WHEN TO FLAG, WHERE TO LOCALIZE: RISK-CONTROLLED DETECTION WITH AUTHORSHIP LOCALIZATION IN MIXED HUMAN–LLM TEXT

Anonymous authors
Paper under double-blind review

ABSTRACT

Documents containing both human-written and LLM-generated text require both deciding whether LLM-generated text exists and identifying where it occurs. Localization methods can in principle address both by treating any predicted LLM segment as evidence that the document contains generated text, but sentence-level errors can then produce false alarms on fully human-written documents. We therefore propose CoMa-LD, a training-free framework for sequential LLM detection and post-alarm localization. CoMa-LD introduces a two-state Human/LLM Markov model into a conformal test martingale: summing evidence over candidate authorship paths determines when to flag, while selecting the highest-weight path identifies where the generated text occurs. The resulting detector provides statistical guarantees that control the probability of ever falsely flagging a fully human-written document. Across multiple domains and authorship transition patterns, CoMa-LD improves both detection and localization over the evaluated baselines while maintaining low false-alarm rates. Our code is available at `https://anonymous.4open.science/r/CoMa-LD-B486/`.

1 INTRODUCTION

Large language models (LLMs) are increasingly used not only to generate complete documents, but also to revise, expand, or rewrite selected parts of human-written text. As a result, many documents may contain interleaved human- and LLM-generated content (Zhang et al., 2024; Zeng et al., 2024). For such documents, it is not enough to decide whether LLM-generated text is present; we also want to know where it occurs. However, applying localization directly to every document can falsely flag a fully human-written document when a few sentences are incorrectly labeled as LLM-generated. In applications such as academic integrity screening and authorship assessment, such false alarms can lead to unnecessary scrutiny or incorrect conclusions about how a document was produced. This motivates a question: can we first decide when to flag a document while controlling false alarms, and then localize the LLM-generated text?

A straightforward solution is to apply a document-level detector first and invoke a separate localizer only after an alarm. Existing work provides methods for both stages: document-level detectors decide whether to flag a document (Mitchell et al., 2023; Hans et al., 2024; Verma et al., 2024), with some providing false-alarm control (Zhu et al., 2025; Chen & Wang, 2025), while localization methods infer sentence-level authorship labels or authorship change points (Lei et al., 2025; Li et al., 2026; Lu et al., 2026). However, a naive cascade delegates false-alarm control to the document-level detector. If the detector fails to flag a mixed-authorship document, the downstream localizer is never invoked, resulting in a localization failure. This calls for a unified sequential formulation in which detection accounts for possible authorship transitions, so that the evidence used to decide when to flag can also support where to localize.

Conformal test martingales (CTM), originally developed for sequential detection of distributional changes, provide a statistical starting point for addressing this requirement (Vovk et al., 2021; 2022; Prinster et al., 2025). Since human-written and LLM-generated sentences follow different distributions (Bao et al., 2024; Hans et al., 2024), we evaluate a direct application of CTM to sequential detection of LLM-generated text. Specifically, each sentence is assigned a detection score, which is

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

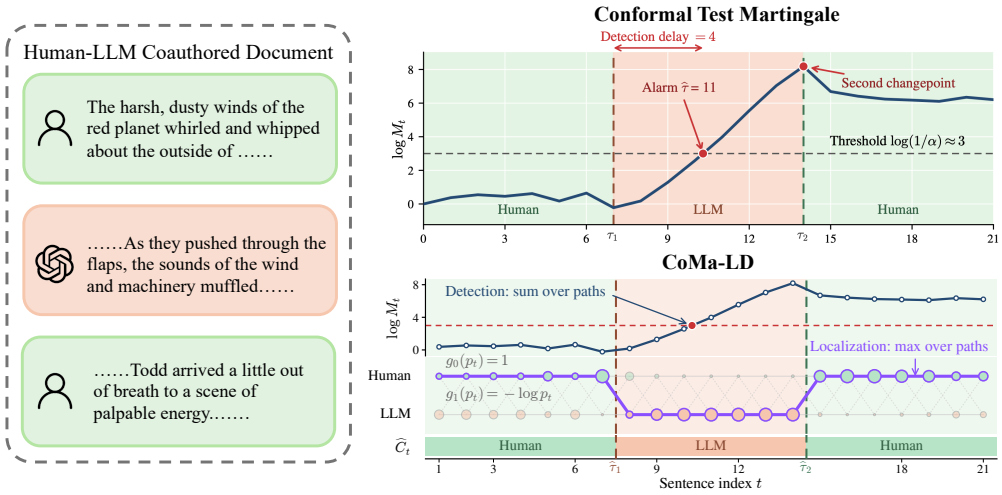


Figure 1: **Detection and localization in mixed Human-LLM text**. **Left:** A document containing alternating Human-written (green) and LLM-generated (orange) segments. **Right, top**: A standard conformal test martingale accumulates martingale evidence M_t against the human reference distribution. Its wealth crosses the detection threshold only after the first Human-to-LLM transition and remains above the threshold after the document returns to Human, so the alarm time does not recover the segment boundaries. **Right, bottom**: CoMa-LD assigns weights to candidate Human/LLM paths. Summing the path weights gives the martingale wealth M_t used for detection, while Viterbi decoding selects the highest-weight path after an alarm to recover the sentence-level authorship labels.

converted into a conformal p-value using human-written calibration data, and the resulting p-values update the martingale wealth through a betting function. As illustrated in Figure 1 (right, top), an alarm is raised when this wealth crosses a threshold. Under joint exchangeability of the calibration and test data, setting the threshold to $1/\alpha$ bounds the probability of ever falsely flagging an all-human document by α. However, the threshold crossing does not identify an authorship switch: the first alarm lags behind the true Human-to-LLM transition, while a subsequent switch back to Human may leave the accumulated wealth high without producing a new threshold crossing. Thus, this direct CTM construction addresses *when to flag*, but does not determine *where the generated segments are located*.

To go beyond *when to flag*, we propose **CoMa-LD** (**Co**nformal **Ma**rtingales for **L**LM **D**etection and Localization) by formulating a two-state Human/LLM Markov model with state-specific betting functions into the standard CTM. As shown in Figure 1 (right, bottom), the transition probabilities and state-specific betting functions jointly define weights over candidate Human/LLM paths. Summing the weights of all candidate paths gives the martingale wealth used for detection, whereas selecting the highest-weight path after an alarm yields the localized authorship sequence. The detection preserves document-level false-alarm control since the resulting martingale wealth is still a valid test martingale. Our contributions are:

- We propose CoMa-LD, a training-free conformal framework that formulates sequential detection and post-alarm sentence localization over Human/LLM paths, so that both tasks are derived from the same weighted path model.
- We establish document-level false-alarm control for sequential detection. CoMa-LD preserves the standard CTM guarantee under joint exchangeability of the human calibration and test sentences, while calibration on held-out human documents provides finite-sample control under exchangeability across documents and allows dependence among sentences.
- We conduct extensive experiments across different domains and human–LLM transitions. CoMa-LD achieves higher detection rates and more accurate localization than the evaluated baselines, while maintaining low empirical false-alarm rates on fully human-written documents.

2 Problem Setup and Background

2.1 Problem Setup

We consider joint document-level detection and sentence-level localization of LLM-generated text, with explicit control of document-level false alarms. Let $X_{1:T} = (X_1, \ldots, X_T)$ denote a document of T sentences. Each sentence X_t has an unobserved label $C_t \in \{0, 1\}$, where $C_t = 0$ indicates human-written text and $C_t = 1$ indicates LLM-generated text. We formulate document-level detection as the hypothesis test

$$H_0 : \ C_t = 0 \text{ for all } t, \qquad H_1 : \ \exists\, t \text{ such that } C_t = 1.$$

Under H_1, LLM-generated text may appear in one or more segments anywhere in the document.

