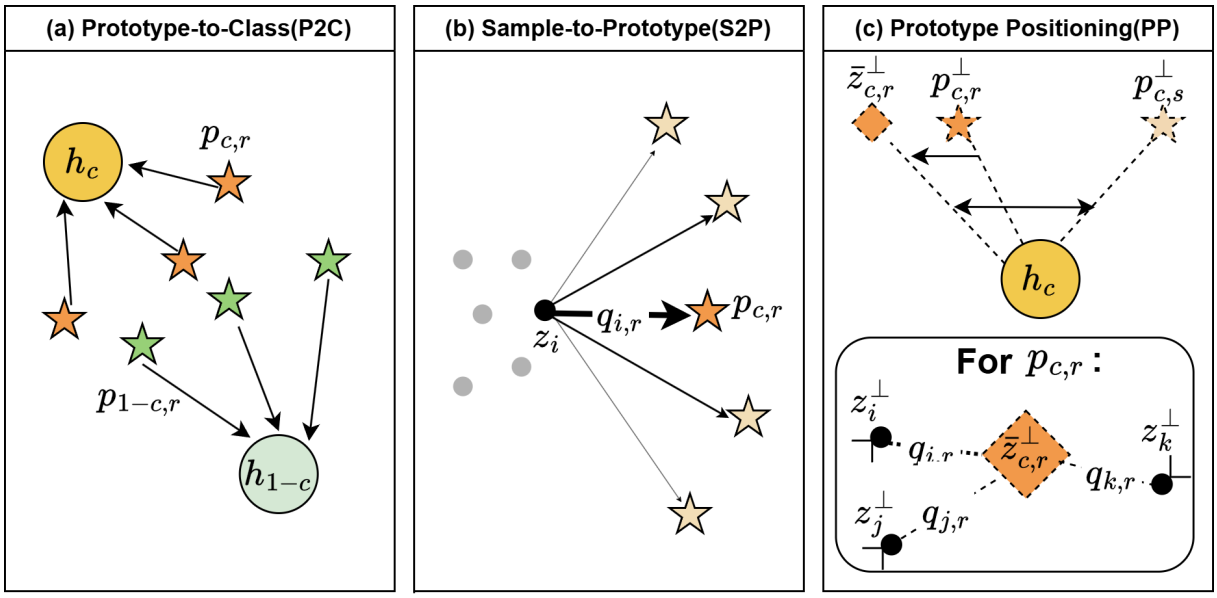


Figure 2: Overview of the training objectives. Circles denote class hubs, stars denote learnable prototypes, dots denote sample embeddings, and diamonds denote weighted sum of sample residuals. Prototype-to-Class loss aligns each prototype with its class hub. Sample-to-Prototype loss attaches each sample to a soft mixture of prototypes from its ground-truth class. Prototype Positioning loss uses the same assignment structure to position prototype slots in the residual space.

3.2 Data-Driven Prototype Initialization

Prototypes are initialized using a data-driven approach to avoid poor local optima. Before training, we extract embeddings for the training set and perform K-Means algorithm per class (McQueen, 1967). The resulting centroids serve as the initial prototypes $\mathcal{P}_c^{(0)} = \{\mathbf{p}_{c,r}^{(0)}\}_{r=1}^{R}$.

After initialization, all prototypes are normalized. During training, prototypes are treated as learnable parameters and optimized together with the encoder. After each optimization step, prototypes are projected back onto the unit hypersphere. This design keeps the prototypes data-driven at initialization while allowing them to adapt to the task-specific training objective.

3.3 Training Objectives

For compact notation, we define

$$\phi(u, v) = \exp(u^\top v/\tau), \tag{3}$$

where τ is a temperature parameter. For a mini-batch B, let $B_c = \{i \in B : y_i = c\}$ denote the set of samples belonging to class c. We compute the class hub as

$$h_c = \text{norm}\left(\frac{1}{|B_c|}\sum_{i\in B_c} z_i\right). \tag{4}$$

The class hub represents the current class-level direction in the mini-batch and is used to organize the prototype space.

Prototype-to-Class Loss. The first objective encourages each prototype to align with the hub of its own class:

$$\mathcal{L}_{\text{P2C}} = -\frac{1}{2R}\sum_{c=0}^{1}\sum_{r=1}^{R}\log\frac{\phi(p_{c,r}, h_c)}{\sum_{d=0}^{1}\phi(p_{c,r}, h_d)}. \tag{5}$$

This loss makes each prototype aware of its class identity before modeling finer prototype-level structure.

Sample-to-Prototype Loss. The second objective attaches each sample to the prototype mixture of its ground-truth class. We first compute the within-class assignment of each sample to the prototypes of its ground-truth class:

$$q_{i,r} = \frac{\phi(z_i, p_{y_i,r})}{\sum_{k=1}^{R}\phi(z_i, p_{y_i,k})}. \tag{6}$$

Using $q_{i,r}$ as a stop-gradient target, the Sample-to-Prototype loss is defined as

$$\mathcal{L}_{\text{S2P}} = -\frac{1}{|B|}\sum_{i\in B}\sum_{r=1}^{R}\text{sg}(q_{i,r})\log\frac{\phi(z_i, p_{y_i,r})}{\sum_{p\in\mathcal{P}}\phi(z_i, p)}. \tag{7}$$

Here, $\mathrm{sg}(\cdot)$ denotes the stop-gradient operator. This loss encourages each sample to remain close to its assigned prototype mixture while separating it from prototypes of the opposite class.

Prototype Positioning Loss. While $\mathcal{L}_{\mathrm{S2P}}$ attaches samples to class-specific prototypes, it does not determine how prototype slots should be arranged around the class hub. To model class-internal diversity, we position prototypes using the residual component after removing the class-hub direction. For samples and prototypes, we define

$$z_i^{\perp} = z_i - (z_i^{\top} h_{y_i}) h_{y_i}, \tag{8}$$

$$p_{c,r}^{\perp} = \mathrm{norm}\left(p_{c,r} - (p_{c,r}^{\top} h_c) h_c\right). \tag{9}$$

Using the same assignment $q_{i,r}$, we form a residual aggregate for each prototype slot:

$$g_{c,r}^{\perp} = \sum_{i \in B_c} q_{i,r} z_i^{\perp}, \qquad \bar{z}_{c,r}^{\perp} = \mathrm{norm}(g_{c,r}^{\perp}). \tag{10}$$

