Steer-to-Detect: Probing Hidden Representations for Detection of LLM-Generated Texts

Luxu Liang
School of Economics and Management
and Shenzhen International Graduate School
Tsinghua University
liang-lx25@mails.tsinghua.edu.cn

Xiang Li
Perelman School of Medicine
University of Pennsylvania
lx10077@upenn.edu

Abstract

The rapid advancement of large language models (LLMs) has made machine-generated text increasingly difficult to distinguish from human-written text. While recent studies explore leveraging internal representations of language models to uncover deeper detection signals, these raw features often exhibit substantial overlap between classes, limiting their discriminative power. To address this challenge, we propose Steer-to-Detect (S2D), a two-stage framework for detecting LLM-generated text. In the first stage, S2D learns a steering vector that is injected into the hidden states of a frozen observer LLM, producing representations with improved class separability. In the second stage, detection is performed via a hypothesis testing procedure based on the steered representations. We establish finite-sample, high-probability guarantees for Type I and Type II errors, providing a theoretical characterization of the procedure. Empirically, S2D achieves strong and consistent performance across a range of settings, including out-of-distribution scenarios and adversarial perturbations.

1 Introduction

Large language models (LLMs) are increasingly deployed across a broad range of domains, including education, finance, legal services, customer support, and writing assistance [1–5]. Despite their widespread adoption, their use has raised growing societal concerns, including the spread of misinformation, academic misconduct, and the erosion of trust in written content [6–9]. A key reason is that LLM-generated text often closely resembles human writing, making it difficult to distinguish between the two. This challenge highlights the need for reliable detection of LLM-generated text.

Early work on this problem suggests that as long as distributional differences exist, reliable detection is possible in principle, particularly with sufficiently long samples [10]. Building on this observation, a prominent line of work seeks to explicitly introduce such differences during generation, most notably through watermarking methods that embed structured patterns identifiable with statistical guarantees [11, 12]. While effective under controlled settings, these approaches rely on access to the generation process and are not applicable when watermarking is absent or not widely adopted. This limitation motivates passive methods, which aim to infer the origin of a text directly from observed samples without any control over the generation process. A detailed taxonomy is provided in Section 1.1.

A key challenge in the passive setting is that the distributional differences between human-written and LLM-generated text can be subtle, particularly for short or moderate-length samples. This raises a natural question: rather than relying solely on these weak signals, can we amplify them to improve detectability? Recent studies suggest that hidden representations of LLMs encode behavior-relevant information that is not fully captured by final-layer outputs [13–17]. These hidden representations provide a richer feature space in which the differences between human- and machine-generated

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

text may become more pronounced. This observation motivates our central question: ***Can we reshape LLM hidden representations to amplify such differences and enable reliable detection of machine-generated text?***

Our Contributions. In this paper, we propose *Steer-to-Detect* (S2D), a novel method for detecting LLM-generated text. At a high level, S2D employs a surrogate model as an "observer" and intervenes in its representation extraction process. Specifically, during the forward pass on input text, we steer hidden representations by adding a universal vector, thereby modifying how representations are formed. The vector is learned to enhance the separability between human-written and LLM-generated text. This approach avoids expensive model fitting and extends to other binary detection tasks.

On the theoretical side, we establish finite-sample control of the Type I (or false positive) error and derive an explicit upper bound on the excess Type II (or false negative) error. In addition, we provide a quantitative characterization of robustness, demonstrating the reliability of the detector in risk-sensitive settings.

Empirically, we conduct extensive experiments and show that S2D achieves strong and consistent performance across both in-distribution (ID) and out-of-distribution (OOD) settings. Further analyses demonstrate robustness to paraphrasing, adversarial perturbations, and variations in input length. In short, we propose a simple yet effective approach that enhances detectability by steering hidden representations.

1.1 Related Works

LLM-generated Text Detection. Active methods modify the generation process and are beyond the scope of this work [29–31], so we focus on passive detection. As summarized in Table 1, passive detectors can be organized along two axes: whether labeled training data are required and whether detection relies on auxiliary rewritten or perturbed variants of the input text. This yields four broad categories: train-free rewrite-based [10, 18], train-free non-rewrite-based [19–21], train-based rewrite-based [22, 24], and train-based non-rewrite-based methods [25–28]. Rewrite-based methods incur additional cost due to auxiliary text generation, whereas non-rewrite-based methods avoid this overhead. Our method falls into the *train-based, non-rewrite-based* category. Prior work in this setting typically relies on final-layer outputs such as logits or derived scores, leaving intermediate representations underexplored [16]. More recent studies have begun to use hidden representations for detection [32, 33], with RepreGuard [33] showing that hidden representations can provide stronger signals than final-layer outputs. Unlike these methods, we do not treat representations as fixed features. Instead, we shape their geometry to better separate human-written and LLM-generated text.

Table 1: Taxonomy of passive detection.

	Rewrite	**Non-rewrite**
Train-free	[10, 18]	[19–21]
Train-based	[22–24]	Ours & [25–28]

Representation Engineering. Representation engineering studies how information encoded in the hidden states of LLMs can be analyzed, interpreted, and manipulated to understand, monitor, or control model behavior [34, 13, 35, 16, 36–42]. This line of work includes both methods for characterizing what hidden representations encode and intervention-based approaches that modify them. Among these, *activation steering* has emerged as a prominent technique that perturbs intermediate activations along task-specific directions (often represented as steering vectors) in representation space, enabling targeted behavioral changes without updating model parameters [43, 44]. Recent studies apply activation steering to improve factual reliability, control generation style, and monitor model behaviors such as hallucinations and toxic content using internal representations [45–49, 41]. Despite these advances, the use of representation-level interventions for LLM-generated text detection remains relatively underexplored. In this work, we repurpose activation steering to reshape the geometry of hidden representations, transforming raw hidden states into features that better separate human-written and LLM-generated text.

2 Our Method: Steer-to-Detect (S2D)

Problem Formulation. Let $\mathcal{X}$ denote the space of all text sequences (possibly of varying length). We assume that human-written and LLM-generated texts are drawn from two distributions, $\mathbb{P}_0$ and

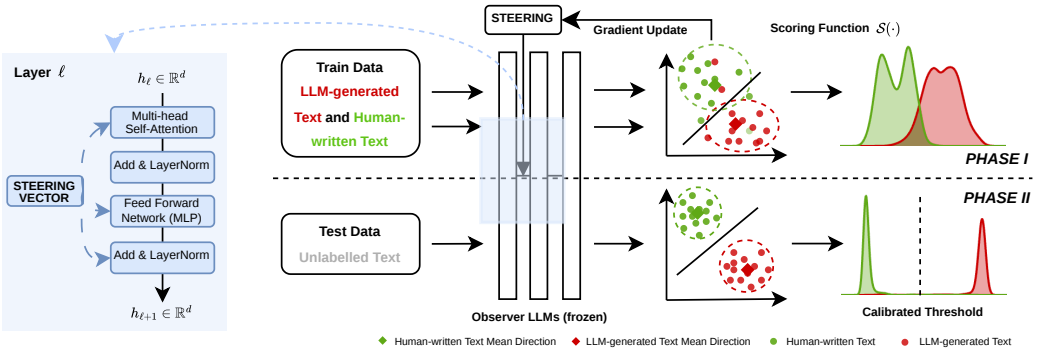


Figure 1: **Overview of Steer-to-Detect** (S2D)**.** *Phase I* (top row) applies a steering vector to reshape the observer LLM's hidden representations, enhancing the separation between human-written and LLM-generated text. *Phase II* (bottom row) scores unseen texts and rejects the null hypothesis when the score exceeds a calibrated threshold.

$\mathbb{P}_1$, respectively. Given an observation $x \in \mathcal{X}$, detection is formulated as the binary hypothesis test:

$$\mathcal{H}_0 : x \sim \mathbb{P}_0 \quad \text{(human-written)} \quad \textit{versus} \quad \mathcal{H}_1 : x \sim \mathbb{P}_1 \quad \text{(LLM-generated)}. \tag{1}$$

2.1 Method Overview

We begin with an overview of our method. We employ a surrogate model as an "observer" and extract hidden representations during the forward pass on input text. Representations of LLM-generated and human-written text exhibit systematic differences, which, although subtle, provide a sufficient signal for detection [33]. Motivated by this observation, we intervene in the representation formation process by injecting a lightweight, learnable steering vector $\mathbf{v}$ into intermediate hidden states during the forward pass. This modifies how representations are formed and enhances their separability between the two types of text. Based on the resulting representations, we construct a hypothesis test for detection. The overall framework is illustrated in Figure 1.

More specifically, let $f_\theta : \mathcal{X} \to \mathbb{S}^{d-1}$ denote the representation map of an observer model, which embeds a text sequence into a d-dimensional vector on the unit hypersphere (to be specified later). To enable intervention in the representation space, we introduce a steering vector $\mathbf{v} \in \mathbb{R}^d$ and denote the resulting steered representation map by $f_{\theta,\mathbf{v}}$.

Our method proceeds in two stages, as introduced earlier. In the first stage, we estimate the steering vector $\mathbf{v}$ from training data $\mathcal{S}_{\text{train}} = \{(x_i, y_i)\}_{i=1}^{n_1}$ and fit parametric class-conditional models to the steered representations. In the second stage, we construct a likelihood-ratio statistic based on the estimated models and use it to test whether a given text is human-written or LLM-generated. We next describe (i) the computation of f_θ and the introduction of $\mathbf{v}$ in Section 2.2, (ii) the learning of $\mathbf{v}$ in Section 2.3, and (iii) the detection procedure in Section 2.4.

2.2 Representation Extraction and Steering

Representation Extraction. We define the mapping f_θ by extracting hidden states from an observer model with L Transformer blocks and hidden dimension d. For a sequence $x \in \mathcal{X}$, let $h_{\ell,t}(x) \in \mathbb{R}^d$ denote the hidden state at layer $\ell (\leq L)$ and token index t. Prior work [50, 13, 16, 33] suggests that tokens near the end of a sequence encode richer contextual information and exhibit more pronounced differences between human-written and LLM-generated text, as later tokens aggregate broader preceding context in causal LLMs. Motivated by this observation, we focus on the final $K \in (0, 1]$ fraction of valid (non-padding) tokens,[1] indexed by $\mathcal{I}_K(x)$. In addition, intermediate layers capture complementary information beyond the final layer [16], so we aggregate hidden states from the last N layers to obtain a more stable sequence-level representation. Mathematically, we define

$$f_\theta(x) = \frac{\bar{m}(x)}{\|\bar{m}(x)\|} \in \mathbb{S}^{d-1} \quad \text{with} \quad \bar{m}(x) := \frac{1}{N \cdot |\mathcal{I}_K(x)|} \sum_{\ell=L-N+1}^{L} \sum_{t \in \mathcal{I}_K(x)} h_{\ell,t}(x) \in \mathbb{R}^d,$$

where $|\mathcal{I}_K(x)|$ denotes the number of selected non-padding tokens.

[1] Non-padding tokens refer to tokens corresponding to actual text content, excluding padding tokens.

Remark 2.1 (Representation Norms and Directionality)**.** *In Appendix E, we provide empirical support for this modeling choice f_θ. In modern Transformer models, representation norms are often approximately uniform after root mean square normalization, leaving directional variation as the primary source of information [17, 51, 52]. For example, in meta-llama/Llama-3.1-8B, the mean norm is around 18.8 across four different datasets; see Figure 5 in the appendix for details. We also observe that $f_\theta(x) \mid y$ exhibits unimodal structure (see Figure 6 in the appendix), which motivates the use of the vMF model in deriving* (2)*.*

Representation Steering. Although the base extractor $f_\theta(\cdot)$ captures high-level semantic information, the resulting class-conditional distributions may still overlap in latent space, even in out-of-distribution settings (see Appendix F.1 for evidence).

To improve separability, we introduce a learnable steering vector $\mathbf{v} \in \mathbb{R}^d$ and define a steered extractor $f_{\theta,\mathbf{v}}(\cdot)$. We implement steering as an additive intervention on intermediate activations of the observer model by injecting a vector $\mathbf{v}$ at a chosen layer ℓ:

$$h_{\ell,t}(\cdot) \leftarrow h_{\ell,t}(\cdot) + \mathbf{v}.$$

In the default setting, the intervention is applied at an intermediate layer $\ell \le L - N + 1$. Importantly, the method is not sensitive to the exact choice of ℓ: as long as the intervention is applied within intermediate layers, performance remains stable (see Section 4.2).

This is the only modification to the model: after the injection, the observer model remains fixed, and the perturbed activations propagate through subsequent Transformer blocks without further intervention, producing a new representation. Intuitively, since the model captures rich semantic structure from large-scale pre-training, the steering vector $\mathbf{v}$ introduces a direction that enhances its intrinsic discriminative capability. Applying the same token selection, layer aggregation, and normalization yields the steered representation $f_{\theta,\mathbf{v}}(\cdot) \in \mathbb{S}^{d-1}$ for the downstream hypothesis test.

2.3 Phase I: Learning the Steering Vector via Likelihood Maximization

Likelihood-Based Modeling. A remaining question is how to learn the steering vector $\mathbf{v}$, which is addressed in Phase I of our method. The steering vector aims to enhance the separability of latent representations across classes. To quantify this, we formulate a classification problem where the feature is given by $f_{\theta,\mathbf{v}}$ and the response is the ground-truth label ($y_i = 1$ for LLM-generated text and $y_i = 0$ for human-written text). More specifically, we model the conditional distribution $f_{\theta,\mathbf{v}}(x_i) \mid y_i$ as a von Mises–Fisher (vMF) distribution, a directional analogue of a Gaussian on the unit sphere.[2] Under a shared-concentration vMF model, the class posterior takes a softmax form. Motivated by this structure, we use the same functional form for Phase-I optimization, with the inverse temperature $\beta > 0$. Specifically, under a uniform class prior, we define

$$q_\beta\left(y_i \mid f_{\theta,\mathbf{v}}(x_i), \boldsymbol{\mu}_0, \boldsymbol{\mu}_1\right) = \frac{\exp\left(\beta \boldsymbol{\mu}_{y_i}^\top f_{\theta,\mathbf{v}}(x_i)\right)}{\sum_{c\in\{0,1\}} \exp\left(\beta \boldsymbol{\mu}_c^\top f_{\theta,\mathbf{v}}(x_i)\right)}. \tag{2}$$

Here, $\boldsymbol{\mu}_c \in \mathbb{S}^{d-1}$ denotes the mean direction for class $c \in \{0,1\}$, while β is an optimization hyperparameter and is distinct from the shared vMF concentration parameter κ. Appendix B derives the corresponding softmax posterior under the vMF model and explains this temperature-scaled formulation. To estimate the steering vector, we collect a labeled training set $\mathcal{S}_{\text{train}} = \{(x_i, y_i)\}_{i=1}^{n_1}$ and optimize the following objective:

$$(\widehat{\mathbf{v}}, \widehat{\boldsymbol{\mu}}_0, \widehat{\boldsymbol{\mu}}_1) = \underset{\mathbf{v}\in\mathbb{R}^d, \boldsymbol{\mu}_0, \boldsymbol{\mu}_1 \in \mathbb{S}^{d-1}}{\arg\max} \frac{1}{n_1} \sum_{i=1}^{n_1} \log q_\beta\left(y_i \mid f_{\theta,\mathbf{v}}(x_i), \boldsymbol{\mu}_0, \boldsymbol{\mu}_1\right). \tag{3}$$

Optimization Procedure. In practice, we adopt a two-timescale optimization scheme to optimize (3). Specifically, the steering vector $\mathbf{v}$ is updated via gradient ascent on a slower timescale (i.e., using a smaller step size η), while the class mean directions $\boldsymbol{\mu}_c$ are updated on a faster timescale (i.e., using an exponential moving average (EMA) with coefficient $\rho \in (0,1)$) over normalized class-specific embeddings [53]. The observer model remains fixed throughout the optimization, and we choose $0 < \eta \ll \rho \le 1$ to introduce two time scales. The full procedure is provided in Appendix A due to space constraints.

[2] Under the vMF model, $p(f_{\theta,\mathbf{v}}(x_i) \mid y_i = c) \propto \exp\left(\kappa \boldsymbol{\mu}_c^\top f_{\theta,\mathbf{v}}(x_i)\right)$, where κ is the concentration parameter. The Phase-I objective retains the resulting softmax form but uses a separate inverse temperature β.

Remark 2.2 (Gradient analysis)**.** *Intuitively, this objective in* (3) *encourages the steering vector to reshape the representation space so that class-wise representations become more separable. In Appendix H.1, we characterize the gradient of the population version of* (3)*, which provides theoretical support for this effect.*

2.4 Phase II: Detection via the Log-Likelihood Ratio Test

Log-Likelihood Ratio Test. Detection naturally reduces to a log-likelihood ratio test. Under the shared-concentration vMF generative model, the Neyman-Pearson (NP) lemma [54] implies that the most powerful test is based on the log-likelihood ratio:

$$\mathcal{S}(x) = \log\left(\frac{p(f_{\theta,\mathbf{v}}(x) \mid y=1, \boldsymbol{\mu}_1, \boldsymbol{\mu}_0)}{p(f_{\theta,\mathbf{v}}(x) \mid y=0, \boldsymbol{\mu}_1, \boldsymbol{\mu}_0)}\right) = \kappa(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^\top f_{\theta,\mathbf{v}}(x). \tag{4}$$

Since $\kappa > 0$ only rescales the score, it does not affect the rejection region after threshold calibration. Thus, the detector is equivalently based on projecting the steered representation onto the discriminative direction $\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0$, with unscaled projection used in practice. In practice, we plug in the estimates $(\widehat{\mathbf{v}}, \widehat{\boldsymbol{\mu}}_0, \widehat{\boldsymbol{\mu}}_1)$ obtained from (3) to form the empirical score $\widehat{\mathcal{S}}(x)$. The resulting detector is $\widehat{\phi}(x) := \mathbb{1}\big(\widehat{\mathcal{S}}(x) > \widehat{\tau}\big)$, which rejects the null hypothesis $\mathcal{H}_0$ when the score exceeds the threshold $\widehat{\tau}$. Accordingly, x is classified as LLM-generated if $\widehat{\phi}(x) = 1$, and as human-written otherwise.

Threshold Calibration. In high-stakes scenarios such as academic integrity assessment, falsely labeling human-written text as LLM-generated corresponds to a Type-I error with substantial practical and ethical consequences. In such settings, controlling the false positive rate (FPR) is more important than optimizing a balanced trade-off between detection power and false alarms. Accordingly, we calibrate the threshold using an independent calibration sample $\mathcal{S}_{\text{cal}} = \{x_i^-\}_{i=1}^{n_2}$ drawn from the null distribution, and define

$$\widehat{\tau}_\alpha := \inf\left\{\tau : \frac{1}{n_2}\sum_{i=1}^{n_2} \mathbb{1}\big(\widehat{\mathcal{S}}(x_i^-) > \tau\big) \le \alpha\right\}. \tag{5}$$

This empirical calibration enforces the target Type-I error level on the calibration set. Moreover, Theorem 3.1 shows that the resulting test achieves approximate control of the population Type-I error in finite samples with high probability.

Remark 2.3 (Threshold from Youden index)**.** *In Appendix C, we describe an alternative threshold selection strategy based on the Youden index, which is commonly used in standard benchmarking settings [33, 55–57]. While this approach is not designed to control the Type I error, our empirical results indicate that it can nevertheless achieve competitive false positive control in practice.*

3 Theoretical Analysis

In this section, we provide a theoretical analysis of our method. We begin by establishing finite-sample guarantees for Type I error control, and then characterize the excess Type II error arising from parameter estimation. Finally, we study the effect of null distribution shifts on the resulting test. All proofs are deferred to Appendix G.

Type-I Error Control. We show that empirical threshold calibration ensures Type I error control with high probability. Let $\widehat{\mathcal{S}}_t$ denote the score function at training iteration t, trained on $\mathcal{S}_{\text{train}}$, and define the detector $\widehat{\phi}(x) := \mathbb{1}\big(\widehat{\mathcal{S}}(x) > \widehat{\tau}_\alpha\big)$, where $\widehat{\tau}_{\alpha,t}$ is the empirical threshold at iteration t, computed from an independent null sample $\mathcal{S}_{\text{cal}} = \{x_i^-\}_{i=1}^{n_2}$ as in (5). Theorem 3.1 shows that this procedure controls the Type I error up to a finite-sample deviation that vanishes as n_2 increases.

Theorem 3.1 (Type-I Error Control)**.** *Let $\alpha, \delta \in (0,1)$ and suppose that, conditional on the training sample, the null score $\widehat{\mathcal{S}}_t(X^-)$ has a continuous distribution. Then, with probability at least $1-\delta$ over the independent calibration sample, $\left|\mathbb{P}_0\left(\widehat{\phi}(X_{\text{test}}^-) = 1\right) - \alpha\right| \le \sqrt{\log(2/\delta)/(2n_2)} + 1/n_2$.*

Remark 3.2 (Ties)**.** *The continuity assumption ensures almost surely distinct calibration scores. With ties, one may conservatively exclude the boundary group for one-sided Type-I control or use randomized tie-breaking for exact empirical calibration.*

Excess Type-II Error. Next, we study the Type-II error. In general, analyzing the power of the detector is challenging due to the complex and implicit behavior of LLM-extracted representations. To facilitate analysis, we consider a simplified setting in which the extracted features $f_{\theta,\mathbf{v}}(X)$ follow a class-conditional vMF model with certain ground-truth parameters $\boldsymbol{\mu}_0$ and $\boldsymbol{\mu}_1$. This assumption allows us to define the corresponding oracle likelihood ratio test (i.e., $\mathcal{S}$ in (4)) based on this true model, and to study the excess Type-II error of the plug-in detector $\widehat{\mathcal{S}}$ relative to this oracle benchmark. In particular, it enables us to disentangle the contributions of estimation and calibration.

Importantly, the oracle and plug-in scores are evaluated at the same steering iterate $\mathbf{v}_t$ and operate on the same steered representation $f_{\theta,\mathbf{v}_t}(x)$. The following result therefore isolates prototype-estimation and threshold-calibration errors, rather than comparing $\mathbf{v}_t$ with a globally optimal steering vector or establishing global convergence of the steering optimization.

Theorem 3.3 (Excess Type-II Error (informal))**.** *Assume the extracted features $f_{\theta,\mathbf{v}}$ follow a vMF distribution, and the EMA coefficient ρ and learning rate η are chosen sufficiently small, satisfying $0 < \eta \ll \rho < 1$. Under regularity conditions, there exists a constant $c \in (0,1)$ such that, with probability at least $1-2\delta$, for any fixed $0 \le t \le T$, the gap $\mathbb{P}_1(\widehat{\phi}(X^+_{\text{test}}) = 0) - \mathbb{P}_1(\phi(X^+_{\text{test}}) = 0)$ is bounded by*

$$\mathcal{O}\Bigg(\underbrace{\left((1-c\rho)^t + \sqrt{\rho \log \frac{4T}{\delta}} + \frac{\eta^2}{\rho^2} \right)^{\frac{1+\bar{\gamma}}{2}}}_{\textit{Estimation Error}} + \underbrace{\sqrt{\frac{\log(2/\delta)}{n_2} + \frac{1}{n_2}}}_{\textit{Calibration Error}} \Bigg)$$

where n_2 is the calibration sample size and $\bar{\gamma}$ characterizes the local concentration of the score distribution (i.e., $\mathcal{S}$ under $\mathbb{P}_0$) around the threshold $\tau^\star_\alpha$.[3]

Theorem 3.3 establishes an upper bound on the excess Type-II error, which characterizes the sub-optimality of the learned score function $\widehat{\mathcal{S}}$. The bound consists of two components: an estimation error arising from learning the ground-truth parameters $\boldsymbol{\mu}_0$ and $\boldsymbol{\mu}_1$, and a calibration error due to estimating the critical value $\widehat{\tau}_{\alpha,t}$.

The second term follows from Theorem 3.1 and reflects the finite-sample effect of threshold calibration; it decreases as the calibration sample size n_2 increases. The first term can be further decomposed into several parts: an optimization term exhibiting geometric decay $(1-c\rho)^t$, a variance term induced by stochastic training $\sqrt{\rho \log(4T/\delta)}$, and a higher-order lag term η^2/ρ^2. When t is sufficiently large and the two-timescale condition $\eta \ll \rho < 1$ holds with ρ sufficiently small, the estimation error becomes negligible.

Effects of Distributional Shift. Prior work has highlighted performance degradation under domain shifts in LLM-generated text detection [58, 59], and proposed database-based strategies to mitigate such a mismatch. However, these approaches do not explicitly characterize the impact of distribution shift. In Appendix H.2, we quantify this effect by analyzing the detector under a Wasserstein perturbation with magnitude $\mathcal{E}$. Proposition H.2 provides finite-sample bounds for both Type-I and Type-II errors under the shifted distributions. In particular, when $\mathcal{E} > 0$, both errors may deteriorate, reflecting the impact of distribution mismatch.

4 Experiment

In this section, we present a comprehensive evaluation of the proposed `S2D` detector. Our code is available at: `https://github.com/LuxLiang/steer-to-detect-release/tree/main`.

Datasets. Following previous setups [33, 60], we evaluate `S2D` on the DetectRL benchmark [61], which covers four domains that are particularly vulnerable to misuse by LLMs, including academic writing (arXiv Archive[4]), news summarization (XSum [62]), creative writing (WritingPrompts [63]), and user reviews (Yelp [64]). For each domain, the benchmark provides 2,800 pairs of human-written and LLM-generated texts. The LLM-generated texts are produced by four widely used LLMs: *GPT-3.5-Turbo* [65], *Claude-Instant* [66], *Google-PaLM* [67], and *Llama-2-70B* [68]. To ensure robust

[3] A larger value of $\bar{\gamma}$ corresponds to less mass near the threshold and hence an easier estimation problem. See Assumption 3 for the detailed definition.

[4] `https://www.kaggle.com/datasets/spsayakpaul/arxiv-paper-abstracts/data.`

evaluation, we repeat the experiment over five independent runs with different random samplings of the training data. In each run, we randomly sample 512 human-written/LLM-generated text pairs to form the training set, while using a fixed, disjoint test set of 1,000 pairs. We further evaluate on the recent OpAI-Bench [69], which contains progressive human-to-AI revision trajectories across four domains and includes texts revised by more recent generators, including *GPT-5.4*, *GPT-5.4-nano*, *Gemini-2.5-Flash*, and *Qwen3-8B*. Full experimental details and results are provided in Appendix F.8.

