Not All Tokens Are Equal: Syntax-Pattern Weighted Likelihood Difference Joint Detection for Neural-Training-Free AI-Generated Text Detection

Tianyu Guo[1], **Shufeng Hao**[2]*, **Jiaqi Li**[1], **Chongyang Shi**[1]
[1]Beijing Institute of Technology, China [2]Taiyuan University of Technology, China
[1]{3220231209, 3220251257, cy_shi}@bit.edu.cn [2]haoshufeng@tyut.edu.cn

Abstract

Existing neural-training-free AI-Generated Text (AIGT) detectors usually aggregate token-level probability signals with uniform weights. However, this ignores an important linguistic regularity: human-written and AI-generated texts exhibit different Part-Of-Speech (POS) n-gram patterns, making certain tokens more informative than others. To address the limitations, we propose **SWLD-Joint** (**S**yntax-pattern **W**eighted **L**og-likelihood **D**ifference **J**oint Detection), a calibration-based neural-training-free detector without gradient-based optimization or neural classifier fine-tuning. SWLD-Joint first estimates POS n-gram template weights from labeled calibration data and then combines them with a POS-trajectory calibration signal to focus on syntactically informative positions. Experiments on DetectRL, RAID, and HC3_Ext show robust gains under domain shift, generator mismatch, and adversarial perturbations, with SWLD-Joint achieving the best or second-best average AUROC on DetectRL and RAID and remaining highly competitive on HC3_Ext. Ablations, parameter analysis, case studies, score distributions, and ROC curves verify the effectiveness of syntax-pattern weighting. Dataset and code are available at https://github.com/Guotianyu-2020/SWLD-Joint-AIGT-Detection.

1 Introduction

Large language models (LLMs), such as ChatGPT, GPT-4, Qwen, and DeepSeek, can now produce fluent and coherent text that is increasingly difficult to distinguish from human writing (Mitchell et al., 2023; Wu et al., 2024). Their misuse in academic plagiarism, misinformation, and content manipulation makes reliable AIGT detection important (Fraser et al., 2025; Tufts et al., 2025).

Existing neural-training-free detectors avoid updating neural model parameters and instead derive detection scores from language-model probability signals. Single-model methods use statistics such as log-likelihood, entropy, or token rank (Solaiman et al., 2019; Hashimoto et al., 2019; Gehrmann et al., 2019; Su et al., 2023); perturbation-based methods estimate probability curvature around the observed text (Mitchell et al., 2023; Bao et al., 2024); and dual-model methods compare probability estimates from two language models, as in Binoculars (Hans et al., 2024) and IRM (Liu et al., 2025). Recent work also shows that detectors remain sensitive to domain shift, false-positive constraints, and paraphrasing attacks (Chen et al., 2025; Zhu et al., 2025; Nguyen-Son et al., 2026). Despite their differences, these methods are largely *syntax-agnostic*: they typically aggregate token-level signals without explicitly distinguishing syntactically informative positions from neutral ones. As illustrated in Figure 1, these methods can dilute useful syntactic evidence when only a subset of tokens appears in highly discriminative syntactic contexts.

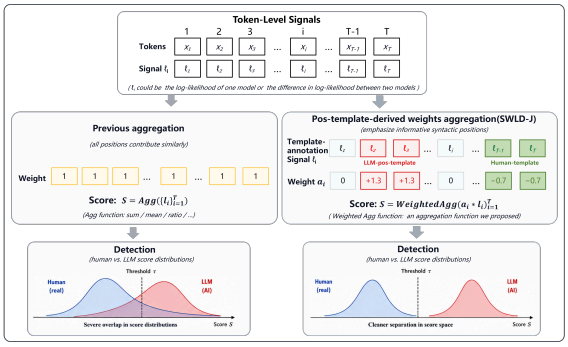


Figure 1: Motivation of SWLD-Joint. Existing neural-training-free detectors aggregate token-level probability signals in a largely syntax-agnostic way, whereas SWLD-Joint assigns larger weights to tokens covered by discriminative POS templates.

Syntactic contexts are informative for AIGT detection. Prior work shows that AI-generated and human-written texts exhibit systematic differences in POS n-gram distributions. For example, Shaib

*Corresponding author.

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 40743–40758
October 24-29, 2026 ©2026 Association for Computational Linguistics

et al. (2024) report that, in summarization, model-generated text contains length-6 POS templates more often than human reference text, and recent linguistic analyses link detector generalization to linguistic feature shifts (Xia et al., 2026). In our setting, we observe similar POS-template preference gaps on DetectRL (Appendix B). These observations suggest that POS templates provide a compact and interpretable signal for identifying generation-specific syntactic regularities.

Based on this insight, we argue that *not all token positions should contribute equally to AIGT detection*. Tokens inside POS patterns strongly preferred by AI-generated or human-written text should receive greater influence than tokens in neutral contexts. We therefore propose **SWLD-Joint**, a calibration-based neural-training-free detector that computes a dual-model syntax-aware likelihood difference score. SWLD-Joint first computes token-level log-likelihood differences between a pre-trained base model and its instruction-tuned counterpart. It then builds a POS n-gram template library from labeled calibration data and converts each human-vs-AI frequency gap into token-level weights. The final score aggregates likelihood differences with these syntax-derived weights, allowing the detector to focus on discriminative syntactic positions without updating neural parameters. The contributions are as follows:

- We propose **SWLD-Joint**, a neural-training-free AIGT detector that injects POS-template statistics into dual-model likelihood-difference aggregation, making token-level probability scoring explicitly syntax-aware.
- The discriminative POS n-gram template library is designed, where it converts labeled calibration statistics into interpretable token weights, and examines both global and local template construction strategies.
- We construct an HC3-based benchmark **HC3_Ext** that extends evaluation to recent generators and DeepSeek-based humanization attacks.

2 Related Work

2.1 Neural-Training-Free AIGT Detection Methods

Neural-training-free AIGT detectors identify machine-generated text without training a neural classifier. They mainly rely on language-model probability signals, including log-likelihood, entropy, token rank, log-rank ratio, entropy-normalized likelihood, probability curvature, and recent domain- or reliability-aware scoring strategies (Solaiman et al., 2019; Hashimoto et al., 2019; Gehrmann et al., 2019; Su et al., 2023; Lu et al., 2024; Xu et al., 2024; Mitchell et al., 2023; Bao et al., 2024; Chen et al., 2025; Zhu et al., 2025). Recent dual-model methods further compare probability estimates from two language models, such as Binoculars, IRM, and LastDE++ (Hans et al., 2024; Liu et al., 2025; Xu et al., 2024). Practical studies also show degradation under unseen domains, unseen models, and adversarial prompting (Tufts et al., 2025).

These methods provide strong training-free baselines, but they aggregate all token positions uniformly: Binoculars averages token probabilities, IRM performs entropy minimization across sequences, and other likelihood-, rank-, or curvature-based scorers likewise treat every token as equally informative. SWLD-Joint retains the dual-model likelihood-difference backbone but departs from this uniform-aggregation assumption by weighting token positions through POS-template-derived syntax weights.

2.2 Syntactic Signals for AIGT Detection

Linguistic studies show that AI-generated and human-written texts differ not only in probability statistics, but also in structural patterns. Shaib et al. (2024) report that generated text exhibits repeated POS n-gram templates, while other work explores linguistic, stylistic, or representation-level regularities for detection and characterization (Tulchinskii et al., 2023; Xia et al., 2026; Wu et al., 2025).

However, these syntactic observations are rarely integrated into neural-training-free likelihood scoring. Our work bridges this gap by converting human-vs-AI POS n-gram frequency differences into token-level weights and folding them directly into the dual-model likelihood difference score, keeping the detector neural-training-free without training a separate feature-based classifier. Unlike post-hoc calibration, which rescales a scalar score globally, our weights are position-specific and therefore preserve local syntactic evidence.