Here, $\bar{z}_{c,r}^{\perp}$ summarizes the residual direction of samples assigned to slot r of class c. Let $\mathcal{P}^{\perp} = \{p_{d,r}^{\perp}\}_{d,r}$ denote the residual prototype bank. The Prototype Positioning loss is defined as

$$\mathcal{L}_{\mathrm{PP}} = -\frac{1}{2R} \sum_{c=0}^{1} \sum_{r=1}^{R} \log \frac{\phi(\bar{z}_{c,r}^{\perp}, p_{c,r}^{\perp})}{\sum_{p^{\perp} \in \mathcal{P}^{\perp}} \phi(\bar{z}_{c,r}^{\perp}, p^{\perp})}. \tag{11}$$

In this contrastive form, the numerator pulls the residual prototype $p_{c,r}^{\perp}$ toward the residual aggregate assigned to the same slot, while the denominator separates it from other residual prototypes. Together with $\mathcal{L}_{\mathrm{P2C}}$, this places same-class prototypes around the shared class hub while encouraging each slot to specialize to a different within-class direction.

Final Training Objective. The final training objective is the sum of the three prototype-aware losses:

$$\mathcal{L}_{\mathrm{train}} = \mathcal{L}_{\mathrm{P2C}} + \mathcal{L}_{\mathrm{S2P}} + \mathcal{L}_{\mathrm{PP}}. \tag{12}$$

The encoder parameters and prototype parameters are optimized jointly with this objective, and prototypes are re-normalized after each optimization step.

3.4 Inference

At inference time, classification is performed by comparing the input embedding with the learned prototype banks. Given an input text x, we compute $z = \mathrm{norm}(f_\theta(x))$. For each class, the score is computed as the maximum similarity to its prototypes:

$$s_c(z) = \max_{r \in \{1,\ldots,R\}} z^{\top} p_{c,r}. \tag{13}$$

We then define the detection score as

$$S(z) = s_1(z) - s_0(z), \tag{14}$$

where a larger value indicates that the input is closer to human-written prototypes than to LLM-generated prototypes. The final prediction is obtained by thresholding this score:

$$\hat{y} = \mathbf{1}\{S(z) > \delta\}, \tag{15}$$

where the threshold δ is selected on the validation set.

4 Experiments

4.1 Experimental Setup

Evaluation Settings. We evaluate MD-ProTector on MAGE, M4, and RAID, which cover distinct deployment scenarios. Detailed dataset statistics, split construction, validation and calibration protocols, and aggregation rules are provided in Appendix A. MAGE is used to evaluate domain and generator generalization: MAGE Cross-Domain Cross-Model (MAGE CDCM) tests mixed domain/generator conditions, while MAGE Unseen Domains and MAGE Unseen Models test leave-one-out domain and leave-one-out generator-family generalization (Li et al., 2024). M4 evaluates multilingual and unseen-language generalization (Wang et al., 2024). RAID evaluates robustness to adversarial attacks and decoding-related variations (Dugan et al., 2024). In MAGE Unseen Domains, one domain is excluded from training and used only for evaluation, and this process is repeated over 10 leave-one-out domains. In MAGE Unseen Models, one generator family is excluded from training and used only for evaluation, and this process is repeated over 7 leave-one-out generator-family scenarios. Each leave-one-out scenario trains a separate detector.

Baselines. We compare MD-ProTector with four encoder-based baselines. Binary CE attaches a linear classification head to the same backbone encoder and is trained with a standard binary cross-entropy objective. SupCon trains the encoder with

Method	MAGE CDCM	RAID	M4
Binary CE	90.79 (83.18/98.40)	86.81 (74.60/99.02)	76.68 (54.56/98.80)
SupCon	94.77 (92.98/96.56)	77.67 (58.94/96.39)	83.84 (72.37/95.30)
DeTeCtive	<u>94.84</u> (91.87/97.81)	<u>87.68</u> (80.17/95.20)	**92.74** (87.92/97.57)
DSVDD	94.43 (95.18/93.67)	86.17 (76.36/95.97)	81.08 (63.13/99.03)
MD-ProTector	**95.14** (95.81/94.47)	**88.18** (82.52/93.84)	<u>86.03</u> (76.87/95.20)

Table 1: Benchmark-level evaluation results. Each cell reports AvgRec with HumanRec/MachineRec in parentheses. MAGE CDCM evaluates mixed domain and generator conditions, RAID evaluates adversarial and decoding robustness, and M4 evaluates language shift. Bold and underline indicate the best and second-best AvgRec within each setting, respectively.

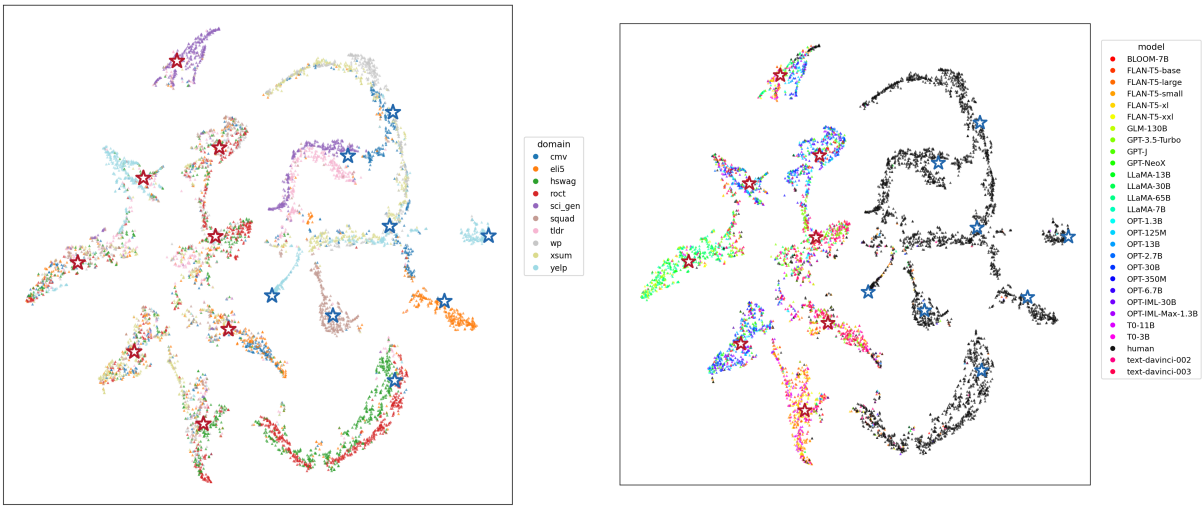
Method	MAGE Unseen Domains	MAGE Unseen Models
Binary CE	67.99 (37.04/98.94)	89.78 (85.85/93.71)
SupCon	75.11 (56.82/93.41)	90.92 (93.49/88.34)
DeTeCtive	76.72 (55.75/97.69)	**91.69** (92.08/91.30)
DSVDD	**79.08** (63.46/94.71)	90.71 (95.22/86.19)
MD-ProTector	<u>78.59</u> (61.46/95.72)	<u>91.34</u> (95.63/87.05)