Baselines. In our primary DetectRL evaluation, we compare S2D against thirteen baselines across train-free and train-based categories. The train-free baselines include: *Likelihood* [70], Log Rank Ratio (*LRR*) [70], Fast-DetectGPT (*FDGPT*) [20], *Binoculars* [21], Implicit Reward Model (*IRM*) [71], *Lastde++* [72], *SpecDetect++* [73], and Normalized Temperature Sensitivity (*NTS*) [74]. The train-based baselines consist of: *RoBERTa-Large* [25], *RAIDAR* [22], Imitate Before Detection (*ImBD*) [75], *RepreGuard* [33], and Learn-to-Distance (*L2D*) [23]. We further conduct controlled comparisons against logistic regression, a linear SVM, and a small MLP trained on the same pooled observer representations without steering, to assess whether a standard discriminative readout alone can account for the observed performance. The complete comparisons are reported in Appendix F.8.

Evaluation Metrics. We evaluate detection performance using AUROC and the true positive rate (TPR) at two fixed false positive rate (FPR) levels, namely TPR@1% and TPR@0.01%[5]. These metrics capture both overall discrimination and performance in the high-precision regime where false positives must be tightly controlled. Further experimental details are provided in Appendix D.

4.1 Detection Performance and Robustness

Detection under ID and OOD settings. We first evaluate detection performance under both in-distribution (ID) and out-of-distribution (OOD) settings, as summarized in Table 2. The detector is trained on text generated by a specific source model and evaluated either on the same generator (ID, diagonal entries) or on different generators (OOD, off-diagonal entries).

Under the ID setting, S2D performs strongly across all generators when the training and test distributions are aligned. RepreGuard and L2D are the most competitive baselines, often achieving TPRs above 99% at low FPR, while other training-based methods are less consistent across generators. L2D relies on rewriting, whereas S2D uses a single forward pass of the frozen observer model at inference time; efficiency comparisons are provided in Appendix F.6. In the OOD setting, performance gaps become more pronounced, and S2D remains competitive against most baselines.

Robustness to Adversarial Attacks. We then evaluate robustness under adversarial manipulations, including three paraphrasing attacks (i.e., *Polish*, *Back Translation*, *DIPPER*) and three perturbation attacks (i.e., *Character*, *Word*, and *Character+Word*), following the previous setup [20]. As shown in Table 3, paraphrasing attacks substantially degrade all detectors, highlighting the challenge of semantic-preserving rewrites. Nevertheless, S2D is highly competitive and often achieves the best or near-best performance, whereas baseline methods experience severe degradation. This robustness stems from S2D's representation reshaping, which enforces a more discriminative feature space and preserves separability even under attacks. In contrast, perturbation-based attacks have a much milder impact, as they primarily introduce local or surface-level noise.

Robustness to Text Length. Our main test set has an average length of approximately 267 tokens, with 95% of samples shorter than 435 tokens. Figure 2a shows that detection generally benefits from additional context within this range. We further examine substantially longer sequences, up to 4096 tokens, using a separate diagnostic set. Performance continues to improve with context length before plateauing, while the score overlap between human and LLM-generated text remains low. In particular, we observe no evidence that repeatedly injecting the same steering vector causes representation collapse or performance degradation within the evaluated range. Full long-context results are reported in Appendix F.4.

Comparison with Watermarking Methods. An interesting question is whether S2D can outperform existing watermarking methods. Although this comparison is inherently unfair—since

[5]Since the DetectRL test sets contain 1,000 human-written samples, TPR@0.01% is below the empirical FPR resolution and is obtained by linear interpolation of the ROC curve. We further examine the accuracy of this interpolation using a substantially larger null set in Appendix F.2.

Train ↓	Detector ↓	Test: ChatGPT			Test: Llama-2-70B			Test: Google-PaLM			Test: Claude-Instant		
		AUROC	*TPR@1%*	*TPR@.01%*	*AUROC*	*TPR@1%*	*TPR@.01%*	*AUROC*	*TPR@1%*	*TPR@.01%*	*AUROC*	*TPR@1%*	*TPR@.01%*
Train-free Methods													
N/A	Likelihood	89.91	73.00	0.40	89.32	80.30	5.80	85.73	69.30	13.10	76.72	5.50	0.40
	LRR	88.95	78.00	0.01	89.51	81.50	38.80	85.66	73.70	21.30	76.75	7.00	0.20
	FDGPT	94.70	51.90	0.06	96.85	70.10	32.90	95.26	66.00	50.40	58.37	4.10	0.70
	Binoculars	94.71	90.67	87.00	99.15	96.70	88.90	94.50	91.40	89.30	87.70	37.50	13.70
	IRM	99.08	90.70	83.10	99.20	98.20	95.50	99.11	91.30	85.20	95.50	69.20	35.10
	Lastde++	94.32	53.10	4.27	96.07	65.77	20.30	94.85	63.07	47.80	69.38	10.23	2.33
	SpecDetect++	94.19	38.90	3.07	94.76	54.33	2.60	89.46	42.30	15.07	69.90	5.70	0.40
	NTS	93.48	56.60	6.00	96.93	76.40	57.30	94.75	71.60	36.80	66.72	8.90	2.30
Train-based Methods													
ChatGPT	RoBERTa-L	$98.99_{\pm.12}$	$98.90_{\pm.25}$	$98.10_{\pm.45}$	$98.82_{\pm.15}$	$63.70_{\pm1.5}$	$44.80_{\pm2.1}$	$98.08_{\pm.22}$	$45.60_{\pm2.0}$	$25.40_{\pm2.5}$	**$98.95_{\pm.18}$**	$76.90_{\pm1.6}$	$15.80_{\pm1.5}$
	RAIDAR	$99.89_{\pm.05}$	$99.90_{\pm.08}$	$99.10_{\pm.15}$	$98.49_{\pm.21}$	$92.20_{\pm.85}$	$32.60_{\pm1.8}$	$94.53_{\pm.60}$	$75.40_{\pm1.4}$	$51.90_{\pm1.9}$	$87.40_{\pm1.1}$	$67.80_{\pm1.9}$	$0.10_{\pm.05}$
	ImBD	$99.94_{\pm.04}$	$99.90_{\pm.06}$	$98.72_{\pm.20}$	$99.98_{\pm.02}$	$99.90_{\pm.05}$	$93.64_{\pm.90}$	**$99.65_{\pm.10}$**	**$95.30_{\pm.65}$**	$84.82_{\pm1.1}$	$86.18_{\pm1.3}$	$25.20_{\pm2.4}$	$9.34_{\pm.80}$
	L2D	$99.99_{\pm.01}$	**$100.0_{\pm.00}$**	$99.90_{\pm.05}$	$99.80_{\pm.11}$	$99.50_{\pm.12}$	$95.50_{\pm.70}$	$86.55_{\pm1.2}$	$57.80_{\pm1.8}$	$31.90_{\pm2.2}$	$83.80_{\pm2.1}$	$40.10_{\pm4.5}$	$12.30_{\pm2.40}$
	RepreGuard	$99.84_{\pm.10}$	$99.80_{\pm.00}$	$99.70_{\pm.00}$	$99.54_{\pm.12}$	$99.43_{\pm.04}$	$99.40_{\pm.00}$	$97.19_{\pm.12}$	$81.40_{\pm.21}$	$79.70_{\pm.41}$	$98.25_{\pm.16}$	$78.53_{\pm1.1}$	$70.40_{\pm.90}$
	S2D (ours)	**$99.99_{\pm.00}$**	**$100.0_{\pm.00}$**	**$99.90_{\pm.00}$**	**$100.0_{\pm.00}$**	**$99.95_{\pm.00}$**	**$99.90_{\pm.05}$**	$98.93_{\pm.25}$	$91.93_{\pm1.2}$	**$90.83_{\pm1.1}$**	$98.31_{\pm.34}$	**$83.02_{\pm3.8}$**	**$79.48_{\pm4.8}$**
Llama-2	RoBERTa-L	$98.97_{\pm.20}$	$98.90_{\pm.30}$	$98.10_{\pm.40}$	$99.90_{\pm.05}$	$99.90_{\pm.04}$	$97.10_{\pm.60}$	$99.82_{\pm.06}$	$96.80_{\pm.50}$	$83.50_{\pm1.2}$	$99.11_{\pm.15}$	$90.70_{\pm.80}$	$57.90_{\pm2.2}$
	RAIDAR	$99.98_{\pm.02}$	**$100.0_{\pm.00}$**	$92.40_{\pm1.2}$	$99.58_{\pm.12}$	$96.80_{\pm.50}$	$70.80_{\pm1.5}$	$97.56_{\pm.40}$	$84.10_{\pm1.1}$	$61.30_{\pm1.8}$	$88.58_{\pm1.0}$	$43.70_{\pm2.0}$	$6.30_{\pm.80}$
	ImBD	$99.99_{\pm.01}$	$99.90_{\pm.05}$	$99.80_{\pm.05}$	$99.99_{\pm.01}$	$99.90_{\pm.04}$	$99.70_{\pm.05}$	**$99.89_{\pm.04}$**	**$97.50_{\pm.40}$**	**$94.51_{\pm.50}$**	$94.39_{\pm.60}$	$49.70_{\pm1.8}$	$27.36_{\pm1.9}$
	L2D	$99.98_{\pm.02}$	$99.90_{\pm.05}$	$95.60_{\pm.80}$	$99.80_{\pm.08}$	$99.30_{\pm.15}$	$77.60_{\pm1.2}$	$93.82_{\pm.90}$	$77.40_{\pm1.3}$	$51.40_{\pm2.1}$	$94.47_{\pm.55}$	$70.80_{\pm1.5}$	$2.40_{\pm.50}$
	RepreGuard	$99.94_{\pm.00}$	$99.78_{\pm.04}$	$99.62_{\pm.08}$	$99.94_{\pm.01}$	$99.40_{\pm.00}$	$99.35_{\pm.09}$	$97.35_{\pm.14}$	$82.85_{\pm.52}$	$81.32_{\pm.78}$	$98.41_{\pm.20}$	$81.05_{\pm.50}$	$74.28_{\pm2.6}$
	S2D (ours)	**$100.0_{\pm.00}$**	**$100.0_{\pm.00}$**	**$99.90_{\pm.00}$**	**$99.99_{\pm.00}$**	**$99.90_{\pm.00}$**	**$99.80_{\pm.00}$**	$99.48_{\pm.09}$	$95.42_{\pm.41}$	$94.50_{\pm.56}$	**$99.36_{\pm.10}$**	**$91.20_{\pm.65}$**	**$88.63_{\pm.95}$**
PaLM	RoBERTa-L	$99.99_{\pm.01}$	$99.90_{\pm.03}$	$99.30_{\pm.15}$	$99.97_{\pm.01}$	$99.90_{\pm.04}$	$80.20_{\pm1.5}$	**$99.98_{\pm.01}$**	$99.60_{\pm.10}$	**$99.50_{\pm.15}$**	$99.70_{\pm.10}$	$99.70_{\pm.08}$	$88.70_{\pm1.1}$
	RAIDAR	$99.96_{\pm.02}$	$99.90_{\pm.03}$	$71.10_{\pm1.8}$	$99.96_{\pm.02}$	$99.90_{\pm.04}$	$60.90_{\pm2.0}$	$99.97_{\pm.02}$	$99.40_{\pm.15}$	$93.00_{\pm.80}$	$99.71_{\pm.08}$	$98.10_{\pm.30}$	$11.30_{\pm1.5}$
	ImBD	$99.85_{\pm.05}$	$96.45_{\pm.50}$	$85.53_{\pm1.1}$	$99.93_{\pm.03}$	$98.20_{\pm.25}$	$95.32_{\pm.80}$	$99.78_{\pm.08}$	$96.00_{\pm.50}$	$88.01_{\pm1.1}$	$93.67_{\pm.70}$	$44.20_{\pm1.8}$	$25.12_{\pm2.1}$
	L2D	**$100.0_{\pm.00}$**	**$100.0_{\pm.00}$**	**$100.0_{\pm.00}$**	$99.92_{\pm.03}$	**$100.0_{\pm.00}$**	$24.50_{\pm2.5}$	**$99.98_{\pm.01}$**	**$99.70_{\pm.08}$**	$95.70_{\pm.60}$	**$99.98_{\pm.01}$**	**$99.90_{\pm.03}$**	**$97.70_{\pm.40}$**
	RepreGuard	$99.92_{\pm.00}$	$99.58_{\pm.13}$	$99.53_{\pm.04}$	$99.94_{\pm.00}$	$99.62_{\pm.11}$	$99.50_{\pm.12}$	$96.95_{\pm.17}$	$85.38_{\pm.80}$	$84.08_{\pm.81}$	$96.28_{\pm.50}$	$76.97_{\pm2.2}$	$72.40_{\pm2.0}$
	S2D (ours)	**$100.0_{\pm.00}$**	**$100.0_{\pm.00}$**	**$100.0_{\pm.00}$**	**$99.99_{\pm.00}$**	**$100.0_{\pm.00}$**	**$99.60_{\pm.00}$**	$99.88_{\pm.02}$	$98.25_{\pm.40}$	$97.60_{\pm.52}$	$99.74_{\pm.11}$	$94.52_{\pm1.6}$	$92.60_{\pm1.8}$
Claude	RoBERTa-L	$80.91_{\pm1.5}$	$14.00_{\pm1.2}$	$9.70_{\pm.80}$	$80.26_{\pm1.4}$	$18.20_{\pm1.6}$	$16.30_{\pm1.5}$	$74.82_{\pm1.8}$	$26.95_{\pm1.8}$	$25.00_{\pm2.0}$	$99.97_{\pm.01}$	$99.80_{\pm.05}$	$99.60_{\pm.10}$
	RAIDAR	$99.91_{\pm.03}$	$99.70_{\pm.10}$	$46.70_{\pm2.2}$	$97.32_{\pm.50}$	$88.90_{\pm1.2}$	$56.20_{\pm2.0}$	$93.60_{\pm.80}$	$73.50_{\pm1.4}$	$20.20_{\pm1.5}$	$99.98_{\pm.01}$	$99.90_{\pm.04}$	$94.70_{\pm.80}$
	ImBD	**$99.99_{\pm.01}$**	$99.90_{\pm.04}$	$99.71_{\pm.10}$	$99.97_{\pm.02}$	$99.90_{\pm.04}$	$84.31_{\pm1.5}$	**$99.88_{\pm.04}$**	**$97.40_{\pm.35}$**	**$91.60_{\pm.80}$**	$99.05_{\pm.15}$	$86.70_{\pm1.1}$	$60.14_{\pm1.8}$
	L2D	$99.92_{\pm.02}$	$99.90_{\pm.04}$	$82.20_{\pm1.5}$	$99.90_{\pm.05}$	$99.80_{\pm.08}$	$99.10_{\pm.20}$	$97.88_{\pm.40}$	$92.20_{\pm.80}$	$79.80_{\pm1.3}$	$99.96_{\pm.02}$	$99.90_{\pm.04}$	$97.60_{\pm.40}$
	RepreGuard	$99.92_{\pm.00}$	$99.47_{\pm.11}$	$99.18_{\pm.19}$	$99.89_{\pm.01}$	$98.82_{\pm.26}$	$98.10_{\pm.19}$	$95.08_{\pm.27}$	$70.38_{\pm.93}$	$68.45_{\pm.54}$	$98.42_{\pm.26}$	$85.85_{\pm.62}$	$81.47_{\pm2.0}$
	S2D (ours)	$99.98_{\pm.03}$	**$99.98_{\pm.04}$**	**$99.93_{\pm.04}$**	**$99.99_{\pm.01}$**	**$100.0_{\pm.00}$**	**$99.98_{\pm.04}$**	$98.92_{\pm.20}$	$92.58_{\pm.69}$	$89.45_{\pm1.6}$	**$99.99_{\pm.00}$**	**$99.95_{\pm.09}$**	**$99.95_{\pm.09}$**

Table 2: **In-distribution and out-of-distribution performance comparison.** Methods are categorized into train-free and train-based settings. Our proposed S2D demonstrates robust generalization, achieving state-of-the-art or highly competitive results across the majority of evaluated scenarios. Best and second-best results are highlighted in **darker** and lighter blue, respectively.

Detector ↓	No Attack		Paraphrasing Attacks						Perturbations					
			Polish		Back Trans.		DIPPER		Character		Word		Char + Word	
	AUROC	*TPR@1%*	*AUROC*	*TPR@1%*	*AUROC*	*TPR@1%*	*AUROC*	*TPR@1%*	*AUROC*	*TPR@1%*	*AUROC*	*TPR@1%*	*AUROC*	*TPR@1%*
Binoculars	95.40	62.80	73.54 (−22.9%)	0.53 (−99.2%)	90.93 (−4.7%)	53.73 (−14.4%)	93.67 (−1.8%)	58.14 (−7.4%)	94.76 (−0.7%)	57.73 (−8.1%)	94.23 (−1.2%)	58.73 (−6.5%)	95.01 (−0.4%)	61.07 (−2.8%)
L2D	95.80	88.33	56.14 (−41.4%)	1.20 (−98.6%)	92.36 (−3.6%)	15.73 (−82.2%)	87.72 (−8.4%)	3.20 (−96.4%)	83.43 (−12.9%)	15.20 (−82.8%)	95.70 (−0.1%)	80.13 (−9.3%)	94.21 (−1.7%)	42.13 (−52.3%)
RepreGuard	97.02	85.30	**83.88** (−13.5%)	10.53 (−87.7%)	91.67 (−5.5%)	45.33 (−46.9%)	94.23 (−2.9%)	73.67 (−13.6%)	96.32 (−0.7%)	83.87 (−1.7%)	92.88 (−4.3%)	78.67 (−7.8%)	94.29 (−2.8%)	85.00 (−0.4%)
S2D (ours)	**98.87**	**96.68**	82.06 (−17.0%)	**23.60** (−75.6%)	**93.45** (−5.5%)	**67.33** (−30.4%)	**98.66** (−0.2%)	**84.91** (−12.2%)	**97.76** (−1.1%)	**90.53** (−6.4%)	**96.73** (−2.2%)	**90.87** (−6.0%)	**95.77** (−3.1%)	**91.47** (−5.4%)

Table 3: **Robustness evaluation.** "No Attack" denotes performance on original text, and values in parentheses indicate the relative percentage change $((\text{Attack} - \text{No attack})\,/\text{No attack} \times 100\%)$ compared to this baseline (positive values indicate performance gains under attack). Best and second-best results are highlighted in **darker** and lighter blue, respectively.

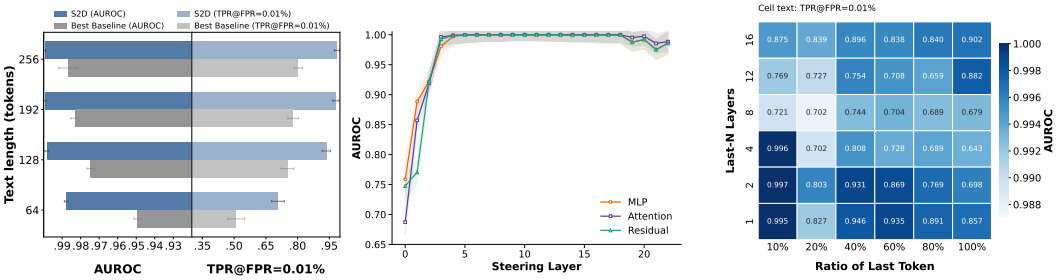
watermarking methods actively embed detectable signals while S2D is a passive detector—it helps position S2D relative to these approaches.[6]

Table 4 compares S2D with three watermarking methods, namely Gumbel-max [76], Green-red [11], and SynthID [77], using their default detectors, which have full knowledge of the watermarking mechanisms. Interestingly, S2D achieves stronger performance at lower temperature ($T = 0.3$), where watermark signals are weaker due to reduced generation entropy. In contrast, watermarking methods perform better at higher temperatures, where the embedded signals are more pronounced. The performance of S2D in this setting remains reasonable: even without knowledge of the watermarking scheme, it still captures distributional artifacts induced by the watermarking generation process. These artifacts are more pronounced in low-entropy settings, where watermark signals are weak, while at higher entropy levels, stronger embedded watermark signals favor watermark-specific detectors.

4.2 Ablation Study

Importance of Steering. We assess the role of steering by comparing S2D with a variant without steering, which does not incorporate the steering vector and instead estimates the vMF parameters

[6]To reflect a realistic setting, S2D is trained once on human-written and non-watermarked LLM-generated text, without retraining for specific watermarking schemes, and is directly applied at evaluation.



(a) Robustness to text length (b) Impact of steering position (c) Token ratio vs. Layer index

Figure 2: **Analysis of S2D performance.** (a) Comparison of detection stability across varying input lengths. (b) Detection performance across steering layers, showing that intermediate layers consistently achieve the best performance. (c) Performance heatmap as a function of last-token selection ratio and the number of aggregated layers.

Temp.	Method	64 tokens			128 tokens			256 tokens		
		Gumbel	Green-red	SynthID	Gumbel	Green-red	SynthID	Gumbel	Green-red	SynthID
		AUROC / TPR@1%FPR (%)								
0.3	Watermark Det.†	**99.9 / 98.0**	82.0 / 12.0	81.8 / 15.9	**99.9 / 99.7**	**99.6 / 95.3**	84.8 / 28.0	99.9 / **99.9**	99.8 /**99.4**	84.5 / 34.0
	S2D (ours)	77.6 / 20.0	**96.7 / 60.7**	**85.1 / 26.0**	96.6 / 62.8	98.4 / 81.3	**96.5 / 69.0**	**100** / 98.5	**99.9** / 98.6	**99.9 / 98.1**
0.7	Watermark Det.†	**99.9 / 97.0**	80.2 / 9.3	**99.3 / 89.8**	**99.9 / 99.8**	**99.6 / 95.7**	**99.7 / 95.8**	**99.9 / 99.9**	**100 / 99.8**	**99.5 / 97.5**
	S2D (ours)	65.4 / 12.3	**90.5 / 33.0**	70.0 / 10.9	89.3 / 44.7	89.0 / 41.7	84.2 / 32.7	99.6 / 91.7	95.9 / 70.4	97.6 / 79.0

† The defaulted detectors provided by the original watermarking algorithms.

Table 4: **Detection performance across different text lengths and temperatures.** Comparison with default watermark detectors. **Bold** indicates better performance within each setting.

directly from training representations. As shown in Figure 4, S2D `w/o steering` exhibits substantially weaker separation between human- and LLM-generated texts, with significant overlap between the two distributions. This highlights the importance of incorporating the steering vector.

Steering Position. We first analyze the impact of the steering position. As shown in Figure 2b, for the default *Llama-3.1-8B* observer, intermediate-layer injection performs substantially better than very early injection, while the exact insertion point within a Transformer block (i.e., residual stream, attention output, or MLP output) has a much smaller effect.

We further examine how this preferred region changes with model depth by performing steering-layer sweeps on *Llama-3.2-1B*, *Meta-Llama-3-8B*, and *Meta-Llama-3-70B*. Across all three scales, strong performance is attained over a range of intermediate depths rather than at a single isolated layer. However, the best-performing layer does not follow a fixed proportional scaling law with network depth. These results suggest that intermediate depth is a useful search region, while the exact steering layer remains model dependent. Full cross-scale results are provided in Appendix F.5.

Token Selection and Layer Aggregation. We analyze the effect of the last-token ratio K and the number of aggregated layers N (introduced in Section 2.2). As shown in Figure 2c, performance exhibits a non-monotonic pattern across both factors. Strong performance is observed in two regimes: the bottom-left region, which uses a small fraction of tokens from a few top layers, and the top-right region, where deeper aggregation is combined with larger token ratios. In contrast, intermediate configurations perform worse. This pattern suggests a trade-off between signal concentration and aggregation. Detection-relevant signals are concentrated in the final layers and later tokens. While incorporating more layers or tokens can be beneficial when aggregation is sufficiently large and helps stabilize the representation, intermediate settings tend to introduce noise without sufficient complementary signal, leading to degraded performance.

Effects of Different Observer Models. As shown in Table 5, the performance of S2D depends noticeably on the choice of observer model. Several observers, including *Falcon-7B*, *Qwen2.5-7B*, and *Llama-3.1-8B*, yield near-perfect AUROC and strong TPR at low FPRs, while others, such as *Gemma-2*, perform substantially worse. Importantly, this variation is not explained by model scale, recency, or instruction tuning alone, e.g., the lightweight *OPT-2.7B* remains highly competitive.

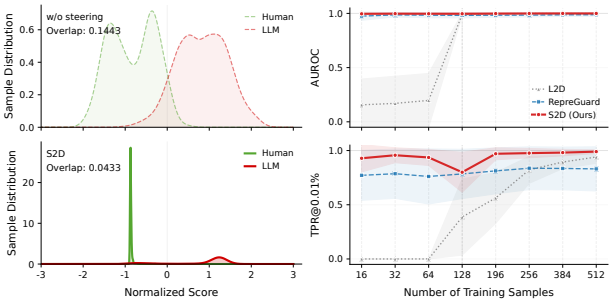


Figure 4: **Detection analysis.** Left: Steering leads to better separability. Right: Detection performance across training sizes. Full results are in Figures 7 and 8 in Appendix.

Observer Model	AUROC	TPR@1%	TPR@.01%
Llama-3.1-8B	99.62 ± 0.46	99.05 ± 0.60	97.98 ± 1.37
Mistral-7B-v0.3	68.52 ± 0.94	30.53 ± 5.31	23.28 ± 6.45
GPT-Neo-2.7B	82.38 ± 2.94	32.65 ± 8.91	11.33 ± 5.67
OPT-2.7B	99.86 ± 0.13	98.30 ± 1.79	84.50 ± 8.24
Qwen2.5-7B	99.93 ± 0.13	99.58 ± 0.65	98.65 ± 1.69
Falcon-7B	99.96 ± 0.05	99.40 ± 0.88	98.28 ± 1.83
Falcon-7B-Inst.	99.96 ± 0.05	99.18 ± 1.45	95.90 ± 2.68
Gemma-2-9B	76.54 ± 14.6	19.25 ± 5.37	3.05 ± 2.89
Gemma-2-9B-Inst.	79.09 ± 20.3	37.73 ± 26.6	18.90 ± 14.1

Table 5: **Observer stability.** Detection performance (mean ± std), averaged across four training datasets. Full results are in Table 9 in Appendix.