Sequential detection. Detection proceeds sentence by sentence, using only the information available at each step. Let $\widehat{\tau} \in \{1, \ldots, T+1\}$ denote the time of the first alarm, with $\widehat{\tau} = T + 1$ if no alarm is raised within the document. Given a false-alarm budget $\alpha \in (0, 1)$, we seek to detect documents satisfying H_1 subject to the document-level false-alarm constraint $\mathbb{P}_{H_0}\big(\widehat{\tau} \le T\big) \le \alpha$.

Post-alarm localization. For a flagged document, we use the complete document $X_{1:T}$ to estimate the sentence labels $C_{1:T}$. We denote the predicted Human/LLM label sequence by

$$\widehat{C}_{1:T} = (\widehat{C}_1, \ldots, \widehat{C}_T) \in \{0, 1\}^T. \tag{1}$$

2.2 Conformal Test Martingales

Conformal p-values. A sequence $(Z_1, \ldots, Z_{n+1})$ is *exchangeable* if its joint distribution is invariant under permutations. Let $\mathcal{A}$ be a nonconformity measure, where larger values indicate greater nonconformity. Given reference scores $s_i = \mathcal{A}(z_i)$, $i = 1, \ldots, n$, and a new score $s = \mathcal{A}(z_{n+1})$, the smoothed conformal p-value is

$$p = \frac{|\{i : s_i > s\}| + U\,(1 + |\{i : s_i = s\}|)}{n + 1}, \tag{2}$$

where $U \sim \mathrm{Unif}(0, 1)$ is independent and randomly breaks ties.

Proposition 1 (Conformal validity)**.** *Under exchangeability, sequential application of equation 2 to the growing reference sequence, using independent random variables U_t, yields*

$$p_t \mid \mathcal{F}_{t-1} \sim \mathrm{Unif}(0, 1), \qquad \mathcal{F}_t = \sigma(p_1, \ldots, p_t),$$

where $(\mathcal{F}_t)_{t \ge 0}$ is the filtration generated by the sequential conformal p-values.

Small p-values indicate unusually large scores relative to the current reference.

Conformal test martingales. A betting function $g_t : [0, 1] \to [0, \infty)$ is chosen using only $\mathcal{F}_{t-1}$ and satisfies $\int_0^1 g_t(p)\, dp = 1$. Starting from $M_0 = 1$, the wealth process

$$M_t = M_{t-1} g_t(p_t) = \prod_{u=1}^{t} g_u(p_u) \tag{3}$$

is a nonnegative martingale under H_0, since $\mathbb{E}_{H_0}[g_t(p_t) \mid \mathcal{F}_{t-1}] = 1$. A betting function assigning greater wealth to small p-values allows M_t to accumulate evidence after a distributional change.

Proposition 2 (Ville's inequality)**.** *For any nonnegative test martingale with $M_0 = 1$ and any $\lambda \ge 1$,*

$$\mathbb{P}_{H_0}\left(\sup_{t \ge 1} M_t \ge \lambda\right) \le \frac{1}{\lambda}. \tag{4}$$

Consequently, the alarm time $\widehat{\tau} = \inf\big\{t \in \{1, \ldots, T\} : M_t \ge 1/\alpha\big\}$ satisfies $\mathbb{P}_{H_0}(\widehat{\tau} \le T) \le \alpha$, providing anytime-valid false-alarm control.

A standard CTM provides a document-level rejection rule for H_0, but does not estimate the latent label sequence $C_{1:T}$. Section 3 addresses this limitation.

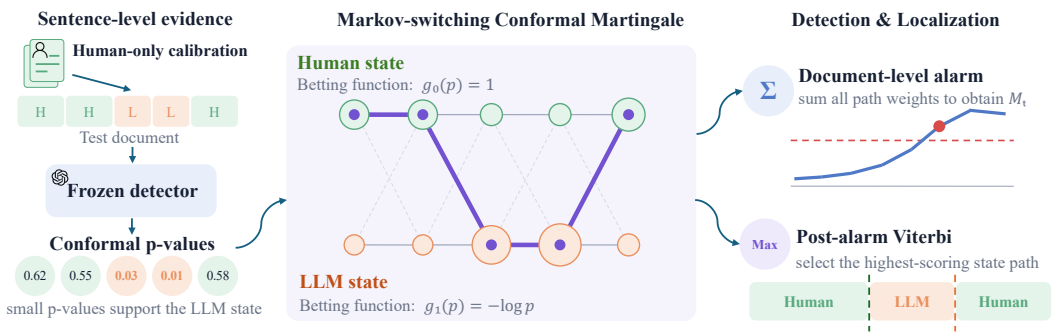


Figure 2: **Overview of CoMa-LD. Left**: A frozen detector computes a score for each test sentence. The score is converted into a conformal p-value using a reference initialized with human calibration scores and updated with previously observed test scores. Small p-values support the LLM state. **Middle**: A two-state Markov-switching conformal martingale represents repeated transitions between Human and LLM text using state-specific betting functions. **Right**: Forward aggregation sums over weighted state paths to obtain the detection statistic. After an alarm, Viterbi decoding maximizes over the same paths using the complete document to predict the label of each sentence.

3 METHOD

Figure 2 summarizes CoMa-LD, our training-free method for sequential LLM-text detection and change-point localization. A fixed sentence-level scoring function converts each test sentence into a detection score, which is calibrated against a reference containing human calibration scores and previously observed test scores to obtain a conformal p-value. CoMa-LD then evaluates a two-state Human/LLM Markov model in two ways: the forward update aggregates evidence over all state paths and raises a document-level alarm using either the Ville or document-calibrated boundary, whereas post-alarm Viterbi decoding selects the highest-scoring path to estimate the label sequence.

3.1 LLM-TEXT DETECTION WITH A MARKOV-SWITCHING CONFORMAL MARTINGALE

Given a document $X = (X_1, \ldots, X_T)$, CoMa-LD processes its sentences sequentially. Let $\mathcal{A}$ be a fixed sentence-level nonconformity function, such as the Fast-DetectGPT statistic (Bao et al., 2024) or negative mean NLL (Gehrmann et al., 2019), with larger scores indicating stronger evidence of LLM generation. We compute $s_t = \mathcal{A}(X_t)$ and use sequential conformal calibration to assess whether it is unusually large relative to the current reference.

Sequential conformal evidence. Let $\mathcal{C} = \{X_1^{\text{cal}}, \ldots, X_n^{\text{cal}}\}$ be a human-only sentence calibration set, with scores $\{s_1^{\text{cal}}, \ldots, s_n^{\text{cal}}\}$. These scores initialize the reference set $\mathcal{R}_0 = \{s_1^{\text{cal}}, \ldots, s_n^{\text{cal}}\}$. At time t, the reference set additionally contains all previously observed test scores, so that $\mathcal{R}_{t-1} = \{s_1^{\text{cal}}, \ldots, s_n^{\text{cal}}, s_1, \ldots, s_{t-1}\}$. As in the standard CTM construction in equation 2, we convert the score s_t into a smoothed conformal p-value p_t:

$$p_t = \frac{|\{r \in \mathcal{R}_{t-1} : r > s_t\}| + U_t\,(1 + |\{r \in \mathcal{R}_{t-1} : r = s_t\}|)}{|\mathcal{R}_{t-1}| + 1}, \qquad U_t \sim \text{Unif}(0,1). \tag{5}$$

After computing p_t, we add s_t to the reference set. A small p_t indicates that the current sentence has an unusually large score relative to the current reference and is used as evidence for the LLM state. Under joint exchangeability of the calibration and test scores, this prequential construction yields valid sequential conformal p-values.

Markov-switching conformal martingale. To model repeated transitions between human-written and LLM-generated text, we associate each sentence with a latent state $C_t \in \{0, 1\}$, where 0 denotes Human and 1 denotes LLM. We assign Markov weights to candidate Human/LLM label sequences

$$P = \begin{pmatrix} 1 - \rho_{01} & \rho_{01} \\ \rho_{10} & 1 - \rho_{10} \end{pmatrix}. \tag{6}$$

Here, $\rho_{01}, \rho_{10} \in [0,1]$ are fixed transition probabilities from Human to LLM and from LLM to Human, respectively. The resulting matrix P is row-stochastic.