Table 2: MAGE leave-one-out evaluation results. Each cell reports AvgRec with HumanRec/MachineRec in parentheses. MAGE Unseen Domains and MAGE Unseen Models report averages over 10 independently trained leave-one-out-domain scenarios and 7 independently trained leave-one-generator-family-out scenarios, respectively. Bold and underline indicate the best and second-best AvgRec within each setting.

supervised contrastive learning and performs prediction in the learned embedding space via KNN. DeTeCtive uses multi-level contrastive learning to capture writing-style structure and performs prediction via KNN (Guo et al., 2024). DSVDD formulates LLM-generated text detection through a one-class objective that compacts machine-generated embeddings and predicts with a distance-based score, treating human-written text as outliers (Zeng et al., 2025).

Evaluation Metrics. Following prior LLM-generated text detection studies (Li et al., 2024; Guo et al., 2024), we use Average Recall (AvgRec) as the primary metric. AvgRec averages recall on human-written and LLM-generated texts, preventing evaluation from being dominated by one class. In the main results, we report AvgRec together with Human Recall and Machine Recall to show both class-balanced performance and class-wise behavior. F1, Accuracy, AUROC, AUPR, and FPR95 are reported in Appendix C for completeness. For threshold-based metrics, the decision threshold is selected on the validation set and fixed for test evaluation.

Implementation Details. Unless otherwise specified, all encoder-based methods use the same backbone encoder within each evaluation setting. Pretrained encoder checkpoints are loaded from the HuggingFace model hub and used with the HuggingFace Transformers implementation (Wolf et al., 2020). We use 125M Unsupervised SimCSE-RoBERTa as the default encoder (Gao et al., 2021). All models are trained with a batch size of 256 using AdamW with a learning rate of 2×10^{-5}. Models are trained for 30 epochs with 2,000 warmup steps. The checkpoint with the best AvgRec on validation set is selected for evaluation. For MD-ProTector, the number of prototypes per class is set to $R = 8$ in the main experiments. All experiments are conducted on a single NVIDIA B200 GPU with mixed BF16 precision.



(a) Colored by domain

(b) Colored by generator model

Figure 3: Prototype visualization on the MAGE CDCM dataset. Both panels show the same t-SNE projection of normalized test embeddings and learned prototypes. The left panel colors samples by domain, and the right panel colors samples by generator model. Dots denote test samples. Blue and red stars indicate human and machine prototypes, respectively.

4.2 Evaluation Results

Tables 1 and 2 report the main evaluation results. Table 1 summarizes benchmark-level settings that test distinct evaluation axes: mixed domain/generator conditions, adversarial and decoding robustness, and language shift. Table 2 reports MAGE leave-one-out evaluations, where each result is averaged over multiple independently trained leave-one-out scenarios.

In Table 1, MD-ProTector achieves the best AvgRec on MAGE CDCM and RAID, and obtains the second-best on M4. This indicates that the proposed prototype-based detector remains competitive across mixed domain/generator conditions, adversarial and decoding robustness, and language shift. In Table 2, although the strongest competing method differs between domain and generator-family shifts, MD-ProTector obtains the second-best AvgRec in both MAGE Unseen Domains and MAGE Unseen Models, indicating stable class-balanced performance across the two leave-one-out shift settings while maintaining high HumanRec.

The class-wise recalls provide additional insight into the behavior of different detectors. Binary CE often obtains very high MachineRec but substantially lower HumanRec, indicating a tendency to favor the machine-generated class under several shifted settings. MD-ProTector mitigates this one-sided behavior in the benchmark-level evaluations, achieving more balanced recalls on MAGE CDCM and RAID and improving HumanRec over Binary CE on M4. For the MAGE leave-one-out settings, MD-ProTector remains competitive while maintaining strong HumanRec, particularly under leave-one-out generator-family generalization.

Full results for all methods, including F1, Accuracy, AUROC, AUPR, FPR95, and MAGE per-scenario results, are provided in Appendix C.

4.3 Ablation Studies

We conduct ablation studies on the MAGE CDCM dataset to analyze the main design choices of MD-ProTector. Details of the loss variants and the encoder backbone comparison are provided in Appendix B.

Prototype Initialization. K-Means initialization outperforms random initialization, confirming that data-driven prototype initialization contributes to enhancing the detection performance.

Training Objective. The training-objective ablation validates the proposed loss design for training a multiple-prototype detector. Removing $\mathcal{L}_{\mathrm{PP}}$ reduces AvgRec, indicating that sample-to-prototype attachment alone is not sufficient to organize prototype slots. Replacing $\mathcal{L}_{\mathrm{PP}}$ with a simple same-class prototype repulsion loss also performs worse, suggesting that data-supported positioning is more effective than simply pushing prototypes apart. The raw-space variant further degrades performance, indicating that removing the class hub direction helps

Variant	AvgRec
Prototype Initialization	
K-Means	**95.14**
Random	94.50
Training Objective ($R = 8,\ \tau = 0.15$)	
Full objective	**95.14**
w/o $\mathcal{L}_{\mathrm{PP}}$	94.78
PP $\rightarrow$ Simple Prototype Repulsion	94.55
PP w/o residual	94.33
Number of Prototypes ($\tau = 0.15$)	
$R = 1$	94.50
$R = 2$	95.03
$R = 4$	94.88
$R = 8$	**95.14**
$R = 16$	94.53
$R = 32$	94.36
Temperature ($R = 8$)	
$\tau = 0.07$	94.91
$\tau = 0.10$	95.07
$\tau = 0.15$	**95.14**
$\tau = 0.20$	95.05
$\tau = 0.50$	93.97
Encoder Backbone	
unsup-simcse-roberta-base	**95.14**
roberta-base	94.66
sup-simcse-roberta-base	94.01
e5-base	93.70
bert-base-uncased	93.60
unsup-simcse-bert-base	93.24
bge-base-en-v1.5	92.86

Table 3: Ablation studies on the MAGE CDCM dataset. All entries report AvgRec. The default configuration uses the full objective, $R = 8$, $\tau = 0.15$, K-Means initialization, and the unsupervised SimCSE-RoBERTa encoder. "PP w/o residual" denotes Prototype Positioning without removing the class hub direction.

prototypes model class-internal variation.