This dependence is consistent with the design of S2D: the method operates on the hidden representations of a frozen observer and can amplify discriminative structure already present in its latent space, but its effectiveness is limited when human- and LLM-generated representations remain strongly entangled. We therefore do not expect all observer models to perform equally well. In practice, we recommend selecting the observer using a small held-out validation set, with pre-steering representation overlap serving as a useful diagnostic of observer suitability.

Shots of Training Dataset. Finally, we investigate the impact of training set size on detection performance using S2D, RepreGuard, and L2D, since these three methods yielded the best overall results in the preceding experiments. As shown in Figure 4 (right), S2D achieves strong performance even in the low-shot regime, while RepreGuard shows moderate sensitivity, and L2D degrades substantially with limited data. This indicates that S2D can leverage intrinsic differences between human-written and LLM-generated text in the representation space, with training mainly serving to reduce distributional overlap. Detailed results are provided in the Appendix F.3.

5 Discussion

We present S2D, a representation-based framework for detecting LLM-generated text. By injecting a lightweight steering vector into hidden representations, S2D improves class separability and formulates detection as a hypothesis test. The resulting detector provides Type-I error control and a bound on excess Type-II error. Experiments demonstrate strong performance across ID and OOD settings, as well as robustness to adversarial perturbations, paraphrasing, varying text lengths, and different observer models. S2D also achieves a favorable trade-off between detection performance and computational efficiency compared with existing methods.

Several limitations suggest directions for future work. First, S2D is a document-level binary detector and does not localize human- and AI-written segments within mixed-authorship text. Second, a single vMF model may not fully capture latent geometry across domains; mixture models or target-domain adaptation may improve robustness. Finally, more flexible steering mechanisms, such as multiple vectors or non-additive interventions, could better capture representation geometry.

Acknowledgments and Disclosure of Funding

The authors declare no competing financial interests.

References

[1] Inyoung Cheong, King Xia, KJ Kevin Feng, Quan Ze Chen, and Amy X Zhang. (A) I am not A lawyer, but...: engaging legal experts towards responsible LLM policies for legal advice. In *Proceedings of the 2024 ACM conference on fairness, accountability, and transparency*, pages 2454–2469, 2024.

[2] Giorgos Iacovides, Thanos Konstantinidis, Mingxue Xu, and Danilo Mandic. FinLlama: LLM-based financial sentiment analysis for algorithmic trading. In *Proceedings of the 5th ACM International Conference on AI in Finance*, pages 134–141, 2024.

[3] Dmitry Kobak, Rita González-Márquez, Emőke-Ágnes Horvát, and Jan Lause. Delving into LLM-assisted writing in biomedical publications through excess vocabulary. *Science Advances*, 11(27):eadt3813, 2025.

[4] Hanchen Su, Wei Luo, Yashar Mehdad, Wei Han, Elaine Liu, Wayne Zhang, Mia Zhao, and Joy Zhang. LLM-friendly knowledge representation for customer support. In *Proceedings of the 31st International Conference on Computational Linguistics: Industry Track*, pages 496–504, 2025.

[5] Devesh Batra, Conor Hamill, John Hartley, Ramin Okhrati, Dale Seddon, Harvey Miller, Raad Khraishi, and Greig Cowan. A review of LLM agent applications in finance and banking. *Available at SSRN 5381584*, 2025.

[6] Jooyoung Lee, Thai Le, Jinghui Chen, and Dongwon Lee. Do language models plagiarize? In *Proceedings of the ACM Web Conference 2023*, pages 3637–3647, 2023.

[7] Aneta Zugecova, Dominik Macko, Ivan Srba, Robert Moro, Jakub Kopál, Katarína Marcinčinová, and Matúš Mesarčík. Evaluation of LLM vulnerabilities to being misused for personalized disinformation generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 780–797, 2025.

[8] Jiashu Xu, Mingyu Ma, Fei Wang, Chaowei Xiao, and Muhao Chen. Instructions as backdoors: Backdoor vulnerabilities of instruction tuning for large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3111–3126, 2024.

[9] Yifan Yang, Qiao Jin, Furong Huang, and Zhiyong Lu. Adversarial prompt and fine-tuning attacks threaten medical large language models. *Nature Communications*, 16(1):9011, 2025.

[10] Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. DetectGPT: Zero-shot machine-generated text detection using probability curvature. In *International conference on machine learning*, pages 24950–24962. PMLR, 2023.

[11] John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. In *International Conference on Machine Learning*, pages 17061–17084. PMLR, 2023.

[12] Xiang Li, Feng Ruan, Huiyuan Wang, Qi Long, and Weijie J Su. A statistical framework of watermarks for large language models: Pivot, detection efficiency and optimal rules. *The Annals of Statistics*, 53(1):322–351, 2025.

[13] Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. Representation engineering: A top-down approach to AI transparency. *arXiv preprint arXiv:2310.01405*, 2023.

[14] Hadas Orgad, Michael Toker, Zorik Gekhman, Roi Reichart, Idan Szpektor, Hadas Kotek, and Yonatan Belinkov. LLMs know more than they show: On the intrinsic representation of LLM hallucinations. *arXiv preprint arXiv:2410.02707*, 2024.

[15] Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 36:41451–41530, 2023.

[16] Oscar Skean, Md Rifat Arefin, Dan Zhao, Niket Patel, Jalal Naghiyev, Yann LeCun, and Ravid Shwartz-Ziv. Layer by layer: Uncovering hidden representations in language models. *arXiv preprint arXiv:2502.02013*, 2025.

[17] Seongheon Park, Xuefeng Du, Min-Hsuan Yeh, Haobo Wang, and Yixuan Li. Steer LLM Latents for Hallucination Detection. In *International Conference on Machine Learning*, pages 47971–47990. PMLR, 2025.

[18] Jingtao Sun and Zhanglong Lv. Zero-shot detection of LLM-generated text via text reorder. *Neurocomputing*, 631:129829, 2025.

[19] Jinyan Su, Terry Zhuo, Di Wang, and Preslav Nakov. DetectLLM: Leveraging log rank information for zero-shot detection of machine-generated text. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12395–12412, 2023.

[20] Guangsheng Bao, Yanbin Zhao, Zhiyang Teng, Linyi Yang, and Yue Zhang. Fast-DetectGPT: Efficient Zero-Shot Detection of Machine-Generated Text via Conditional Probability Curvature. In *The Twelfth International Conference on Learning Representations*, 2024.

[21] Abhimanyu Hans, Avi Schwarzschild, Valeriia Cherepanova, Hamid Kazemi, Aniruddha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Spotting LLMs with binoculars: Zero-shot detection of machine-generated text. *arXiv preprint arXiv:2401.12070*, 2024.

[22] Chengzhi Mao, Carl Vondrick, Hao Wang, and Junfeng Yang. Raidar: generative AI detection via rewriting. *arXiv preprint arXiv:2401.12970*, 2024.

[23] Hongyi Zhou, Jin Zhu, Erhan Xu, Kai Ye, Ying Yang, and Chengchun Shi. Learn-to-Distance: Distance Learning for Detecting LLM-Generated Text. *arXiv preprint arXiv:2601.21895*, 2026.

[24] Yifei Huang, Jiuxin Cao, Hanyu Luo, Xin Guan, and Bo Liu. Magret: Machine-generated text detection with rewritten texts. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 8336–8346, 2025.

[25] Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, et al. Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203*, 2019.

[26] Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. 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, 2020.

[27] Sandra Mitrović, Davide Andreoletti, and Omran Ayoub. ChatGPT or human? Detect and explain. Explaining decisions of machine learning model for detecting short ChatGPT-generated text. *arXiv preprint arXiv:2301.13852*, 2023.

[28] Hongyi Zhou, Jin Zhu, Pingfan Su, Kai Ye, Ying Yang, Shakeel AOB Gavioli-Akilagun, and Chengchun Shi. AdaDetectGPT: Adaptive detection of LLM-generated text with statistical guarantees. *arXiv preprint arXiv:2510.01268*, 2025.

[29] Zhiguang Yang, Gejian Zhao, and Hanzhou Wu. Watermarking for large language models: A survey. *Mathematics*, 13(9):1420, 2025.

[30] Xuhong Wang, Haoyu Jiang, Yi Yu, Jingru Yu, Yilun Lin, Ping Yi, Yingchun Wang, Yu Qiao, Li Li, and Fei-Yue Wang. Building intelligence identification system via large language model watermarking: a survey and beyond. *Artificial Intelligence Review*, 58(8):249, 2025.

[31] Peigen Ye, Huali Ren, Zhengdao Li, Anli Yan, Hongyang Yan, Shaowei Wang, and Jin Li. Securing large language models: A survey of watermarking and fingerprinting techniques. *ACM Computing Surveys*, 58(7):1–35, 2026.

[32] Xiao Yu, Kejiang Chen, Qi Yang, Weiming Zhang, and Nenghai Yu. Text fluoroscopy: Detecting LLM-generated text through intrinsic features. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15838–15846, 2024.

[33] Xin Chen, Junchao Wu, Shu Yang, Runzhe Zhan, Zeyu Wu, Ziyang Luo, Di Wang, Min Yang, Lidia S Chao, and Derek F Wong. Repreguard: Detecting LLM-generated text by revealing hidden representation patterns. *Transactions of the Association for Computational Linguistics*, 13:1812–1831, 2025.

[34] Nadir Durrani, Hassan Sajjad, Fahim Dalvi, and Yonatan Belinkov. Analyzing individual neurons in pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4865–4880, 2020.

[35] Anna Hedström, Salim I Amoukou, Tom Bewley, Saumitra Mishra, and Manuela Veloso. To steer or not to steer? mechanistic error reduction with abstention for language models. *arXiv preprint arXiv:2510.13290*, 2025.

[36] Elena Voita, Javier Ferrando, and Christoforos Nalmpantis. Neurons in large language models: Dead, n-gram, positional. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1288–1301, 2024.

[37] Asma Ghandeharioun, Avi Caciularu, Adam Pearce, Lucas Dixon, and Mor Geva. Patchscopes: a unifying framework for inspecting hidden representations of language models. In *Proceedings of the 41st International Conference on Machine Learning*, pages 15466–15490, 2024.

[38] Hao Sun, Huailiang Peng, Qiong Dai, Xu Bai, and Yanan Cao. LayerNavigator: Finding promising intervention layers for efficient activation steering in large language models. In *Neural Information Processing Systems*, 2025. URL `https://openreview.net/forum?id=wj4lM45xQR`.

[39] Dung V Nguyen, Hieu M Vu, Nhi Y Pham, Lei Zhang, and Tan M Nguyen. Activation steering with a feedback controller. *arXiv preprint arXiv:2510.04309*, 2025.

[40] Praveen Venkateswaran and Danish Contractor. Spotlight your instructions: Instruction-following with dynamic attention steering. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3752–3770, 2026.

[41] Daniel Beaglehole, Adityanarayanan Radhakrishnan, Enric Boix-Adsera, and Mikhail Belkin. Toward universal steering and monitoring of AI models. *Science*, 391(6787):787–792, 2026.

[42] Parmida Davarmanesh, Ashia Wilson, and Adityanarayanan Radhakrishnan. Efficient and accurate steering of large language models through attention-guided feature learning. *arXiv preprint arXiv:2602.00333*, 2026.

[43] Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*, 2023.

[44] Shawn Im and Sharon Li. A unified understanding and evaluation of steering methods. *arXiv preprint arXiv:2502.02716*, 2025.

[45] Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15504–15522, 2024.

[46] Junfei Wu, Yue Ding, Guofan Liu, Tianze Xia, Ziyue Huang, Dianbo Sui, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. SHARP: Steering hallucination in LVLMs via representation engineering. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 14357–14372, 2025.

[47] Leheng Sheng, Changshuo Shen, Weixiang Zhao, Junfeng Fang, Xiaohao Liu, Zhenkai Liang, Xiang Wang, An Zhang, and Tat-Seng Chua. Alphasteer: Learning refusal steering with principled null-space constraint. *arXiv preprint arXiv:2506.07022*, 2025.

[48] Xinchi Qiu, Lei Yu, Yuchen Zhang, Aobo Yang, Narine Kokhlikyan, Nicola Cancedda, Diego Garcia-Olano, et al. Hallucination reduction with casal: Contrastive activation steering for amortized learning. *arXiv preprint arXiv:2510.02324*, 2025.

[49] Tim Tian Hua, Andrew Qin, Samuel Marks, and Neel Nanda. Steering evaluation-aware language models to act like they are deployed. *arXiv preprint arXiv:2510.20487*, 2025.

[50] Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*, 2023.

[51] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[52] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[53] Haobo Wang, Ruixuan Xiao, Yixuan Li, Lei Feng, Gang Niu, Gang Chen, and Junbo Zhao. Pico: Contrastive label disambiguation for partial label learning. In *International conference on learning representations*, 2021.

[54] Erich Leo Lehmann and Joseph P Romano. *Testing statistical hypotheses*. Springer, 2005.

[55] Marcus D Ruopp, Neil J Perkins, Brian W Whitcomb, and Enrique F Schisterman. Youden index and optimal cut-point estimated from observations affected by a lower limit of detection. *Biometrical Journal: Journal of Mathematical Methods in Biosciences*, 50(3):419–430, 2008.

[56] Jingjing Yin and Lili Tian. Joint confidence region estimation for area under roc curve and youden index. *Statistics in medicine*, 33(6):985–1000, 2014.

[57] Xinhua Liu. Classification accuracy and cut point selection. *Statistics in medicine*, 31(23): 2676–2686, 2012.

[58] Minjia Mao, Dongjun Wei, Xiao Fang, and Michael Chau. A General Method for Detecting Information Generated by Large Language Models. *arXiv preprint arXiv:2506.21589*, 2025.

[59] Hongyi Zhou, Jin Zhu, Ying Yang, and Chengchun Shi. Detecting LLM-Generated Text with Performance Guarantees. *arXiv preprint arXiv:2601.06586*, 2026.

[60] Shengchao Liu, Xiaoming Liu, Chengzhengxu Li, Zhaohan Zhang, Guoxin Ma, Yu Lan, and Shuai Xiao. MGT-Prism: Enhancing Domain Generalization for Machine-Generated Text Detection via Spectral Alignment. *arXiv preprint arXiv:2508.13768*, 2025.

[61] Junchao Wu, Runzhe Zhan, Derek Wong, Shu Yang, Xinyi Yang, Yulin Yuan, and Lidia Chao. DetectRL: Benchmarking LLM-generated text detection in real-world scenarios. *Advances in Neural Information Processing Systems*, 37:100369–100401, 2024.

[62] Shashi Narayan, Shay B Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 1797–1807, 2018.

[63] Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, 2018.

[64] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. *Advances in Neural Information Processing Systems*, 28, 2015.

[65] OpenAI. Introducing ChatGPT. `https://openai.com/index/chatgpt/`, 2023. OpenAI Blog.

[66] Anthropic. Releasing claude instant 1.2, 2023. URL `https://www.anthropic.com/news/releasingclaude-instant-1-2`. Anthropic Blog.

[67] Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023.

[68] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open Foundation and Fine-Tuned Chat Models. *arXiv preprint arXiv:2307.09288*, 2023. URL `https://arxiv.org/abs/2307.09288`.

[69] Sondos Mahmoud Bsharat, Jiacheng Liu, Xiaohan Zhao, Tianjun Yao, Xinyi Shang, Yi Tang, Jiacheng Cui, Ahmed Elhagry, Salwa K. Al Khatib, Hao Li, Salman Khan, and Zhiqiang Shen. Operation-guided progressive human-to-ai text transformation benchmark for multi-granularity ai-text detection. *arXiv preprint arXiv:2606.06481*, 2026.

[70] Sebastian Gehrmann, Hendrik Strobelt, and Alexander M Rush. 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, 2019.

[71] Runheng Liu, Heyan Huang, Xingchen Xiao, and Zhijing Wu. Zero-shot detection of LLM-generated text via implicit reward model. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[72] Yihuai Xu, Yongwei Wang, Yifei Bi, Huangsen Cao, Zhouhan Lin, Yu Zhao, and Fei Wu. Training-free LLM-generated text detection by mining token probability sequences. In *International Conference on Learning Representations (ICLR)*, 2025.

[73] Haitong Luo, Weiyao Zhang, Suhang Wang, Wenji Zou, Chungang Lin, Xuying Meng, and Yujun Zhang. SpecDetect: Simple, fast, and training-free detection of LLM-generated text via spectral analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 32356–32364, 2026. doi: 10.1609/aaai.v40i38.40510.

[74] Shixuan Ma, Jiahao Li, Zhendong Mao, and Quan Wang. Zero-shot detection of LLM-generated text using temperature sensitivity. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 37664–37679, San Diego, California, United States, 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-long.1748.

[75] Jiaqi Chen, Xiaoye Zhu, Tianyang Liu, Ying Chen, Chen Xinhui, Yiwen Yuan, Chak Tou Leong, Zuchao Li, Long Tang, Lei Zhang, et al. Imitate before detect: Aligning machine stylistic preference for machine-revised text detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 23559–23567, 2025.

[76] Scott Aaronson and H Kirchner. Watermarking of large language models. In *Large language models and transformers workshop at Simons Institute for the Theory of Computing*, volume 2023, 2023.

[77] Sumanth Dathathri, Abigail See, Sumedh Ghaisas, Po-Sen Huang, Rob McAdam, Johannes Welbl, Vandana Bachani, Alex Kaskasoli, Robert Stanforth, Tatiana Matejovicova, et al. Scalable watermarking for identifying large language model outputs. *Nature*, 634(8035):818–823, 2024.

[78] William J Youden. Index for rating diagnostic tests. *Cancer*, 3(1):32–35, 1950.

[79] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[80] Pascal Massart. The tight constant in the Dvoretzky-Kiefer-Wolfowitz inequality. *The Annals of Probability*, pages 1269–1283, 1990.

[81] Wolfgang Polonik. Measuring mass concentrations and estimating density contour clusters-an excess mass approach. *The annals of Statistics*, pages 855–881, 1995.

[82] Enno Mammen and Alexandre B Tsybakov. Smooth discrimination analysis. *The Annals of Statistics*, 27(6):1808–1829, 1999.

[83] Xin Tong. A plug-in approach to neyman-pearson classification. *Journal of Machine Learning Research*, 14(1):3011–3040, 2013.

[84] David A Freedman. On tail probabilities for martingales. *the Annals of Probability*, pages 100–118, 1975.

A Algorithm

Algorithm 1: Overall training pipeline for S2D

Input: Frozen observer model f_θ, training set $\mathcal{S}_{\text{train}}$, null calibration set $\mathcal{S}_{\text{cal}}$ (human-written text only); steering layer ℓ_s; inverse temperature β; EMA coefficient ρ; learning rate η; epochs E; batch size B.

Output: Steering vector $\mathbf{v}$, class mean directions $\widehat{\boldsymbol{\mu}}_0, \widehat{\boldsymbol{\mu}}_1$, and calibrated threshold $\widehat{\tau}$.

Initialize: $\mathbf{v} \leftarrow \mathbf{0} \in \mathbb{R}^d$, and uniformly sample $\widehat{\boldsymbol{\mu}}_c \in \mathbb{S}^{d-1}$ for $c \in \{0, 1\}$;

for $e = 1$ **to** E **do**

  Shuffle $\mathcal{S}_{\text{train}}$ and divide into mini-batches;

  for *each mini-batch* $\mathcal{B} \subset \mathcal{S}_{train}$ **do**

    (1) Forward with steering: For all $x \in \mathcal{B}$, inject $\mathbf{v}$ at layer ℓ_s:

$$\tilde{h}_{\ell_s,t}(x) \leftarrow h_{\ell_s,t}(x) + \mathbf{v}, \quad \text{for all valid token position } t.$$

    Extract the steered representation $f_{\theta,\mathbf{v}}(x) \in \mathbb{S}^{d-1}$ as described in Section 2.2;

    (2) Temperature-Scaled Objective: Compute the batch objective using the softmax form in (2):

$$\mathcal{L}_{\mathcal{B}}(\mathbf{v}) \leftarrow \frac{1}{|\mathcal{B}|} \sum_{(x,y)\in\mathcal{B}} \log q_\beta \left(y \mid f_{\theta,\mathbf{v}}(x), \widehat{\boldsymbol{\mu}}_0, \widehat{\boldsymbol{\mu}}_1\right).$$

    (3) Steering Update: Take a gradient update step: $\mathbf{v} \leftarrow \mathbf{v} + \eta \nabla_{\mathbf{v}} \mathcal{L}_{\mathcal{B}}(\mathbf{v})$;

    (4) Mean Directions Update: For $c \in \{0, 1\}$, compute the batch mean representation $\bar{\mathbf{z}}_c(\mathbf{v})$ and apply EMA:

$$\widehat{\boldsymbol{\mu}}_c \leftarrow \frac{(1-\rho)\widehat{\boldsymbol{\mu}}_c + \rho\, \bar{\mathbf{z}}_c(\mathbf{v})}{\|(1-\rho)\widehat{\boldsymbol{\mu}}_c + \rho\, \bar{\mathbf{z}}_c(\mathbf{v})\|},$$

    where $\bar{\mathbf{z}}_c(\mathbf{v}) = \frac{1}{|\mathcal{B}_c|} \sum_{x\in\mathcal{B}_c} f_{\theta,\mathbf{v}}(x)$, and $\mathcal{B}_c = \{(x, y) \in \mathcal{B} : y = c\}$ denotes the subset of samples in the mini-batch belonging to class c, with $\mathcal{B} = \mathcal{B}_0 \cup \mathcal{B}_1$.

Threshold Calibration: for *each* $x \in \mathcal{S}_{cal}$ **do**

  Compute the detection score $\widehat{\mathcal{S}}(x) = (\widehat{\boldsymbol{\mu}}_1 - \widehat{\boldsymbol{\mu}}_0)^\top f_{\theta,\mathbf{v}}(x)$;

Set the threshold $\widehat{\tau}$ using $\mathcal{S}_{\text{cal}}$ according to the chosen calibration scheme;

return $(\mathbf{v}, \widehat{\boldsymbol{\mu}}_0, \widehat{\boldsymbol{\mu}}_1, \widehat{\tau})$

B Derivation of the Discriminative Posterior Probability

In this section, we derive the class posterior probability $p(y_i \mid f_{\theta,\mathbf{v}}(x_i), \boldsymbol{\mu}_0, \boldsymbol{\mu}_1)$ under the vMF model. In the vMF model, given a text x_i with label $y_i = c \in \{0, 1\}$, its steered representation $f_{\theta,\mathbf{v}}(x_i)$ follows the following distribution on the unit hypersphere $\mathbb{S}^{d-1}$, with likelihood

$$p(f_{\theta,\mathbf{v}}(x_i) \mid y_i = c) = C_d(\kappa) \exp\left(\kappa \boldsymbol{\mu}_c^\top f_{\theta,\mathbf{v}}(x_i)\right),$$

where $C_d(\kappa) = \frac{\kappa^{d/2-1}}{(2\pi)^{d/2} I_{d/2-1}(\kappa)}$ is the normalization constant and $\boldsymbol{\mu}_c$ is the class mean direction. We assume a shared concentration parameter κ across classes and a uniform prior $p(y_i = 0) = p(y_i = 1) = \frac{1}{2}$. By Bayes' theorem,

$$p(y_i \mid f_{\theta,\mathbf{v}}(x_i)) = \frac{p(f_{\theta,\mathbf{v}}(x_i) \mid y_i) p(y_i)}{\sum_{c\in\{0,1\}} p(f_{\theta,\mathbf{v}}(x_i) \mid y_i = c) p(y_i = c)}.$$

Substituting the vMF likelihood and the uniform prior gives

$$p(y_i \mid f_{\theta,\mathbf{v}}(x_i)) = \frac{C_d(\kappa) \exp\left(\kappa \boldsymbol{\mu}_{y_i}^\top f_{\theta,\mathbf{v}}(x_i)\right) \cdot \frac{1}{2}}{\sum_{c\in\{0,1\}} C_d(\kappa) \exp\left(\kappa \boldsymbol{\mu}_c^\top f_{\theta,\mathbf{v}}(x_i)\right) \cdot \frac{1}{2}}.$$

Since $C_d(\kappa)$ and the prior $\frac{1}{2}$ are identical across classes, they cancel out, yielding the class posterior

$$p\left(y_i \mid f_{\theta,\mathbf{v}}(x_i), \boldsymbol{\mu}_0, \boldsymbol{\mu}_1\right) = \frac{\exp\left(\kappa \boldsymbol{\mu}_{y_i}^\top f_{\theta,\mathbf{v}}(x_i)\right)}{\sum_{c\in\{0,1\}} \exp\left(\kappa \boldsymbol{\mu}_c^\top f_{\theta,\mathbf{v}}(x_i)\right)}.$$

Thus, the shared-concentration vMF model induces a softmax posterior with scale κ. In Phase I, we retain this functional form while replacing the distributional concentration κ with a tunable inverse temperature β, which yields the optimization objective in (2).

C Alternative Threshold Calibration via the Youden Index

In standard benchmarking settings, where a balanced trade-off between the true positive rate (TPR) and false positive rate (FPR) is desired, one may select the threshold by maximizing the Youden index [78] on a calibration set. Let the calibration sample be given by $\mathcal{S}_{\mathrm{cal}} = \{(x_i, y_i)\}_{i=1}^{n_2}$, where $y_i \in \{0, 1\}$ indicates whether x_i is LLM-generated ($y_i = 1$) or human-written ($y_i = 0$). For a threshold τ, the empirical TPR and FPR are defined as

$$\widehat{\mathrm{TPR}}(\tau) := \frac{1}{n_+} \sum_{i:y_i=1} \mathbb{1}\big(\widehat{\mathcal{S}}(x_i) \geq \tau\big), \qquad \widehat{\mathrm{FPR}}(\tau) := \frac{1}{n_-} \sum_{i:y_i=0} \mathbb{1}\big(\widehat{\mathcal{S}}(x_i) \geq \tau\big),$$

where $n_+ = \sum_{i=1}^{n_2} \mathbb{1}(y_i = 1)$ and $n_- = \sum_{i=1}^{n_2} \mathbb{1}(y_i = 0)$. The Youden-based threshold is then given by

$$\widehat{\tau}_{\mathrm{Y}} := \arg\max_{\tau} \left(\widehat{\mathrm{TPR}}(\tau) + 1 - \widehat{\mathrm{FPR}}(\tau)\right).$$

Equivalently, this criterion maximizes $\widehat{\mathrm{TPR}}(\tau) - \widehat{\mathrm{FPR}}(\tau)$, thereby favoring thresholds that achieve high detection power while maintaining a low FPR.