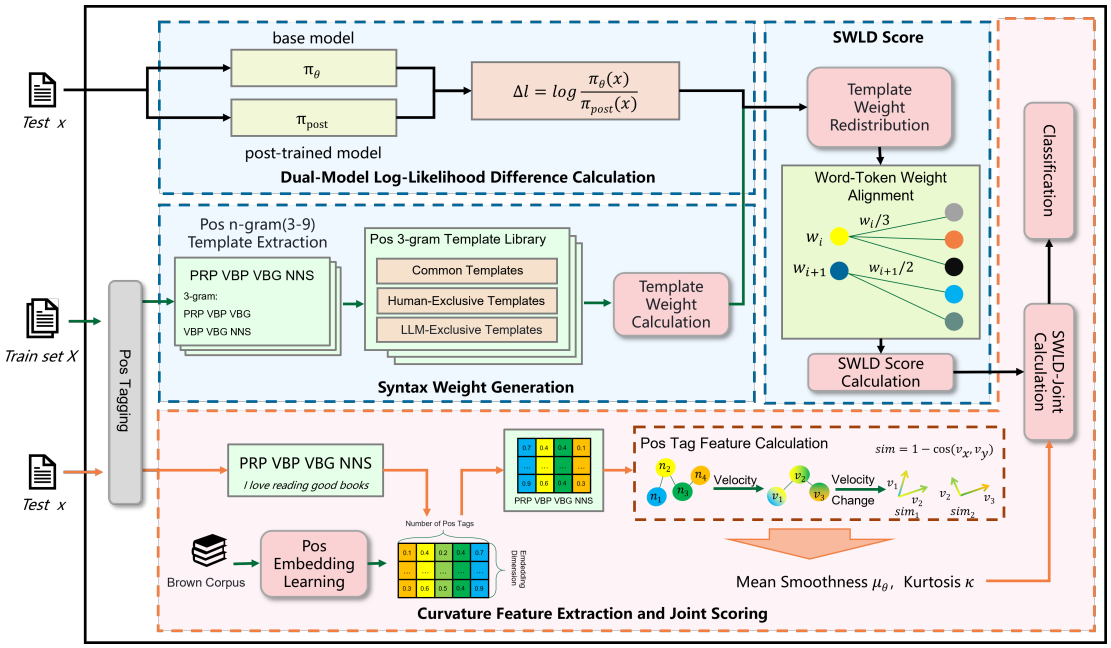


Figure 2: Overview of SWLD-Joint. The framework consists of five modules: (1) dual-model log-likelihood difference calculation, which compares a base model and its instruction-tuned counterpart; (2) syntax weight generation, which builds a discriminative POS n-gram template library from labeled calibration data; (3) SWLD score calculation, which aligns word-level template weights to subword tokens and computes the syntax-weighted likelihood-difference score; (4) curvature feature extraction and joint scoring, which uses global POS-trajectory statistics as a lightweight calibration signal; and (5) classification based on the final ranking score.

3 Method

3.1 Overview

Figure 2 illustrates the overall architecture of SWLD-Joint. To make the method description consistent with the framework, we organize this section according to the main modules in the figure.

First, the **dual-model log-likelihood difference calculation** module computes token-level likelihood shifts by comparing a pre-trained base model with its instruction-tuned counterpart. Second, the **syntax weight generation** module POS-tags the input text, extracts POS n-gram templates from labeled calibration data, and assigns each template a discriminative human-vs-AI weight. Third, the **SWLD score calculation** module redistributes word-level template weights to subword tokens and computes a syntax-pattern weighted likelihood-difference score. Finally, the **curvature feature extraction and joint scoring** module uses global POS-trajectory statistics as a lightweight calibration signal. No neural parameters are updated during detection; both language-model inference passes use `torch.no_grad()`, and the additional cost comes only from POS tagging, dictionary lookup, and simple feature calculation.

3.2 Problem Formulation

Given a text sequence $\mathbf{x} = (x_1, x_2, \ldots, x_T)$, where x_t is the t-th subword token, and a binary label $y \in \{0, 1\}$ ($y = 0$: human; $y = 1$: LLM-generated), the goal is to design a scoring function $S : \mathcal{X} \to \mathbb{R}$ such that human and AI texts have separable score distributions. In this work, we focus on threshold-free ranking-based evaluation with AUROC, avoiding any test-set threshold tuning.

We assume access to a pre-trained base model M_{base} and its instruction-tuned counterpart M_{post}. No parameters of either model are updated during detection. The only corpus statistics used by SWLD-Joint are POS n-gram frequencies estimated from labeled calibration data.

3.3 Dual-Model Log-Likelihood Difference Calculation

Instruction tuning and alignment shift the token probability distribution of a base model toward instruction-following and human-preferred outputs. Following dual-model detection methods, we capture this shift using token-level log-likelihood difference:

$$\Delta\ell_t = \log p_{M_{\text{post}}}(x_t \mid \mathbf{x}_{<t}) - \log p_{M_{\text{base}}}(x_t \mid \mathbf{x}_{<t}), \quad (1)$$

for $t = \{2, \ldots, T\}$; the first position is set to zero because it has no left context. The unweighted sum $\sum_t \Delta\ell_t$ is the core signal used by IRM-style dual-model detectors. SWLD-Joint retains this signal but changes how token positions are aggregated: instead of treating all tokens equally, it uses syntax-derived weights to emphasize positions covered by discriminative POS templates.

3.4 Syntax Weight Generation

This module constructs a POS n-gram template library from labeled calibration data and converts each template into a discriminative weight.

POS tagging. For each input text, we use an English Penn Treebank POS tagger to obtain a word-level POS sequence $\{(w_i, p_i)\}_{i=1}^{m}$, where w_i is the i-th word and p_i is its POS tag.

POS n-gram template extraction. From the labeled calibration split, we extract all POS n-grams of length n from human and AI texts. For each POS pattern k, we define its human and AI frequencies as f_k^{human} and f_k^{llm}. Patterns appearing fewer than 10 times in total are removed to reduce low-frequency noise.

Discriminative template weight. Each template is assigned a log probability-ratio weight:

$$w_k = \log_2 \frac{P_k^{\text{llm}}}{P_k^{\text{human}}}, \tag{2}$$

where P_k^{human} and P_k^{llm} are the smoothed normalized probabilities of template k in human-written and AI-generated texts, respectively. Specifically, let

$$N_{\text{human}} = \sum_k f_k^{\text{human}}, \quad N_{\text{llm}} = \sum_k f_k^{\text{llm}}. \tag{3}$$

With smoothing $\epsilon = 10^{-7}$, the normalized probabilities are

$$P_k^{\text{human}} = \frac{f_k^{\text{human}}}{N_{\text{human}}} + \epsilon, \quad P_k^{\text{llm}} = \frac{f_k^{\text{llm}}}{N_{\text{llm}}} + \epsilon. \tag{4}$$

A positive w_k indicates an AI-preferred syntactic pattern; a negative w_k indicates a human-preferred pattern. The magnitude $|w_k|$ reflects the strength of the preference gap. The base-2 logarithm gives an intuitive bit-scale interpretation: $w_k = 1$ means that the pattern is twice as likely in AI-generated text.

Word-level template weight assignment. We slide a window of length n over the word-level POS sequence. If the subsequence $p_i \cdots p_{i+n-1}$ matches template k, each word in the window receives an additive contribution:

$$a_j^{\text{word}} \mathrel{+}= \frac{w_k}{n}, \quad j = i, \ldots, i+n-1. \tag{5}$$

A word covered by multiple matched templates accumulates their contributions. Positions not covered by any matched template retain weight zero.

Global and local template libraries. Global templates aggregate calibration subsets into one library, capturing general human-vs-AI syntactic preferences. Local templates build a separate library for each generator or domain subset, capturing target-subset-specific syntactic preferences when labeled calibration data from that subset is available. In tables, the global-template variant of SWLD-Joint is denoted **SWLD-G** and the local-template variant is denoted **SWLD-L**.

For SWLD-L, the POS template library is constructed separately for each evaluation subset using only the labeled calibration split of that subset. Therefore, SWLD-L corresponds to a target-subset calibrated setting rather than a fully zero-shot unseen-generator setting. For SWLD-G, calibration subsets are pooled to construct a single global template library. In all cases, test examples are disjoint from calibration and validation examples, and test labels are never used for template construction or hyperparameter selection. The no-curvature variant is reported only in the ablation (Section 4.5) to quantify component contributions.

3.5 SWLD Score Calculation

Since language models operate on subword tokens whereas POS templates are defined at the word level, we first redistribute word-level syntax weights to subword tokens.

Word-token weight alignment. We align word-level POS labels to subword tokens using the tokenizer's offset mapping. If word w_i is split into c_i subword tokens and receives a word-level syntactic weight a_i^{word}, each corresponding subword token receives a_i^{word}/c_i. This gives the aligned token-level weights $\{a_t\}_{t=1}^{T}$.

Syntax-pattern weighted likelihood difference. Given token-level log-likelihood differences $\{\Delta\ell_t\}_{t=1}^{T}$ and aligned syntactic weights

$\{a_t\}_{t=1}^{T}$, the SWLD score is:

$$S_{\mathrm{SWLD}}(\mathbf{x}) = \frac{\sum_{t=1}^{T} a_t \cdot \Delta\ell_t}{\sum_{t=1}^{T} \Delta\ell_t}. \quad (6)$$

If $|\sum_t \Delta\ell_t| < 10^{-10}$, the score is set to zero for numerical stability.