Configuration Choices. The temperature and prototype counts support the default configuration. The best AvgRec is obtained at $\tau = 0.15$ and $R = 8$, while overly high temperature or large prototype banks reduce performance.

Encoder Backbone. The encoder ablation shows that unsupervised SimCSE-RoBERTa provides the strongest representation space among the evaluated backbones. These results suggest that while MD-ProTector can be instantiated with different encoder families, its absolute performance benefits from a suitable pretrained representation space.

4.4 Prototype Visualization

To examine the learned prototype structure, we visualize test samples and prototypes on the MAGE CDCM dataset using t-SNE (van der Maaten and Hinton, 2008). Figure 3 shows the same embedding projection colored by domain and by generator model.

Rather than being concentrated in a single area, prototypes are spread across several sample groups. This is consistent with the inference rule of MD-ProTector, where each input is compared with the nearest prototype from each class. The visualization therefore suggests that the learned prototype bank provides multiple reference points for different parts of the embedding space, rather than relying on one global representative.

The two color views also show that these prototypes are not tied to a single metadata label. In the domain-colored view, some regions are dominated by one domain, while other regions contain samples from multiple domains. In the generator-colored view, generator labels are also spread across several regions rather than forming a simple one-prototype-per-generator pattern. This suggests that the prototypes serve as local reference points in the learned representation space, rather than explicit domain or generator identifiers.

5 Conclusion

We presented MD-ProTector, an encoder-based detector that represents human-written and LLM-generated text with class-specific data-driven prototypes. Its training objective combines Prototype-to-Class, Sample-to-Prototype, and Prototype Positioning losses to preserve the human–machine decision while allowing same-class prototypes to cover different local writing patterns. Experiments across mixed, adversarial, language-shift, and MAGE leave-one-out settings show that MD-ProTector is competitive across diverse distribution shifts, and ablation studies validate the role of Prototype Positioning, data-driven initialization, and the default prototype configuration. These results suggest that modeling multiple prototypes for each class provides an effective decision structure for robust LLM-generated text detection with encoder-based models.

Limitations

This work assumes a fixed-label binary detection setting in which each input is classified as either human-written or LLM-generated, and does not address more complex scenarios such as partial generation, human–machine co-editing, or estimating degrees of machine involvement. In addition, the number of prototypes per class is treated as an empirically fixed design choice. Adaptive mechanisms for adjusting prototype cardinality based on data characteristics are not explored. Finally, our experiments are conducted under fixed training, validation, and test splits. In practical deployment, the distribution of generators, prompts, writing styles, and adversarial perturbations may change over time. While MD-ProTector initializes and optimizes prototypes from training data, continual prototype adaptation under temporal distribution drift remains future work.

Ethics Statement

The datasets utilized do not include private data or non-public personally identifiable information. The development of reliable text detection systems is crucial for maintaining trust in digital information. However, we acknowledge that detection technologies can potentially be used in a dual-use manner—adversaries might use our detector as a discriminator to train more sophisticated generators that evade detection. Additionally, while we strove to use diverse datasets, the "Human" class in our training data is sourced from web texts (e.g., Reddit, Wikipedia), which may contain inherent biases. Users should be cautious when deploying this model in sensitive contexts, as false positives could unfairly penalize human writers.

References

Anton Bakhtin, Sam Gross, Myle Ott, Yuntian Deng, Marc'Aurelio Ranzato, and Arthur Szlam. 2019. Real or fake? learning to discriminate machine from human generated text. *arXiv preprint arXiv:1906.03351*.

Guangsheng Bao, Yanbin Zhao, Zhiyang Teng, Linyi Yang, and Yue Zhang. 2024. Fast-detectGPT: Efficient zero-shot detection of machine-generated text via conditional probability curvature. In *The Twelfth International Conference on Learning Representations*, pages 5046–5068, Vienna, Austria.

Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. 2019. This looks like that: Deep learning for interpretable image recognition. In *Advances in Neural Information Processing Systems*, volume 32, pages 8928–8939, Vancouver, BC, Canada. Curran Associates, Inc.

Jun-Kun Chen, Jilin Mei, Liang Chen, Fangzhou Zhao, Yan Xing, and Yu Hu. 2024. Proto-OOD: Enhancing OOD object detection with prototype feature similarity. *arXiv preprint arXiv:2409.05466*.

Miranda Christ, Sam Gunn, and Or Zamir. 2024. Undetectable watermarks for language models. In *The Thirty Seventh Annual Conference on Learning Theory*, pages 1125–1139, Edmonton, Canada. Proceedings of Machine Learning Research.

Yin Cui, Feng Zhou, Yuanqing Lin, and Serge J. Belongie. 2016. Fine-grained categorization and dataset bootstrapping using deep metric learning with humans in the loop. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*, pages 1153–1162, Las Vegas, NV, USA. IEEE Computer Society.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Zhijin Dong, Hongzhi Liu, Boyuan Ren, Weimin Xiong, and Zhonghai Wu. 2024. Reconstruction-based multi-normal prototypes learning for weakly supervised anomaly detection. *arXiv preprint arXiv:2408.14498*.

Liam Dugan, Alyssa Hwang, Filip Trhlík, Andrew Zhu, Josh Magnus Ludan, Hainiu Xu, Daphne Ippolito, and Chris Callison-Burch. 2024. RAID: A shared benchmark for robust evaluation of machine-generated text detectors. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12463–12492, Bangkok, Thailand. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander Rush. 2019. GLTR: Statistical detection and visualization of generated text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 111–116, Florence, Italy. Association for Computational Linguistics.

Xun Guo, Shan Zhang, Yongxin He, Ting Zhang, Wanquan Feng, Haibin Huang, and Chongyang Ma. 2024. DeTeCtive: Detecting ai-generated text via multi-level contrastive learning. In *Advances in Neural Information Processing Systems*, volume 37, pages 88320–88347. Curran Associates, Inc.

Dat Hong, Tong Wang, and Stephen Baek. 2023. ProtoryNet-interpretable text classification via prototype trajectories. *Journal of Machine Learning Research*, 24(264):1–39.

Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. 2020. Automatic detection of generated text is easiest when humans are fooled. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1808–1822, Online. Association for Computational Linguistics.

Yulong Jia, Jiaming Li, Ganlong Zhao, Shuangyin Liu, Weijun Sun, Liang Lin, and Guanbin Li. 2025. Enhancing out-of-distribution detection via diversified multi-prototype contrastive learning. *Pattern Recognition*, 161:111214.

Hemanth Kandula, Chak Fai Li, Haoling Qiu, Damianos Karakos, Hieu Man, Thien Huu Nguyen, and Brian Ulicny. 2025. BBN-U.Oregon's ALERT system at GenAI content detection task 3: Robust authorship style representations for cross-domain machine-generated text detection. In *Proceedings of the 1st Workshop on GenAI Content Detection (GenAIDetect)*, pages 358–364, Abu Dhabi, UAE. International Conference on Computational Linguistics.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023. A watermark for large language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pages 17061–17084, Vienna, Austria. Proceedings of Machine Learning Research.

Kristian Kuznetsov, Eduard Tulchinskii, Laida Kushnareva, German Magai, Serguei Barannikov, Sergey Nikolenko, and Irina Piontkovskaya. 2024. Robust AI-generated text detection by restricted embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 17036–17055, Miami, Florida, USA. Association for Computational Linguistics.

Soonchan Kwon and Beakcheol Jang. 2025. A comprehensive survey of fake text detection on misinformation and lm-generated texts. *IEEE Access*, 13:25301–25324.

Junnan Li, Pan Zhou, Caiming Xiong, and Steven Hoi. 2021. Prototypical contrastive learning of unsupervised representations. In *9th International Conference on Learning Representations*, pages 5576–5591, Virtual Event, Austria.

Yafu Li, Qintong Li, Leyang Cui, Wei Bi, Zhilin Wang, Longyue Wang, Linyi Yang, Shuming Shi, and Yue Zhang. 2024. MAGE: Machine-generated text detection in the wild. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 36–53, Bangkok, Thailand. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Haodong Lu, Dong Gong, Shuo Wang, Jason Xue, Lina Yao, and Kristen Moore. 2024. Learning with mixture of prototypes for out-of-distribution detection. In *The Twelfth International Conference on Learning Representations*, pages 6039–6061, Vienna, Austria.

James B McQueen. 1967. Some methods of classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. DetectGPT: Zero-shot machine-generated text detection using probability curvature. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pages 24950–24962, Honolulu, Hawaii. Proceedings of Machine Learning Research.

Ayat A. Najjar, Huthaifa I. Ashqar, Omar A. Darwish, and Eman M. Hammad. 2025. Detecting ai-generated text in educational content: Leveraging machine learning and explainable AI for academic integrity. *arXiv preprint arXiv:2501.03203*.

Juan Diego Rodriguez, Todd Hay, David Gros, Zain Shamsi, and Ravi Srinivasan. 2022. Cross-domain detection of GPT-2-generated technical text. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1213–1233, Seattle, United States. Association for Computational Linguistics.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, volume 30, pages 4077–4087, Long Beach, CA, USA. Curran Associates, Inc.

Adaku Uchendu, Thai Le, Kai Shu, and Dongwon Lee. 2020. Authorship attribution for neural text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8384–8395, Online. Association for Computational Linguistics.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(86):2579–2605.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *ArXiv*, abs/2212.03533.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Toru Sasaki, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. M4: Multi-generator, multi-domain, and multilingual black-box machine-generated text detection. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1369–1407, St. Julian's, Malta. Association for Computational Linguistics.

Zecong Wang, Jiaxi Cheng, Chen Cui, and Chenhao Yu. 2023. Implementing BERT and fine-tuned roberta to detect AI generated news by chatgpt. *arXiv preprint arXiv:2306.07401*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Junchao Wu, Shu Yang, Runzhe Zhan, Yulin Yuan, Lidia Sam Chao, and Derek Fai Wong. 2025. A survey on LLM-generated text detection: Necessity, methods, and future directions. *Computational Linguistics*, 51(1):275–338.

Junchao Wu, Runzhe Zhan, Derek F. Wong, Shu Yang, Xinyi Yang, Yulin Yuan, and Lidia S. Chao. 2024. DetectRL: Benchmarking llm-generated text detection in real-world scenarios. In *Advances in Neural Information Processing Systems*, volume 37, pages 100369–100401, Vancouver, BC, Canada. Curran Associates, Inc.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. 2024. C-Pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 641–649, Washington, DC, USA. Association for Computing Machinery.

Cong Zeng, Shengkun Tang, Yuanzhou Chen, Zhiqiang Shen, Wenchao Yu, Xujiang Zhao, Haifeng Chen, Wei Cheng, and zhiqiang xu. 2025. Human texts are outliers: Detecting LLM-generated texts via out-of-distribution detection. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, San Diego, California, USA.

Yue Zhang, Yiyi Chen, Chaowei Fang, Qian Wang, Jiayi Wu, and Jingmin Xin. 2025. Learning from open-set noisy labels based on multi-prototype modeling. *Pattern Recognition*, 157:110902.

A Evaluation Set and Protocol Details

This section reports dataset statistics and protocol details for MAGE, M4, and RAID. All datasets are cast as binary human-written versus machine-generated text detection, and human-written text is treated as the positive class in our metric implementation. Validation data are used only for checkpoint selection, threshold calibration, and hyperparameter selection; test labels are not used for model selection or threshold tuning.

MAGE Splits. MAGE CDCM uses the cross-domain cross-model setting of MAGE (Li et al., 2024). The processed split contains 10 source domains: CMV, ELI5, HellaSwag, ROC, SciGen, SQuAD, TLDR, WP, XSum, and Yelp. Machine-generated texts are produced by 27 model variants grouped into seven Generator Model Families: LLaMA, BigScience, FLAN-T5, GLM-130B, EleutherAI, OpenAI, and OPT. For MAGE Unseen Domains, each leave-one-domain-out scenario excludes one source domain from training and validation and evaluates on that domain. For MAGE Unseen Models, each leave-one-generator-family-out scenario excludes one Generator Model Family from training and validation and evaluates on that family. Specifically, we used LLaMA-7B, BLOOM-7B, FLAN-T5-Small, GLM-130B, GPT-J, GPT-3.5-Turbo and OPT-125M model for the unseen-model setting evaluation. The test sizes for the leave-one-out scenarios are listed in Table 5.