Empirical comparison of Type-I error control and power. To further understand the practical behavior of different calibration strategies, we compare the threshold defined in (5), denoted by $\widehat{\tau}_\alpha$, with the Youden-based threshold $\widehat{\tau}_{\mathrm{Y}}$ in terms of both their realized Type-I error and empirical power.

We use *meta-llama/Llama-3.1-8B* as the observer model. Both the training set and the calibration set are drawn from a mixed dataset comprising outputs from four LLMs together with the corresponding human-written texts, while evaluation is conducted on four test sets generated by *ChatGPT-3.5-Turbo*, *Llama-2-70B*, *Google-PaLM*, and *Claude-Instant*, respectively. For each calibration strategy, the threshold is computed using the same mixed calibration set, and we report the realized Type-I error and empirical detection power. Since $\widehat{\tau}_{\mathrm{Y}}$ does not depend on a target level α, it yields a single operating point once the calibration set is fixed, whereas $\widehat{\tau}_\alpha$ explicitly adapts to the desired Type-I error level. We focus on the representative operating regime $\alpha = 1\%$.

Table 6: Realized type-I error (%) and empirical detection power (%) under different calibration strategies at target level $\alpha = 1\%$.

Method	ChatGPT-3.5-Turbo		Llama-2-70B		Google-PaLM		Claude-Instant	
	Type-I Error	Power	Type-I Error	Power	Type-I Error	Power	Type-I Error	Power
$\widehat{\tau}_{\mathrm{Y}}$	0.20	99.70	0.00	99.50	0.20	85.70	0.20	99.80
$\widehat{\tau}_\alpha$	0.90	100.00	0.80	99.90	0.80	92.10	1.30	99.90

As shown in Table 6, the threshold $\widehat{\tau}_\alpha$ achieves realized Type-I errors close to the target level of 1% across all test sets, with FPRs ranging from 0.80% to 1.30%, which is consistent with the finite-sample guarantee in Theorem 3.1. At the same time, it maintains strong empirical power, reaching at least 99.90% on three generators and substantially improving performance on *Google-PaLM*. By contrast, the Youden-based threshold $\widehat{\tau}_{\mathrm{Y}}$ yields a single fixed operating point and does not explicitly target a prescribed Type-I error level. In our experiments, it behaves conservatively, with realized FPRs between 0.00% and 0.20%, all well below the target level of 1%. Although this still gives strong detection performance on some generators, it also reduces power by imposing a stricter threshold than necessary; this is most evident on *Google-PaLM*, where the TPR decreases from 92.10% under

$\widehat{\tau}_\alpha$ to 85.70% under $\widehat{\tau}_Y$. Overall, these results show that while $\widehat{\tau}_Y$ is suitable for a fixed balanced operating point, $\widehat{\tau}_\alpha$ is more flexible with respect to the desired false positive budget and can yield better power by avoiding unnecessarily conservative thresholding.

D Experiment Details

This section outlines the experimental setup and evaluation criteria.

Evaluation Metrics. To comprehensively evaluate the discriminative capability of each detector under different security requirements, we adopt the Area Under the Receiver Operating Characteristic curve (AUROC) and the True Positive Rate (TPR) at fixed False Positive Rate (FPR) thresholds, namely TPR@FPR=1% and TPR@FPR=0.01%. Since the empirical ROC curve is defined over discrete test samples, we use linear interpolation to report TPR at the specified FPR levels. In particular, with 1,000 human-written test samples, the smallest nonzero empirical FPR is 0.1%, so TPR@0.01% is an interpolated low-FPR summary rather than a directly observed operating point. Appendix F.2 provides a large-sample diagnostic of this interpolation.

Setup. To ensure comparability, we standardize backbone models across all detectors. We use *meta-llama/Llama-3.1-8B* as the base model for scoring, rewriting, and representation extraction. For training-free methods, the same open-source LLM is used as a surrogate to compute statistics (e.g., Likelihood, LRR, FDGPT). In Binoculars, *meta-llama/Llama-3.1-8B* serves as the observer and *meta-llama/Llama-3.1-8B-Instruct* as the performer. For training-based approaches (e.g., RoBERTa), we report both models fine-tuned on our data and publicly available checkpoints[7]. For rewrite-based methods, rewrites are generated by *meta-llama/Llama-3.1-8B-Instruct* with 4 samples, following [23].

For the experiments in Sections 4.1 and 4.2 (excluding the ID/OOD experiments), the detectors are trained on a mixed dataset composed of outputs from four LLMs and evaluated under diverse settings. We consistently use *meta-llama/Llama-3.1-8B* as the observer model, and *meta-llama/Llama-3.1-8B-Instruct* for methods requiring rewriting. The rewriting process incurs a substantial computational cost, amounting to approximately 300 GPU hours on NVIDIA A100 80GB GPUs. For experiments comparing with watermarking methods, we employ *Qwen/Qwen2.5-7B* as the base model to generate watermarked text using three different watermarking schemes.

For our method, unless otherwise specified, we adopt a fixed configuration across all experiments. The steering vector is injected at layer 11 through the residual connection. Representations are extracted from the last 8 transformer layers, where we average the final 25% tokens in each layer to obtain the representation. Training is conducted using AdamW [79] with a learning rate of 1×10^{-3}, batch size 8, and 10 epochs. The EMA decay for centroid updates is set to 0.9, and the optimization inverse temperature is set to $\beta = 2.5$. All hyperparameters are kept fixed across datasets and models without extensive tuning.

E Empirical Motivation for Representation Modeling

Our modeling choices for f_θ are supported by the following two empirical properties of hidden representations observed across models and domains:

1. **Norm concentration.** By pooling hidden states from the final 20% of tokens across the last $N = 8$ layers, we find that the resulting L_2 norms are tightly concentrated for *EleutherAI/GPT-J-6B*, *Qwen/Qwen2.5-7B*, and *meta-llama/Llama-3.1-8B* (Figure 5). This supports a hyperspherical approximation, where semantic information is primarily encoded in direction rather than magnitude.
2. **Directional unimodality.** On the unit sphere, the normalized representations exhibit a unimodal directional structure within each class (human or LLM), as evidenced by the distribution of $\widehat{\mu}^\top f_\theta(x)$ relative to the estimated class mean directions $\widehat{\mu}$ (Figure 6). Moreover, the estimated concentration parameters $\widehat{\kappa}$ are stable across domains and remain similar between the human and LLM classes.

[7] `https://huggingface.co/openai-community/roberta-large-openai-detector`

These observations support the use of a von Mises–Fisher (vMF) model as a simple and effective approximation for the directional geometry, with a shared concentration parameter κ.

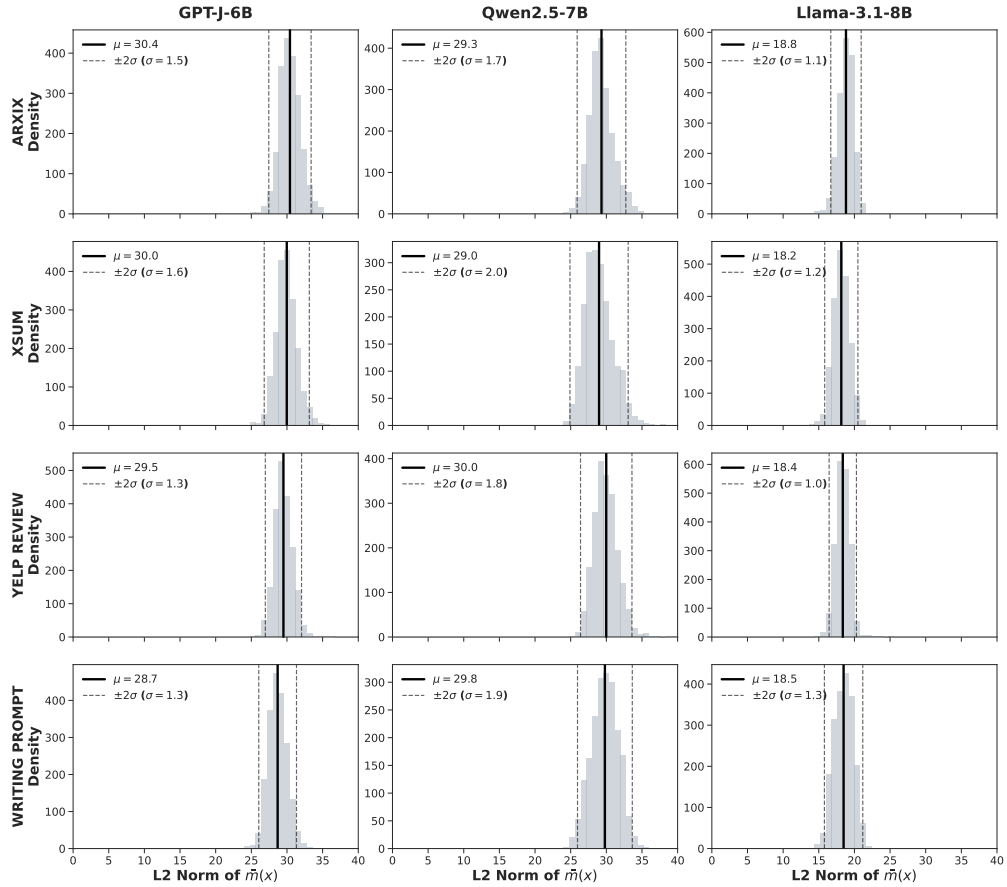


Figure 5: Empirical distributions of L_2 norms of representations obtained from the last 8 layers and the final 20% of tokens, across different models and domains. Columns correspond to *EleutherAI/GPT-J-6B*, *Qwen/Qwen2.5-7B*, and *meta-llama/Llama-3.1-8B*, while rows correspond to the Arxiv, XSum, Yelp, and Writing datasets. The solid and dashed red lines denote the mean and the $\pm 2\sigma$ intervals, respectively, with exact values reported in the upper-left legends.

F Additional Experiment results

F.1 Score Distributions

In this section, we visualize score distributions from four LLMs under different settings. We use *meta-llama/Llama-3.1-8B* as the observer model. The training data is generated by *GPT-3.5-Turbo*. The evaluated datasets are generated by *GPT-3.5-Turbo*, *Claude-Instant*, *Google-PaLM*, and *Llama-2-70B*.

In Figure 7, the leftmost column shows scores from RepreGuard [33], the middle column corresponds to raw representations without steering, and the rightmost column shows scores produced by our method. Although the first two approaches exploit representation-level separability, a non-negligible overlap between the two classes remains, which limits detection performance. This issue is particularly pronounced under OOD settings: RepreGuard exhibits 15.24% and 17.25% overlap on datasets generated by *Claude-Instant* and *Google-PaLM*, respectively, while S2D without steering yields 14.10% and 18.27%. In contrast, our method achieves substantially clearer separation between human-written and LLM-generated text, even under OOD settings, leading to improved discriminability.

F.2 Large-Sample Validation of Low-FPR Interpolation

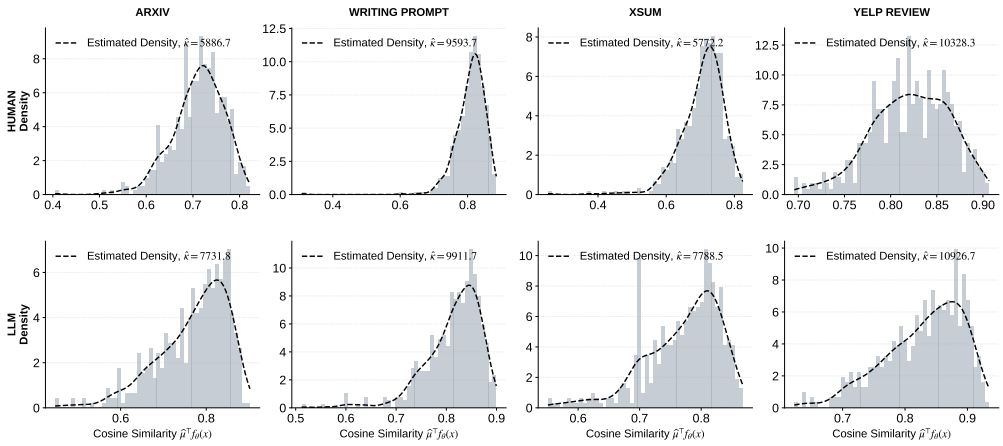


Figure 6: Empirical distributions of projected representations $\widehat{\mu}^\top f_\theta(x)$ across domains. Histograms show empirical frequencies, and dashed curves represent the corresponding estimated densities. The estimated concentration parameter κ is reported for each case.

Each DetectRL test set contains 1,000 human-written samples, so the smallest nonzero empirical FPR is 0.1%. Therefore, TPR@0.01% in the main experiments is obtained by linear interpolation of the empirical ROC curve. We conduct an additional large-sample experiment to examine the effect of this interpolation. Specifically, we collect 21686 human-written texts and 20,000 LLM-generated texts from C-ReD, EvoBench, and OpAI-Bench. The resulting empirical FPR resolution is approximately $1/21686 \approx 0.0046\%$, allowing us to evaluate performance directly near FPR $= 0.01\%$. We compare the interpolated and directly evaluated TPR@0.01% for S2D, RepreGuard, and ImBD. As shown in Table 7, the interpolated and direct estimates differ by at most 0.09 percentage points across the three methods. This suggests that, in this larger-sample evaluation, linear interpolation has little effect on the reported TPR@0.01% values.

Table 7: Large-sample validation of TPR@0.01%. All values are percentages.

Method	Interp.	Direct	$\lvert\mathrm{Gap}\rvert$
S2D	38.61	38.52	0.09
RepreGuard	13.99	13.99	0.00
ImBD	11.12	11.10	0.02

F.3 Shots of Training Dataset

We evaluate the impact of training set size on detection performance. The training sets are randomly sampled from a mixed-LLM dataset with sizes $\{16, 32, 64, 128, 196, 256, 384, 512\}$ pairs. We use *Llama-3.1-8B* as the base model and *Llama-3.1-8B-Instruct* as the rewriter, and evaluate performance on four test sets (*ChatGPT-3.5-Turbo*, *Claude-Instant*, *Google-PaLM*, and *Llama-2-70B*). As shown in Figure 8, S2D achieves strong data efficiency and consistent improvement over baselines, especially in low-resource settings. We observe a slight performance drop at the earliest stage as training size increases, since a very small number of samples can perturb the estimated score distribution. Nevertheless, even with as few as 16 pairs, S2D can still exploit the inherent score gap between human-written and LLM-generated texts. As more training data becomes available, the estimated distribution stabilizes, resulting in clearer separation and improved performance.

F.4 Long-Context Steering Stability

Since the same steering vector is injected at every valid token, we examine whether long sequences cause representation saturation or collapse. We train S2D on 512 mixed DetectRL human–LLM pairs using *Meta-Llama-3-8B* and evaluate a fixed steering checkpoint on a separate long-context diagnostic set constructed from OpAI-Bench, C-ReD, and EvoBench, with 400 human–LLM pairs at each sequence length. As shown in Table 8, performance improves with

Table 8: Long-context stability of S2D.

Tokens	AUROC	TPR@1%	Overlap
256	0.9107	0.610	0.105
512	0.9667	0.790	0.096
1024	0.9940	0.960	0.055
2048	0.9998	0.990	0.035
4096	0.9999	0.990	0.036

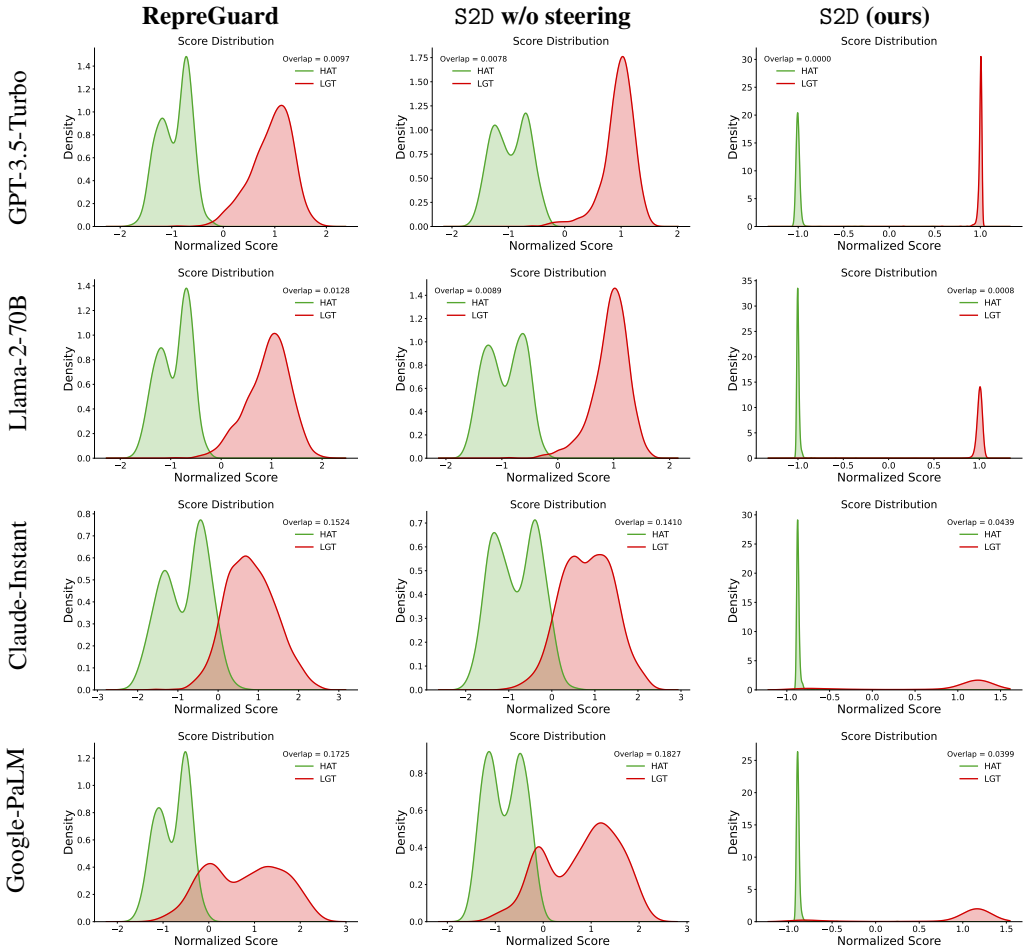


Figure 7: Score distributions of different detection methods involving hidden representations across datasets. Columns denote different methods and rows denote datasets generated by different LLMs. HAT and LGT denote human-authored text and LLM-generated text, respectively.

context length and plateaus beyond 2048 tokens, while score overlap decreases from 0.105 to about 0.035. We therefore observe no evidence of length-induced representation collapse up to 4,096 tokens. This result is an empirical diagnostic rather than a guarantee for arbitrarily long contexts.

F.5 Different Observers

Table 9 shows that strong observer models (e.g., *Llama-3.1-8B*, *Qwen2.5-7B*, and *Falcon-7B*) achieve consistently near-saturated performance across all generators, with high AUROC and stable TPR even at very low false positive rates. In contrast, weaker models (e.g., *Mistral-7B-v0.3*, *GPT-Neo-2.7B*, and *Gemma-2-9B*) exhibit both lower accuracy and larger variance across generators, suggesting sensitivity to distributional shifts and less discriminative features. Notably, instruction-tuned variants do not consistently improve performance, suggesting that detection effectiveness in S2D is primarily determined by representation quality, rather than model scale or alignment alone.

Observer Model↓	ChatGPT			Llama-2-70B			Google-PaLM			Claude-Instant			Avg.		
	AUROC	*TPR@1%*	*TPR@0.01%*	*AUROC*	*TPR@1%*	*TPR@0.01%*	*AUROC*	*TPR@1%*	*TPR@0.01%*	*AUROC*	*TPR@1%*	*TPR@0.01%*	*AUROC*	*TPR@1%*	*TPR@0.01%*
Llama-3.1-8B	99.99	99.90	99.80	98.95	98.90	98.10	99.83	98.50	97.40	99.70	98.90	96.60	99.62	99.05	97.98
Mistral-7B-v0.3	68.85	29.20	24.20	67.55	33.80	26.60	67.97	35.40	28.40	69.72	23.70	13.90	68.52	30.53	23.28
GPT-Neo-2.7B	84.44	33.70	8.30	79.97	41.50	19.40	79.63	20.40	6.70	85.46	35.00	10.90	82.38	32.65	11.33
OPT-2.7B	99.95	99.80	82.60	99.93	99.30	94.10	99.66	95.80	74.50	99.88	98.30	86.80	99.86	98.30	84.50
Qwen2.5-7B	99.99	99.90	99.80	99.99	99.90	98.90	99.73	98.60	96.20	99.99	99.90	99.70	99.93	99.58	98.65
Falcon-7B	99.99	99.90	99.80	99.99	99.90	99.60	99.89	98.10	95.70	99.98	99.70	98.00	99.96	99.40	98.28
Falcon-7B-Instruct	99.98	99.90	99.60	99.99	99.90	95.70	99.89	97.00	93.20	99.97	99.90	95.10	99.96	99.18	95.90
Gemma-2-9B	88.01	25.70	5.90	76.25	13.90	0.10	56.10	20.90	5.20	85.79	16.50	1.00	76.54	19.25	3.05
Gemma-2-9B-Instruct	96.25	62.70	32.50	95.66	60.10	29.70	54.10	15.60	7.30	70.34	12.50	6.10	79.09	37.73	18.90

Table 9: Impact of observer model. Detection performance of S2D across various observer models.

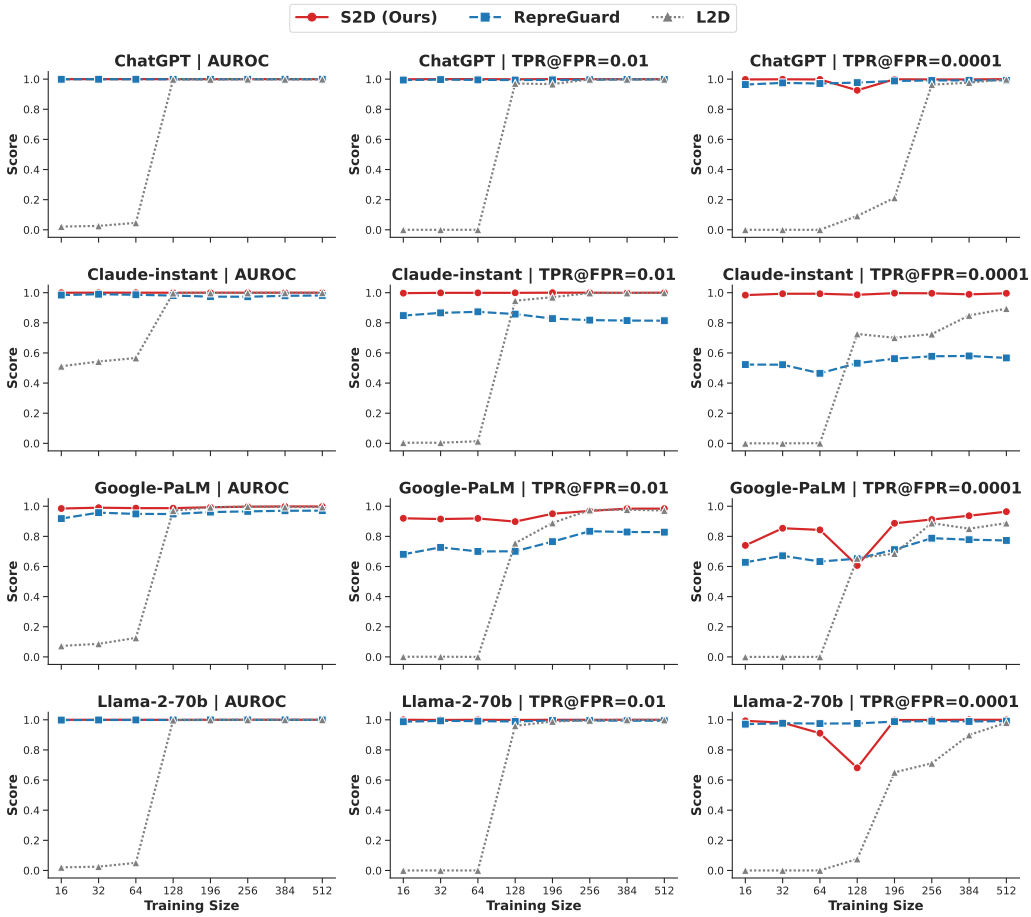


Figure 8: Detection performance across different training set sizes.

Table 10: Steering-layer sweeps across model scales. Here, $r = \ell/L$ denotes normalized injection depth. TPR@0.01% and overlap are reported at the best validated layer as mean $\pm$ standard deviation over three seeds.

Observer	Swept layers	High-performing interval	Best layer ℓ/L	TPR@0.01%	Overlap (%)
Llama-3.2-1B	2, 4, 6, 8, 10, 12, 14	4–6 ($r = 0.25$–0.375)	4/16	94.09 ± 2.94	2.29 ± 0.10
Meta-Llama-3-8B	4, 8, 12, 16, 20, 24, 28	8–16 ($r = 0.25$–0.50)	12/32	97.14 ± 0.66	1.92 ± 0.36
Meta-Llama-3-70B	10, 20, 30, 40, 50, 60, 70	20–50 ($r = 0.25$–0.625)	40/80	97.52 ± 0.54	1.97 ± 0.20

Steering Depth Across Model Scales. We examine how the preferred steering depth changes with model scale by sweeping injection layers on *Llama-3.2-1B*, *Meta-Llama-3-8B*, and *Meta-Llama-3-70B* under the same experimental protocol. Since AUROC is nearly saturated, we report TPR@0.01% and human–LLM score overlap. We define the high-performing interval as the largest contiguous region among the swept layers whose mean TPR@0.01% is within 10 percentage points of the model-specific maximum and whose overlap is at most 2.5%.