The denominator measures the total alignment-induced likelihood shift of the sequence, while the numerator measures how much of this shift is concentrated on positions covered by discriminative syntactic templates. Therefore, S_{SWLD} is a syntax-conditioned concentration score: it asks whether the likelihood shift occurs at syntactically informative positions rather than being uniformly aggregated over all tokens. This reduces the dilution effect caused by low-information tokens and makes the score more sensitive to human-vs-AI syntactic preferences.

3.6 Curvature Feature Extraction and Joint Scoring

The syntax-pattern weighted component captures local POS n-gram evidence. SWLD-Joint further incorporates a lightweight global calibration signal from the overall trajectory of POS transitions. This module corresponds to the **Curvature Feature Extraction and Joint Scoring** block in Figure 2. Implementation details for POS embedding training are provided in Section 4.2.

Given a POS sequence $\{p_i\}_{i=1}^{m}$, we map each POS tag to a continuous embedding vector $\mathbf{v}_i$. We then compute velocity vectors $\mathbf{u}_i = \mathbf{v}_{i+1} - \mathbf{v}_i$ and adjacent cosine similarities:

$$\theta_i = \frac{\mathbf{u}_i \cdot \mathbf{u}_{i+1}}{\|\mathbf{u}_i\| \cdot \|\mathbf{u}_{i+1}\|}, \quad i = 1, \ldots, m-2. \quad (7)$$

Positions with zero-norm velocity are skipped. From the valid values $\{\theta_i\}$, we extract the mean smoothness $\mu_\theta = \mathrm{mean}(\{\theta_i\})$ and Fisher kurtosis $\kappa = \mathrm{kurt}(\{\theta_i\})$. If fewer than three valid values remain, the score falls back to S_{SWLD}.

The final joint score is:

$$S_{\mathrm{joint}}(\mathbf{x}) = S_{\mathrm{SWLD}}(\mathbf{x}) \cdot \frac{1 + \tanh(\kappa)}{1 + \mu_\theta}. \quad (8)$$

The $\tanh(\kappa)$ term bounds the influence of extreme kurtosis values (factor range $[0, 2]$), while the denominator $(1 + \mu_\theta)$ adjusts the score according to the smoothness of POS transitions. Multiplicative fusion preserves the sign of S_{SWLD} and requires no additional balancing hyperparameter. The ablation in Section 4.5 shows that the main gain comes from syntax-pattern weighted likelihood aggregation, while the curvature module provides only a modest but complementary calibration signal.

3.7 Classification

The final output of SWLD-Joint is a scalar score. For threshold-free evaluation, we directly use $S_{\mathrm{joint}}(\mathbf{x})$ to rank human-written and AI-generated texts and report AUROC. In deployment scenarios where binary labels are required, a threshold can be selected on a validation set, but this paper reports only threshold-free ranking performance.

4 Experiments

4.1 Datasets and Setup

Datasets. We evaluate on three benchmarks: DetectRL (Wu et al., 2024), RAID (Dugan et al., 2024), and HC3_Ext. DetectRL is used as the primary benchmark and contains three scenarios: Multi-Domain (ArXiv, Writing Prompt, XSum, Yelp Review), Multi-LLM (ChatGPT, Claude-instant, Google-PaLM, Llama-2-70B), and Multi-Attack (Data Mixing, Paraphrase, Perturbation, Prompt). RAID provides a broader per-generator robustness test across 11 generators. HC3_Ext is built to complement existing benchmarks with newer models: it keeps HC3 questions and human answers, regenerates the machine-written side with Claude Sonnet 4, DeepSeek-R1, and Qwen3-235B, and adds a DeepSeek-humanized split where 40% of LLM answers are lightly rewritten while preserving meaning and paragraph structure. Construction details and full per-subset results are provided in Appendix A and Appendix C.

4.2 Implementation Details

All datasets use a 6:1:3 calibration/validation/test split. The calibration split is used **only** to compute POS-template statistics (Eq. 2–4); the validation split is used **only** to select the POS n-gram length n and to orient baseline score directions; the test split is used **only** for final evaluation and **never** influences template construction or hyperparameter selection. For SWLD-L, template statistics are computed **solely** from the calibration portion of the target subset itself, which is why we describe SWLD-L as *target-subset calibrated* rather than

zero-shot. We use Qwen3-4B-Base as M_{base} and Qwen3-4B as M_{post}; both models share the same tokenizer and are loaded in float16 on separate GPUs. We search $n \in \{3, 4, 5, 6, 7, 8, 9\}$ and use $n = 6$ as the default setting in ablations. For the curvature module, we train a Word2Vec Skip-gram model on Brown Corpus POS sequences with $d = 128$ and window size 12. We report AUROC as the primary threshold-free metric, tune no decision threshold on the test set, and fix all random seeds at 42.

4.3 Baselines

Single-model baselines. Likelihood (Hashimoto et al., 2019), Entropy (Lu et al., 2024), Log-Rank (Gehrmann et al., 2019), LRR (Su et al., 2023), and LastDE (Xu et al., 2024).

Dual-model baselines. IRM (Liu et al., 2025), Binoculars (Hans et al., 2024), Fast-DetectGPT (Bao et al., 2024), and LastDE++ (Xu et al., 2024). All dual-model baselines use the same Qwen3-4B-Base/Qwen3-4B pair as SWLD-Joint.

Table 1: Dataset-level average AUROC (%). Best results are in **bold**, and second-best results are underlined. DetectRL is averaged over all available evaluation subsets. SWLD-G and SWLD-L denote global- and local-template variants.

Method	DetectRL	RAID	HC3_Ext
Likelihood	73.07	71.52	96.08
Entropy	73.61	67.40	**98.04**
Log-Rank	74.19	72.74	96.08
LRR	76.01	76.90	93.88
LastDE	62.08	77.63	62.21
IRM	75.27	66.85	78.45
Binoculars	71.61	74.07	93.16
F-DetectGPT	69.80	75.77	48.96
LastDE++	60.53	72.40	41.91
SWLD-G	**95.49**	80.05	97.43
SWLD-L	92.15	**83.02**	97.27

4.4 Main Results

Overall summary. Table 1 reports average AUROC scores across DetectRL, RAID, and HC3_Ext. Overall, SWLD-Joint variants achieve the best or second-best average AUROC on DetectRL and RAID, and remain highly competitive on HC3_Ext, where Entropy attains the highest average AUROC (98.04% vs. 97.43% for SWLD-G). SWLD-G performs best on DetectRL (95.49%) and remains close to the saturated upper range on HC3_Ext (97.43%), while SWLD-L achieves the best result on RAID (83.02%). These results show that syntax-pattern weighted scoring provides stable gains under different evaluation settings by redirecting dual-model likelihood signals toward token positions covered by discriminative POS templates. The two variants further show complementary strengths: SWLD-G captures generalizable human–AI syntactic regularities from pooled calibration data, whereas SWLD-L adapts to generator-specific syntactic styles through local calibration.

Primary benchmark: DetectRL. Table 2 reports scenario-level average AUROC on DetectRL. SWLD-G achieves the best result in all three settings: 96.41% on Multi-Domain, 95.87% on Multi-LLM, and 94.19% on Multi-Attack. Compared with the strongest non-SWLD baseline in each setting, the improvements are 18.94, 19.43, and 19.43 points, respectively.

These gains show that SWLD-Joint is not only effective in one specific evaluation condition. In Multi-Domain, the advantage indicates that syntax-pattern signals are less tied to topic or domain vocabulary. In Multi-LLM, the improvement suggests that different generators still leave shared POS-template regularities after instruction tuning. In Multi-Attack, SWLD-G remains strong under perturbed or rewritten inputs, showing that syntax-pattern weighted scoring captures structural evidence beyond surface probability fluctuations. Notably, SWLD-G also outperforms SWLD-L on DetectRL, suggesting that pooled calibration data provides more stable template statistics for this mixed benchmark. Full per-subset DetectRL results are provided in Appendix C.1.

Table 2: DetectRL scenario-level average AUROC (%). Best results are in **bold**, and second-best results are underlined.

Method	Multi-Domain	Multi-LLM	Multi-Attack	Avg.
Likelihood	73.83	73.33	72.06	73.07
Entropy	73.99	74.20	72.64	73.61
Log-Rank	74.96	74.47	73.13	74.19
LRR	76.86	76.44	74.71	76.01
LastDE	61.23	59.84	65.15	62.08
IRM	77.47	73.59	74.76	75.27
Binoculars	72.88	71.68	70.26	71.61
F-DetectGPT	71.62	69.51	68.27	69.80
LastDE++	59.80	59.78	62.01	60.53
SWLD-L	92.99	90.78	92.87†	92.15†
SWLD-G	**96.41**	**95.87**	**94.19**	**95.49**

† SWLD-L excludes Data Mixing because this merged subset has no local template; its Multi-Attack and overall averages use 3 and 11 subsets, respectively.