M4 Split. M4 is a multi-generator, multi-domain, and multilingual benchmark for black-box machine-generated text detection (Wang et al., 2024). We use the SemEval-2024 M4 Subtask A multilingual protocol with the multilingual train, development, and test splits. The development split is used for validation and calibration and is not merged into training. The training split spans nine sources, while the multilingual test split contains Arabic, German, Italian, and Outfox. The test-time generator labels include BLOOMZ, ChatGPT, Cohere, Davinci, Dolly, JAIS-30B, and LLaMA2 fine-tuned.

RAID Split. RAID is designed to evaluate detector robustness across domains, generators, decoding strategies, and adversarial perturbations (Dugan

MAGE CDCM			
Split	Total	Human	Machine
Train	319,071	93,318	225,753
Valid	56,792	28,799	27,993
Test	56,819	28,741	28,078

M4 Multilingual			
Split	Total	Human	Machine
Train	172,417	83,846	88,571
Valid	4,000	2,000	2,000
Test	42,378	20,238	22,140

RAID			
Split	Total	Human	Machine
Train	303,247	8,571	294,676
Valid	33,694	952	32,742
Test	112,317	3,232	109,085

Table 4: Split sizes after preprocessing. Each panel reports the total number of examples and the class-wise counts for one evaluation setting. MAGE CDCM denotes the cross-domain cross-model setting of MAGE. For M4, Valid corresponds to the official development split and is not merged into training. For RAID, Train and Valid are obtained by a deterministic 90%/10% split of the processed RAID training data.

MAGE Unseen Domains			
Domain	Total	Human	Machine
CMV	4,917	2,403	2,514
ELI5	6,351	3,156	3,195
HellaSwag	6,347	3,292	3,055
ROC	6,462	3,275	3,187
SciGen	4,789	2,538	2,251
SQuAD	5,004	2,508	2,496
TLDR	4,977	2,535	2,442
WP	6,236	3,099	3,137
XSum	6,537	3,283	3,254
Yelp	5,199	2,652	2,547

MAGE Unseen Models			
Generator Model Family	Total	Human	Machine
GLM-130B	1,838	919	919
LLaMA	7,420	3,710	3,710
BigScience	5,386	2,693	2,693
FLAN-T5	9,320	4,660	4,660
OpenAI	13,284	6,642	6,642
EleutherAI	2,884	1,442	1,442
OPT	16,024	8,012	8,012

Table 5: Test sizes for the MAGE leave-one-out evaluations. The left panel lists leave-one-domain-out scenarios, and the right panel lists leave-one-generator-family-out scenarios. Each row corresponds to an independently trained leave-one-out evaluation scenario.

et al., 2024). In our experiments, we use the RAID split preprocessed by Zeng et al. (2025). We apply a 90%/10% split to the processed RAID training data, stratified by binary label. The resulting train split is used for fitting, valid is used for checkpoint selection and threshold calibration, and the processed RAID test split is used for final evaluation.

RAID covers eight domains: abstracts, books, news, poetry, recipes, Reddit, reviews, and Wiki. The machine-generated side spans several generator families, including GPT-series, Cohere, LLaMA, Mistral, and MPT variants, alongside human-written text. The train and validation splits contain clean examples, while the final test split contains both clean and perturbed examples. The perturbations include paraphrase, homoglyph, whitespace, zero-width-space, synonym, and spelling or formatting perturbations. Thus, RAID evaluates test-time robustness to adversarial and decoding-related variation, while model selection is performed only on clean validation data.

MAGE Leave-one-out Settings. For MAGE Unseen Domains and MAGE Unseen Models, each leave-one-out scenario trains a separate detector. Aggregate metrics are computed as arithmetic means over scenario-level results:

$$m_{\mathrm{UD}} = \frac{1}{10} \sum_{d \in \mathcal{D}_{\mathrm{UD}}} m_d, \qquad m_{\mathrm{UM}} = \frac{1}{7} \sum_{g \in \mathcal{G}_{\mathrm{UM}}} m_g.$$

Thus, MAGE leave-one-out results are scenario-wise macro averages. All metrics in Appendix C are reported as percentages.

B Ablation Details

This section provides additional details for the ablation studies in Section 4.3.

Prototype Repulsion Variant. For the loss ablation, we compare Prototype Positioning with a direct same-class prototype repulsion loss. Since all prototypes are normalized, inner products correspond to cosine similarity. The repulsion loss is defined as

$$\mathcal{L}_{\mathrm{rep}} = \frac{1}{2R(R-1)} \sum_{c=0}^{1} \sum_{r=1}^{R} \sum_{k \neq r} \left[\max(0, p_{c,r}^{\top} p_{c,k}) \right]^2 . \tag{16}$$

This loss penalizes same-class prototype pairs only when their cosine similarity is positive. It therefore encourages same-class prototypes to move apart without using assignment-weighted sample directions.

Prototype Positioning without Residuals. We also evaluate a raw-space variant of Prototype Positioning. Instead of removing the class-hub direction from samples and prototypes, this variant forms the prototype-indexed aggregate directly in the original embedding space:

$$g_{c,r} = \sum_{i \in B_c} q_{i,r} z_i, \qquad \bar{z}_{c,r} = \mathrm{norm}(g_{c,r}). \quad (17)$$

The raw-space positioning loss is defined as

$$\mathcal{L}_{\text{PP-raw}} = -\frac{1}{2R} \sum_{c=0}^{1} \sum_{r=1}^{R} \log \frac{\phi(\bar{z}_{c,r}, p_{c,r})}{\sum_{p \in \mathcal{P}} \phi(\bar{z}_{c,r}, p)}. \quad (18)$$

This variant tests whether prototype positioning remains effective without separating the class-level direction from the within-class variation.

Encoder Backbone Ablation. For the encoder ablation, we compare off-the-shelf backbones from the SimCSE, RoBERTa, BERT, E5, and BGE families (Gao et al., 2021; Liu et al., 2019; Devlin et al., 2019; Wang et al., 2022; Xiao et al., 2024). All variants use the same prototype configuration except for the encoder backbone. The evaluated encoders are unsupervised SimCSE-RoBERTa, RoBERTa-base, supervised SimCSE-RoBERTa, E5-base, BERT-base-uncased, unsupervised SimCSE-BERT-base, and BGE-base-en-v1.5.