The two states use different betting functions $g_0(p_t) = 1$ and $g_1(p_t) = -\log p_t$. The Human state places a neutral bet, whereas the LLM state gains evidence from small p-values. Both functions are nonnegative and integrate to one over $[0,1]$, which ensures fair betting under uniform p-values.

Let W_t^c denote the total weight of all state paths ending in state c at time t. The fixed initial weights satisfy $\pi_c \geq 0$ and $\pi_0 + \pi_1 = 1$. Unlike the standard CTM in equation 3, CoMa-LD tracks wealth separately for the Human and LLM states. We introduce an auxiliary initial state C_0, which does not correspond to a sentence. Starting from $W_0^c = \pi_c$, we update the state weights and aggregate evidence as

$$\begin{aligned} \widetilde{W}_t^c &= \sum_{c' \in \{0,1\}} W_{t-1}^{c'} P_{c'c}, \qquad W_t^c = \widetilde{W}_t^c g_c(p_t), \\ M_t &= W_t^0 + W_t^1 = \sum_{c_{0:t} \in \{0,1\}^{t+1}} \pi_{c_0} \prod_{j=1}^{t} \left[P_{c_{j-1},c_j} \cdot g_{c_j}(p_j) \right]. \end{aligned} \tag{7}$$

Thus, M_t aggregates evidence over all possible Human/LLM state paths $c_{0:t}$ without enumerating them explicitly. Under joint exchangeability of the calibration and test scores under H_0, $(M_t)_{t \geq 0}$ is a valid test martingale and the probability of crossing this boundary at any time is at most α.

3.2 LLM-TEXT LOCALIZATION WITH VITERBI DECODING

The forward martingale determines whether a document contains sufficient evidence of LLM-generated text, but its threshold-crossing time is not treated as an estimate of a source boundary. If a document does not trigger an alarm, CoMa-LD returns the all-Human label sequence. For an alerted document, we apply Viterbi decoding to the complete sequence of conformal p-values once the full document is available.

The localizer reuses the state-specific betting values $g_c(p_t)$ computed during detection and applies Viterbi decoding with the same transition matrix P. For a candidate state path $c_{0:T}$, CoMa-LD selects the path with the largest contribution to the aggregate evidence M_T, equivalently maximizing its log weight:

$$\begin{aligned} \widehat{C}_{0:T} &= \arg \max_{c_{0:T} \in \{0,1\}^{T+1}} \left\{ \pi_{c_0} \prod_{t=1}^{T} \left[P_{c_{t-1},c_t} \cdot g_{c_t}(p_t) \right] \right\} \\ &= \arg \max_{c_{0:T} \in \{0,1\}^{T+1}} \left\{ \log \pi_{c_0} + \sum_{t=1}^{T} \left[\log P_{c_{t-1},c_t} + \log g_{c_t}(p_t) \right] \right\}. \end{aligned} \tag{8}$$

The localization output consists of the sentence labels $\widehat{C}_{1:T}$ in equation 1. The term $\log g_{c_t}(p_t)$ measures how well sentence t supports state c_t, whereas $\log P_{c_{t-1},c_t}$ accounts for consistency between neighboring labels. In particular, the Human state contributes $\log g_0(p_t) = 0$, while a small p_t increases the score of the LLM state through $\log g_1(p_t) = \log(-\log p_t)$. The transition probabilities determine the relative weights of retaining the current label and switching to the other label.

Enumerating state sequences in equation 8 would require exponential time. Viterbi decoding records the highest score of any partial path ending in each state. With $\delta_0(c) = \log \pi_c$, the recursion is

$$\delta_t(c) = \log g_c(p_t) + \max_{c' \in \{0,1\}} \{ \delta_{t-1}(c') + \log P_{c'c} \}. \tag{9}$$

At each step, we store the maximizing predecessor and backtrack from the highest-scoring terminal state to recover $\widehat{C}_{1:T}$. The complete inference procedure is summarized in Algorithm 1 in Appendix B. The computational complexity analysis is provided in Appendix C.

3.3 VALIDITY AND DOCUMENT-LEVEL CALIBRATION

CoMa-LD provides two complementary false-alarm guarantees. Under joint exchangeability of the calibration and test scores under H_0, its aggregate evidence process is a valid test martingale and

therefore admits an anytime-valid Ville boundary. Since exact exchangeability may not hold among sentences from a natural document, we additionally calibrate the running maximum of the process using held-out human documents. This calibration requires exchangeability across documents but does not require exchangeability among sentences within a document.

Ville validity. Let $\mathcal{F}_t = \sigma(p_1, \ldots, p_t)$ denote the information generated by the sequential conformal p-values. Under H_0, suppose that the calibration and test scores are jointly exchangeable. The smoothed prequential construction in equation 5 then satisfies

$$p_t \mid \mathcal{F}_{t-1} \sim \mathrm{Unif}(0, 1).$$

Since both betting functions integrate to one, it follows that $\mathbb{E}[g_c(p_t) \mid \mathcal{F}_{t-1}] = 1$ for $c \in \{0, 1\}$.

Proposition 3 (Martingale validity)**.** *Suppose that the sequential conformal p-values are conditionally uniform under H_0 with respect to $\mathcal{F}_{t-1}$ for $t = 1, \ldots, T$. Let P be a fixed row-stochastic transition matrix, and let the fixed initial weights satisfy $\pi_c \geq 0$ and $\pi_0 + \pi_1 = 1$. Then the aggregate process $(M_t)_{t=0}^T$ defined in equation 7 is a nonnegative test martingale with $M_0 = 1$.*

Proof: Refer to Appendix D.1 for a proof. □

Applying Ville's inequality in equation 4 to Proposition 3 yields

$$\mathbb{P}_{H_0}\left(\max_{1 \leq t \leq T} M_t \geq \frac{1}{\alpha}\right) \leq \alpha. \tag{10}$$

Equivalently, $\mathbb{P}_{H_0}(\widehat{\tau}_{\mathrm{Ville}} \leq T) \leq \alpha$. This provides anytime-valid false-alarm control across all sentence positions within the document under the stated assumptions.

Document-level calibration. Sentence scores within a natural document can remain dependent even when the document is human-written. For false-alarm control without assuming joint exchangeability, we use a separate calibration set of fully human documents, $\mathcal{D}_{\mathrm{doc}}^{\mathrm{cal}} = \{D_1, \ldots, D_N\}$. Before calibration, we fix the sentence-level scoring function, the transition matrix P, the initial weights (π_0, π_1), the betting functions g_0, g_1, and the sentence-level calibration set $\mathcal{C}$.

For each calibration document D_i of length T_i, we reset the reference and state weights to their initial values and run CoMa-LD. Let $M_t^{(i)}$ denote the resulting aggregate evidence process. We record the maximum log evidence

$$R_i = \max_{1 \leq t \leq T_i} \log M_t^{(i)}, \qquad i = 1, \ldots, N. \tag{11}$$

The same notation applies to the new test document D_{N+1}, whose length is T_{N+1}. Using the maximum, rather than only the final value, calibrates the event that the detector crosses its boundary at any point in the document.

Let $R_{(1)} \leq \cdots \leq R_{(N)}$ denote the ordered calibration statistics. For a target $\alpha \in (0, 1)$, we define

$$k_\alpha = \lceil (N+1)(1-\alpha) \rceil, \qquad q_\alpha^{\mathrm{doc}} = \begin{cases} R_{(k_\alpha)}, & k_\alpha \leq N, \\ +\infty, & k_\alpha = N+1. \end{cases} \tag{12}$$

For a document of length T, the document-calibrated alarm time is $\widehat{\tau}_{\mathrm{DC}} = \inf\{t \in \{1, \ldots, T\} : \log M_t > q_\alpha^{\mathrm{doc}}\}$. The threshold q_α^{doc} is determined by the held-out human documents and may be higher or lower than the corresponding log-Ville threshold $\log(1/\alpha)$.