In Table 10, across all three scales, strong performance is attained over a range of intermediate depths rather than at a single layer. The best normalized depths shift from 0.25 to 0.375 and 0.50, while the high-performing region broadens from 0.25–0.375 to 0.25–0.625. Thus, the results do not support an exact proportional depth-scaling rule; instead, intermediate layers provide a robust search region whose effective range may broaden as model depth increases.

F.6 Computational Efficiency

We evaluate the computational efficiency of S2D in terms of both inference latency and training cost, and compare it against representative baselines including L2D, RepreGuard, and Binoculars.

Experimental Setup. All efficiency profiling is conducted on a single NVIDIA A100 (80GB) GPU. To ensure a fair evaluation, the training phase for all trainable methods is conducted on a mixed dataset comprising 512 text pairs generated by multiple LLMs, and all inference metrics are evaluated over a correspondingly mixed test set. Our test set has an average length of approximately 267 tokens, with 95% of the samples shorter than 435 tokens.

For all methods, we adopt *meta-llama/Llama-3.1-8B* as the observer/proxy model. For Binoculars, *meta-llama/Llama-3.1-8B* is used as the observer while *meta-llama/Llama-3.1-8B-Instruct* serves as the performer. For L2D, the number of rewrites is set to 4. Unless otherwise specified, we report per-sample inference latency with a batch size of 1. Peak memory usage is measured using `torch.cuda.max_memory_allocated()`. Importantly, for the rewrite-based baseline (L2D), the reported training and inference times strictly *exclude* the time required for the auxiliary LLM to autoregressively generate the rewrites, as these were pre-computed offline.

Method	Rewrite?	Efficiency				Performance (%)	
		Train Time (s) ↓	Train Cost (GB) ↓	Infer. Time (s) ↓	Infer. Cost (GB) ↓	AUROC ↑	TPR@1% ↑
Binoculars	No	-	-	0.50	58.0	87.70	74.70
L2D	Yes	1213.12	45.0	2.03	43.0	**98.96**	97.60
RepreGuard	No	835.62	42.0	0.32	**38.0**	98.42	81.47
S2D (ours)	No	**759.46**	**40.0**	**0.30**	39.0	98.90	**97.75**

Table 11: Comparison of computational cost and detection performance evaluated on a mixed test set. *Train Cost* and *Infer Cost* denote the peak memory allocated during the respective phases. Binoculars is a train-free method, hence its training costs are omitted. Note that for L2D, the heavy computational time required to generate rewrites is excluded from the timing metrics.

Results. During **inference**, S2D requires only a single forward pass through the frozen observer LLM. Even when L2D is evaluated under the favorable assumption that rewrite generation is excluded, S2D still achieves a **6.7× speedup** (0.30s vs. 2.03s, $2.03/0.30 \approx 6.7$). Compared with Binoculars, S2D substantially reduces inference latency and lowers memory usage by avoiding the need to load two LLMs simultaneously. Compared with RepreGuard, S2D incurs virtually no additional runtime overhead while maintaining comparable memory usage, indicating that steering introduces negligible deployment cost. During **training**, S2D keeps the observer LLM frozen and optimizes only a lightweight steering vector. As a result, it achieves the lowest training time among the evaluated trainable detectors while maintaining low training memory cost. Overall, these results show that S2D provides a scalable and deployable detection framework without sacrificing accuracy.

F.7 Interpretability Analysis of the Steering Vector

We further examine what the learned steering vector $\mathbf{v}$ represents and how it affects the observer model. For this analysis, we use the steering vector learned with *Meta-Llama-3-8B* and evaluate it on an independent held-out set of 512 human–LLM text pairs from DetectRL.

Vocabulary Projection. Let $\mathbf{w}_j$ denote the unembedding direction of token j. We first examine the alignment between $\mathbf{w}_j$ and $\mathbf{v}$ using $\mathbf{w}_j^\top \mathbf{v}$ and $\cos(\mathbf{w}_j, \mathbf{v})$. The most aligned tokens are heterogeneous, including multilingual fragments and incomplete subwords, and the maximum absolute cosine similarity is only 0.074. This suggests that $\mathbf{v}$ does not correspond to a small set of identifiable lexical features. Moreover, since $\mathbf{v}$ is injected at an intermediate layer and subsequently transformed by the remaining frozen Transformer blocks, direct vocabulary projection provides only a coarse interpretation of its effect.

Effect on Output Logits. We next examine the change in output logits induced by steering. Let $\Delta\ell_j(x) := \ell_j(x; \mathbf{v}) - \ell_j(x; \mathbf{0})$ denote the change in the output logit of token j. Tokens whose changes differ most between LLM-generated and human-written text include `incumbent`, `subsequently`, `aforementioned`, `plethora`, `concurrently`, and `rationale`, showing some association with formal written expression. However, direct vocabulary alignment poorly predicts these realized changes. The Spearman correlations between $\cos(\mathbf{w}_j, \mathbf{v})$ and the overall average change $\mathbb{E}[\Delta\ell_j(X)]$, and between $\cos(\mathbf{w}_j, \mathbf{v})$ and the class-conditional difference $\mathbb{E}[\Delta\ell_j(X) \mid Y = 1] - \mathbb{E}[\Delta\ell_j(X) \mid$

$Y = 0]$, are 0.061 and -0.035, respectively. In addition, the corresponding top-50 token sets have no overlap. Thus, the effect of steering cannot be explained well by nearest-token projections alone.

Effect on the Final Representation. Finally, we examine how steering changes the representation used by the detector. Recall from (4) that the detection score depends on the projection of $f_{\theta,\mathbf{v}}(x)$ onto the direction $\widehat{\boldsymbol{\mu}}_1 - \widehat{\boldsymbol{\mu}}_0$. We therefore consider $\langle \widehat{\boldsymbol{\mu}}_1 - \widehat{\boldsymbol{\mu}}_0,\ f_{\theta,\mathbf{v}}(X) - f_\theta(X) \rangle$. Its average value is -1.671 for human-written text and 0.388 for LLM-generated text, yielding an LLM-minus-human contrast of 2.059. Hence, although the same vector $\mathbf{v}$ is injected for every input, its effect after propagation through the remaining Transformer layers is class dependent rather than a uniform shift of the representation space.

Overall, these results suggest that $\mathbf{v}$ is better viewed as a context-dependent intervention direction, whose effect emerges through the subsequent frozen Transformer layers and is reflected in the final representation used for detection.

F.8 Additional Experiments

We further evaluate S2D on the recent OpAI-Bench [69] to examine whether the conclusions from DetectRL continue to hold for more recent generators and more challenging human-to-AI revision settings. OpAI-Bench covers four domains, including abstracts, essays, news, and reports, and contains texts produced or revised by *GPT-5.4*, *GPT-5.4-nano*, *Gemini-2.5-Flash*, and *Qwen3-8B*. Our setup uses 512 training pairs and 302 test documents. In addition to the baselines considered in the main experiments, we include several recent training-free detectors: Implicit Reward Model (*IRM*) [71], *Lastde++* [72], *SpecDetect++* [73], and Normalized Temperature Sensitivity (*NTS*) [74].

Evaluation on recent generators and detectors. Table 13 compares both training-free and train-based detectors on OpAI-Bench. Each entry reports AUROC / TPR@1% / TPR@0.1%, in percentage points. For training-free methods, no source-generator training is required. For train-based methods, we report results separately for each training generator; the corresponding diagonal entry is in-distribution, while the remaining columns measure cross-generator transfer. The recent training-free methods improve substantially over several classical likelihood-based detectors at low false-positive rates, with *IRM* generally providing the strongest low-FPR performance among this group. Nevertheless, a substantial gap remains between training-free detection and the strongest train-based approaches.

Across the train-based settings, S2D is generally the strongest method or competitive with the strongest baseline, particularly at low FPRs under cross-generator transfer. For example, when trained on *Gemini-2.5-Flash*, S2D achieves TPR@0.1% of 53.65, 74.92, and 89.40 on *GPT-5.4*, *GPT-5.4-nano*, and *Qwen3-8B*, respectively, substantially exceeding the corresponding values of the other train-based detectors. Similar gains are observed for several other training–test combinations. The advantage is not uniform across every metric, as RoBERTa-Large or RepreGuard can match or slightly outperform S2D in a few settings. Overall, these results show that the strong performance of S2D persists when both the generators and competing detectors are updated.

Table 12: Comparison with discriminative probes. Each entry reports AUROC / TPR@1%.

Revision	Method	ID	OOD
v4	LogReg	0.92 / 0.45	0.87 / 0.34
	Linear SVM	0.91 / 0.43	0.87 / 0.33
	MLP	0.92 / **0.50**	0.88 / 0.37
	S2D	**0.94** / 0.44	**0.90 / 0.38**
v6	LogReg	0.98 / 0.84	**0.96 / 0.71**
	Linear SVM	0.98 / 0.84	**0.96** / 0.70
	MLP	0.98 / 0.85	0.95 / 0.70
	S2D	**0.99 / 0.86**	**0.96 / 0.71**
v8	LogReg	**0.99** / 0.89	0.96 / 0.79
	Linear SVM	**0.99** / 0.90	0.96 / 0.79
	MLP	**0.99** / 0.91	0.96 / 0.77
	S2D	**0.99 / 0.92**	**0.97 / 0.81**

Comparison with discriminative probes. We next examine whether the performance of S2D can be reproduced by directly fitting a standard discriminative classifier to the observer representations. We compare against logistic regression, a linear SVM, and a small MLP. All methods use the same frozen Llama-3-8B observer, representation pooling strategy, and training data, while the probe baselines operate on the corresponding pooled representations without steering. We evaluate on the OpAI-Bench revision trajectories, where v4, v6, and v8 correspond to progressively stronger AI revisions of the same human-written documents, with approximately 50%, 75%, and 100% of the sentences edited, respectively. As shown in Table 12, S2D is generally the strongest method or tied

Table 13: Detection performance on OpAI-Bench. Each entry reports AUROC / TPR@1% / TPR@0.1%, in percentage points. For train-based methods, results are mean ± standard deviation. "–" denotes training-free methods that do not use a source generator.

Train	Method	Gemini-2.5-Flash	GPT-5.4	GPT-5.4-nano	Qwen3-8B
–	Likelihood	71.42 / 0.10 / 0.00	49.45 / 0.03 / 0.00	71.37 / 0.15 / 0.00	59.45 / 0.04 / 0.00
	LRR	77.73 / 3.66 / 0.76	51.09 / 0.22 / 0.00	72.25 / 0.22 / 0.00	60.00 / 0.22 / 0.00
	Fast-DetectGPT	73.66 / 8.78 / 2.53	49.79 / 1.33 / 0.30	76.62 / 12.92 / 3.06	47.34 / 1.66 / 0.40
	Binoculars	52.12 / 0.07 / 0.00	59.18 / 1.66 / 0.43	36.85 / 0.07 / 0.00	74.54 / 3.98 / 0.63
	IRM	72.33 / 17.93 / 6.70	59.52 / 6.32 / 1.49	66.28 / 8.30 / 1.69	74.02 / 28.19 / 10.95
	Lastde++	67.88 / 4.66 / 1.01	46.62 / 1.46 / 0.54	71.03 / 6.98 / 1.85	41.79 / 0.46 / 0.00
	SpecDetect++	58.36 / 2.07 / 0.70	43.59 / 0.76 / 0.00	62.46 / 4.40 / 1.75	38.43 / 0.45 / 0.11
	NTS	57.81 / 2.03 / 0.73	51.36 / 1.36 / 0.30	51.91 / 2.00 / 0.20	51.36 / 1.68 / 0.40
Gemini	RoBERTa-L	100.00±0.00 / 100.00±0.00 / 99.44±0.01	79.51±0.03 / 36.11±0.04 / 16.64±0.07	87.10±0.02 / 38.75±0.04 / 15.88±0.05	97.80±0.01 / 76.00±0.02 / 49.87±0.04
	ImBD	81.06±0.02 / 13.91±0.02 / 10.70±0.02	68.81±0.01 / 3.87±0.06 / 2.26±0.05	47.36±0.03 / 0.27±0.02 / 0.10±0.07	79.58±0.02 / 12.59±0.03 / 9.51±0.05
	RepreGuard	97.82±0.04 / 84.77±0.02 / 77.07±0.02	76.41±0.03 / 17.12±0.14 / 6.07±0.12	84.98±0.04 / 34.36±0.04 / 21.01±0.10	91.90±0.03 / 57.30±0.02 / 48.16±0.02
	S2D	**100.00**±0.00 / **100.00**±0.00 / **100.00**±0.00	**89.81**±0.02 / **60.53**±0.05 / **53.65**±0.06	**94.75**±0.01 / **80.97**±0.03 / **74.92**±0.06	97.25±0.00 / **92.92**±0.01 / **89.40**±0.03
GPT-5.4	RoBERTa-L	**98.76**±0.02 / 83.86±0.22 / 59.14±0.12	98.38±0.01 / 70.28±0.13 / **44.93**±0.25	97.38±0.02 / **69.62**±0.21 / 41.85±0.26	95.32±0.02 / 62.98±0.19 / 41.22±0.10
	ImBD	72.38±0.03 / 2.34±0.24 / 1.71±0.10	87.99±0.02 / 16.72±0.15 / 11.03±0.23	83.28±0.03 / 9.07±0.23 / 7.77±0.28	83.43±0.03 / 11.31±0.21 / 6.39±0.08
	RepreGuard	97.12±0.02 / 80.75±0.32 / **72.64**±0.02	80.46±0.16 / 21.09±0.36 / 11.54±0.20	87.89±0.02 / 44.95±0.01 / 29.18±1.20	92.14±0.52 / 57.42±0.36 / 47.41±1.02
	S2D	97.67±0.01 / **92.13**±0.23 / 71.58±0.11	**98.56**±0.00 / **81.04**±0.14 / 38.35±0.24	**98.36**±0.01 / 68.29±0.22 / **49.94**±0.27	**96.47**±0.01 / **73.09**±0.20 / **56.38**±0.09
GPT-5.4-nano	RoBERTa-L	99.29±0.01 / 90.44±0.01 / 83.32±0.02	93.88±0.05 / 61.28±0.08 / 51.83±0.03	**99.67**±0.01 / 95.04±0.02 / 90.77±0.02	97.17±0.02 / 84.12±0.02 / 78.35±0.02
	ImBD	64.24±0.02 / 1.99±0.02 / 0.86±0.03	83.40±0.03 / 13.25±0.10 / 7.78±0.01	86.34±0.02 / 19.91±0.01 / 13.68±0.03	86.53±0.03 / 18.89±0.03 / 14.05±0.03
	RepreGuard	97.55±0.02 / 82.15±0.02 / 77.35±0.02	81.13±0.02 / 27.33±0.12 / 23.64±0.10	90.24±0.04 / 56.27±0.02 / 49.08±0.02	92.69±0.03 / 57.32±0.03 / 52.09±0.04
	S2D	**99.97**±0.00 / **99.47**±0.00 / **98.57**±0.01	**95.93**±0.04 / **70.32**±0.09 / **59.93**±0.02	99.64±0.00 / **96.20**±0.01 / **94.16**±0.01	**98.16**±0.01 / **93.53**±0.01 / **89.69**±0.01
Qwen3-8B	RoBERTa-L	99.74±0.01 / 97.04±0.01 / 68.90±0.02	89.82±0.02 / 36.47±0.04 / 10.37±0.06	94.95±0.02 / 64.28±0.04 / 26.62±0.08	**99.42**±0.01 / **94.38**±0.01 / 75.63±0.02
	ImBD	72.73±0.02 / 4.98±0.02 / 4.67±0.03	74.36±0.01 / 4.00±0.06 / 3.38±0.08	64.85±0.03 / 1.68±0.02 / 1.36±0.10	90.36±0.02 / 32.81±0.02 / 30.07±0.03
	RepreGuard	96.53±0.02 / 78.81±0.02 / 62.18±0.03	71.61±0.12 / 11.71±0.23 / 3.65±0.12	80.13±0.03 / 23.87±0.01 / 8.29±0.14	91.61±0.02 / 61.85±0.02 / 44.35±0.02
	S2D	**100.00**±0.00 / **100.00**±0.00 / **99.46**±0.01	**94.26**±0.01 / **56.79**±0.05 / **41.04**±0.07	**96.69**±0.01 / **82.52**±0.03 / **65.49**±0.09	99.20±0.00 / 91.36±0.00 / **88.01**±0.01

with the strongest probe. The advantage is particularly consistent under cross-generator evaluation: `S2D` achieves the highest OOD AUROC and TPR@1% for `v4` and `v8`, while matching the best probe for `v6`. The gains are not uniform across all metrics; for example, the MLP achieves a higher ID TPR@1% on `v4`. Overall, directly fitting a discriminative classifier to the unsteered representations does not consistently recover the performance of `S2D`, particularly under cross-generator transfer. However, since `S2D` combines steering with a vMF-based readout whereas the probe baselines use unsteered representations with different classifiers, this comparison does not fully disentangle the individual contributions of steering and the downstream classifier.

G Omitted Theoretical Details and Proofs

G.1 Proof of Theorem 3.1

Proof of Theorem 3.1. Condition on the training sample $\mathcal{S}_{\text{train}}$, the estimated scoring function $\widehat{\mathcal{S}}_t$ is fixed, and the only remaining randomness comes from the calibration sample $\mathcal{S}_{\text{cal}} = \{x_i^-\}_{i=1}^{n_2}$ with $x_i^- \overset{\text{i.i.d.}}{\sim} \mathbb{P}_0$.

Define the true survival function $\mathcal{T}(\tau) := \mathbb{P}_0\big(\widehat{\mathcal{S}}_t(X_{\text{test}}^-) > \tau\big)$ (where X_{test}^- follows the same distribution of samples in $\mathcal{S}_{\text{cal}}$) and its empirical counterpart $\widehat{\mathcal{T}}_{n_2}(\tau) := \frac{1}{n_2}\sum_{i=1}^{n_2} \mathbb{1}\big(\widehat{\mathcal{S}}_t(x_i^-) > \tau\big)$. By the definition of the empirical threshold $\widehat{\tau}_{\alpha,t}$ in Equation (5), it is the infimum of τ such that $\widehat{\mathcal{T}}_{n_2}(\tau) \le \alpha$. On one hand, by definition, we have $\widehat{\mathcal{T}}_{n_2}(\widehat{\tau}_{\alpha,t}) \le \alpha$. On the other hand, by the continuity assumption, the calibration scores are almost surely distinct, so $\widehat{\mathcal{T}}_{n_2}(\tau)$ is a step function whose jumps have size $1/n_2$. The infimum definition therefore guarantees that the empirical probability does not fall below α by more than $1/n_2$. Therefore, $\alpha - \widehat{\mathcal{T}}_{n_2}(\widehat{\tau}_{\alpha,t}) \le \frac{1}{n_2}$. It implies that

$$\left|\widehat{\mathcal{T}}_{n_2}(\widehat{\tau}_{\alpha,t}) - \alpha\right| \le \frac{1}{n_2}.$$

By applying the triangle inequality, we can bound the absolute difference between the population Type-I error and the target level α

$$\begin{aligned}
|\mathcal{T}(\widehat{\tau}_{\alpha,t}) - \alpha| &\le \left|\mathcal{T}(\widehat{\tau}_{\alpha,t}) - \widehat{\mathcal{T}}_{n_2}(\widehat{\tau}_{\alpha,t})\right| + \left|\widehat{\mathcal{T}}_{n_2}(\widehat{\tau}_{\alpha,t}) - \alpha\right| \\
&\le \sup_{\tau\in\mathbb{R}} \left|\mathcal{T}(\tau) - \widehat{\mathcal{T}}_{n_2}(\tau)\right| + \frac{1}{n_2}.
\end{aligned}$$

By using the Dvoretzky–Kiefer–Wolfowitz (DKW) inequality [80], the uniform deviation between the true and empirical functions is bounded by

$$\mathbb{P}\left(\sup_{\tau\in\mathbb{R}}\left|\mathcal{T}(\tau)-\widehat{\mathcal{T}}_{n_2}(\tau)\right|>\sqrt{\frac{\log(2/\delta)}{2n_2}}\,\middle|\,\mathcal{S}_{\text{train}}\right)\le 2e^{-2n_2\left(\sqrt{\frac{\log(2/\delta)}{2n_2}}\right)^2}=\delta.$$

Hence, conditional on $\mathcal{S}_{\text{train}}$, with probability at least $1-\delta$, we have

$$|\mathcal{T}(\widehat{\tau}_{\alpha,t})-\alpha|\le\sqrt{\frac{\log(2/\delta)}{2n_2}}+\frac{1}{n_2}.$$

Since $\mathbb{P}_0(\widehat{\mathcal{S}}_t(X^-_{\text{test}})>\widehat{\tau}_{\alpha,t})=\mathcal{T}(\widehat{\tau}_{\alpha,t})$, we can rewrite the conditional probability as:

$$\mathbb{P}\left(\left|\mathbb{P}_0\big(\widehat{\mathcal{S}}_t(X^-_{\text{test}})>\widehat{\tau}_{\alpha,t}\big)-\alpha\right|\le\sqrt{\frac{\log(2/\delta)}{2n_2}}+\frac{1}{n_2}\,\middle|\,\mathcal{S}_{\text{train}}\right)\ge 1-\delta,$$

which yields

$$\mathbb{P}\left(\left|\mathbb{P}_0\big(\widehat{\mathcal{S}}_t(X^-_{\text{test}})>\widehat{\tau}_{\alpha,t}\big)-\alpha\right|\le\sqrt{\frac{\log(2/\delta)}{2n_2}}+\frac{1}{n_2}\right)\ge 1-\delta,$$

which completes the proof. □

G.2 Formal Statement for Theorem 3.3

Theorem 3.3 characterizes the excess Type II error of the detector trained on $\mathcal{S}_{\text{train}}$. To establish this result, we introduce several additional technical assumptions. We first present and discuss these assumptions, followed by the formal statement of Theorem 3.3.

Two-Time-Scale Prototype Tracking. Algorithm 1 can be viewed as a two-time-scale stochastic approximation scheme for solving the empirical objective (6):

$$\mathcal{L}_{\text{vMF}}(\mathbf{v},\boldsymbol{\mu}_0,\boldsymbol{\mu}_1):=\frac{1}{n_1}\sum_{i=1}^{n_1}\log q_\beta\left(y_i\mid f_{\theta,\mathbf{v}}(x_i),\boldsymbol{\mu}_0,\boldsymbol{\mu}_1\right),\tag{6}$$

where $\mathcal{S}_{\text{train}}=\{(x_i,y_i)\}_{i=1}^{n_1}$ consists of i.i.d. samples from $\mathcal{P}$.

We restate the key update steps for clarity and subsequent analysis. At iteration t, let $\mathbf{v}_t$ denote the current steering vector. For each class c, define

$$\bar{\mathbf{z}}_{c,t}(\mathbf{v}_t)=\frac{1}{|\{(x,y):x\in\mathcal{B}_t,y=c\}|}\sum_{(x,y):x\in\mathcal{B}_t,y=c}f_{\theta,\mathbf{v}_t}(x)$$

as the average representation of class c over a fresh mini-batch $\mathcal{B}_t$. The update rule is given by

$$\textbf{Fast Process (Mean Direction Tracking)}\quad\widehat{\boldsymbol{\mu}}_{c,t+1}=\Pi_{\mathbb{S}^{d-1}}\left((1-\rho)\widehat{\boldsymbol{\mu}}_{c,t}+\rho\,\bar{\mathbf{z}}_{c,t}(\mathbf{v}_t)\right),\tag{7a}$$

$$\textbf{Slow Process (Steering Vector Update)}\quad\mathbf{v}_{t+1}=\mathbf{v}_t+\eta\nabla_{\mathbf{v}}\mathcal{L}_{\text{vMF}}\big(\mathbf{v}_t,\widehat{\boldsymbol{\mu}}_{0,t},\widehat{\boldsymbol{\mu}}_{1,t}\big),\tag{7b}$$

where $\Pi_{\mathbb{S}^{d-1}}(\mathbf{x})=\mathbf{x}/\|\mathbf{x}\|$ denotes projection onto the unit sphere. Here, $\widehat{\boldsymbol{\mu}}_{c,t}$ serves as an estimator of the population mean direction for class c at iteration t. We consider a two-time-scale regime with $0<\eta\ll\rho\le 1$, under which $\{\widehat{\boldsymbol{\mu}}_{c,t}\}_{c=0}^1$ evolves on the fast timescale while $\mathbf{v}_t$ evolves on the slow timescale.

Assumptions. Assumption 1 introduces a generative model under which the representation follows a vMF distribution at the population optimum. This assumption facilitates the theoretical analysis.

Assumption 1 (Local vMF Model)**.** *For every* $\mathbf{v}$ *and each class* $c\in\{0,1\}$*, define* $Z_{\mathbf{v}}=f_{\theta,\mathbf{v}}(X)$ *and assume*

$$Z_{\mathbf{v}}\mid Y=c\sim\text{vMF}(\boldsymbol{\mu}_c(\mathbf{v}),\kappa),$$

that is, its density is given by $p(z\mid Y=c)=C_d(\kappa)\exp\big(\kappa\,\boldsymbol{\mu}_c(\mathbf{v})^\top z\big)$ *for any* $z\in\mathbb{S}^{d-1}$*, where* $C_d(\kappa)$ *is the normalization constant.*

Remark G.1 (Gradient Noises). *Let $\mathcal{F}_t$ denote the filtration generated by the optimization trajectory up to iteration t. Under Assumption 1, the conditional expectation of the mini-batch average satisfies*

$$\mathbb{E}[\bar{\mathbf{z}}_{c,t}(\mathbf{v}_t) \mid \mathcal{F}_t] = A_d(\kappa)\boldsymbol{\mu}_c(\mathbf{v}_t),$$

where $A_d(\kappa) = \frac{I_{d/2}(\kappa)}{I_{d/2-1}(\kappa)}$, and $I_\nu(\cdot)$ denotes the modified Bessel function of the first kind of order ν. Based on the observation, we can define the following gradient noise

$$\boldsymbol{\zeta}_{c,t+1} := \bar{\mathbf{z}}_{c,t}(\mathbf{v}_t) - A_d(\kappa)\boldsymbol{\mu}_c(\mathbf{v}_t), \tag{8}$$

for a class $c \in \{0,1\}$. We then have that $\mathbb{E}\big[\boldsymbol{\zeta}_{c,t+1} \mid \mathcal{F}_t\big] = 0$ and $\|\boldsymbol{\zeta}_{c,t+1}\| \le 2$ almost surely due to the fact that $\|\bar{\mathbf{z}}_{c,t}(\mathbf{v}_t)\|, A_d(\kappa)$, and $\|\boldsymbol{\mu}_c(\mathbf{v}_t)\| \le 1$.