C Full Results

We provide full results for all methods, including aggregate evaluation settings and detailed MAGE leave-one-out scenarios. Each table reports AvgRec, HumanRec, MachineRec, F1, Accuracy, AUROC, AUPR, and FPR95. MAGE-UD and MAGE-UM denote MAGE Unseen Domains and MAGE Unseen Models, respectively.

Setting	Scenario	AvgRec	HumanRec	MachineRec	F1	Acc	AUROC	AUPR	FPR95↓
Aggregate settings									
MAGE CDCM	Mixed	90.79	83.18	98.40	90.05	90.71	98.36	98.52	4.78
M4	Language shift	76.68	54.56	98.80	70.00	77.67	95.80	95.55	18.63
RAID	Adversarial/decoding	86.81	74.60	99.02	71.80	98.31	89.48	77.32	93.90
MAGE-UD	Average	67.99	37.04	98.94	51.14	67.55	89.30	90.79	56.25
MAGE-UM	Average	89.78	85.85	93.71	89.41	89.78	97.03	96.93	12.37
MAGE Unseen Domains									
UD	ROC	53.78	7.85	99.72	14.52	53.16	88.75	88.25	46.38
UD	HellaSwag	66.87	37.24	96.50	53.02	65.76	91.04	89.00	28.25
UD	XSum	55.86	12.85	98.86	22.55	55.67	70.42	74.85	96.22
UD	Yelp	67.70	35.78	99.61	52.56	67.05	91.23	93.50	69.61
UD	TLDR	61.80	24.54	99.06	39.12	61.10	94.20	93.10	19.25
UD	SciGen	69.66	40.39	98.93	57.15	67.91	93.65	94.50	24.21
UD	WP	75.59	51.73	99.46	67.94	75.74	87.75	92.07	91.81
UD	CMV	79.57	60.13	99.01	74.62	80.01	91.82	94.42	81.50
UD	SQuAD	67.68	35.49	99.88	52.34	67.61	89.16	92.44	83.20
UD	ELI5	81.43	64.45	98.40	77.62	81.53	94.94	95.79	22.10
MAGE Unseen Models									
UM	LLaMA	91.33	85.74	96.93	90.82	91.33	98.17	97.84	7.12
UM	FLAN-T5	85.04	83.61	86.48	84.82	85.04	92.67	92.62	33.43
UM	EleutherAI	92.02	84.26	99.79	91.35	92.02	99.37	99.57	0.76
UM	GLM-130B	91.62	86.07	97.17	91.13	91.62	98.25	98.32	6.31
UM	OPT	90.88	86.50	95.26	90.46	90.88	97.60	97.31	9.67
UM	BigScience	89.77	82.14	97.40	88.92	89.77	97.84	97.88	8.76
UM	OpenAI	87.78	92.65	82.91	88.35	87.78	95.28	95.00	20.52

Table 6: Full results for Binary CE.

Setting	Scenario	AvgRec	HumanRec	MachineRec	F1	Acc	AUROC	AUPR	FPR95↓
Aggregate settings									
MAGE CDCM	Mixed	94.77	92.98	96.56	94.71	94.75	95.06	96.79	31.27
M4	Language shift	83.84	72.37	95.30	81.54	84.35	89.23	92.21	72.72
RAID	Adversarial/decoding	77.67	58.94	96.39	42.00	95.32	78.04	45.77	88.26
MAGE-UD	Average	75.11	56.82	93.41	67.01	74.84	75.15	83.68	84.16
MAGE-UM	Average	90.92	93.49	88.34	91.40	90.92	91.89	93.90	32.59
MAGE Unseen Domains									
UD	ROC	55.72	12.06	99.37	21.41	55.12	55.71	75.80	94.35
UD	HellaSwag	72.64	52.58	92.70	65.99	71.89	72.77	82.82	90.23
UD	XSum	65.99	35.58	96.40	51.14	65.86	66.00	79.24	92.52
UD	Yelp	50.08	37.22	62.94	43.07	49.82	49.15	51.03	95.43
UD	TLDR	72.57	47.14	97.99	63.24	72.09	72.68	85.15	90.73
UD	SciGen	84.35	72.66	96.05	82.49	83.65	84.52	91.29	82.44
UD	WP	92.04	85.87	98.21	91.51	92.08	92.13	95.49	65.25
UD	CMV	91.64	86.27	97.02	91.10	91.76	91.98	95.07	64.68
UD	SQuAD	79.92	60.89	98.96	75.20	79.88	79.99	89.49	87.35
UD	ELI5	86.17	77.92	94.43	84.90	86.22	86.53	91.35	78.62
MAGE Unseen Models									
UM	LLaMA	93.48	93.29	93.67	93.46	93.48	94.18	95.78	30.22
UM	FLAN-T5	80.98	94.25	67.70	83.21	80.98	85.20	87.79	41.14
UM	EleutherAI	96.05	92.30	99.79	95.89	96.05	96.08	97.99	35.18
UM	GLM-130B	94.29	92.38	96.19	94.18	94.29	94.51	96.31	36.86
UM	OPT	93.26	93.58	92.94	93.28	93.26	93.82	95.35	27.57
UM	BigScience	93.15	92.94	93.35	93.13	93.15	93.39	95.07	33.84
UM	OpenAI	85.21	95.69	74.72	86.61	85.21	86.06	88.98	23.31

Table 7: Full results for SupCon.