Proposition 4 (Document-level false-alarm control)**.** *Conditioned on the fixed sentence-level calibration set $\mathcal{C}$, suppose that $D_1, \ldots, D_N, D_{N+1}$ are exchangeable human documents. All detector parameters are fixed before document calibration, and each document is processed with the same initialization and scoring procedure, using i.i.d. auxiliary randomization. Then*

$$\mathbb{P}\left(\exists\, t \leq T_{N+1} : \log M_t^{(N+1)} > q_\alpha^{\mathrm{doc}} \,\middle|\, \mathcal{C}\right) \leq \frac{N + 1 - k_\alpha}{N+1} \leq \alpha. \tag{13}$$

Proof: Refer to Appendix D.2 for a proof. □

Under document-level exchangeability, Proposition 4 provides finite-sample marginal control of the probability of any alarm within a human document, allowing arbitrary dependence among its sentences.

Table 1: WD and false-alarm performance on WritingPrompts and CNN/DailyMail in the black-box setting. Bold and underlined values indicate the best and second-best results within each dataset.

		Mixed documents: WD ↓				Human-only: FPR ↓
	Method	$K=1$	$K=2$	$K=3$	$K=5$	$K=0$
WritingPrompts	DetectGPT-Seg	$0.922_{\pm 0.016}$	$0.906_{\pm 0.010}$	$0.884_{\pm 0.007}$	$0.775_{\pm 0.028}$	$1.000_{\pm 0.000}$
	FastDetectGPT-Seg	$0.894_{\pm 0.007}$	$0.883_{\pm 0.008}$	$0.865_{\pm 0.009}$	$0.766_{\pm 0.028}$	$0.997_{\pm 0.006}$
	PaLD	$0.732_{\pm 0.017}$	$0.652_{\pm 0.013}$	$0.691_{\pm 0.021}$	$0.621_{\pm 0.017}$	$1.000_{\pm 0.000}$
	VCP	$0.278_{\pm 0.024}$	$0.330_{\pm 0.007}$	$0.367_{\pm 0.014}$	$0.384_{\pm 0.003}$	$0.153_{\pm 0.045}$
	WCP_{len}	$0.276_{\pm 0.018}$	$0.329_{\pm 0.012}$	$0.351_{\pm 0.012}$	$0.374_{\pm 0.001}$	$0.403_{\pm 0.040}$
	GCP	$0.338_{\pm 0.029}$	$0.394_{\pm 0.013}$	$0.420_{\pm 0.007}$	$0.430_{\pm 0.006}$	$0.593_{\pm 0.049}$
	LepskiCP	$0.486_{\pm 0.022}$	$0.475_{\pm 0.016}$	$0.479_{\pm 0.017}$	$0.438_{\pm 0.005}$	$0.427_{\pm 0.068}$
	CoMa-LD	$\underline{0.263_{\pm 0.004}}$	$\mathbf{0.282_{\pm 0.011}}$	$\underline{0.291_{\pm 0.015}}$	$\underline{0.321_{\pm 0.004}}$	$\underline{0.080_{\pm 0.017}}$
	CoMa-LD$_{\text{DC}}$	$\mathbf{0.262_{\pm 0.023}}$	$\underline{0.284_{\pm 0.011}}$	$\mathbf{0.288_{\pm 0.013}}$	$\mathbf{0.320_{\pm 0.004}}$	$\mathbf{0.040_{\pm 0.017}}$
CNN/DailyMail	DetectGPT-Seg	$0.939_{\pm 0.008}$	$0.910_{\pm 0.014}$	$0.852_{\pm 0.021}$	$0.723_{\pm 0.018}$	$1.000_{\pm 0.000}$
	FastDetectGPT-Seg	$0.833_{\pm 0.022}$	$0.801_{\pm 0.025}$	$0.739_{\pm 0.025}$	$0.637_{\pm 0.052}$	$1.000_{\pm 0.000}$
	PaLD	$0.703_{\pm 0.011}$	$0.619_{\pm 0.019}$	$0.645_{\pm 0.017}$	$0.565_{\pm 0.022}$	$1.000_{\pm 0.000}$
	VCP	$0.206_{\pm 0.021}$	$0.249_{\pm 0.032}$	$0.305_{\pm 0.027}$	$0.355_{\pm 0.024}$	$0.170_{\pm 0.035}$
	WCP_{len}	$0.226_{\pm 0.005}$	$0.254_{\pm 0.017}$	$0.296_{\pm 0.019}$	$0.357_{\pm 0.025}$	$0.340_{\pm 0.087}$
	GCP	$0.233_{\pm 0.010}$	$0.282_{\pm 0.016}$	$0.325_{\pm 0.014}$	$0.370_{\pm 0.018}$	$0.710_{\pm 0.035}$
	LepskiCP	$0.509_{\pm 0.004}$	$0.486_{\pm 0.006}$	$0.495_{\pm 0.008}$	$0.436_{\pm 0.005}$	$0.233_{\pm 0.046}$
	CoMa-LD	$\underline{0.187_{\pm 0.012}}$	$\underline{0.198_{\pm 0.013}}$	$\underline{0.217_{\pm 0.011}}$	$\mathbf{0.271_{\pm 0.033}}$	$\underline{0.043_{\pm 0.023}}$
	CoMa-LD$_{\text{DC}}$	$\mathbf{0.158_{\pm 0.011}}$	$\mathbf{0.187_{\pm 0.006}}$	$\mathbf{0.211_{\pm 0.006}}$	$\underline{0.284_{\pm 0.019}}$	$\mathbf{0.017_{\pm 0.021}}$

4 Experiments

4.1 Experimental Settings

Setup. We evaluate on WritingPrompts (WP) (Fan et al., 2018) and CNN/DailyMail (CNN) (Nallapati et al., 2016), using 100 documents per dataset and setting. Mixed documents contain $K \in \{1, 2, 3, 5\}$ transitions between alternating human-written and LLM-generated segments; $K = 0$ denotes human-only documents. The black-box setting uses Qwen3-8B (Qwen Team, 2025) as the generator and Llama-3.2-3B-Instruct (Llama Team, 2024) as the scoring and reference model for Fast-DetectGPT. CoMa-LD uses 100 held-out human sentences, $\alpha = 0.05$, and $\rho_{01} = \rho_{10} = 0.15$. CoMa-LD uses the Ville threshold; CoMa-LD$_{\text{DC}}$ calibrates its threshold on 500 separate human documents. Results are means and standard deviations over three seeds. Implementation details and additional model combinations are provided in Appendices F and G. All experiments are implemented in PyTorch and run on NVIDIA L40s GPUs.

Baselines. Localization baselines include VCP, WCP_{len}, and GCP (Li et al., 2026); LepskiCP (Lu et al., 2026); PaLD (Lei et al., 2025); and the sentence-level adaptations DetectGPT-Seg (Mitchell et al., 2023) and FastDetectGPT-Seg (Bao et al., 2024). Detection baselines include Fast-DetectGPT (Bao et al., 2024), MCP (Zhu et al., 2025), and Online Betting (Chen & Wang, 2025). All methods use the analytic Fast-DetectGPT statistic with the same frozen scoring model, except DetectGPT-Seg, which retains its perturbation-based score.

Metrics. Document-level true-positive rate (TPR) and false-positive rate (FPR) measure the fractions of mixed and human-only documents flagged, respectively. For localization baselines, a document is flagged if any part is labeled LLM-generated. We assess localization using WindowDiff (WD) (Pevzner & Hearst, 2002), which compares boundary counts within sliding windows, and sentence-level precision, recall, and F1 for the LLM class. Lower WD indicates better segmentation. Bold and underlined entries mark the best and second-best results within each comparison.

4.2 In-Domain Black-Box Evaluation

We evaluate black-box detection and localization using independent calibration and test splits from the same dataset. In Table 1, both CoMa-LD variants achieve lower WD than the localization baselines across the reported settings, indicating better recovery of the segmentation structure in mixed documents. This advantage persists across different numbers of human–LLM transitions. Both variants also produce fewer false alarms on fully human-written documents. Document-level calibration

Table 2: Sentence level precision, recall, and F1 for the LLM class with $K = 2$.