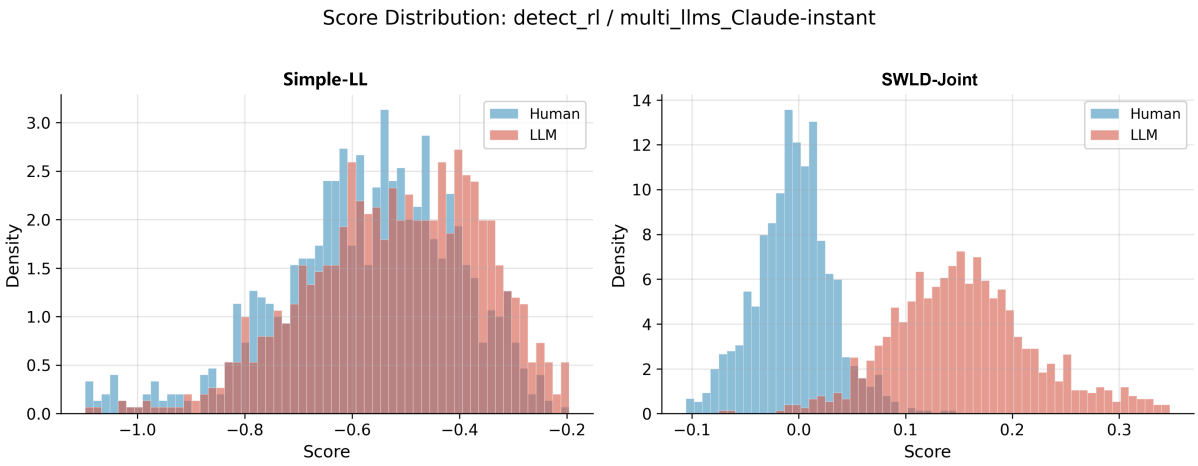


Figure 3: Score distribution on DetectRL / Claude-instant. Syntax-pattern weighting produces clearer human–LLM separation than the unweighted likelihood-difference signal.

4.5 Ablation Study

Table 3 compares two variants with SWLD-Joint by using $n = 6$ and global templates. Simple ΔLL directly uses the unweighted dual-model likelihood-difference signal; SWLD w/o Curv. adds POS-template weighted aggregation but removes curvature calibration.

Table 3: Ablation results measured by AUROC (%). Best results are in **bold**, and second-best results are underlined.

Variant	DetectRL	HC3_Ext	RAID
Simple ΔLL	75.50	84.35	70.48
SWLD w/o Curv.	<u>94.89</u>	**93.63**	<u>79.25</u>
SWLD-Joint	**95.05**	<u>93.61</u>	**79.48**

Compared with Simple ΔLL, SWLD w/o Curv. improves AUROC by 19.52 points on DetectRL, 9.68 points on HC3_Ext, and 8.77 points on RAID. Adding curvature calibration brings smaller changes: +0.16 on DetectRL and +0.23 on RAID, but -0.02 on HC3_Ext. This suggests that global POS-trajectory statistics are not the dominant factor, but can provide a lightweight complementary signal in some settings.

4.6 Parameter Sensitivity and Template Coverage

Figure 4 analyzes the effect of POS n-gram length. AUROC shows an inverted-U trend across the three benchmarks, peaking around $n = 6$ and 7. This pattern reflects a trade-off between template specificity and template coverage. Short templates cover many tokens but are often too generic to distinguish human and AI syntax, while very long templates are more specific but become sparse and less reliable. The best performance is therefore obtained when POS templates are long enough to encode discriminative syntactic contexts but still frequent enough to provide stable evidence.

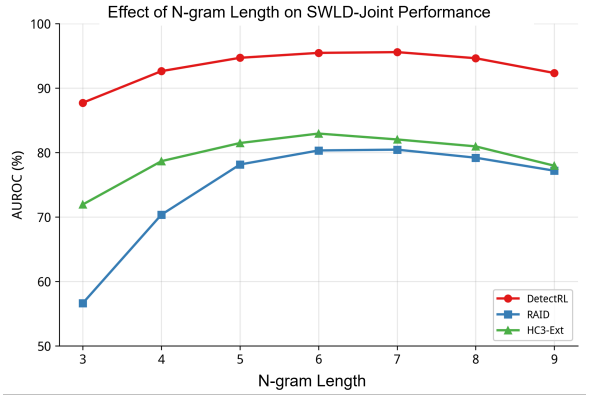


Figure 4: Effect of POS n-gram length on AUROC. Performance generally peaks around $n = 6$ and 7, reflecting a balance between syntactic specificity and statistical reliability.

Figure 5 supports this interpretation. The syntax token ratio decreases monotonically as n increases, from near-complete coverage at short n-grams to sparse activation at long n-grams. However, higher coverage does not directly lead to higher AUROC: $n = 3$ and 4 cover most tokens but perform worse than $n = 6$ and 7, showing that many short-template matches are weakly informative. When n reaches 8 and 9, the sharp drop in coverage leaves too few activated positions for stable aggregation. Together, the two figures show that SWLD-Joint works best in a middle regime where POS templates are both discriminative and sufficiently frequent.

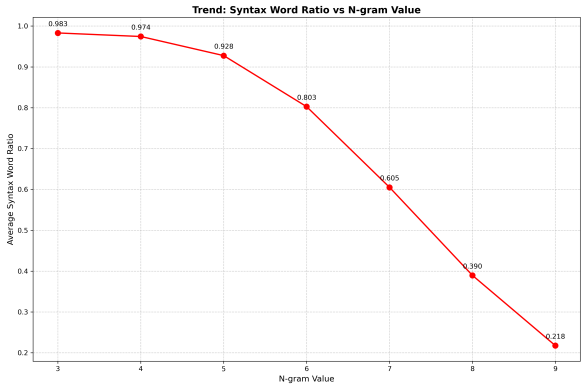


Figure 5: Syntax token ratio by POS n-gram length. Longer POS templates become sparser and cover fewer token positions, reducing the amount of syntax-activated evidence available for scoring.

4.7 Case Study

We examine the DetectRL Claude-instant subset, a challenging case where baselines show weak separability, while SWLD-G reaches 97.93% AUROC (Appendix 6). As shown in Figure 3, syntax-pattern weighting produces clearer human–LLM separation than the unweighted likelihood-difference signal. We further inspect one LLM-generated and one human-written news article from this subset, and then discuss a failure case to clarify the limitation of syntax-pattern weighting.

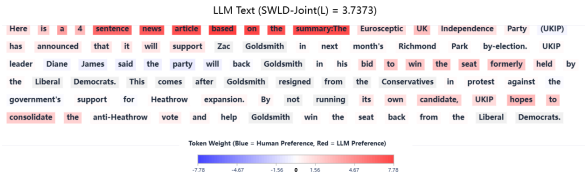


Figure 6: Token-level activation visualization for the LLM-generated news example. Red activations indicate AI-preferred POS templates.

For the LLM-generated example, Figure 6 shows strong positive activations around regular infinitival and nominal-extension structures. For instance, "in his bid to win the seat" activates templates such as `IN PRP$ NN TO VB DT` and `PRP$ NN TO VB DT NN`, while "to consolidate the anti-Heathrow vote and help" activates `TO VB DT JJ NN CC`. These patterns do not encode topic words; rather, they describe recurring POS-level organizations in which information is extended through a noun phrase followed by a `TO VB` complement, or through an infinitival clause expanded by a determiner–adjective–noun object and coordination. Such structures provide regular syntactic continuation paths for generated news-style text. Since these POS templates receive positive weights, likelihood shifts at their covered positions push the score toward the LLM side.

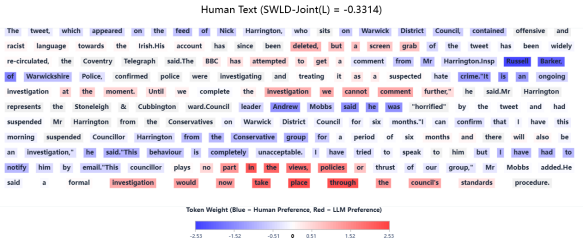


Figure 7: Token-level activation visualization for the human-written news example. Blue activations indicate human-preferred POS templates.

For the human-written example, Figure 7 shows negative activations around source attribution and quotation structures. For example, `NNP NNP , IN NNP NNP` corresponds to appositive affiliation phrases such as "Russell Barker, of Warwickshire Police," while `NNP NNP VBD PRP VBD "` captures a named speaker, a reporting verb, and the start of a quoted clause. These templates reflect local news-reporting conventions at the POS level: proper-name chains identify sources, appositive phrases specify affiliations, reporting verbs introduce attributed statements, and quotation boundaries mark transitions between narration and direct speech. Because these templates occur more often in human-written calibration texts, they receive negative weights and shift the score toward the human side.