Assumption 2 (Local Tracking Conditions). *Define $\boldsymbol{\mu}_{c,t} := \boldsymbol{\mu}_c(\mathbf{v}_t)$ and assume that*

1. ***(Bounded drift)** There exists $C_\mu > 0$ such that $\|\boldsymbol{\mu}_{c,t+1} - \boldsymbol{\mu}_{c,t}\| \le C_\mu \eta$.*
2. ***(Initialization)** $\|\widehat{\boldsymbol{\mu}}_{c,0} - \boldsymbol{\mu}_{c,0}\| \le 1/4$.*

Theorem G.2 (Local Mean Direction Tracking). *Fix $c \in \{0,1\}$ and write $\boldsymbol{\mu}_{c,t} := \boldsymbol{\mu}_c(\mathbf{v}_t)$. Consider the recursion*

$$\mathbf{z}_{c,t+1} = (1-\rho)\widehat{\boldsymbol{\mu}}_{c,t} + \rho\bar{\mathbf{z}}_{c,t}(\mathbf{v}_t), \qquad \widehat{\boldsymbol{\mu}}_{c,t+1} = \frac{\mathbf{z}_{c,t+1}}{\|\mathbf{z}_{c,t+1}\|}.$$

Under Assumptions 1 and 2, there exist constants $c, C, \rho_0, \gamma_0 > 0$ such that, if

$$0 < \rho \le \rho_0, \qquad \frac{\eta}{\rho} \le \gamma_0, \qquad \rho \log\frac{2T}{\delta} \le 1,$$

and

$$\|\widehat{\boldsymbol{\mu}}_{c,0} - \boldsymbol{\mu}_{c,0}\|^2 + C\left(\sqrt{\rho\log\frac{2T}{\delta}} + \frac{\eta^2}{\rho^2}\right) \le \frac{1}{16},$$

then, with probability at least $1-\delta$,

1. *For all $0 \le t \le T$, $\|\widehat{\boldsymbol{\mu}}_{c,t} - \boldsymbol{\mu}_{c,t}\|^2 \le (1-c\rho)^t\|\widehat{\boldsymbol{\mu}}_{c,0} - \boldsymbol{\mu}_{c,0}\|^2 + C\left(\sqrt{\rho\log\frac{2T}{\delta}} + \frac{\eta^2}{\rho^2}\right)$.*
2. *For all $0 \le t \le T$, $\|\widehat{\boldsymbol{\mu}}_{c,t} - \boldsymbol{\mu}_{c,t}\| \le \frac{1}{4}$, $\|\mathbf{z}_{c,t+1}\| \ge \frac{1}{4}$, $\langle\widehat{\boldsymbol{\mu}}_{c,t}, \boldsymbol{\mu}_{c,t}\rangle > 0$.*

Theorem G.2 establishes that the proposed exponential moving average update $\widehat{\boldsymbol{\mu}}_{c,t}$ can accurately track the evolving class prototypes $\boldsymbol{\mu}_{c,t}$ under the local vMF model. Despite stochastic gradient noise and slow drift in the true mean direction, the estimated prototype remains close to the target with high probability. The result shows a contraction behavior up to a steady-state error, where the averaging parameter and the drift rate govern the tracking accuracy. Its proof is deferred in Appendix G.4.

Since the score function depends on the class prototypes only through their difference, the tracking error bound in Theorem G.2 directly translates into a uniform bound on the deviation between the plug-in and oracle score functions (i.e., $\widehat{\mathcal{S}}_t$ and $\mathcal{S}_t$), as formalized in the following corollary.

Corollary G.3. *Define the oracle and plug-in log-likelihood ratios with respect to the same steered representation $f_{\theta,\mathbf{v}_t}(x)$ as*

$$\mathcal{S}_t(x) = \kappa(\boldsymbol{\mu}_{1,t} - \boldsymbol{\mu}_{0,t})^\top f_{\theta,\mathbf{v}_t}(x), \qquad \widehat{\mathcal{S}}_t(x) = \kappa(\widehat{\boldsymbol{\mu}}_{1,t} - \widehat{\boldsymbol{\mu}}_{0,t})^\top f_{\theta,\mathbf{v}_t}(x).$$

Under Assumptions 1 and 2, for any $\delta \in (0,1)$, suppose that the conditions of Theorem G.2 hold for both classes with failure probability $\delta/2$. Then, with probability at least $1-\delta$, there exist constants $c \in (0,1)$ and $C > 0$ such that the following uniform bound holds for all $0 \le t \le T$:

$$\sup_{x\in\mathcal{X}}\left|\widehat{\mathcal{S}}_t(x) - \mathcal{S}_t(x)\right| \le r_t(\delta,\rho,\eta),$$

where

$$r_t(\delta,\rho,\eta) := \kappa C\sqrt{(1-c\rho)^t + \sqrt{\rho\log\frac{4T}{\delta}} + \frac{\eta^2}{\rho^2}}.$$

Proof of Corollary G.3. Apply Theorem G.2 separately to $c = 0$ and $c = 1$, each with failure probability $\delta/2$. By a union bound, with probability at least $1 - \delta$, the tracking bounds hold simultaneously for both classes and for all $0 \le t \le T$.

By definition, we have

$$\begin{aligned}\left|\widehat{\mathcal{S}}_t(x) - \mathcal{S}_t(x)\right| &= \left|\kappa(\widehat{\boldsymbol{\mu}}_{1,t} - \widehat{\boldsymbol{\mu}}_{0,t})^\top f_{\theta,\mathbf{v}}(x) - \kappa(\boldsymbol{\mu}_{1,t} - \boldsymbol{\mu}_{0,t})^\top f_{\theta,\mathbf{v}}(x)\right| \\ &\overset{(a)}{\le} \kappa\left(\|\widehat{\boldsymbol{\mu}}_{0,t} - \boldsymbol{\mu}_{0,t}\| + \|\widehat{\boldsymbol{\mu}}_{1,t} - \boldsymbol{\mu}_{1,t}\|\right) \\ &\overset{(b)}{\le} 2\kappa\sqrt{(1-c\rho)^t \cdot \frac{1}{16} + C\left(\sqrt{\rho\log\frac{4T}{\delta}} + \frac{\eta^2}{\rho^2}\right)} \\ &\overset{(c)}{\le} \kappa C\sqrt{(1-c\rho)^t + \sqrt{\rho\log\frac{4T}{\delta}} + \frac{\eta^2}{\rho^2}}.\end{aligned}$$

where (a) follows from the triangle inequality and the fact that $\|f_{\theta,\mathbf{v}}(x)\| \le 1$, (b) applies Theorem G.2, and (c) enlarges the constant C properly. □

Assumption 3 (Local Mass Condition Around the Threshold)**.** *There exist constants $c_0, C_0 > 0$, exponents $\underline{\gamma}, \bar{\gamma} > 0$ satisfying $\underline{\gamma} \le 1 + \bar{\gamma}$, and $\varepsilon_0 > 0$ such that for all $\varepsilon \in [0, \varepsilon_0]$,*

$$\mathbb{P}_0(|\mathcal{S}(X) - \tau_\alpha^\star| \le \varepsilon) \le C_0\varepsilon^{\bar{\gamma}},$$

and

$$\min\left\{\mathbb{P}_0(\tau_\alpha^\star - \varepsilon \le \mathcal{S}(X) < \tau_\alpha^\star),\, \mathbb{P}_0(\tau_\alpha^\star < \mathcal{S}(X) \le \tau_\alpha^\star + \varepsilon)\right\} \ge c_0\varepsilon^{\underline{\gamma}}.$$

This assumption imposes regularity conditions on the null score distribution in a neighborhood of $\tau_\alpha^\star$. The upper bound controls the concentration of probability mass near the threshold, corresponding to a standard margin-type condition in classification theory [81–83, 28]. The lower bound ensures that there is sufficient probability mass around the threshold, preventing the distribution from being too sparse in this region.

Theorem G.4 (Formal Statement of Theorem 3.3)**.** *Let Assumptions 1-3 hold, and suppose that the conditions of Theorem G.2 hold for both classes with failure probability $\delta/2$. Suppose further that the locality condition in Lemma G.6 holds. Then there exists a constant $c \in (0,1)$ such that, with probability at least $1 - 2\delta$ over the randomness in the training procedure and the calibration sample, the excess Type-II error satisfies*

$$\mathbb{P}_1(\widehat{\mathcal{R}}_t) - \mathbb{P}_1(\mathcal{R}_t^\star) \le \mathcal{O}\Bigg(\underbrace{\left((1-c\rho)^t + \sqrt{\rho\log(4T/\delta)} + \eta^2/\rho^2\right)^{\frac{1+\bar{\gamma}}{2}}}_{\textit{Estimation Error}} + \underbrace{\sqrt{\frac{\log(2/\delta)}{n_2}} + \frac{1}{n_2}}_{\textit{Calibration Error}}\Bigg)$$

where $t \in [1, T]$ is the iteration index within the total horizon T, $\rho \in (0,1)$ is the EMA coefficient satisfying $\rho \le 1/c$, $\eta > 0$ is the steering learning rate, n_2 is the calibration sample size, $\delta \in (0,1)$ is the confidence parameter, and $\underline{\gamma}, \bar{\gamma}$ are the constants in Assumption 3.

G.3 Proof of Theorem G.4

Proof of Theorem G.4. For notational simplicity, we suppress the time index t. The excess Type-II error admits the decomposition given in Lemma G.5. We prove Lemma G.5 in Appendix G.5.

Lemma G.5. *Let p_0 and p_1 denote the density functions of $\mathbb{P}_0$ and $\mathbb{P}_1$, respectively. Define the score function $\mathcal{S}(x) = \log\frac{p_1(x)}{p_0(x)}$, and the acceptance regions*

$$\mathcal{R}^\star = \{x : \mathcal{S}(x) \le \tau_\alpha^\star\}, \qquad \widehat{\mathcal{R}} = \left\{x : \widehat{\mathcal{S}}(x) \le \widehat{\tau}_\alpha\right\}.$$

Then,

$$\mathbb{P}_1(\widehat{\mathcal{R}}) - \mathbb{P}_1(\mathcal{R}^\star) = \underbrace{\int_{\widehat{\mathcal{R}}\Delta\mathcal{R}^\star}\left|\frac{p_1(x)}{p_0(x)} - e^{\tau_\alpha^\star}\right|\mathrm{d}\mathbb{P}_0(x)}_{\textit{Estimation Error}} + \underbrace{e^{\tau_\alpha^\star}\left(\alpha - \mathbb{P}_0(\widehat{\mathcal{R}}^c)\right)}_{\textit{Price of Conservativeness}}. \tag{9}$$

Here, $\widehat{\mathcal{R}}\Delta\mathcal{R}^\star$ denotes the symmetric difference, i.e., $\widehat{\mathcal{R}}\Delta\mathcal{R}^\star = (\widehat{\mathcal{R}} \setminus \mathcal{R}^\star) \cup (\mathcal{R}^\star \setminus \widehat{\mathcal{R}})$.

From (9), there are two terms in the decomposition of the excess Type-II error.

(i) The second term quantifies the gap between empirical Type I error and the target level α. By Theorem 3.1, with probability at least $1-\delta$, we have $\left|\mathbb{P}_0(\widehat{\mathcal{R}}^c) - \alpha\right| \le \Delta_{n_2}(\delta)$ with $\Delta_{n_2}(\delta) := \sqrt{\frac{\log(2/\delta)}{2n_2}} + \frac{1}{n_2}$, as a result of which,

$$e^{\tau_\alpha^\star}\left(\alpha - \mathbb{P}_0(\widehat{\mathcal{R}}^c)\right) \le e^{\tau_\alpha^\star}\Delta_{n_2}(\delta). \tag{10}$$

(ii) The first term on the RHS of (9) is due to the estimation error between the oracle score $\mathcal{S}$ and the estimated $\widehat{\mathcal{S}}$. To bound it, we have to bound the discrepancy of the acceptance regions in the score space. By Corollary G.3, with probability at least $1-\delta$, we have $\sup_{x\in\mathcal{X}} |\widehat{\mathcal{S}}(x) - \mathcal{S}(x)| \le r(\delta,\rho,\eta)$.

Since both $\boldsymbol{\mu}_c$ and $f_{\theta,\mathbf{v}}(x)$ lie on the unit hypersphere, the log-scores are deterministically bounded within $[-2\kappa, 2\kappa]$. Therefore, the exponential mapping is Lipschitz continuous with constant $e^{2\kappa}$ over this domain. For any $x \in \widehat{\mathcal{R}}\Delta\mathcal{R}^\star$, it follows that

$$\left|\frac{p_1(x)}{p_0(x)} - e^{\tau_\alpha^\star}\right| = \left|e^{\mathcal{S}(x)} - e^{\tau_\alpha^\star}\right| \le e^{2\kappa}\left|\mathcal{S}(x) - \tau_\alpha^\star\right|. \tag{11}$$

Lemma G.6. *Suppose*

$$\left|\mathbb{P}_0(\widehat{\mathcal{R}}^c) - \alpha\right| \le \Delta_{n_2}(\delta), \qquad \sup_{x\in\mathcal{X}}\left|\widehat{\mathcal{S}}(x) - \mathcal{S}(x)\right| \le r(\delta,\rho,\eta),$$

and

$$2r(\delta,\rho,\eta) + \left(\frac{\Delta_{n_2}(\delta)}{c_0}\right)^{1/\underline{\gamma}} \le \varepsilon_0.$$

Then, under Assumption 3,

$$\widehat{\mathcal{R}}\Delta\mathcal{R}^\star \subseteq \left\{x : |\mathcal{S}(x) - \tau_\alpha^\star| < 2r(\delta,\rho,\eta) + \left(\frac{\Delta_{n_2}(\delta)}{c_0}\right)^{1/\underline{\gamma}}\right\}. \tag{12}$$

Consequently,

$$\mathbb{P}_0\left(\widehat{\mathcal{R}}\Delta\mathcal{R}^\star\right) \le C_0\left[2r(\delta,\rho,\eta) + \left(\frac{\Delta_{n_2}(\delta)}{c_0}\right)^{1/\underline{\gamma}}\right]^{\bar{\gamma}}. \tag{13}$$

Now, we are ready to analyze the first term.

$$\begin{aligned}
&\int_{\widehat{\mathcal{R}}\Delta\mathcal{R}^\star}\left|\frac{p_1(x)}{p_0(x)} - e^{\tau_\alpha^\star}\right| \mathrm{d}\mathbb{P}_0(x)\\
&\overset{(11)}{\le} e^{2\kappa}\int_{\widehat{\mathcal{R}}\Delta\mathcal{R}^\star}\left|\mathcal{S}(x) - \tau_\alpha^\star\right| \mathrm{d}\mathbb{P}_0(x)\\
&\overset{(12)}{\le} e^{2\kappa}\left[2r(\delta,\rho,\eta) + \left(\frac{\Delta_{n_2}(\delta)}{c_0}\right)^{1/\underline{\gamma}}\right]\mathbb{P}_0\left(\widehat{\mathcal{R}}\Delta\mathcal{R}^\star\right)\\
&\overset{(13)}{\le} e^{2\kappa}C_0\left[2r(\delta,\rho,\eta) + \left(\frac{\Delta_{n_2}(\delta)}{c_0}\right)^{1/\underline{\gamma}}\right]^{1+\bar{\gamma}}.
\end{aligned} \tag{14}$$

With probability at least $1-2\delta$, we have both Theorem 3.1 and Corollary G.3 hold. Conditioned on this joint event, combining (10) and (14) with the decomposition (9) yields

$$\begin{aligned}
&\mathbb{P}_1(\widehat{\mathcal{R}}_t) - \mathbb{P}_1(\mathcal{R}_t^\star)\\
&\le e^{2\kappa}C_0\left[2r_t(\delta,\rho,\eta) + \left(\frac{1}{c_0}\left(\sqrt{\frac{\log(2/\delta)}{2n_2}} + \frac{1}{n_2}\right)\right)^{1/\underline{\gamma}}\right]^{1+\bar{\gamma}} + e^{\tau_{\alpha,t}^\star}\left(\sqrt{\frac{\log(2/\delta)}{2n_2}} + \frac{1}{n_2}\right)
\end{aligned}$$

By applying the basic inequality $\sqrt{a+b+c} \le \sqrt{a} + \sqrt{b} + \sqrt{c}$ and $(x+y)^p \le 2^{(p-1)_+}(x^p + y^p)$ to decouple the terms and using Corollary G.3, we abstract the multiplicative constants (including $e^{2\kappa}$, C_0, C_1, and $e^{\tau^\star_{\alpha,t}}$) into the asymptotic notation to obtain the final explicit finite-sample bound

$$\mathbb{P}_1(\widehat{\mathcal{R}}_t) - \mathbb{P}_1(\mathcal{R}_t^\star) \le \mathcal{O}\left(\left((1-c\rho)^t + \sqrt{\rho \log(4T/\delta)} + \eta^2/\rho^2 \right)^{\frac{1+\bar{\gamma}}{2}} + \sqrt{\frac{\log(2/\delta)}{n_2}} + \frac{1}{n_2} \right).$$

When $\Delta_{n_2}(\delta) \le 1$, the condition $\underline{\gamma} \le 1 + \bar{\gamma}$ implies $\Delta_{n_2}(\delta)^{(1+\bar{\gamma})/\underline{\gamma}} \le \Delta_{n_2}(\delta)$. When $\Delta_{n_2}(\delta) > 1$, the claimed bound is trivial since the excess Type-II error is at most one. This completes the proof. □

G.4 Proof of Theorem G.2

Proof of Theorem G.2. Fix $c \in \{0, 1\}$ and suppress the class index throughout the proof. Write

$$\boldsymbol{\mu}_t := \boldsymbol{\mu}_{c,t}, \qquad \widehat{\boldsymbol{\mu}}_t := \widehat{\boldsymbol{\mu}}_{c,t}, \qquad \bar{\mathbf{z}}_t := \bar{\mathbf{z}}_{c,t}(\mathbf{v}_t), \qquad A := A_d(\kappa).$$

By Assumption 1,

$$\bar{\mathbf{z}}_t = A\boldsymbol{\mu}_t + \boldsymbol{\zeta}_{t+1}, \qquad \mathbb{E}[\boldsymbol{\zeta}_{t+1} \mid \mathcal{F}_t] = 0, \qquad \|\boldsymbol{\zeta}_{t+1}\| \le 2 \quad \text{a.s.}$$

Hence

$$\mathbf{z}_{t+1} = (1-\rho)\widehat{\boldsymbol{\mu}}_t + \rho A\boldsymbol{\mu}_t + \rho\boldsymbol{\zeta}_{t+1}, \qquad \widehat{\boldsymbol{\mu}}_{t+1} = \frac{\mathbf{z}_{t+1}}{\|\mathbf{z}_{t+1}\|}.$$

Define

$$\mathbf{e}_t := \widehat{\boldsymbol{\mu}}_t - \boldsymbol{\mu}_t, \qquad \mathbf{y}_t := (1-\rho)\widehat{\boldsymbol{\mu}}_t + \rho A\boldsymbol{\mu}_t.$$

Step 1: Bootstrap region and deterministic bounds. Define the bootstrap event

$$\mathcal{E}_t := \left\{ \|\mathbf{e}_s\| \le \frac{1}{4} \text{ for all } 0 \le s \le t \right\}.$$

Clearly, $\mathcal{E}_t \in \mathcal{F}_t := \sigma(\{\boldsymbol{\zeta}_s\}_{s=1}^t)$ is $\mathcal{F}_t$-measurable. On $\mathcal{E}_t$, Lemma G.8 yields

$$\|\Pi(\mathbf{y}_t) - \boldsymbol{\mu}_t\| \le (1 - c_0\rho)\|\mathbf{e}_t\|,$$

for some $c_0 \in (0, 1)$ and all $0 \le t \le T-1$, provided $\rho \le \rho_0$ is small enough.

Also, on $\mathcal{E}_t$, $\langle \widehat{\boldsymbol{\mu}}_t, \boldsymbol{\mu}_t \rangle = 1 - \frac{1}{2}\|\mathbf{e}_t\|^2 \ge 1 - \frac{1}{32} = \frac{31}{32}$. Therefore,

$$\|\mathbf{y}_t\| \ge \langle \mathbf{y}_t, \boldsymbol{\mu}_t \rangle = (1-\rho)\langle \widehat{\boldsymbol{\mu}}_t, \boldsymbol{\mu}_t \rangle + \rho A \ge (1-\rho)\frac{31}{32} + \rho A.$$

Shrinking ρ_0 if necessary, we may ensure $\|\mathbf{y}_t\| \ge \frac{1}{2}$, $\forall\, 0 < \rho \le \rho_0$. Since $\mathbf{z}_{t+1} = \mathbf{y}_t + \rho\boldsymbol{\zeta}_{t+1}$ and $\|\boldsymbol{\zeta}_{t+1}\| \le 2$, we further obtain $\|\mathbf{z}_{t+1}\| \ge \|\mathbf{y}_t\| - \rho\|\boldsymbol{\zeta}_{t+1}\| \ge \frac{1}{2} - 2\rho \ge \frac{1}{4}$ for ρ_0 smaller if needed. In short, we show that as long as $\|\mathbf{e}_t\| \le \frac{1}{4}$, we then have $\|\mathbf{y}_t\| \ge \frac{1}{2}$, $\|\mathbf{z}_{t+1}\| \ge \frac{1}{4}$, and $\langle \widehat{\boldsymbol{\mu}}_t, \boldsymbol{\mu}_t \rangle \ge 0$.

Step 2: One-step square-error decomposition. For any $\gamma > 0$, Young's inequality gives

$$\|\mathbf{e}_{t+1}\|^2 = \|\widehat{\boldsymbol{\mu}}_{t+1} - \boldsymbol{\mu}_{t+1}\|^2 \le (1+\gamma)\|\widehat{\boldsymbol{\mu}}_{t+1} - \boldsymbol{\mu}_t\|^2 + \left(1 + \frac{1}{\gamma}\right)\|\boldsymbol{\mu}_{t+1} - \boldsymbol{\mu}_t\|^2. \tag{15}$$

By Assumption 2, $\|\boldsymbol{\mu}_{t+1} - \boldsymbol{\mu}_t\| \le C_\mu \eta$. We now analyze the first term of (15):

$$\|\widehat{\boldsymbol{\mu}}_{t+1} - \boldsymbol{\mu}_t\|^2 = \|\Pi(\mathbf{z}_{t+1}) - \boldsymbol{\mu}_t\|^2.$$

For simplicity, we set $\mathbf{a}_t := \Pi(\mathbf{y}_t) - \boldsymbol{\mu}_t$. Then

$$\Pi(\mathbf{z}_{t+1}) - \boldsymbol{\mu}_t = \mathbf{a}_t + \big(\Pi(\mathbf{y}_t + \rho\boldsymbol{\zeta}_{t+1}) - \Pi(\mathbf{y}_t)\big).$$

Hence, it follows that

$$\begin{aligned} \|\Pi(\mathbf{z}_{t+1}) - \boldsymbol{\mu}_t\|^2 = \|\mathbf{a}_t\|^2 &+ 2\left\langle \mathbf{a}_t,\, \Pi(\mathbf{y}_t + \rho\boldsymbol{\zeta}_{t+1}) - \Pi(\mathbf{y}_t) \right\rangle \\ &+ \left\| \Pi(\mathbf{y}_t + \rho\boldsymbol{\zeta}_{t+1}) - \Pi(\mathbf{y}_t) \right\|^2. \end{aligned} \tag{16}$$

Step 3: Taylor expansion of the projection map. Since $\|\mathbf{y}_t\| \geq 1/2$ on $\mathcal{E}_t$, the projection map $\Pi(\mathbf{x}) = \mathbf{x}/\|\mathbf{x}\|$ is C^2 on a neighborhood of the trajectory. Therefore,

$$\Pi(\mathbf{y}_t + \rho\boldsymbol{\zeta}_{t+1}) - \Pi(\mathbf{y}_t) = \rho J_t \boldsymbol{\zeta}_{t+1} + \mathbf{R}_{t+1},$$

where J_t is the Jacobian matrix

$$J_t := D\Pi(\mathbf{y}_t) = \frac{1}{\|\mathbf{y}_t\|}\Big(I - \Pi(\mathbf{y}_t)\Pi(\mathbf{y}_t)^\top\Big),$$

and the remainder satisfies

$$\|\mathbf{R}_{t+1}\| \leq C_R\rho^2\|\boldsymbol{\zeta}_{t+1}\|^2 \leq 4C_R\rho^2$$

for a universal constant $C_R > 0$. Moreover, since $\|\mathbf{y}_t\| \geq 1/2$, we have $\|J_t\| \leq 2$. Substituting this expansion into (16), we obtain

$$\|\Pi(\mathbf{z}_{t+1}) - \boldsymbol{\mu}_t\|^2 = \|\mathbf{a}_t\|^2 + 2\rho\langle \mathbf{a}_t, J_t\boldsymbol{\zeta}_{t+1}\rangle + \Xi_{t+1}, \tag{17}$$

where Ξ_{t+1} collects the higher-order terms:

$$\Xi_{t+1} := 2\langle \mathbf{a}_t, \mathbf{R}_{t+1}\rangle + \|\rho J_t\boldsymbol{\zeta}_{t+1} + \mathbf{R}_{t+1}\|^2.$$

Since $\|\mathbf{a}_t\| \leq 2$, $\|J_t\| \leq 2$, $\|\boldsymbol{\zeta}_{t+1}\| \leq 2$, and $\|\mathbf{R}_{t+1}\| \leq 4C_R\rho^2$, there exists a universal constant $C_1 > 0$ such that

$$|\Xi_{t+1}| \leq C_1\rho^2.$$

Let $\mathbb{1}_{\mathcal{E}_t}$ denote the indicator of $\mathcal{E}_t$ and define

$$\Delta_{t+1} := 2\rho\langle \mathbf{a}_t, J_t\boldsymbol{\zeta}_{t+1}\rangle \mathbb{1}_{\mathcal{E}_t}.$$

Since $\mathbf{a}_t$, J_t, and $\mathbb{1}_{\mathcal{E}_t}$ are $\mathcal{F}_t$-measurable and $\mathbb{E}[\boldsymbol{\zeta}_{t+1} \mid \mathcal{F}_t] = 0$, $\{\Delta_{t+1}\}$ is a martingale difference sequence with respect to $\{\mathcal{F}_t\}$. Moreover, there exist constants $C_2, C_3 > 0$ such that, almost surely,

$$|\Delta_{t+1}| \leq C_2\rho, \qquad \mathbb{E}[\Delta_{t+1}^2 \mid \mathcal{F}_t] \leq C_3\rho^2.$$

Hence, by (17), on $\mathcal{E}_t$,

$$\|\Pi(\mathbf{z}_{t+1}) - \boldsymbol{\mu}_t\|^2 \leq \|\mathbf{a}_t\|^2 + \Delta_{t+1} + C_1\rho^2.$$

Step 4: One-step recursion on the bootstrap region. By Lemma G.8,

$$\|\mathbf{a}_t\|^2 \leq (1 - c_0\rho)^2\|\mathbf{e}_t\|^2 \leq (1 - c_0\rho)\|\mathbf{e}_t\|^2.$$

Combining this with (15), we obtain that on $\mathcal{E}_{t-1}$,

$$\|\mathbf{e}_{t+1}\|^2 \leq (1+\gamma)(1 - c_0\rho)\|\mathbf{e}_t\|^2 + (1+\gamma)C_1\rho^2 + \left(1 + \frac{1}{\gamma}\right)C_\mu^2\eta^2 + (1+\gamma)\Delta_{t+1}.$$

We then choose $\gamma := \frac{c_0\rho}{4}$. As a result, for ρ_0 sufficiently small, we could simultaneously have that (i) $(1+\gamma)(1 - c_0\rho) \leq 1 - c_1\rho$ for some constant $c_1 > 0$, (ii) $(1+\gamma)C_1\rho^2 \leq C_4\rho^2$, and (iii) $\left(1 + \frac{1}{\gamma}\right)C_\mu^2\eta^2 \leq C_5\frac{\eta^2}{\rho}$. As a result of the above notation simplification, we may rewrite the recursion as

$$\|\mathbf{e}_{t+1}\|^2 \leq (1 - c_1\rho)\|\mathbf{e}_t\|^2 + C_4\rho^2 + C_5\frac{\eta^2}{\rho} + \left(1 + \frac{c_0\rho}{4}\right)\Delta_{t+1}, \tag{18}$$

where $\{\Delta_{t+1}\}$ is a martingale difference sequence satisfying $|\Delta_{t+1}| \leq C_2\rho$ and $\mathbb{E}[\Delta_{t+1}^2 \mid \mathcal{F}_t] \leq C_3\rho^2$.