Method	WritingPrompts			CNN/DailyMail		
	Precision ↑	Recall ↑	F1 ↑	Precision ↑	Recall ↑	F1 ↑
VCP	$0.525_{\pm 0.022}$	$0.395_{\pm 0.024}$	$0.451_{\pm 0.022}$	$0.641_{\pm 0.026}$	$0.613_{\pm 0.044}$	$0.627_{\pm 0.034}$
WCP_{len}	$0.511_{\pm 0.003}$	$0.557_{\pm 0.077}$	$0.531_{\pm 0.037}$	$0.638_{\pm 0.043}$	$0.725_{\pm 0.044}$	$0.678_{\pm 0.038}$
LepskiCP	$0.495_{\pm 0.014}$	$0.620_{\pm 0.019}$	$0.551_{\pm 0.014}$	$0.574_{\pm 0.023}$	$0.471_{\pm 0.015}$	$0.517_{\pm 0.005}$
CoMa-LD	$0.781_{\pm 0.010}$	$\mathbf{0.672_{\pm 0.030}}$	$\mathbf{0.722_{\pm 0.021}}$	$0.834_{\pm 0.019}$	$\mathbf{0.853_{\pm 0.076}}$	$\mathbf{0.842_{\pm 0.026}}$
CoMa-LD$_{\text{DC}}$	$\mathbf{0.784_{\pm 0.015}}$	$0.637_{\pm 0.014}$	$0.703_{\pm 0.009}$	$\mathbf{0.906_{\pm 0.038}}$	$0.747_{\pm 0.041}$	$0.818_{\pm 0.009}$

Table 3: Document-level detection results at the target false-alarm level $\alpha = 0.05$.

	Method	Mixed documents: TPR ↑				Human-only: FPR ↓
		$K = 1$	$K = 2$	$K = 3$	$K = 5$	$K = 0$
WritingPrompts	Fast-DetectGPT	$0.307_{\pm 0.018}$	$0.128_{\pm 0.028}$	$0.269_{\pm 0.053}$	$0.275_{\pm 0.009}$	$0.060_{\pm 0.020}$
	MCP	$0.323_{\pm 0.012}$	$0.077_{\pm 0.033}$	$0.125_{\pm 0.026}$	$0.120_{\pm 0.029}$	$0.050_{\pm 0.017}$
	Online Betting	$0.181_{\pm 0.052}$	$0.245_{\pm 0.056}$	$0.387_{\pm 0.038}$	$0.440_{\pm 0.048}$	$0.060_{\pm 0.017}$
	CoMa-LD	$\mathbf{0.673_{\pm 0.014}}$	$\mathbf{0.707_{\pm 0.024}}$	$\mathbf{0.793_{\pm 0.044}}$	$\mathbf{0.747_{\pm 0.030}}$	$0.080_{\pm 0.017}$
	CoMa-LD$_{\text{DC}}$	$0.597_{\pm 0.030}$	$0.656_{\pm 0.029}$	$\mathbf{0.793_{\pm 0.045}}$	$0.741_{\pm 0.061}$	$\mathbf{0.040_{\pm 0.017}}$
CNN/DailyMail	Fast-DetectGPT	$0.784_{\pm 0.062}$	$0.552_{\pm 0.126}$	$0.739_{\pm 0.092}$	$0.723_{\pm 0.129}$	$0.177_{\pm 0.104}$
	MCP	$0.699_{\pm 0.086}$	$0.571_{\pm 0.141}$	$0.752_{\pm 0.106}$	$0.725_{\pm 0.147}$	$0.177_{\pm 0.095}$
	Online Betting	$0.288_{\pm 0.068}$	$0.341_{\pm 0.086}$	$0.507_{\pm 0.086}$	$0.411_{\pm 0.107}$	$0.090_{\pm 0.020}$
	CoMa-LD	$\mathbf{0.880_{\pm 0.094}}$	$\mathbf{0.899_{\pm 0.054}}$	$\mathbf{0.891_{\pm 0.026}}$	$\mathbf{0.768_{\pm 0.125}}$	$0.043_{\pm 0.023}$
	CoMa-LD$_{\text{DC}}$	$0.795_{\pm 0.045}$	$0.821_{\pm 0.065}$	$0.816_{\pm 0.085}$	$0.651_{\pm 0.060}$	$\mathbf{0.017_{\pm 0.021}}$

further reduces FPR while preserving similar WD on WritingPrompts and improving WD in most CNN/DailyMail settings.

Table 2 complements these segmentation results with sentence-level evaluation at $K = 2$. Both variants achieve higher LLM-class precision and F1 than the evaluated baselines on both datasets. The improvements therefore extend beyond segmentation quality to identifying which sentences are LLM-generated, supporting the effectiveness of CoMa-LD at both levels of localization.

For document-level detection, Table 3 shows that CoMa-LD achieves the highest TPR across all evaluated settings, demonstrating strong sensitivity to LLM-generated content even when human-written and generated segments alternate repeatedly. CoMa-LD$_{\text{DC}}$ further reduces false alarms, achieving the lowest FPR on both datasets.

4.3 Cross-Domain Calibration

We evaluate CoMa-LD on CNN using human calibration data from CNN or WP. Cross-domain calibration uses WP for both the sentence-reference set and the document-calibration set of CoMa-LD$_{\text{DC}}$. Mixed-document metrics are averaged equally over $K \in \{1, 2, 3, 5\}$, and the no-change-point (No-CP) rate denotes the fraction of mixed documents with no predicted change point. As shown in Table 4, although cross-domain calibration increases WD and FPR relative to in-domain calibration, both variants retain lower WD and FPR than VCP. These results suggest that CoMa-LD remains effective in this setting even when the calibration and test domains differ.

Table 4: Calibration-domain comparison on CNN data.

Method	Calib.	Mixed documents		Human-only
		Mean WD ↓	No-CP rate ↓	FPR ↓
VCP	—	$0.279_{\pm 0.025}$	$0.349_{\pm 0.096}$	$0.170_{\pm 0.035}$
CoMa-LD	CNN	$\mathbf{0.218_{\pm 0.014}}$	$0.145_{\pm 0.070}$	$\mathbf{0.043_{\pm 0.023}}$
	WP	$0.220_{\pm 0.029}$	$\mathbf{0.105_{\pm 0.018}}$	$0.063_{\pm 0.055}$
CoMa-LD$_{\text{DC}}$	CNN	$\mathbf{0.210_{\pm 0.009}}$	$0.229_{\pm 0.042}$	$\mathbf{0.017_{\pm 0.021}}$
	WP	$0.224_{\pm 0.023}$	$\mathbf{0.135_{\pm 0.031}}$	$0.047_{\pm 0.042}$

4.4 Ablation Studies

Localization ablation. We compare Viterbi decoding with independent sentence labeling, which assigns each sentence the state with the larger betting score, while keeping p-values and document-level alarm decisions fixed. As shown in Figure 3, Viterbi decoding achieves substantially lower WD

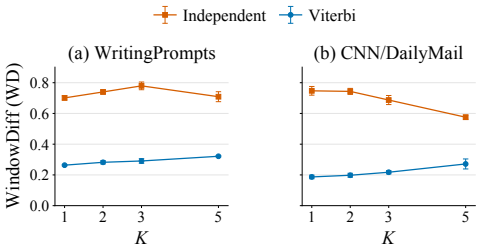


Figure 3: Localization ablation

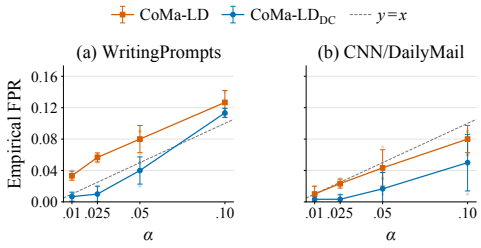


Figure 4: Empirical FPR across α.

across both datasets and all transition settings. These results support modeling neighboring labels through Markov transitions and decoding the full sequence jointly, rather than making independent sentence-level decisions.