We also analyze a human-written restaurant review that receives a high SWLD score and is misclassified as LLM-generated. Many sentences follow highly regular review-style templates, e.g., "The service and waitress was nice and attentive", "The egg drop soup was fine", and "My salmon was dry and flavorless". These fragments activate several AI-preferred POS templates, including `NN VBD JJ CC JJ ,`, `VBD JJ CC JJ ,` `RB`, and `DT NN NN NN VBD JJ`. This case shows that some human-written texts can be formulaic at the POS level. Their structure resembles the regular syntactic continuation patterns often observed in LLM-generated text. The detailed visualization is provided in Appendix D.4.

Overall, the case study shows that SWLD-Joint makes dual-model likelihood signals interpretable by locating whether likelihood shifts occur inside AI-preferred or human-preferred POS templates. The failure case also clarifies its boundary: syntax weighting may over-score human texts whose genre

naturally uses highly regular evaluative templates.

5 Conclusion

We presented SWLD-Joint, a syntax-pattern weighted joint detector for neural-training-free AIGT detection. Instead of uniformly aggregating token-level likelihood differences, SWLD-Joint constructs a POS n-gram template library, uses human-vs-AI frequency gaps to assign discriminative weights to token positions, and jointly calibrates the result with global POS-trajectory statistics. Experiments on three benchmarks show strong AUROC gains under domain shift, generator mismatch, and surface-level adversarial perturbations. Ablation and case analyses confirm that POS-template weighted likelihood aggregation is the dominant source of improvement, while the failure analysis also shows that highly formulaic human writing can activate AI-preferred syntactic templates. We also introduce HC3_Ext, which extends HC3 with three recent generators and a DeepSeek-humanized adversarial split. Future work will explore multilingual settings and richer syntactic, dependency, and discourse-level structures to better distinguish formulaic human writing from LLM-generated regularity.

Limitations

SWLD-Joint has several limitations. 1) It relies on English POS tagging and does not explicitly model translationese effects that may arise when LLM outputs are generated via upstream translation from different source languages. Its current implementation therefore cannot be directly transferred to multilingual settings without reliable language-specific POS taggers and calibration corpora, and calibration data collected from a single source-language distribution may not transfer well to outputs produced via other translation pipelines. 2) The method is neural-training-free but not uniformly calibration-free across its variants. SWLD-G is zero-shot and requires no calibration data, whereas SWLD-L uses labeled human/AI calibration data from the target subset to estimate POS-template statistics. Thus, SWLD-L should be viewed as a target-subset calibrated setting rather than a fully zero-shot or unsupervised detector. 3) The method mainly handles surface-level perturbations. Adversaries that explicitly control POS sequences or paraphrase text under syntactic constraints may reduce the effectiveness of syntax-pattern weighting.

Ethical Considerations

AIGT detection can support information integrity and academic honesty, but it also carries risks. False positives may unfairly flag human authors, especially non-native writers or users with atypical syntactic styles. Therefore, SWLD-Joint should be used as one signal among multiple sources of evidence, not as sole proof of AI authorship. Publishing detection methods may also motivate adaptive evasion; nevertheless, we believe transparent research on robust and interpretable detection is beneficial for the community. HC3_Ext is constructed from publicly available human responses and LLM-generated answers and does not include personally identifiable information.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 62372043).

References

Guangsheng Bao, Yanbin Zhao, Zhiyang Teng, Linyi Yang, and Yue Zhang. 2024. Fast-detectgpt: Efficient zero-shot detection of machine-generated text via conditional probability curvature. In *The Twelfth International Conference on Learning Representations (ICLR)*.

Zhihui Chen, Kai He, Yucheng Huang, Yunxiao Zhu, and Mengling Feng. 2025. DivScore: Zero-shot detection of LLM-generated text in specialized domains. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 19231–19253, Suzhou, China. Association for Computational Linguistics.

Liam Dugan, Alyssa Hwang, Filip Trhlik, Josh Magnus Ludan, Andrew Zhu, Hainiu Xu, Daphne Ippolito, and Chris Callison-Burch. 2024. Raid: A shared benchmark for robust evaluation of machine-generated text detectors.

Kathleen C Fraser, Hillary Dawkins, and Svetlana Kiritchenko. 2025. Detecting ai-generated text: Factors influencing detectability with current methods. *Journal of Artificial Intelligence Research*, 82:2233–2278.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander M Rush. 2019. Gltr: Statistical detection and visualization of generated text. In *Proceedings of the 57th annual meeting of the association for computational linguistics: system demonstrations*, pages 111–116.

Abhimanyu Hans, Avi Schwarzschild, Valeriia Cherepanova, Hamid Kazemi, Aniruddha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2024. Spotting llms with binoculars: Zero-shot detection of machine-generated text. In *Forty-first International Conference on Machine Learning (ICML)*.

Tatsunori B Hashimoto, Hugh Zhang, and Percy Liang. 2019. Unifying human and statistical evaluation for natural language generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1689–1701.

Runheng Liu, Heyan Huang, Xingchen Xiao, and Zhijing Wu. 2025. Zero-shot detection of llm-generated text via implicit reward model. In *39th Conference on Neural Information Processing Systems (NeurIPS 2025)*.

Yijian Lu, Aiwei Liu, Dianzhi Yu, Jingjing Li, and Irwin King. 2024. An entropy-based text watermarking detection method. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11724–11735.

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

Hoang-Quoc Nguyen-Son, Minh-Son Dao, and Koji Zettsu. 2026. SearchLLM: Detecting LLM paraphrased text by measuring the similarity with regeneration of the candidate source via search engine. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1755–1772, Rabat, Morocco. Association for Computational Linguistics.

Chantal Shaib, Yanai Elazar, Junyi Jessy Li, and Byron C Wallace. 2024. Detection and measurement of syntactic templates in generated text. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6416–6431.

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

Jinyan Su, Terry Zhuo, Di Wang, and Preslav Nakov. 2023. 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.

Brian Tufts, Xuandong Zhao, and Lei Li. 2025. A practical examination of AI-generated text detectors for large language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 4839–4856, Albuquerque, New Mexico. Association for Computational Linguistics.

Eduard Tulchinskii, Kristian Kuznetsov, Laida Kushnareva, Daniil Cherniavskii, Sergey Nikolenko, Evgeny Burnaev, Serguei Barannikov, and Irina Piontkovskaya. 2023. Intrinsic dimension estimation for robust detection of ai-generated texts. *Advances in Neural Information Processing Systems*, 36:39257–39276.

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

Junxi Wu, Jinpeng Wang, Zheng Liu, Bin Chen, Dongjian Hu, Hao Wu, and Shu-Tao Xia. 2025. MoSEs: Uncertainty-aware AI-generated text detection via mixture of stylistics experts with conditional thresholds. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 5786–5805, Suzhou, China. Association for Computational Linguistics.

Yuxi Xia, Kinga Stańczak, and Benjamin Roth. 2026. Explaining generalization of AI-generated text detectors through linguistic analysis. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6524–6546, Rabat, Morocco. Association for Computational Linguistics.

Yihuai Xu, Yongwei Wang, Yifei Bi, Huangsen Cao, Zhouhan Lin, Yu Zhao, and Fei Wu. 2024. Training-free llm-generated text detection by mining token probability sequences. *arXiv preprint arXiv:2410.06072*.

Xiaowei Zhu, Yubing Ren, Yanan Cao, Xixun Lin, Fang Fang, and Yangxi Li. 2025. Reliably bounding false positives: A zero-shot machine-generated text detection framework via multiscaled conformal prediction. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12298–12319, Vienna, Austria. Association for Computational Linguistics.

Appendix

A Construction Details and Motivation of HC3_Ext

HC3_Ext is constructed on top of the original HC3 benchmark. The motivation is twofold. First, HC3 is a representative AIGT benchmark, but its machine-written side mainly reflects an earlier ChatGPT-era generation setting. As LLM outputs from newer models become more fluent and stylistically diverse, a detector evaluated only on the original HC3 may overestimate its robustness to modern generators.

We keep the original HC3 questions and human-written answers, and regenerate the machine-written side using three recent LLMs: Claude Sonnet 4, DeepSeek-R1, and Qwen3-235B. This preserves the original question distribution and human reference side while updating the machine-written side to more recent generators. We further add a DeepSeek-humanized adversarial split to examine whether detectors remain stable when LLM answers are rewritten without changing their meaning or paragraph structure.

A.1 Direct LLM Answer Generation

For each HC3 record, we extract the original question field and query each target LLM independently. Each question is processed as an independent single-turn query, and previous samples are not used as dialogue history.