Setting	Scenario	AvgRec	HumanRec	MachineRec	F1	Acc	AUROC	AUPR	FPR95↓
Aggregate settings									
MAGE CDCM	Mixed	94.84	91.87	97.81	94.71	94.80	95.81	97.36	23.07
M4	Language shift	92.74	87.92	97.57	92.27	92.96	92.74	95.38	59.62
RAID	Adversarial/decoding	87.68	80.17	95.20	46.85	94.77	91.08	62.01	67.70
MAGE-UD	Average	76.72	55.75	97.69	67.09	76.42	79.49	88.39	82.17
MAGE-UM	Average	91.69	92.08	91.30	91.85	91.69	92.87	94.61	27.14
MAGE Unseen Domains									
UD	ROC	53.43	7.08	99.78	13.20	52.80	54.32	75.86	94.53
UD	HellaSwag	78.62	66.01	91.23	75.81	78.15	82.84	87.35	81.69
UD	XSum	61.16	24.03	98.28	38.23	60.99	64.78	78.90	92.83
UD	Yelp	77.81	56.30	99.33	71.74	77.38	80.09	89.81	87.32
UD	TLDR	70.70	42.17	99.22	59.01	70.16	73.52	86.22	90.49
UD	SciGen	80.49	63.83	97.16	76.74	79.49	83.38	90.87	83.99
UD	WP	88.55	78.28	98.82	87.23	88.61	91.12	95.10	70.38
UD	CMV	90.66	82.40	98.93	89.80	90.85	92.74	95.86	63.16
UD	SQuAD	78.77	58.33	99.20	73.31	78.72	82.38	90.72	85.63
UD	ELI5	86.99	79.02	94.96	85.84	87.04	89.74	93.26	71.69
MAGE Unseen Models									
UM	LLaMA	93.77	93.58	93.96	93.76	93.77	95.41	96.27	6.73
UM	FLAN-T5	86.63	90.88	82.38	87.18	86.63	87.98	90.19	41.95
UM	EleutherAI	95.46	90.98	99.93	95.25	95.46	96.46	98.21	28.71
UM	GLM-130B	93.58	90.32	96.84	93.36	93.58	94.63	96.43	38.17
UM	OPT	93.32	93.00	93.65	93.30	93.32	94.24	95.53	16.50
UM	BigScience	93.32	90.61	96.03	93.13	93.32	94.38	96.07	35.10
UM	OpenAI	85.77	95.21	76.33	87.00	85.77	87.01	89.55	22.82

Table 8: Full results for DeTeCtive.

Setting	Scenario	AvgRec	HumanRec	MachineRec	F1	Acc	AUROC	AUPR	FPR95↓
Aggregate settings									
MAGE CDCM	Mixed	94.43	95.18	93.67	94.54	94.44	98.47	98.56	6.15
M4	Language shift	81.08	63.13	99.03	76.90	81.89	95.20	96.32	12.53
RAID	Adversarial/decoding	86.17	76.36	95.97	48.89	95.41	87.47	51.98	96.52
MAGE-UD	Average	79.08	63.46	94.71	72.57	78.90	92.11	92.02	33.25
MAGE-UM	Average	90.71	95.22	86.19	91.38	90.71	96.71	96.79	13.33
MAGE Unseen Domains									
UD	ROC	59.50	20.06	98.93	33.13	58.96	87.08	86.51	45.09
UD	HellaSwag	83.92	80.01	87.82	83.65	83.77	92.36	90.10	23.54
UD	XSum	62.12	28.21	96.04	42.69	61.97	83.12	82.25	63.92
UD	Yelp	81.10	66.52	95.68	77.95	80.80	91.98	93.14	48.02
UD	TLDR	70.76	45.05	96.48	60.70	70.28	87.36	87.99	58.11
UD	SciGen	85.64	78.17	93.11	84.84	85.20	95.24	95.45	18.84
UD	WP	86.65	76.93	96.37	85.19	86.71	97.27	97.19	11.57
UD	CMV	89.23	84.14	94.31	88.53	89.34	96.60	96.71	16.91
UD	SQuAD	83.67	70.33	97.00	81.16	83.63	95.26	95.62	25.00
UD	ELI5	88.27	85.17	91.36	87.84	88.29	94.86	95.24	21.56
MAGE Unseen Models									
UM	LLaMA	94.30	95.66	92.94	94.38	94.30	98.35	98.34	6.44
UM	FLAN-T5	79.39	94.87	63.91	82.15	79.39	91.83	91.91	36.52
UM	EleutherAI	97.16	95.08	99.24	97.10	97.16	99.04	99.34	0.55
UM	GLM-130B	94.72	95.10	94.34	94.74	94.72	98.40	98.52	5.66
UM	OPT	91.20	94.22	88.18	91.46	91.20	96.92	97.01	13.27
UM	BigScience	92.65	94.65	90.64	92.79	92.65	97.44	97.55	9.84
UM	OpenAI	85.54	96.99	74.09	87.02	85.54	94.96	94.85	21.02

Table 9: Full results for DSVDD.

Setting	Scenario	AvgRec	HumanRec	MachineRec	F1	Acc	AUROC	AUPR	FPR95↓
Aggregate settings									
MAGE CDCM	Mixed	95.14	95.81	94.47	95.23	95.15	98.41	98.37	4.89
M4	Language shift	86.03	76.87	95.20	84.42	86.45	92.35	93.08	57.00
RAID	Adversarial/decoding	88.18	82.52	93.84	42.25	93.51	95.41	71.41	27.78
MAGE-UD	Average	78.59	61.46	95.72	71.13	78.34	86.90	89.10	59.41
MAGE-UM	Average	91.34	95.63	87.05	91.91	91.34	96.48	96.16	11.44
MAGE Unseen Domains									
UD	ROC	56.68	13.92	99.44	24.33	56.10	86.11	86.41	55.26
UD	HellaSwag	75.84	59.51	92.18	71.37	75.23	78.73	82.41	89.89
UD	XSum	61.29	25.68	96.90	39.89	61.13	68.73	72.74	92.58
UD	Yelp	83.29	71.68	94.90	81.19	83.05	88.71	91.76	78.03
UD	TLDR	75.06	53.89	96.23	68.42	74.66	81.49	86.25	87.85
UD	SciGen	83.82	72.89	94.76	82.11	83.17	95.89	95.21	12.67
UD	WP	91.29	85.25	97.32	90.71	91.32	96.67	97.32	16.67
UD	CMV	90.30	85.06	95.54	89.67	90.42	95.56	96.41	22.37
UD	SQuAD	80.18	62.76	97.60	76.00	80.14	85.27	89.54	80.55
UD	ELI5	88.17	84.00	92.33	87.61	88.19	91.87	92.99	58.20
MAGE Unseen Models									
UM	LLaMA	94.20	96.01	92.40	94.31	94.20	98.36	97.77	6.45
UM	FLAN-T5	83.12	96.12	70.13	85.06	83.12	93.13	91.96	26.07
UM	EleutherAI	96.78	94.24	99.31	96.69	96.78	99.07	99.32	0.96
UM	GLM-130B	94.23	95.32	93.14	94.29	94.23	96.47	96.52	6.42
UM	OPT	93.02	94.88	91.15	93.14	93.02	96.48	96.32	9.09
UM	BigScience	93.00	95.95	90.05	93.20	93.00	97.68	97.71	8.47
UM	OpenAI	85.05	96.90	73.20	86.63	85.05	94.19	93.50	22.63

Table 10: Full results for MD-ProTector.