False-alarm control across target levels. We compare Ville and document-calibrated thresholds across target false-alarm levels α. As shown in Figure 4, the Ville-based detector exceeds the target levels on WritingPrompts, whereas document-level calibration reduces the mean FPR to below the targets except for a slight exceedance at $\alpha = 0.10$. On CNN/DailyMail, both variants remain at or below the target levels, with document-level calibration yielding lower FPR.

Markov-switching detection. We compare standard CTM with CoMa-LD while keeping the full-document Viterbi decoding localizer fixed. Both detectors are separately calibrated on the same 100 human calibration sentences at $\alpha = 0.05$. Miss rate denotes the fraction of mixed documents without an alarm, which are labeled entirely human. As shown in Table 5, CoMa-LD substantially reduces missed detections on both datasets while maintaining similarly low FPR. The accompanying reduction in WD shows that improved detection also benefits overall localization. CoMa-LD also achieves comparable or shorter alarm delays among successfully detected mixed documents.

Table 5: Detector ablation.

Data	Detector	FPR ↓	Miss rate ↓	WD ↓	Alarm delay ↓
WP	CTM	$\mathbf{0.063_{\pm 0.015}}$	$0.561_{\pm 0.021}$	$0.342_{\pm 0.001}$	$10.43_{\pm 0.41}$
	CoMa-LD	$0.080_{\pm 0.017}$	$\mathbf{0.270_{\pm 0.021}}$	$\mathbf{0.289_{\pm 0.006}}$	$9.64_{\pm 0.38}$
CNN	CTM	$0.120_{\pm 0.075}$	$0.308_{\pm 0.146}$	$0.290_{\pm 0.004}$	$8.20_{\pm 0.46}$
	CoMa-LD	$\mathbf{0.043_{\pm 0.023}}$	$\mathbf{0.141_{\pm 0.073}}$	$\mathbf{0.218_{\pm 0.014}}$	$8.85_{\pm 1.92}$

5 Conclusion

This work approached LLM-text detection through departures from exchangeability with human reference data. Rather than directly classifying individual sentences, CoMa-LD calibrates their detection scores and accumulates evidence of such departures across a document. Its Markov-switching formulation and Viterbi localizer further connect this evidence to the localization of repeated human–LLM transitions. Our experiments show that this approach improves localization while reducing false alarms, and our theoretical analysis establishes false-alarm control under explicit exchangeability assumptions. Together, these findings support exchangeability testing as a useful foundation for detecting LLM-generated content in mixed documents. Extending this approach to mismatched human references and more diverse editing patterns remains an important direction for future work.

6 AI Use Statement

We used generative AI to produce LLM-written text segments for experimental evaluation, as described in the experimental setup. We also used generative AI tools to polish the language and improve the clarity of the manuscript. The authors reviewed the AI-assisted manuscript revisions and generated experimental data and take responsibility for the methodology, analyses, and final content of this work.

7 Reproducibility Statement

Experimental details, including data construction, calibration procedures, and scoring functions, are provided in Appendix F, with additional results in Appendix G. Together with our anonymous code repository, these materials support reproducibility.

References

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, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=Bpcgcr8E8Z.

Can Chen and Jun-Kun Wang. Online detection of llm-generated texts via sequential hypothesis testing by betting. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/chen25bn.html.

Angela Fan, Mike Lewis, and Yann N. Dauphin. Hierarchical neural story generation. In Iryna Gurevych and Yusuke Miyao (eds.), *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pp. 889–898. Association for Computational Linguistics, 2018. doi: 10.18653/V1/P18-1082. URL https://aclanthology.org/P18-1082/.

Valentina Fedorova, Alex Gammerman, Ilia Nouretdinov, and Volodya Vovk. Plug-in martingales for testing exchangeability on-line. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*. icml.cc / Omnipress, 2012. URL http://icml.cc/2012/papers/808.pdf.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander M. Rush. GLTR: statistical detection and visualization of generated text. In Marta R. Costa-jussà and Enrique Alfonseca (eds.), *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28 - August 2, 2019, Volume 3: System Demonstrations*, pp. 111–116. Association for Computational Linguistics, 2019. doi: 10.18653/V1/P19-3019. URL https://doi.org/10.18653/v1/p19-3019.

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. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 17519–17537. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/hans24a.html.

Lei Jiang, Desheng Wu, and Xiaolong Zheng. SenDetEX: Sentence-level AI-generated text detection for human-AI hybrid content via style and context fusion. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, Suzhou, China, November 4-9, 2025*, pp. 5287–5302. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.EMNLP-MAIN.268. URL https://doi.org/10.18653/v1/2025.emnlp-main.268.

Eric Lei, Hsiang Hsu, and Chun-Fu Chen. PaLD: Detection of text partially written by large language models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=rWjZWHYPcz.

Mengchu Li, Jin Zhu, Jinglai Li, and Chengchun Shi. Segmenting human-llm co-authored text via change point detection. *CoRR*, abs/2605.03723, 2026. doi: 10.48550/ARXIV.2605.03723. URL https://doi.org/10.48550/arXiv.2605.03723.

Llama Team. The Llama 3 herd of models. *CoRR*, abs/2407.21783, 2024. doi: 10.48550/ARXIV.2407.21783. URL https://doi.org/10.48550/arXiv.2407.21783.

Yangjun Lu, Hongyi Zhou, Fabian Spill, Kai Ye, Chengchun Shi, and Jin Zhu. Detecting llm-generated tokens in human-llm coauthored text. *CoRR*, abs/2607.21458, 2026. doi: 10.48550/ARXIV.2607.21458. URL https://doi.org/10.48550/arXiv.2607.21458.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. DetectGPT: Zero-shot machine-generated text detection using probability curvature. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 24950–24962. PMLR, 2023. URL https://proceedings.mlr.press/v202/mitchell23a.html.

Ramesh Nallapati, Bowen Zhou, Cícero Nogueira dos Santos, Çaglar Gülçehre, and Bing Xiang. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In Yoav Goldberg and Stefan Riezler (eds.), *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016*, pp. 280–290. ACL, 2016. doi: 10.18653/V1/K16-1028. URL https://doi.org/10.18653/v1/k16-1028.

Leyi Pan, Aiwei Liu, Yijian Lu, Zitian Gao, Yichen Di, Lijie Wen, Irwin King, and Philip S. Yu. WaterSeeker: Pioneering efficient detection of watermarked segments in large documents. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Findings of the Association for Computational Linguistics: NAACL 2025, Albuquerque, New Mexico, USA, April 29 - May 4, 2025*, volume NAACL 2025 of *Findings of ACL*, pp. 2866–2882. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.FINDINGS-NAACL.156. URL https://doi.org/10.18653/v1/2025.findings-naacl.156.

Lev Pevzner and Marti A. Hearst. A critique and improvement of an evaluation metric for text segmentation. *Comput. Linguistics*, 28(1):19–36, 2002. doi: 10.1162/089120102317341756. URL https://doi.org/10.1162/089120102317341756.

Drew Prinster, Xing Han, Anqi Liu, and Suchi Saria. WATCH: adaptive monitoring for AI deployments via weighted-conformal martingales. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/prinster25a.html.

Qwen Team. Qwen3 technical report. *CoRR*, abs/2505.09388, 2025. doi: 10.48550/ARXIV.2505.09388. URL https://doi.org/10.48550/arXiv.2505.09388.

Tom Sander, Hongyan Chang, Tomás Soucek, Tuan Tran, Valeriu Lacatusu, Sylvestre-Alvise Rebuffi, Alexandre Mourachko, Surya Parimi, Christophe Ropers, Rashel Moritz, Vanessa Stark, Hady Elsahar, and Pierre Fernandez. TextSeal: A localized LLM watermark for provenance & distillation protection. *CoRR*, abs/2605.12456, 2026. doi: 10.48550/ARXIV.2605.12456. URL https://doi.org/10.48550/arXiv.2605.12456.