Due to API failures, empty generations, or invalid responses, the final number of valid generated answers is slightly smaller than the number of original HC3 records. Table 4 reports the final statistics after generation and filtering. The human side keeps the original HC3 human answers, where each record may contain multiple human-written responses. The machine side contains valid answers generated by the corresponding target model.

Table 4: Statistics of HC3_Ext after generation and filtering.

Generator	Records	Human Answers	Valid LLM Answers
Claude Sonnet 4	24,322	58,546	17,025
DeepSeek-R1	24,280	58,434	18,439
Qwen3-235B	24,293	58,467	17,017

For labeling, all original HC3 human-written answers are labeled as *human*, while the newly generated answers from Claude Sonnet 4, DeepSeek-R1, and Qwen3-235B are labeled as *LLM*. Samples with failed, empty, or invalid LLM generations are removed from the machine-written side.

A.2 DeepSeek-Humanized Adversarial Split

To further evaluate robustness against surface-level rewriting, we construct a DeepSeek-humanized adversarial split. Starting from LLM-labeled samples, we randomly select 40% of the LLM answers and rewrite them with the DeepSeek API. The rewriting model is `deepseek-V4`. We fix the random seed to 42 and use the following decoding parameters: temperature $= 0.9$, top-p $= 0.95$, and maximum output length $= 4096$ tokens. Failed rewriting requests are retried up to three times; if all attempts fail, the original text is retained and the failure is recorded in the metadata.

The system prompt used for the humanization rewrite is shown below.

```
You are a text rewriting expert. Your task is to rewrite
↪  an AI-generated response into a more human-like,
↪  naturally written explanation, primarily by
↪  replacing words with their synonyms and making minor
↪  phrasing adjustments. This is for adversarial dataset
↪  creation against AIGC detectors. Follow these rules:

1. Replace a significant portion of content words (nouns,
↪  verbs, adjectives, adverbs) with common, everyday
↪  synonyms. Aim for at least 6-8 synonym replacements
↪  per paragraph, but do not change any proper nouns,
↪  numbers, or domain-specific terms.

2. Keep the exact same information, logical flow,
↪  sentence count, and paragraph structure. Do NOT add
↪  new facts or remove existing ones.

3. You may occasionally adjust the sentence rhythm by
↪  splitting a very long sentence into two, or
↪  combining two very short ones, but the overall
↪  structure should remain unchanged.

4. You may insert 1-2 natural filler words or personal
↪  touches (e.g., 'actually', 'you know', 'honestly')
↪  in the entire text, but keep it minimal -- the main
↪  change should be synonym replacement.

5. The final text must remain grammatically correct,
↪  fluent, and natural. It should still read like a
↪  high-quality human-written explanation, not a messy
↪  chat log.

6. The total length must stay within +/-10% of the
↪  original.

7. Output must be only the rewritten English text. No
↪  explanations, no Chinese.
```

In the implementation, the user message is:

```
Please rewrite the following AI-generated answer:

{original_text}
```

where `{original_text}` denotes the LLM-generated answer to be rewritten. The implementation uses the same meaning as the original user message in the script, whose purpose is to ask the model to rewrite the following AI-generated answer.

This construction gives two complementary HC3_Ext evaluation settings. The direct-

generation setting tests detection on answers produced by recent LLMs under simple question prompting. The DeepSeek-humanized setting tests whether detectors remain robust when LLM answers are lightly rewritten to reduce obvious model-specific surface cues while preserving the original semantic content and overall structure.

B POS Template Statistics

Figure 8 provides dataset-level evidence for the syntactic preference gap used by SWLD-Joint. The upper panel shows that several high-frequency POS 6-grams appear much more often in AI-generated text than in human-written text. The lower panel further shows that this gap is directional: some templates are strongly AI-preferred, while a smaller set of templates is human-preferred. This supports the central design choice of SWLD-Joint: POS templates should not be treated as neutral linguistic units, but as class-preferential patterns that can identify syntactically informative token positions.

The conclusion is that the syntactic difference is not merely a frequency difference over isolated tags. Instead, AI-generated text repeatedly activates specific multi-token POS templates, especially regularized local structures. This motivates using log-ratio template weights to emphasize discriminative contexts during likelihood-difference aggregation.

C Full Per-Subset AUROC Results

This section reports complete per-subset AUROC results on DetectRL, RAID, and HC3_Ext. These tables supplement the dataset-level summaries in the main paper. We report AUROC because it is threshold-free and avoids test-set threshold tuning.

C.1 Full DetectRL Results

DetectRL evaluates three types of distribution shift: domain shift, generator mismatch, and adversarial perturbation. Across all three scenarios, SWLD-G achieves the best average AUROC, showing that global POS-template statistics provide a transferable syntactic signal.

On Multi-Domain, SWLD-G achieves 96.41% average AUROC, outperforming the best non-SWLD baseline, IRM, by 18.94 points. The gain is consistent across all four domains. This suggests that pooled global templates capture transferable human-vs-AI syntactic preferences rather than overfitting to a single domain.

Table 5: DetectRL Multi-Domain AUROC (%). **Bold**: best; underline: second best.

Method	ArXiv	Writing	XSum	Yelp	Avg.
Likelihood	67.13	81.85	60.09	86.24	73.83
Entropy	63.19	85.24	61.19	86.33	73.99
Log-Rank	68.76	82.05	62.03	87.00	74.96
LRR	72.64	80.82	66.42	87.56	76.86
LastDE	63.94	60.16	55.69	65.14	61.23
IRM	86.95	64.85	94.23	63.85	77.47
Binoculars	66.90	79.35	60.40	84.88	72.88
F-DetectGPT	71.67	64.27	78.33	72.21	71.62
LastDE++	62.82	44.59	74.50	57.27	59.80
SWLD-L	87.41	96.46	94.30	93.78	92.99
SWLD-G	**92.28**	**98.70**	**96.59**	**98.08**	**96.41**

Table 6: DetectRL Multi-LLM AUROC (%). **Bold**: best; underline: second best.

Method	ChatGPT	Claude	PaLM	Llama-2	Avg.
Likelihood	78.27	50.58	79.01	85.47	73.33
Entropy	80.86	59.36	73.81	82.78	74.20
Log-Rank	78.63	52.30	80.25	86.72	74.47
LRR	77.64	58.20	81.60	88.34	76.44
LastDE	56.34	46.04	66.82	70.18	59.84
IRM	64.53	68.06	83.38	78.39	73.59
Binoculars	74.70	44.93	80.80	86.30	71.68
F-DetectGPT	63.47	45.25	84.00	85.31	69.51
LastDE++	52.18	40.79	74.26	71.87	59.78
SWLD-L	96.32	92.46	84.09	90.27	90.78
SWLD-G	**99.08**	**97.93**	**90.24**	**96.24**	**95.87**

On Multi-LLM, SWLD-G reaches 95.87% average AUROC, outperforming the best non-SWLD baseline, LRR, by 19.43 points. The Claude-instant subset is the most challenging case for probability-only baselines: several methods fall close to random performance, while SWLD-G reaches 97.93%. This indicates that syntax-pattern weighting remains effective when the detector model and the generation model are mismatched.

Table 7: DetectRL Multi-Attack AUROC (%). **Bold**: best; underline: second best.

Method	Data Mix.	Para.	Pert.	Prompt	Avg.
Likelihood	69.66	68.87	59.56	90.13	72.06
Entropy	69.09	65.68	65.02	90.77	72.64
Log-Rank	70.90	69.77	61.54	90.31	73.13
LRR	72.74	70.95	66.59	88.57	74.71
LastDE	61.67	69.56	43.31	86.06	65.15
IRM	77.71	81.78	65.58	73.97	74.76
Binoculars	67.88	70.25	54.86	88.04	70.26
F-DetectGPT	67.07	76.30	50.82	78.90	68.27
LastDE++	59.18	72.77	45.47	70.63	62.01
SWLD-L	–	93.40	94.55	90.66	92.87[†]
SWLD-G	**91.02**	**95.24**	**95.51**	**94.99**	**94.19**

[†] Average over three subsets; Data Mixing has no local template because it merges multiple subsets.