Step 5: Unrolling the recursion. Iterating (18) yields that, on $\mathcal{E}_{t-1}$,

$$\|\mathbf{e}_t\|^2 \leq (1-c_1\rho)^t\|\mathbf{e}_0\|^2 + \left(C_4\rho^2 + C_5\frac{\eta^2}{\rho}\right)\sum_{j=0}^{t-1}(1-c_1\rho)^j + \left(1 + \frac{c_0\rho}{4}\right)\sum_{s=0}^{t-1}(1-c_1\rho)^{t-1-s}\Delta_{s+1}.$$

Since $\sum_{j=0}^{t-1}(1 - c_1\rho)^j \leq \frac{1}{c_1\rho}$, we obtain that on $\mathcal{E}_{t-1}$,

$$\|\mathbf{e}_t\|^2 \leq (1 - c_1\rho)^t\|\mathbf{e}_0\|^2 + C_6\rho + C_7\frac{\eta^2}{\rho^2} + \left(1 + \frac{c_0\rho}{4}\right)\sum_{s=0}^{t-1}(1 - c_1\rho)^{t-1-s}\Delta_{s+1}. \tag{19}$$

Now, we apply the concentration inequality in Lemma G.9 to the martingale sequence $\{\Delta_{t+1}\}$ (with $c_\star = c_1$). There exists a constant $C_8 > 0$ such that, with probability at least $1 - \delta/2$,

$$\left(1 + \frac{c_0\rho}{4}\right) \sup_{1\le t\le T} \left|\sum_{s=0}^{t-1} (1 - c_1\rho)^{t-1-s} \Delta_{s+1}\right| \le C_8 \left(\sqrt{\rho \log \frac{2T}{\delta}} + \rho \log \frac{2T}{\delta}\right). \tag{20}$$

We denote the event in (20) as $\mathcal{E}_\#$. Then $\mathbb{P}(\mathcal{E}_\#) \ge 1 - \delta$. Since $\rho \le 1$ and $\rho \log(2T/\delta) \le 1$, after enlarging constants we can bound the right-hand side by $C_9\sqrt{\rho \log \frac{2T}{\delta}}$.

Combining this with (19), we conclude that on the event $\mathcal{E}_\# \cap \mathcal{E}_{t-1}$, for suitable constants $c, C > 0$,

$$\|\mathbf{e}_t\|^2 \le (1 - c\rho)^t \|\mathbf{e}_0\|^2 + C\left(\sqrt{\rho \log \frac{2T}{\delta}} + \frac{\eta^2}{\rho^2}\right). \tag{21}$$

Step 6: Induction on Bootstrap events. By the condition in Theorem G.2,

$$\|\mathbf{e}_0\|^2 + C\left(\sqrt{\rho \log \frac{2T}{\delta}} + \frac{\eta^2}{\rho^2}\right) \le \frac{1}{16}. \tag{22}$$

As a result of this condition, $\mathcal{E}_0 = \{\|\mathbf{e}_0\| \le \frac{1}{4}\}$ is true. Now, we want to show that for any $t \ge 1$, on the event $\mathcal{E}_\# \cap \mathcal{E}_{t-1}$, we must have that the event $\mathcal{E}_t$ is also true. It suffices to show that

$$\|\mathbf{e}_t\|^2 \le \frac{1}{16},$$

which naturally follows by combining (22) and (21). We then complete the proof by induction. □

G.5 Proof of Lemma G.5

Proof of Lemma G.5. We split this region $\widehat{\mathcal{R}}\Delta\mathcal{R}^\star$ into two disjoint parts: $\mathcal{R}^\star \cap \widehat{\mathcal{R}}^c$ (where the optimal classifier accepts but the estimated one rejects) and $(\mathcal{R}^\star)^c \cap \widehat{\mathcal{R}}$ (where the estimated one accepts but the optimal rejects). It then follows that

$$\begin{aligned}
&\int_{(\mathcal{R}^\star \cap \widehat{\mathcal{R}}^c) \cup ((\mathcal{R}^\star)^c \cap \widehat{\mathcal{R}})} \left|\frac{p_1(x)}{p_0(x)} - e^{\tau_\alpha^\star}\right| \mathrm{d}\mathbb{P}_0(x) \\
&= \int_{\mathcal{R}^\star \cap \widehat{\mathcal{R}}^c} \left|\frac{p_1(x)}{p_0(x)} - e^{\tau_\alpha^\star}\right| \mathrm{d}\mathbb{P}_0(x) + \int_{(\mathcal{R}^\star)^c \cap \widehat{\mathcal{R}}} \left|\frac{p_1(x)}{p_0(x)} - e^{\tau_\alpha^\star}\right| \mathrm{d}\mathbb{P}_0(x) \\
&= \int_{\mathcal{R}^\star \cap \widehat{\mathcal{R}}^c} \left(e^{\tau_\alpha^\star} - \frac{p_1(x)}{p_0(x)}\right) \mathrm{d}\mathbb{P}_0(x) + \int_{(\mathcal{R}^\star)^c \cap \widehat{\mathcal{R}}} \left(\frac{p_1(x)}{p_0(x)} - e^{\tau_\alpha^\star}\right) \mathrm{d}\mathbb{P}_0(x).
\end{aligned}$$

Note that

$$\begin{aligned}
&\int_{\mathcal{R}^\star} \left(e^{\tau_\alpha^\star} - \frac{p_1(x)}{p_0(x)}\right) \mathrm{d}\mathbb{P}_0(x) + \int_{\widehat{\mathcal{R}}} \left(\frac{p_1(x)}{p_0(x)} - e^{\tau_\alpha^\star}\right) \mathrm{d}\mathbb{P}_0(x) \\
&= \underbrace{\int_{\mathcal{R}^\star \cap \widehat{\mathcal{R}}} \left(e^{\tau_\alpha^\star} - \frac{p_1(x)}{p_0(x)} + \frac{p_1(x)}{p_0(x)} - e^{\tau_\alpha^\star}\right) \mathrm{d}\mathbb{P}_0(x)}_{=0} \\
&\quad + \int_{\mathcal{R}^\star \cap \widehat{\mathcal{R}}^c} \left(e^{\tau_\alpha^\star} - \frac{p_1(x)}{p_0(x)}\right) \mathrm{d}\mathbb{P}_0(x) + \int_{(\mathcal{R}^\star)^c \cap \widehat{\mathcal{R}}} \left(\frac{p_1(x)}{p_0(x)} - e^{\tau_\alpha^\star}\right) \mathrm{d}\mathbb{P}_0(x).
\end{aligned}$$

Using the property that $\int_A \frac{p_1(x)}{p_0(x)} \mathrm{d}\mathbb{P}_0(x) = \int_A \mathrm{d}\mathbb{P}_1(x) = \mathbb{P}_1(A)$ for any measurable set A, we have

$$\begin{aligned}
&\int_{(\mathcal{R}^\star \cap \widehat{\mathcal{R}}^c) \cup ((\mathcal{R}^\star)^c \cap \widehat{\mathcal{R}})} \left|\frac{p_1(x)}{p_0(x)} - e^{\tau_\alpha^\star}\right| \mathrm{d}\mathbb{P}_0(x) \\
&= \int_{\mathcal{R}^\star} \left(e^{\tau_\alpha^\star} - \frac{p_1(x)}{p_0(x)}\right) \mathrm{d}\mathbb{P}_0(x) + \int_{\widehat{\mathcal{R}}} \left(\frac{p_1(x)}{p_0(x)} - e^{\tau_\alpha^\star}\right) \mathrm{d}\mathbb{P}_0(x) \\
&= \mathbb{P}_1(\widehat{\mathcal{R}}) - \mathbb{P}_1(\mathcal{R}^\star) + e^{\tau_\alpha^\star}\left(\mathbb{P}_0(\mathcal{R}^\star) - \mathbb{P}_0(\widehat{\mathcal{R}})\right),
\end{aligned}$$

which, together with the fact that $\mathbb{P}_0((\mathcal{R}^\star)^c) = \alpha$, implies that $\mathbb{P}_0(\mathcal{R}^\star) = 1 - \alpha$. Recalling $\mathbb{P}_0(\widehat{\mathcal{R}}) = 1 - \mathbb{P}_0(\widehat{\mathcal{R}}^c)$, we obtain $\mathbb{P}_0(\mathcal{R}^\star) - \mathbb{P}_0(\widehat{\mathcal{R}}) = \mathbb{P}_0(\widehat{\mathcal{R}}^c) - \alpha$. We then complete the proof. □

G.6 Proof of Lemma G.6

Proof of Lemma G.6. By the uniform error bound, we have

$$\{\mathcal{S}(X) > \widehat{\tau}_\alpha + r(\delta,\rho,\eta)\} \subseteq \{\widehat{\mathcal{S}}(X) > \widehat{\tau}_\alpha\} \subseteq \{\mathcal{S}(X) > \widehat{\tau}_\alpha - r(\delta,\rho,\eta)\},$$

which implies

$$\mathbb{P}_0(\mathcal{S}(X) > \widehat{\tau}_\alpha + r(\delta,\rho,\eta)) \le \mathbb{P}_0(\widehat{\mathcal{R}}^c) \le \mathbb{P}_0(\mathcal{S}(X) > \widehat{\tau}_\alpha - r(\delta,\rho,\eta))\,.$$

Since $\alpha = \mathbb{P}_0\left(\mathcal{S}(X) > \tau^\star_\alpha\right)$, it follows that $\mathbb{P}_0(\mathcal{S}(X) \in [\tau^\star_\alpha, \widehat{\tau}_\alpha - r(\delta,\rho,\eta))) \le \Delta_{n_2}(\delta)$ whenever $\widehat{\tau}_\alpha - r(\delta,\rho,\eta) \ge \tau^\star_\alpha$, and similarly $\mathbb{P}_0(\mathcal{S}(X) \in [\widehat{\tau}_\alpha + r(\delta,\rho,\eta), \tau^\star_\alpha)) \le \Delta_{n_2}(\delta)$ whenever $\widehat{\tau}_\alpha + r(\delta,\rho,\eta) \le \tau^\star_\alpha$.

Now let

$$u_\delta := \left(\frac{\Delta_{n_2}(\delta)}{c_0}\right)^{1/\underline{\gamma}}.$$

Assume $u_\delta < \varepsilon_0$. By the one-sided lower bounds in Assumption 3, for every $\varepsilon \in [0, \varepsilon_0]$,

$$\mathbb{P}_0(\tau^\star_\alpha < \mathcal{S}(X) \le \tau^\star_\alpha + \varepsilon) \ge c_0\varepsilon^{\underline{\gamma}} \quad \text{and} \quad \mathbb{P}_0(\tau^\star_\alpha - \varepsilon \le \mathcal{S}(X) < \tau^\star_\alpha) \ge c_0\varepsilon^{\underline{\gamma}}.$$

Suppose first that $\widehat{\tau}_\alpha - r(\delta,\rho,\eta) > \tau^\star_\alpha + u_\delta$. Then one can choose $\varepsilon \in (u_\delta, \varepsilon_0]$ such that

$$\tau^\star_\alpha + \varepsilon < \widehat{\tau}_\alpha - r(\delta,\rho,\eta).$$

The right-sided lower bound therefore gives

$$\mathbb{P}_0(\tau^\star_\alpha < \mathcal{S}(X) \le \tau^\star_\alpha + \varepsilon) > \Delta_{n_2}(\delta),$$

contradicting the preceding upper bound on the probability mass between $\tau^\star_\alpha$ and $\widehat{\tau}_\alpha - r(\delta,\rho,\eta)$. The case $\widehat{\tau}_\alpha + r(\delta,\rho,\eta) < \tau^\star_\alpha - u_\delta$ is analogous, using the left-sided lower bound. Hence,

$$|\widehat{\tau}_\alpha - \tau^\star_\alpha| \le r(\delta,\rho,\eta) + u_\delta.$$

Next, consider any $x \in \widehat{\mathcal{R}} \Delta \mathcal{R}^\star$.

(i) If $x \in (\mathcal{R}^\star)^c \cap \widehat{\mathcal{R}}$, then $\mathcal{S}(x) > \tau^\star_\alpha$ and $\widehat{\mathcal{S}}(x) \le \widehat{\tau}_\alpha$. Using the uniform score bound gives

$$\mathcal{S}(x) \le \widehat{\tau}_\alpha + r(\delta,\rho,\eta) \le \tau^\star_\alpha + 2r(\delta,\rho,\eta) + \left(\frac{\Delta_{n_2}(\delta)}{c_0}\right)^{1/\underline{\gamma}}.$$

(ii) If $x \in \mathcal{R}^\star \cap \widehat{\mathcal{R}}^c$, then $\mathcal{S}(x) \le \tau^\star_\alpha$ and $\widehat{\mathcal{S}}(x) > \widehat{\tau}_\alpha$, and hence

$$\tau^\star_\alpha - 2r(\delta,\rho,\eta) - \left(\frac{\Delta_{n_2}(\delta)}{c_0}\right)^{1/\underline{\gamma}} < \mathcal{S}(x) \le \tau^\star_\alpha.$$

Combining the two cases, we complete the proof. □

G.7 Useful Lemmas

Lemma G.7. *Let $m > 0$. For any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ satisfying $\|\mathbf{x}\| \ge m$ and $\|\mathbf{y}\| \ge m$,*

$$\|\Pi(\mathbf{x}) - \Pi(\mathbf{y})\| \le \frac{2}{m}\|\mathbf{x} - \mathbf{y}\|.$$

Proof of Lemma G.7. Using $\Pi(\mathbf{x}) - \Pi(\mathbf{y}) = \frac{\mathbf{x}-\mathbf{y}}{\|\mathbf{x}\|} + \mathbf{y}\left(\frac{1}{\|\mathbf{x}\|} - \frac{1}{\|\mathbf{y}\|}\right)$, we obtain

$$\|\Pi(\mathbf{x}) - \Pi(\mathbf{y})\| \le \frac{\|\mathbf{x} - \mathbf{y}\|}{\|\mathbf{x}\|} + \frac{\big|\|\mathbf{y}\| - \|\mathbf{x}\|\big|}{\|\mathbf{x}\|} \le \frac{2}{m}\|\mathbf{x} - \mathbf{y}\|.$$

□

Lemma G.8. *There exist constants $c_0, \rho_0 \in (0,1)$, depending only on A, such that for every $0 < \rho \le \rho_0$, every $t \ge 0$, and every $\mathbf{u} \in \mathbb{S}^{d-1}$ satisfying $\|\mathbf{u} - \boldsymbol{\mu}_t\| \le \frac{1}{4}$,*

$$\big\|\Pi\big((1-\rho)\mathbf{u} + \rho A\boldsymbol{\mu}_t\big) - \boldsymbol{\mu}_t\big\| \le (1 - c_0\rho)\|\mathbf{u} - \boldsymbol{\mu}_t\|.$$

Proof of Lemma G.8. Fix t and $\mathbf{u} \in \mathbb{S}^{d-1}$ with $\|\mathbf{u} - \boldsymbol{\mu}_t\| \le 1/4$. Write

$$\mathbf{u} = \alpha\boldsymbol{\mu}_t + \beta\boldsymbol{\nu}, \qquad \boldsymbol{\nu} \perp \boldsymbol{\mu}_t, \qquad \alpha^2 + \beta^2 = 1.$$

Since $\|\mathbf{u} - \boldsymbol{\mu}_t\|^2 = 2(1-\alpha)$, the condition $\|\mathbf{u} - \boldsymbol{\mu}_t\| \le 1/4$ implies $\alpha \ge 1 - \frac{1}{32} = \frac{31}{32}$. Define

$$\mathbf{x} := (1-\rho)\mathbf{u} + \rho A\boldsymbol{\mu}_t = a\boldsymbol{\mu}_t + b\boldsymbol{\nu},$$

where $a := (1-\rho)\alpha + \rho A$ and $b := (1-\rho)\beta$. Then

$$\Pi(\mathbf{x}) = \frac{a\boldsymbol{\mu}_t + b\boldsymbol{\nu}}{r} \quad \text{with} \quad r := \sqrt{a^2 + b^2}.$$

Therefore, we have that

$$\|\Pi(\mathbf{x}) - \boldsymbol{\mu}_t\|^2 = 2\left(1 - \frac{a}{r}\right) = \frac{2b^2}{r(r+a)}.$$

Using the facts that $\|\mathbf{u} - \boldsymbol{\mu}_t\|^2 = 2(1-\alpha)$ and $\beta^2 = 1 - \alpha^2$, we get

$$\frac{\|\Pi(\mathbf{x}) - \boldsymbol{\mu}_t\|^2}{\|\mathbf{u} - \boldsymbol{\mu}_t\|^2} = \frac{(1-\rho)^2(1+\alpha)}{r(r+a)} =: \Psi(\alpha, \rho).$$

Now $\alpha \in [31/32, 1]$, and one checks that $r^2 = (1-\rho)^2 + 2\rho(1-\rho)A\alpha + \rho^2A^2$ by definition. Hence, Ψ is continuous on the compact set $[31/32, 1] \times [0, \rho_1]$ for any fixed $\rho_1 < 1$, and $\Psi(\alpha, 0) = 1$ for all α. Moreover,

$$\frac{\partial}{\partial\rho}\Psi(\alpha,\rho)\bigg|_{\rho=0} = -A(1+\alpha) \le -A \cdot \frac{63}{32} < 0,$$

uniformly over $\alpha \in [31/32, 1]$. Therefore, by compactness and continuity, there exist constants $c_0, \rho_0 \in (0,1)$, depending only on A, such that for all $\alpha \in [31/32, 1]$ and all $0 < \rho \le \rho_0$,

$$\Psi(\alpha, \rho) \le (1 - c_0\rho)^2.$$

Taking square roots yields that

$$\big\|\Pi\big((1-\rho)\mathbf{u} + \rho A\boldsymbol{\mu}_t\big) - \boldsymbol{\mu}_t\big\| \le (1 - c_0\rho)\|\mathbf{u} - \boldsymbol{\mu}_t\|.$$

□

Lemma G.9 (Weighted martingale concentration)**.** *Let $\{\Delta_{t+1}\}_{t\ge0}$ be a martingale difference sequence with respect to $\{\mathcal{F}_t\}$. Assume that there exist constants $B, V > 0$ such that, almost surely,*

$$|\Delta_{t+1}| \le B\rho, \qquad \mathbb{E}[\Delta_{t+1}^2 \mid \mathcal{F}_t] \le V\rho^2 \qquad \forall\, t \ge 0.$$

Fix $c_\star, \rho \in (0,1)$ and define, for each $t \ge 1$,

$$S_t := \sum_{s=0}^{t-1} (1 - c_\star\rho)^{t-1-s}\Delta_{s+1}.$$

Then there exists a constant $C > 0$, depending only on $B, V, c_\star$, such that for any $\delta \in (0,1)$,

$$\mathbb{P}\left(\sup_{1\le t\le T} |S_t| \le C\left(\sqrt{\rho \log\frac{T}{\delta}} + \rho\log\frac{T}{\delta}\right)\right) \ge 1 - \delta.$$

Proof of Lemma G.9. Fix $t \ge 1$. Define

$$\xi_{t,s+1} := (1 - c_\star\rho)^{t-1-s}\Delta_{s+1}, \qquad 0 \le s \le t-1.$$

Then $\{\xi_{t,s+1}\}_{s=0}^{t-1}$ is a martingale difference sequence with respect to $\{\mathcal{F}_s\}$, and

$$|\xi_{t,s+1}| \le (1 - c_\star\rho)^{t-1-s}B\rho.$$

Moreover,

$$\sum_{s=0}^{t-1} \mathbb{E}[\xi_{t,s+1}^2 \mid \mathcal{F}_s] \le V\rho^2 \sum_{s=0}^{t-1} (1-c_\star\rho)^{2(t-1-s)} \le V\rho^2 \sum_{j=0}^{\infty} (1-c_\star\rho)^{2j}.$$

Since

$$\sum_{j=0}^{\infty} (1-c_\star\rho)^{2j} = \frac{1}{1-(1-c_\star\rho)^2} \le \frac{1}{c_\star\rho},$$

it follows that

$$\sum_{s=0}^{t-1} \mathbb{E}[\xi_{t,s+1}^2 \mid \mathcal{F}_s] \le \frac{V}{c_\star}\rho.$$

Applying Freedman's inequality [84] yields that, for any $u > 0$,

$$\mathbb{P}(|S_t| \ge u) \le 2\exp\left(-\frac{u^2}{2\left(\frac{V}{c_\star}\rho + \frac{1}{3}B\rho\, u\right)}\right).$$

Taking a union bound over $1 \le t \le T$ and choosing

$$u = C\Big(\sqrt{\rho \log\frac{T}{\delta}} + \rho\log\frac{T}{\delta}\Big).$$

for a sufficiently large constant C proves the claim. □

H Additional Theory

H.1 Gradient Analysis of the Population Training Objective

The training procedure in Phase I follows a two-timescale scheme: the steering vector $\mathbf{v}$ is updated on a slow timescale, while the class mean directions are updated on a fast timescale. To make the role of this dynamics clear, we analyze the corresponding population objective.

For a fixed steering vector $\mathbf{v}$, we then define

$$\mathcal{J}(\mathbf{v}) := \max_{\boldsymbol{\mu}_0,\boldsymbol{\mu}_1 \in \mathbb{S}^{d-1}} \mathbb{E}_{x,y}[\log q_\beta(y \mid f_{\theta,\mathbf{v}}(x), \boldsymbol{\mu}_0, \boldsymbol{\mu}_1)] = \mathbb{E}_{x,y}[\log q_\beta(y \mid f_{\theta,\mathbf{v}}(x), \boldsymbol{\mu}_0(\mathbf{v}), \boldsymbol{\mu}_1(\mathbf{v}))]\,.$$

where $\boldsymbol{\mu}_c(\mathbf{v})$ denotes the population-optimal mean direction for class c induced by the steered representation map $f_{\theta,\mathbf{v}}$.

The following proposition characterizes how optimizing $\mathcal{J}(\mathbf{v})$ reshapes the latent geometry. For this analysis, we work under Assumption 1, a uniform class prior $p(y=0) = p(y=1) = 1/2$, and the matched case $\beta = \kappa$. Under these conditions, the temperature-scaled posterior q_β in (2) coincides with the class posterior induced by the shared-concentration vMF model.

Proposition H.1. *The gradient of $\mathcal{J}(\mathbf{v})$ admits the decomposition*

$$\nabla_{\mathbf{v}}\mathcal{J}(\mathbf{v}) = \beta\,\bar{\omega}(\mathbf{v})\,\nabla_{\mathbf{v}}\|\boldsymbol{\mu}_1(\mathbf{v}) - \boldsymbol{\mu}_0(\mathbf{v})\|^2 + \frac{\beta}{2}\big(\boldsymbol{\mu}_1(\mathbf{v}) - \boldsymbol{\mu}_0(\mathbf{v})\big)^\top\big(\Delta_1(\mathbf{v}) - \Delta_0(\mathbf{v})\big).$$

Here, $\Delta_0(\mathbf{v})$ and $\Delta_1(\mathbf{v})$ are residual terms defined in (23) *as*

$$\Delta_c(\mathbf{v}) := \mathbb{E}_x\big[q_\beta(1-c \mid x)\big(\nabla_{\mathbf{v}} f_{\theta,\mathbf{v}}(x) - \nabla_{\mathbf{v}}\boldsymbol{\mu}_c(\mathbf{v})\big) \mid y = c\big]\,, \tag{23}$$

and $\bar{\omega}(\mathbf{v}) \ge 0$ is a data-dependent weight given by

$$\bar{\omega}(\mathbf{v}) = \frac{1}{8}\left(1 - \mathbb{E}_x\left[\tanh^2\left(\frac{\beta}{2}\big(\boldsymbol{\mu}_1(\mathbf{v}) - \boldsymbol{\mu}_0(\mathbf{v})\big)^\top f_{\theta,\mathbf{v}}(x)\right)\right]\right).$$

Proposition H.1 decomposes the gradient of the population objective into two components. The first term drives an increase in the separation between the class mean directions $\|\boldsymbol{\mu}_1(\mathbf{v}) - \boldsymbol{\mu}_0(\mathbf{v})\|^2$, with a data-dependent weight that emphasizes samples near the decision boundary. The second term

captures a residual effect arising from the mismatch between the sample-level response $\nabla_{\mathbf{v}} f_{\theta,\mathbf{v}}(x)$ and the induced movement of the class means.