Irina Tolstykh, Aleksandra Tsybina, Sergey Yakubson, Aleksandr Gordeev, Vladimir Dokholyan, and Maksim Kuprashevich. GigaCheck: Detecting LLM-generated content via object-centric span localization. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics, ACL 2026, San Diego, California, United States, July 2-7, 2026*, pp. 4349–4364. Association for Computational Linguistics, 2026. doi: 10.18653/V1/2026.FINDINGS-ACL.213. URL https://doi.org/10.18653/v1/2026.findings-acl.213.

Vivek Verma, Eve Fleisig, Nicholas Tomlin, and Dan Klein. Ghostbuster: Detecting text ghostwritten by large language models. In Kevin Duh, Helena Gómez-Adorno, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pp. 1702–1717. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.NAACL-LONG.95. URL https://doi.org/10.18653/v1/2024.naacl-long.95.

Vladimir Vovk, Ilia Nouretdinov, and Alex Gammerman. Testing exchangeability on-line. In Tom Fawcett and Nina Mishra (eds.), *Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003), August 21-24, 2003, Washington, DC, USA*, pp. 768–775. AAAI Press, 2003. URL http://www.aaai.org/Library/ICML/2003/icml03-100.php.

Vladimir Vovk, Ivan Petej, Ilia Nouretdinov, Ernst Ahlberg, Lars Carlsson, and Alex Gammerman. Retrain or not retrain: conformal test martingales for change-point detection. In Lars Carlsson, Zhiyuan Luo, Giovanni Cherubin, and Khuong An Nguyen (eds.), *Conformal and Probabilistic Prediction and Applications, 8-10 September 2021, Virtual Event*, volume 152 of *Proceedings of Machine Learning Research*, pp. 191–210. PMLR, 2021. URL https://proceedings.mlr.press/v152/vovk21b.html.

Vladimir Vovk, Ilia Nouretdinov, and Alex Gammerman. Conformal testing: binary case with markov alternatives. In Ulf Johansson, Henrik Boström, Khuong An Nguyen, Zhiyuan Luo, and Lars Carlsson (eds.), *Conformal and Probabilistic Prediction with Applications, 24-26 August 2022, Brighton, UK*, volume 179 of *Proceedings of Machine Learning Research*, pp. 207–218. PMLR, 2022. URL https://proceedings.mlr.press/v179/vovk22a.html.

Zijie Zeng, Shiqi Liu, Lele Sha, Zhuang Li, Kaixun Yang, Sannyuya Liu, Dragan Gasevic, and Guangliang Chen. Detecting AI-generated sentences in human-AI collaborative hybrid texts: Challenges, strategies, and insights. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI 2024, Jeju, South Korea, August 3-9, 2024*, pp. 7545–7553. ijcai.org, 2024. URL https://www.ijcai.org/proceedings/2024/835.

Qihui Zhang, Chujie Gao, Dongping Chen, Yue Huang, Yixin Huang, Zhenyang Sun, Shilin Zhang, Weiye Li, Zhengyan Fu, Yao Wan, and Lichao Sun. LLM-as-a-coauthor: Can mixed human-written and machine-generated text be detected? In Kevin Duh, Helena Gómez-Adorno, and Steven Bethard (eds.), *Findings of the Association for Computational Linguistics: NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, volume NAACL 2024 of *Findings of ACL*, pp. 409–436. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.FINDINGS-NAACL.29. URL https://doi.org/10.18653/v1/2024.findings-naacl.29.

Xuandong Zhao, Chenwen Liao, Yuxiang Wang, and Lei Li. Efficiently identifying watermarked segments in mixed-source texts. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pp. 6304–6316. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.ACL-LONG.316. URL https://doi.org/10.18653/v1/2025.acl-long.316.

Hongyi Zhou, Jin Zhu, Pingfan Su, Kai Ye, Ying Yang, Shakeel Gavioli-Akilagun, and Chengchun Shi. AdaDetectGPT: Adaptive detection of llm-generated text with statistical guarantees. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/808bc5e3d076c1125a87f81b42d5e52d-Abstract-Conference.html.

Xiaowei Zhu, Yubing Ren, Yanan Cao, Xixun Lin, Fang Fang, and Yangxi Li. Reliably bounding false positives: A zero-shot machine-generated text detection framework via multiscaled conformal prediction. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher

Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pp. 12298–12319. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.ACL-LONG.601. URL https://doi.org/10.18653/v1/2025.acl-long.601.

A More Related Work

We provide an extended discussion of related work. Table 6 compares representative methods in terms of training requirements, watermark dependence, detection, and localization capabilities.

LLM-text detection and false-alarm control. Existing detectors either construct scores from language-model probabilities or learn classifiers from labeled examples. DetectGPT estimates probability curvature through text perturbations (Mitchell et al., 2023), Fast-DetectGPT uses conditional probability curvature (Bao et al., 2024), and Binoculars compares scores from two pretrained language models (Hans et al., 2024). Ghostbuster trains a classifier on selected combinations of features from weaker language models (Verma et al., 2024). Beyond score construction, MCP applies length-aware conformal calibration to control false positives (Zhu et al., 2025). Chen & Wang (2025) develop an online detector that accumulates evidence across texts from an unknown source through sequential betting, controlling false alarms under the stated null assumptions. These methods provide passage-level or source-level decisions, but do not themselves identify generated segments within a mixed document.

Localization in mixed human-LLM text. Localization methods predict local labels or identify generated segments within a document. PaLD uses existing classifier scores to estimate the proportion of LLM-generated text and locate generated segments (Lei et al., 2025). VCP, WCP, and GCP identify changes in detection-score distributions and use the resulting segmentation for localization (Li et al., 2026). Lu et al. (2026) smooth neighboring token-level detection scores using adaptively selected bandwidths. Learned approaches include SenDetEX, which combines style and context features for sentence-level classification (Jiang et al., 2025), and GigaCheck, which uses fine-tuned language-model representations for document classification and a span prediction network for locating multiple generated segments (Tolstykh et al., 2026). WaterSeeker, the Geometric Cover Detector (GCD) paired with the Adaptive Online Locator (AOL), and TextSeal detect and localize watermarked segments using signals embedded during generation (Pan et al., 2025; Zhao et al., 2025; Sander et al., 2026). CoMa-LD connects sequential document-level detection with sentence-level localization using shared conformal evidence, without additional detector training or generation-time watermarks.

Conformal test martingales. Conformal test martingales accumulate evidence against exchangeability by betting on sequential conformal p-values (Vovk et al., 2003; 2021; 2022). Plug-in martingales adapt their betting functions by estimating the density of past conformal p-values (Fedorova et al., 2012). WATCH uses weighted conformal martingales for adaptive monitoring and diagnosis of distribution shifts (Prinster et al., 2025).

B Algorithm

Algorithm 1 summarizes CoMa-LD inference. The alarm rule is fixed before processing the test document:

$$\mathsf{Alarm}(M) = \begin{cases} \mathbf{1}\{M \geq 1/\alpha\}, & \text{Ville threshold,} \\ \mathbf{1}\{\log M > q_\alpha^{\mathrm{doc}}\}, & \text{document-calibrated threshold.} \end{cases}$$

The document-calibrated threshold is obtained from equation 12.

C Complexity

Computational complexity. Let n be the number of sentence-level calibration scores and T the document length. The prequential reference contains $n + t - 1$ scores before sentence t and grows

Table 6: Capabilities and requirements of representative LLM-text detection and localization methods. Training-free means that no additional detector training or fine-tuning is required; calibration and online betting updates are allowed. Watermark-free methods do not require watermarks embedded during text generation. False-alarm guarantees are assessed under each method's stated assumptions.