On Multi-Attack, SWLD-G achieves 94.19% average AUROC, outperforming the best non-SWLD baseline, IRM, by 19.43 points. The Perturbation subset is especially revealing: LastDE drops to 43.31% and Fast-DetectGPT to 50.82%, whereas SWLD-G remains at 95.51%. This suggests that

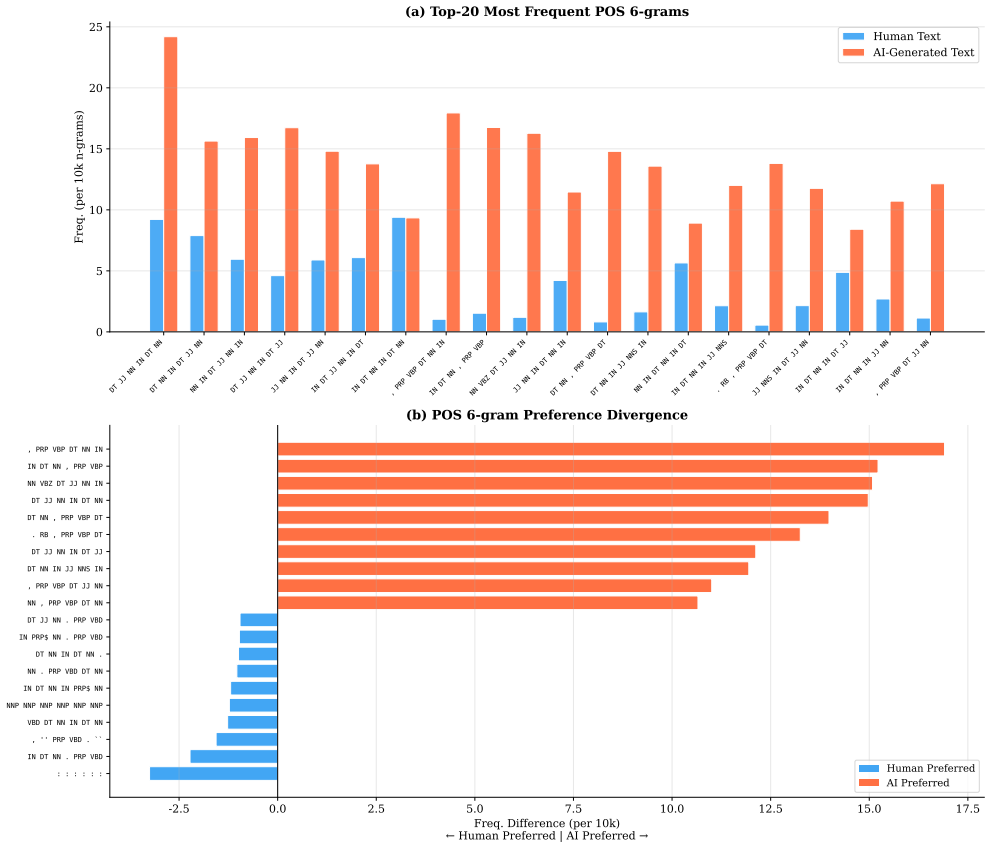


Figure 8: POS 6-gram statistics on DetectRL. The top panel shows the top-20 most frequent POS 6-grams in human-written and AI-generated texts. The bottom panel shows POS 6-gram preference divergence, where positive values indicate AI-preferred templates and negative values indicate human-preferred templates.

POS-template signals are less sensitive to surface-level token corruption than raw probability statistics.

Overall, the DetectRL results show that SWLD-G is the strongest variant on this benchmark. Global templates are effective because DetectRL contains diverse domains, generators, and attacks, allowing pooled calibration statistics to capture broadly transferable syntactic preferences.

C.2 Full RAID Results

On RAID, SWLD-L achieves the best average AUROC of 83.02%, followed by SWLD-G with 80.05%. This differs from DetectRL, where the global variant is stronger. The reason is that RAID contains many generator-specific subsets, and local templates can better capture generator-specific syntactic preferences when labeled target-subset calibration data are available.

The per-generator results also show that syntax-pattern weighting is not uniformly dominant in every subset. Fast-DetectGPT and LastDE++ remain strong on GPT-3 and GPT-2, where probability-space anomalies are easier to capture. In contrast, SWLD-L is strongest on ChatGPT, GPT-4, Llama-Chat, Mistral, Mistral-Chat, MPT, and MPT-Chat. This indicates that syntax-based calibration is especially useful for newer or instruction/chat-oriented generators, while some older generators can still be detected effectively by probability-only signals.

C.3 Full HC3_Ext Results

HC3_Ext is a high-separability setting. Entropy achieves the best average AUROC of 98.04%, suggesting that this QA-style dataset contains a strong probability-space gap between human and LLM answers. Nevertheless, SWLD-G remains highly competitive with 97.43% average AUROC and obtains the best score on DeepSeek-R1. In contrast, Fast-DetectGPT and LastDE++ collapse on all three newer generators, with average AUROC scores of 48.96% and 41.91%, respectively. This shows that syntax-pattern weighting is not always the top signal when probability statistics already saturate, but it is substantially more stable than several probability-curvature or dual-model extensions under modern generator shifts.

Table 8: RAID per-generator AUROC (%). **Bold**: best; underline: second best per row.

Generator	Like.	Ent.	LogR.	LRR	LastDE	IRM	Bino.	F-DGPT	LDE++	SWLD-G	SWLD-L
ChatGPT	83.28	81.08	83.88	86.64	82.45	61.85	84.98	79.22	68.84	88.25	**91.66**
GPT-4	73.23	71.57	73.51	74.87	69.81	62.09	73.52	64.12	57.08	86.66	**89.54**
GPT-3	79.64	67.82	80.77	83.94	85.31	85.00	85.56	**87.84**	81.54	71.96	77.91
GPT-2	63.81	58.86	65.63	70.98	81.39	68.28	65.01	74.38	**81.70**	73.26	77.90
Cohere	67.99	58.31	68.45	69.83	72.99	70.10	73.22	**77.61**	70.56	72.93	73.59
Cohere-Chat	73.61	66.42	74.37	76.96	78.76	77.97	78.21	**82.85**	75.89	78.43	79.14
Llama-Chat	83.54	82.28	84.41	88.72	81.04	61.98	84.89	79.31	68.34	88.12	**90.40**
Mistral	63.97	60.97	65.70	71.35	76.48	59.35	65.41	68.56	72.89	74.40	**77.79**
Mistral-Chat	78.95	76.83	80.15	84.97	81.97	71.71	80.99	79.19	72.45	85.97	**87.45**
MPT	52.37	51.83	55.53	65.04	70.94	50.09	54.26	67.57	77.18	76.72	**82.38**
MPT-Chat	66.37	65.38	67.78	72.64	72.79	66.99	68.68	72.81	69.95	83.83	**85.44**
Avg.	71.52	67.40	72.74	76.90	77.63	66.85	74.07	75.77	72.40	80.05	**83.02**

Table 9: HC3_Ext per-generator AUROC (%). **Bold**: best; underline: second best.

Generator	Like.	Ent.	LogR.	LRR	LastDE	IRM	Bino.	F-DGPT	LDE++	SWLD-G	SWLD-L
Claude Sonnet 4	96.29	**98.11**	96.34	94.24	63.18	75.11	92.83	49.68	42.96	96.42	96.49
DeepSeek-R1	97.14	98.65	97.09	94.96	67.27	88.54	94.71	42.78	36.48	**98.67**	98.46
Qwen3-235B	94.80	**97.36**	94.79	92.42	56.19	71.69	91.96	54.42	46.28	97.19	96.85
Avg.	96.08	**98.04**	96.08	93.88	62.21	78.45	93.16	48.96	41.91	97.43	97.27

D Additional Visual Analyses

D.1 Template Weight Visualization

Figure 9 displays the top-15 AI-preferred and human-preferred POS n-gram templates for two representative subsets. Unlike Appendix B, which focuses on corpus-level frequency divergence, this analysis focuses on templates with the strongest class preference after log-ratio weighting.

The template-weight visualizations show two consistent patterns. AI-preferred templates are dominated by regular nominal, prepositional, and possessive structures, reflecting more repeated local syntactic organization. In contrast, human-preferred templates contain more proper nouns, numbers, punctuation-heavy segments, parenthetical structures, and event-reporting patterns. This explains why SWLD-Joint is interpretable: high or low scores can be traced to concrete POS-template matches rather than opaque neural representations.

D.2 Score Distributions

Figure 10 compares Simple ΔLL and SWLD-Joint on two representative subsets. The plot title for the DeepSeek-R1 subset uses the original internal dataset name "my_llm", which corresponds to HC3_Ext in this paper.

The distributions show that the improvement of SWLD-Joint is not a threshold-specific artifact. In HC3_Ext / DeepSeek-R1, Simple ΔLL already provides partial separation, but SWLD-Joint makes the human and LLM clusters more clearly separated. In DetectRL / Claude-instant, Simple ΔLL produces heavy overlap, whereas SWLD-Joint separates the two distributions into distinct score regions. This explains the large AUROC gain on the Claude-instant subset and supports the claim that syntax-pattern weighting changes the overall ranking structure.