This decomposition clarifies the optimization dynamics: when the residual term is small, the first term pushes the representation toward greater class separation, while the residual term captures a correction that depends on local variability.

Proof of Proposition H.1. Denote by

$$\mathcal{L}_\beta(\mathbf{v}, \boldsymbol{\mu}_0, \boldsymbol{\mu}_1) := \log q_\beta(y \mid f_{\theta,\mathbf{v}}(x), \boldsymbol{\mu}_0, \boldsymbol{\mu}_1)\,.$$

Then $\mathcal{J}(\mathbf{v}) = \mathbb{E}_{x,y}\big[\mathcal{L}_\beta\big(\mathbf{v}, \boldsymbol{\mu}_0(\mathbf{v}), \boldsymbol{\mu}_1(\mathbf{v})\big)\big]$. Differentiating with respect to $\mathbf{v}$ gives

$$\nabla_{\mathbf{v}}\mathcal{J}(\mathbf{v}) = \mathbb{E}_{x,y}\big[\nabla_f \mathcal{L}_\beta^\top \nabla_{\mathbf{v}} f_{\theta,\mathbf{v}}(x)\big] + \sum_{c\in\{0,1\}} \Big(\nabla_{\mathbf{v}}\boldsymbol{\mu}_c(\mathbf{v})\Big)^\top \nabla_{\boldsymbol{\mu}_c}\mathcal{J}(\mathbf{v}).$$

Since $\boldsymbol{\mu}_c(\mathbf{v})$ is the optimal class mean direction on $\mathbb{S}^{d-1}$, the first-order optimality condition implies that $\nabla_{\boldsymbol{\mu}_c}\mathcal{J}(\mathbf{v})$ is normal to $\mathbb{S}^{d-1}$ at $\boldsymbol{\mu}_c(\mathbf{v})$. On the other hand, $\|\boldsymbol{\mu}_c(\mathbf{v})\| = 1$ for all $\mathbf{v}$, and differentiating $\|\boldsymbol{\mu}_c(\mathbf{v})\|^2 = 1$ yields $\boldsymbol{\mu}_c(\mathbf{v})^\top \nabla_{\mathbf{v}}\boldsymbol{\mu}_c(\mathbf{v}) = 0$, which shows that $\nabla_{\mathbf{v}}\boldsymbol{\mu}_c(\mathbf{v})$ lies in the tangent space of $\mathbb{S}^{d-1}$ at $\boldsymbol{\mu}_c(\mathbf{v})$. Consequently,

$$\Big(\nabla_{\mathbf{v}}\boldsymbol{\mu}_c(\mathbf{v})\Big)^\top \nabla_{\boldsymbol{\mu}_c}\mathcal{J}(\mathbf{v}) = 0,$$

which implies that

$$\nabla_{\mathbf{v}}\mathcal{J}(\mathbf{v}) = \mathbb{E}_{x,y}\Big[\big(\nabla_f \mathcal{L}_\beta\big)^\top \nabla_{\mathbf{v}} f_{\theta,\mathbf{v}}(x)\Big]\,. \tag{24}$$

For simplicity, write $q_\beta(y \mid x) := q_\beta(y \mid f_{\theta,\mathbf{v}}(x), \boldsymbol{\mu}_0(\mathbf{v}), \boldsymbol{\mu}_1(\mathbf{v}))$. Under the matched case $\beta = \kappa$, this coincides with the class posterior induced by the shared-concentration vMF model, and

$$q_\beta(y \mid x) = \frac{\exp\{\beta \boldsymbol{\mu}_y(\mathbf{v})^\top f_{\theta,\mathbf{v}}(x)\}}{\exp\{\beta \boldsymbol{\mu}_0(\mathbf{v})^\top f_{\theta,\mathbf{v}}(x)\} + \exp\{\beta \boldsymbol{\mu}_1(\mathbf{v})^\top f_{\theta,\mathbf{v}}(x)\}}.$$

A direct calculation gives

$$\nabla_f \mathcal{L}_\beta = \beta\Big(\boldsymbol{\mu}_y(\mathbf{v}) - q_\beta(0 \mid x)\boldsymbol{\mu}_0(\mathbf{v}) - q_\beta(1 \mid x)\boldsymbol{\mu}_1(\mathbf{v})\Big),$$

which can be rewritten as

$$\nabla_f \mathcal{L}_\beta = \beta\big(y - q_\beta(1 \mid x)\big)\big(\boldsymbol{\mu}_1(\mathbf{v}) - \boldsymbol{\mu}_0(\mathbf{v})\big).$$

Substituting this into (24) and conditioning on y yields

$$\begin{aligned}\nabla_{\mathbf{v}}\mathcal{J}(\mathbf{v}) = \frac{\beta}{2}\big(\boldsymbol{\mu}_1(\mathbf{v}) - \boldsymbol{\mu}_0(\mathbf{v})\big)^\top \Big(&\mathbb{E}_x[q_\beta(0 \mid x)\nabla_{\mathbf{v}} f_{\theta,\mathbf{v}}(x) \mid y = 1] \\ &- \mathbb{E}_x[q_\beta(1 \mid x)\nabla_{\mathbf{v}} f_{\theta,\mathbf{v}}(x) \mid y = 0]\Big),\end{aligned} \tag{25}$$

where we used $p(y = 0) = p(y = 1) = 1/2$. For each $c \in \{0, 1\}$, introduce the residual term

$$\Delta_c(\mathbf{v}) := \mathbb{E}_x\big[q_\beta(1 - c \mid x)\big(\nabla_{\mathbf{v}} f_{\theta,\mathbf{v}}(x) - \nabla_{\mathbf{v}}\boldsymbol{\mu}_c(\mathbf{v})\big) \mid y = c\big]\,. \tag{23}$$

Then, we obtain

$$\mathbb{E}_x[q_\beta(0 \mid x)\nabla_{\mathbf{v}} f_{\theta,\mathbf{v}}(x) \mid y = 1] = \mathbb{E}_x[q_\beta(0 \mid x) \mid y = 1]\nabla_{\mathbf{v}}\boldsymbol{\mu}_1(\mathbf{v}) + \Delta_1(\mathbf{v}),$$

and

$$\mathbb{E}_x[q_\beta(1 \mid x)\nabla_{\mathbf{v}} f_{\theta,\mathbf{v}}(x) \mid y = 0] = \mathbb{E}_x[q_\beta(1 \mid x) \mid y = 0]\nabla_{\mathbf{v}}\boldsymbol{\mu}_0(\mathbf{v}) + \Delta_0(\mathbf{v}).$$

Substituting these identities into (25) gives

$$\begin{aligned}\nabla_{\mathbf{v}}\mathcal{J}(\mathbf{v}) = \frac{\beta}{2}\big(\boldsymbol{\mu}_1(\mathbf{v}) - \boldsymbol{\mu}_0(\mathbf{v})\big)^\top \Big(&\mathbb{E}_x[q_\beta(0 \mid x) \mid y = 1]\nabla_{\mathbf{v}}\boldsymbol{\mu}_1(\mathbf{v}) \\ &- \mathbb{E}_x[q_\beta(1 \mid x) \mid y = 0]\nabla_{\mathbf{v}}\boldsymbol{\mu}_0(\mathbf{v})\Big) \\ &+ \frac{\beta}{2}\big(\boldsymbol{\mu}_1(\mathbf{v}) - \boldsymbol{\mu}_0(\mathbf{v})\big)^\top \big(\Delta_1(\mathbf{v}) - \Delta_0(\mathbf{v})\big).\end{aligned} \tag{26}$$

Under the uniform class prior and the matched case $\beta = \kappa$, $q_\beta(y \mid x) = p(y \mid x)$, so

$$\begin{aligned}\mathbb{E}_x[q_\beta(0 \mid x) \mid y = 1] &= \int p(0 \mid x) p(x \mid y = 1)\, dx \\ &= 2 \int p(0 \mid x) p(1 \mid x) p(x)\, dx \\ &= 2\mathbb{E}_x[q_\beta(0 \mid x) q_\beta(1 \mid x)].\end{aligned}$$

Similarly,

$$\mathbb{E}_x[q_\beta(1 \mid x) \mid y = 0] = 2\mathbb{E}_x[q_\beta(0 \mid x) q_\beta(1 \mid x)].$$

Using these identities in (26), we obtain

$$\begin{aligned}\nabla_{\mathbf{v}} \mathcal{J}(\mathbf{v}) = \beta\, \mathbb{E}_x[q_\beta(0 \mid x) q_\beta(1 \mid x)] \big(\boldsymbol{\mu}_1(\mathbf{v}) - \boldsymbol{\mu}_0(\mathbf{v})\big)^\top \big(\nabla_{\mathbf{v}} \boldsymbol{\mu}_1(\mathbf{v}) - \nabla_{\mathbf{v}} \boldsymbol{\mu}_0(\mathbf{v})\big) \\ + \frac{\beta}{2} \big(\boldsymbol{\mu}_1(\mathbf{v}) - \boldsymbol{\mu}_0(\mathbf{v})\big)^\top \big(\Delta_1(\mathbf{v}) - \Delta_0(\mathbf{v})\big).\end{aligned}$$

By using

$$\nabla_{\mathbf{v}} \|\boldsymbol{\mu}_1(\mathbf{v}) - \boldsymbol{\mu}_0(\mathbf{v})\|^2 = 2\big(\boldsymbol{\mu}_1(\mathbf{v}) - \boldsymbol{\mu}_0(\mathbf{v})\big)^\top \big(\nabla_{\mathbf{v}} \boldsymbol{\mu}_1(\mathbf{v}) - \nabla_{\mathbf{v}} \boldsymbol{\mu}_0(\mathbf{v})\big),$$

we obtain

$$\nabla_{\mathbf{v}} \mathcal{J}(\mathbf{v}) = \frac{\beta}{2}\, \mathbb{E}_x[q_\beta(0 \mid x) q_\beta(1 \mid x)]\, \nabla_{\mathbf{v}} \|\boldsymbol{\mu}_1(\mathbf{v}) - \boldsymbol{\mu}_0(\mathbf{v})\|^2 + \frac{\beta}{2} \big(\boldsymbol{\mu}_1(\mathbf{v}) - \boldsymbol{\mu}_0(\mathbf{v})\big)^\top \big(\Delta_1(\mathbf{v}) - \Delta_0(\mathbf{v})\big).$$

Finally, using the logistic form of q_β,

$$\begin{aligned}\frac{1}{2}\mathbb{E}_x[q_\beta(0 \mid x) q_\beta(1 \mid x)] &= \mathbb{E}_x\left[\frac{\exp\Big\{\beta\big(\boldsymbol{\mu}_0(\mathbf{v}) + \boldsymbol{\mu}_1(\mathbf{v})\big)^\top f_{\theta,\mathbf{v}}(x)\Big\}}{2\left(\sum_{c\in\{0,1\}} \exp\{\beta \boldsymbol{\mu}_c(\mathbf{v})^\top f_{\theta,\mathbf{v}}(x)\}\right)^2}\right] \\ &= \frac{1}{8}\left(1 - \mathbb{E}_x\left[\tanh^2\left(\frac{\beta}{2}\big(\boldsymbol{\mu}_1(\mathbf{v}) - \boldsymbol{\mu}_0(\mathbf{v})\big)^\top f_{\theta,\mathbf{v}}(x)\right)\right]\right),\end{aligned} \tag{27}$$

which yields

$$\nabla_{\mathbf{v}} \mathcal{J}(\mathbf{v}) = \beta\, \bar{\omega}(\mathbf{v})\, \nabla_{\mathbf{v}} \|\boldsymbol{\mu}_1(\mathbf{v}) - \boldsymbol{\mu}_0(\mathbf{v})\|^2 + \frac{\beta}{2} \big(\boldsymbol{\mu}_1(\mathbf{v}) - \boldsymbol{\mu}_0(\mathbf{v})\big)^\top \big(\Delta_1(\mathbf{v}) - \Delta_0(\mathbf{v})\big).$$

This completes the proof. □

H.2 Distribution Shift on Two Types of Errors

In practice, the deployment distribution may differ from the source distribution used for training and calibration. We consider this mismatch as a distribution shift between the source and deployment distributions. In this section, we quantify its impact via a Wasserstein perturbation and study how it affects Type I error control.

Assumption 4. *We denote by $\mathbb{P}_c$ the source distribution of class c used for training and calibration, and by $\widetilde{\mathbb{P}}_c$ the corresponding class-conditional distribution under deployment. Assume there exists a constant $\mathcal{E} > 0$ such that, for a fixed representation map $f_{\theta,\mathbf{v}} : \mathcal{X} \to \mathbb{S}^{d-1}$, the following holds for each class $c \in \{0, 1\}$:*

$$\mathcal{W}_1\Big((f_{\theta,\mathbf{v}})_\# \mathbb{P}_c,\ (f_{\theta,\mathbf{v}})_\# \widetilde{\mathbb{P}}_c\Big) \le \mathcal{E},$$

where $\mathcal{W}_1(\mu, \nu) := \inf_{\gamma \in \Pi(\mu,\nu)} \mathbb{E}_{(z,\tilde{z})\sim\gamma}[\|z - \tilde{z}\|]$ denotes the 1-Wasserstein distance on $\mathbb{S}^{d-1}$, and $(f_{\theta,\mathbf{v}})_\# \mathbb{P}_c$ denotes the pushforward measure of $\mathbb{P}_c$ under $f_{\theta,\mathbf{v}}$.

Proposition H.2. *Suppose Assumption 4 holds. For any $\delta \in (0, 1)$, with probability at least $1-\delta$ over the randomness of training and calibration samples, the Type-I error under the shifted distribution $\widetilde{\mathbb{P}}_0$ satisfies*

$$\widetilde{\mathbb{P}}_0\big(\widehat{\mathcal{S}}_t(X) > \widehat{\tau}_{\alpha,t}\big) - \alpha \le \inf_{\varepsilon > 0} \left(\sqrt{\frac{\log(2/\delta)}{2n_2}} + \frac{1}{n_2} + \mathbb{P}_0\left(\widehat{\tau}_{\alpha,t} - \varepsilon \le \widehat{\mathcal{S}}_t(X) \le \widehat{\tau}_{\alpha,t}\right) + \frac{2\kappa\mathcal{E}}{\varepsilon}\right) \wedge 1.$$

The Type-II error under the shifted distribution $\widetilde{\mathbb{P}}_1$ satisfies

$$\widetilde{\mathbb{P}}_1\big(\widehat{\mathcal{S}}_t(X) \le \widehat{\tau}_{\alpha,t}\big) \le \inf_{\varepsilon > 0} \left(\mathbb{P}_1\big(\widehat{\mathcal{S}}_t(X) \le \widehat{\tau}_{\alpha,t} + \varepsilon\big) + \frac{2\kappa\mathcal{E}}{\varepsilon}\right) \wedge 1.$$

Proposition H.2 characterizes how the Type-I and Type-II errors depend on the magnitude of the distribution shift $\mathcal{E}$. When $\mathcal{E} = 0$, the bounds recover the in-distribution guarantees in Theorem 3.1. When $\mathcal{E} > 0$, both errors incur an additional penalty that scales with $\mathcal{E}$, reflecting performance degradation under shift. The parameter ε captures a trade-off between the local probability mass near the threshold and the shift-induced error. Overall, the result indicates that the proposed method is robust to moderate shifts, while its performance degrades gracefully as the distribution shift increases.

Proof of Proposition H.2. For simplicity of notation, we drop the time index t in the proof. Let $f_{\theta,\mathbf{v}}(x) \in \mathbb{S}^{d-1}$ denote the representation for $x \in \mathcal{X}$. The deployed scoring function is defined as $\widehat{\mathcal{S}}(x) = \kappa\,(\widehat{\boldsymbol{\mu}}_1 - \widehat{\boldsymbol{\mu}}_0)^\top f_{\theta,\mathbf{v}}(x)$. Since the estimated mean directions $\widehat{\boldsymbol{\mu}}_1, \widehat{\boldsymbol{\mu}}_0$ lie on the unit hypersphere $\mathbb{S}^{d-1}$, by the triangle inequality, $\|\widehat{\boldsymbol{\mu}}_1 - \widehat{\boldsymbol{\mu}}_0\| \le \|\widehat{\boldsymbol{\mu}}_1\| + \|\widehat{\boldsymbol{\mu}}_0\| \le 2$. The gradient of $\widehat{\mathcal{S}}$ with respect to the representation f is given by

$$\|\nabla_f \widehat{\mathcal{S}}(x)\| = \kappa \|\widehat{\boldsymbol{\mu}}_1 - \widehat{\boldsymbol{\mu}}_0\| \le 2\kappa. \tag{28}$$

Let $\pi \in \Pi(\mathbb{P}_0, \widetilde{\mathbb{P}}_0)$ be the optimal coupling such that

$$\mathbb{E}_{(X,\widetilde{X})\sim\pi}[\|f_{\theta,\mathbf{v}}(X) - f_{\theta,\mathbf{v}}(\widetilde{X})\|] = \mathcal{W}_1\left((f_{\theta,\mathbf{v}})_{\#}\mathbb{P}_0, (f_{\theta,\mathbf{v}})_{\#}\widetilde{\mathbb{P}}_0\right) \le \mathcal{E}.$$

For any $\varepsilon > 0$, we decompose the Type-I error probability under the shifted distribution as

$$\begin{aligned}\widetilde{\mathbb{P}}_0\big(\widehat{\mathcal{S}}(\widetilde{X}) > \widehat{\tau}_\alpha\big) \le{}& \mathbb{P}_\pi\Big(\widehat{\mathcal{S}}(\widetilde{X}) > \widehat{\tau}_\alpha,\ \|f_{\theta,\mathbf{v}}(X) - f_{\theta,\mathbf{v}}(\widetilde{X})\| \le \frac{\varepsilon}{2\kappa}\Big) \\ &+ \mathbb{P}_\pi\Big(\|f_{\theta,\mathbf{v}}(X) - f_{\theta,\mathbf{v}}(\widetilde{X})\| > \frac{\varepsilon}{2\kappa}\Big).\end{aligned} \tag{29}$$

For the first term on the RHS of (29), by using (28), we obtain

$$|\widehat{\mathcal{S}}(X) - \widehat{\mathcal{S}}(\widetilde{X})| \le 2\kappa \|f_{\theta,\mathbf{v}}(X) - f_{\theta,\mathbf{v}}(\widetilde{X})\| \le \varepsilon.$$

Thus, $\widehat{\mathcal{S}}(\widetilde{X}) > \widehat{\tau}_\alpha$ implies $\widehat{\mathcal{S}}(X) > \widehat{\tau}_\alpha - \varepsilon$. By Theorem 3.1, with probability at least $1 - \delta$ over the calibration sample $\mathcal{S}_{\mathrm{cal}}$,

$$\mathbb{P}_0\Big(\widehat{\mathcal{S}}(X) > \widehat{\tau}_\alpha\Big) \le \alpha + \sqrt{\frac{\log(2/\delta)}{2n_2}} + \frac{1}{n_2}.$$

On this high-probability event, we therefore have

$$\mathbb{P}_\pi\Big(\widehat{\mathcal{S}}(\widetilde{X}) > \widehat{\tau}_\alpha,\ \|f_{\theta,\mathbf{v}}(X) - f_{\theta,\mathbf{v}}(\widetilde{X})\| \le \frac{\varepsilon}{2\kappa}\Big) \le \mathbb{P}_0\big(\widehat{\mathcal{S}}(X) > \widehat{\tau}_\alpha\big) + \mathbb{P}_0\big(\widehat{\tau}_\alpha - \varepsilon \le \widehat{\mathcal{S}}(X) \le \widehat{\tau}_\alpha\big). \tag{30}$$

For the second term on the RHS of (29), applying Markov's inequality yields

$$\mathbb{P}_\pi\Big(\|f_{\theta,\mathbf{v}}(X) - f_{\theta,\mathbf{v}}(\widetilde{X})\| > \frac{\varepsilon}{2\kappa}\Big) \le \frac{2\kappa}{\varepsilon}\mathbb{E}_\pi\big[\|f_{\theta,\mathbf{v}}(X) - f_{\theta,\mathbf{v}}(\widetilde{X})\|\big] \le \frac{2\kappa\mathcal{E}}{\varepsilon}.$$

Substituting (30) into (29) and applying Markov's inequality, we obtain, for any $\varepsilon > 0$,

$$\begin{aligned}\widetilde{\mathbb{P}}_0\big(\widehat{\mathcal{S}}(\widetilde{X}) > \widehat{\tau}_\alpha\big) &\le \mathbb{P}_0\big(\widehat{\mathcal{S}}(X) > \widehat{\tau}_\alpha\big) + \mathbb{P}_0\Big(\widehat{\tau}_\alpha - \varepsilon \le \widehat{\mathcal{S}}(X) \le \widehat{\tau}_\alpha\Big) + \frac{2\kappa\mathcal{E}}{\varepsilon} \\ &\le \alpha + \sqrt{\frac{\log(2/\delta)}{2n_2}} + \frac{1}{n_2} + \mathbb{P}_0\Big(\widehat{\tau}_\alpha - \varepsilon \le \widehat{\mathcal{S}}(X) \le \widehat{\tau}_\alpha\Big) + \frac{2\kappa\mathcal{E}}{\varepsilon}.\end{aligned}$$

Taking the infimum over $\varepsilon > 0$ gives the stated Type-I bound.

Now we turn to the Type-II error. Let $\pi' \in \Pi(\mathbb{P}_1, \widetilde{\mathbb{P}}_1)$ be an optimal coupling satisfying

$$\mathbb{E}_{(X,\widetilde{X})\sim\pi'}\Big[\|f_{\theta,\mathbf{v}}(X) - f_{\theta,\mathbf{v}}(\widetilde{X})\|\Big] \le \mathcal{E}.$$

For any $\varepsilon > 0$,

$$\begin{aligned}\widetilde{\mathbb{P}}_1\big(\widehat{\mathcal{S}}(\widetilde{X}) > \widehat{\tau}_\alpha\big) &\ge \mathbb{P}_{\pi'}\Big(\widehat{\mathcal{S}}(\widetilde{X}) > \widehat{\tau}_\alpha,\ \|f_{\theta,\mathbf{v}}(X) - f_{\theta,\mathbf{v}}(\widetilde{X})\| \le \frac{\varepsilon}{2\kappa}\Big) \\ &\ge \mathbb{P}_{\pi'}\Big(\widehat{\mathcal{S}}(X) > \widehat{\tau}_\alpha + \varepsilon,\ \|f_{\theta,\mathbf{v}}(X) - f_{\theta,\mathbf{v}}(\widetilde{X})\| \le \frac{\varepsilon}{2\kappa}\Big) \\ &\ge \mathbb{P}_1\big(\widehat{\mathcal{S}}(X) > \widehat{\tau}_\alpha + \varepsilon\big) - \mathbb{P}_{\pi'}\Big(\|f_{\theta,\mathbf{v}}(X) - f_{\theta,\mathbf{v}}(\widetilde{X})\| > \frac{\varepsilon}{2\kappa}\Big) \\ &\ge \mathbb{P}_1\big(\widehat{\mathcal{S}}(X) > \widehat{\tau}_\alpha + \varepsilon\big) - \frac{2\kappa\mathcal{E}}{\varepsilon}.\end{aligned}$$

Equivalently,

$$\widetilde{\mathbb{P}}_1\big(\widehat{\mathcal{S}}(\widetilde{X}) \le \widehat{\tau}_\alpha\big) \le \mathbb{P}_1\big(\widehat{\mathcal{S}}(X) \le \widehat{\tau}_\alpha + \varepsilon\big) + \frac{2\kappa\mathcal{E}}{\varepsilon}.$$

Taking the infimum over $\varepsilon > 0$ gives the stated Type-II bound. □

I Broader Impacts

The proposed S2D framework contributes to improving the reliability of distinguishing LLM-generated text from human-written text, which has implications for mitigating risks such as misinformation, academic misconduct, and erosion of trust in digital content. By leveraging hidden representations and providing a statistically grounded detection procedure with controlled Type-I error, our method is particularly relevant in high-stakes scenarios where false accusations (e.g., misclassifying human-written text as machine-generated) carry ethical consequences. In addition, the representation-based nature of S2D allows it to operate without requiring access to the generation process, making it broadly applicable in real-world settings where watermarking is unavailable.

More broadly, such capabilities may influence how content is created and consumed, potentially shaping emerging norms around disclosure and authenticity. They may also inform policy discussions on transparency and responsible AI use, including whether and how generated content should be labeled. Overall, advances in this area contribute to a broader effort to maintain trust, accountability, and informed decision-making in an increasingly AI-mediated information landscape.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract are completely based on our work.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discussed limitations and the potential future work together in Section 5.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: The paper clearly states the assumptions required for each theoretical result and provides corresponding proofs (primarily in the Appendix G).

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Disclosed all the experiment information in Section 4 and Appendix D.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: All data are open-source, information is provided in the content, and code link is provided in Section 4.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We detailed all the experiment information in Section 4 and Appendix D.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: In our experiments, we evaluate our method and baseline methods on five runs with different random samplings of the training data. We report the average performance across the datasets.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Provided in Section D.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in this paper conforms in every respect with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Discussed in Appendix I.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This paper does not involve the release of data, models, or other assets that pose a high risk of misuse.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All the data, code, and models used in the paper are properly credited.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: This work involves the use of large language models to extract hidden representations and to rewrite texts. We provided the detailed usage of LLMs in Section 4 and Appendix D.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.