D.3 ROC Curves

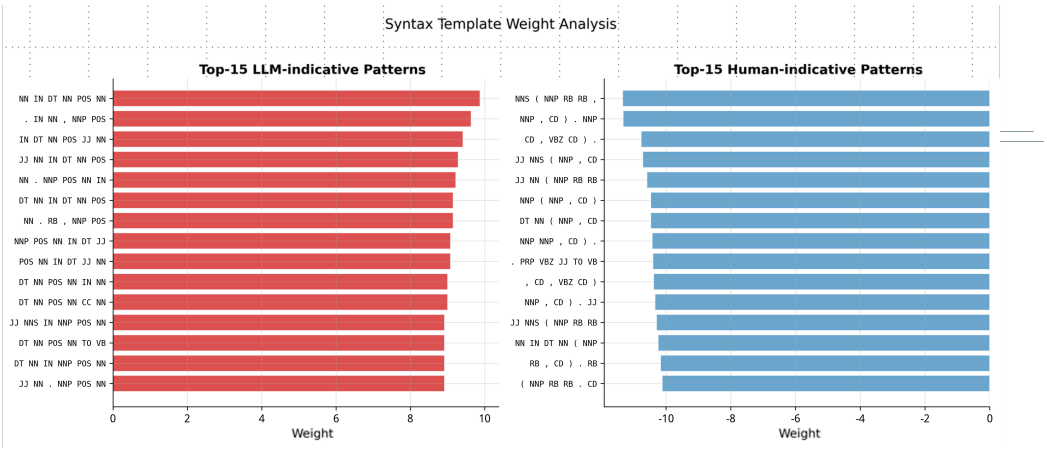
Figure 11 reports ROC curves on two challenging subsets: RAID / MPT and DetectRL / Claude-instant.

The ROC curves provide a threshold-free view of the same trend. On RAID / MPT, SWLD-L achieves the strongest curve, consistent with the per-generator result in Table 8. On DetectRL / Claude-instant, SWLD-G dominates the baselines across a broad range of false-positive rates, consistent with its 97.93% AUROC in Table 6. These curves confirm that the improvement is not restricted to a narrow operating threshold.

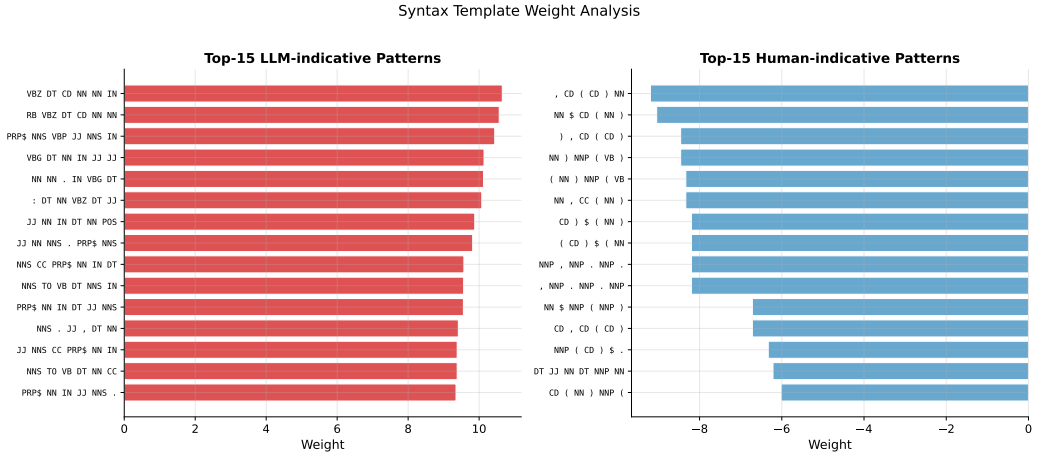
D.4 Failure Case Visualization

Figure 12 shows a human-written restaurant review that receives a high SWLD score and is therefore misclassified as LLM-generated. This example complements the successful cases in Section 4.7 by showing when syntax-pattern weighting can fail.

The review contains clear human-like cues, including personal experience, informal fragments, minor grammatical errors, and concrete details about dishes and service. For example, the writer mentions the daughter's sesame chicken, dry salmon, a gift certificate, and a comparison with Panda Express. These details make the text pragmatically personal and experience-based.

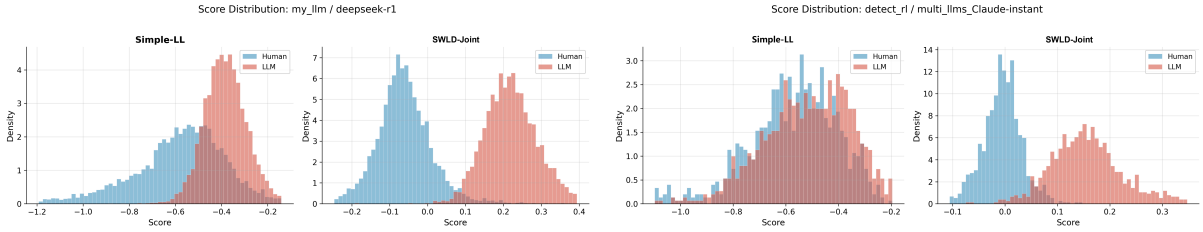


(a) DetectRL / Claude-instant.



(b) DetectRL Multi-Domain / ArXiv.

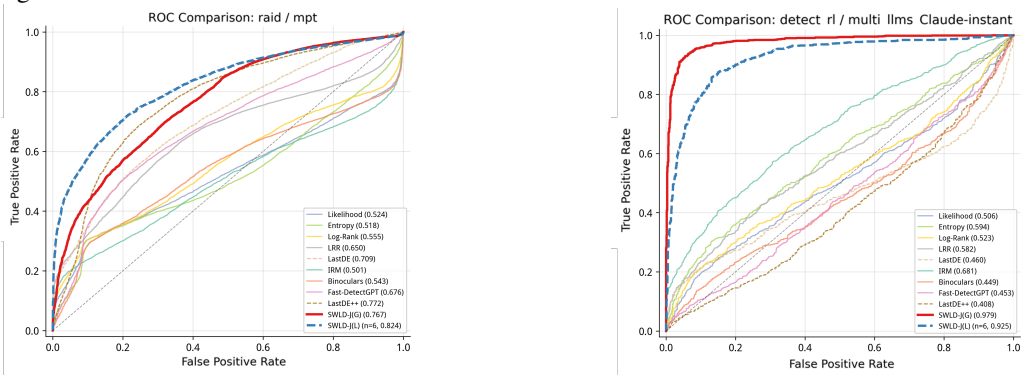
Figure 9: Top-15 AI-preferred and human-preferred POS templates for two representative subsets.



(a) HC3_Ext / DeepSeek-R1.

(b) DetectRL / Claude-instant.

Figure 10: Score distributions on representative subsets. Syntax-pattern weighting produces clearer separation than the unweighted likelihood-difference score.



(a) RAID / MPT.

(b) DetectRL / Claude-instant.

Figure 11: ROC curves on two challenging subsets.

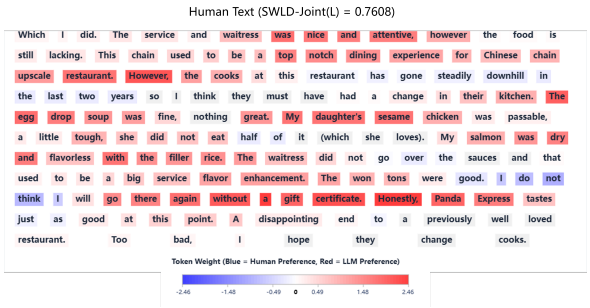


Figure 12: Failure case visualization for a human-written restaurant review. Although the text is written by a human, many local evaluative structures activate AI-preferred POS templates, causing the score to shift toward the LLM side.

However, the POS-template signal focuses on local syntactic organization rather than factual specificity. Several sentences follow repeated consumer-review evaluation patterns, such as a noun phrase followed by a copular verb and adjectival assessment. Examples include “The service and waitress was nice and attentive”, “The egg drop soup was fine”, and “My salmon was dry and flavorless”. These fragments activate positive templates such as `NN VBD JJ CC JJ ,`, `VBD JJ CC JJ ,` `RB`, and `DT NN NN NN VBD JJ`. Since these templates are more frequent in LLM-generated calibration texts, their positive weights push the document score toward the LLM side.

This failure case reveals a genre-level limitation. Human-written restaurant reviews can be highly formulaic because reviewers often list multiple items and attach short evaluative judgments to each one. Such item-by-item organization resembles the regularized syntactic continuation patterns of LLM-generated answers. Although the sample contains informal and human-specific cues, these signals are not fully captured by POS n-gram templates. As a result, the accumulated positive template evidence dominates the final score.