Method	Requirements		Detection		Localization	
	Training-free	Watermark-free	Online detection	False-alarm control	Fine-grained labels	Multiple segments
SenDetEX (Jiang et al., 2025)	–	✓	–	–	✓	✓
GigaCheck (Tolstykh et al., 2026)	–	✓	–	–	✓	✓
WaterSeeker (Pan et al., 2025)	✓	–	–	–	✓	✓
GCD + AOL (Zhao et al., 2025)	✓	–	–	✓	✓	✓
TextSeal (Sander et al., 2026)	✓	–	–	✓	✓	✓
DetectGPT (Mitchell et al., 2023)	✓	✓	–	–	–	–
Fast-DetectGPT (Bao et al., 2024)	✓	✓	–	–	–	–
MCP (Zhu et al., 2025)	✓	✓	–	✓	–	–
Online Betting (Chen & Wang, 2025)	✓	✓	✓	✓	–	–
DetectGPT-Seg (Mitchell et al., 2023)	✓	✓	–	–	✓	✓
FastDetectGPT-Seg (Bao et al., 2024)	✓	✓	–	–	✓	✓
PaLD (Lei et al., 2025)	✓	✓	–	–	✓	✓
VCP (Li et al., 2026)	✓	✓	–	–	✓	✓
WCP_{len} (Li et al., 2026)	✓	✓	–	–	✓	✓
GCP (Li et al., 2026)	✓	✓	–	–	✓	✓
LepskiCP (Lu et al., 2026)	✓	✓	–	–	✓	✓
CoMa-LD	✓	✓	✓	✓	✓	✓
CoMa-LD$_{\text{DC}}$	✓	✓	✓	✓	✓	✓

Online detection denotes sequential evidence accumulation with an alarm rule. Fine-grained labels include sentence- or token-level Human/LLM predictions. Multiple segments means that more than one LLM-generated segment can be localized within a document. A dash indicates that the described method does not provide the corresponding capability or guarantee.

to $n+T$ scores, requiring $O(n+T)$ memory. Using a balanced order-statistic tree, each conformal p-value and subsequent reference update can be computed in $O(\log(n+t))$ time, after $O(n \log n)$ initialization. Given the p-values, the two-state forward detector requires $O(T)$ time and $O(1)$ working memory, while Viterbi decoding requires $O(T)$ time and $O(T)$ memory for backpointers. Thus, excluding sentence scoring, the full procedure can be implemented in $O(n \log n + T \log(n+T))$ time and $O(n+T)$ memory. For N document-level calibration examples with lengths $T_1, \ldots, T_N$, the forward updates and running maxima require $O(\sum_{i=1}^{N} T_i)$ time once the conformal p-values are available. In practice, the computational cost is dominated by sentence scoring with the detector language model rather than by the Markov updates or Viterbi decoding.

D Proof

D.1 Proof of Proposition 3

Proof. We separate the update into Markov propagation and state-specific betting. First, define the total weight immediately after the Markov transition as $\widetilde{M}_t = \widetilde{W}_t^0 + \widetilde{W}_t^1$. Using equation 7 and the fact that every row of P sums to one,

$$\begin{aligned}\widetilde{M}_t &= \sum_{c\in\{0,1\}} \sum_{c'\in\{0,1\}} W_{t-1}^{c'} P_{c'c} \\ &= \sum_{c'\in\{0,1\}} W_{t-1}^{c'} \sum_{c\in\{0,1\}} P_{c'c} = \sum_{c'\in\{0,1\}} W_{t-1}^{c'} = M_{t-1}. \end{aligned} \tag{14}$$

Thus, the Markov transition only redistributes capital between the two states and leaves the total capital unchanged.

Algorithm 1 CoMa-LD

Require: Document $X_{1:T}$; human calibration scores $\{s_i^{\mathrm{cal}}\}_{i=1}^n$
Require: Fixed scoring function $\mathcal{A}$, transition matrix P, initial weights π, and alarm rule Alarm
Ensure: First alarm time $\widehat{\tau}$ and sentence labels $\widehat{C}_{1:T}$

$g_0(p) \leftarrow 1$, $g_1(p) \leftarrow -\log p$
$\mathcal{R} \leftarrow \{s_1^{\mathrm{cal}}, \ldots, s_n^{\mathrm{cal}}\}$; $\widehat{\tau} \leftarrow T+1$
$W_0^c \leftarrow \pi_c$, $\delta_0(c) \leftarrow \log \pi_c$ for $c \in \{0,1\}$
for $t = 1, \ldots, T$ **do**
  $s_t \leftarrow \mathcal{A}(X_t)$; draw $U_t \sim \mathrm{Unif}(0,1)$
  $p_t \leftarrow \dfrac{|\{r \in \mathcal{R} : r > s_t\}| + U_t\,(1 + |\{r \in \mathcal{R} : r = s_t\}|)}{|\mathcal{R}| + 1}$ ▷ **Conformal evidence**
  Add s_t to $\mathcal{R}$, retaining multiplicities
  $W_t^c \leftarrow g_c(p_t) \sum_{c' \in \{0,1\}} W_{t-1}^{c'} P_{c'c}$ for $c \in \{0,1\}$
  $M_t \leftarrow W_t^0 + W_t^1$ ▷ **Detection: aggregate path weights**
  if $\widehat{\tau} = T+1$ and $\mathsf{Alarm}(M_t) = 1$ **then**
    $\widehat{\tau} \leftarrow t$; raise a document-level alarm
  for $c \in \{0,1\}$ **do** ▷ **Localization: Viterbi update**
    $\psi_t(c) \leftarrow \arg\max_{c' \in \{0,1\}} \{\delta_{t-1}(c') + \log P_{c'c}\}$
    $\delta_t(c) \leftarrow \log g_c(p_t) + \delta_{t-1}(\psi_t(c)) + \log P_{\psi_t(c),c}$
▷ **After the complete document has been processed**
if $\widehat{\tau} = T+1$ **then**
  return $(\widehat{\tau}, (0, \ldots, 0))$
▷ **Localization: recover the label sequence**
$\widehat{C}_T \leftarrow \arg\max_{c \in \{0,1\}} \delta_T(c)$
for $t = T, T-1, \ldots, 2$ **do**
  $\widehat{C}_{t-1} \leftarrow \psi_t(\widehat{C}_t)$
return $(\widehat{\tau}, \widehat{C}_{1:T})$

The state-specific bets are then applied according to $W_t^c = \widetilde{W}_t^c g_c(p_t)$. Because the propagated weights are $\mathcal{F}_{t-1}$-measurable and each betting function has conditional mean one,

$$\begin{aligned}\mathbb{E}[M_t \mid \mathcal{F}_{t-1}] &= \sum_{c \in \{0,1\}} \widetilde{W}_t^c\, \mathbb{E}[g_c(p_t) \mid \mathcal{F}_{t-1}] \\ &= \sum_{c \in \{0,1\}} \widetilde{W}_t^c = \widetilde{M}_t = M_{t-1}. \end{aligned} \tag{15}$$

Finally, M_t is nonnegative because the transition probabilities, state weights, and betting functions are all nonnegative, and $M_0 = \pi_0 + \pi_1 = 1$. Therefore, $(M_t)_{t \geq 0}$ is a nonnegative test martingale. □

D.2 Proof of Proposition 4

Proof. Condition on $\mathcal{C}$ and the fixed detector parameters. Since the documents are exchangeable and processed by the same procedure with independent, identically distributed auxiliary randomization, their maximum log-evidence statistics $R_1, \ldots, R_{N+1}$ are exchangeable.

Let J denote the ascending rank of R_{N+1} among these $N+1$ statistics, with ties broken uniformly at random for the proof. By exchangeability, J is uniform on $\{1, \ldots, N+1\}$. For $k_\alpha \leq N$, the event $R_{N+1} > R_{(k_\alpha)}$ implies $J > k_\alpha$, since at least k_α calibration statistics are strictly smaller than R_{N+1}. Therefore,

$$\begin{aligned}&\mathbb{P}\left(\exists\, t \leq T_{N+1} : \log M_t^{(N+1)} > q_\alpha^{\mathrm{doc}} \,\middle|\, \mathcal{C}\right) \\ &\quad = \mathbb{P}\left(R_{N+1} > R_{(k_\alpha)} \,\middle|\, \mathcal{C}\right) \\ &\quad \leq \mathbb{P}(J > k_\alpha \mid \mathcal{C}) = \frac{N+1-k_\alpha}{N+1} \leq \alpha, \end{aligned} \tag{16}$$

Table 7: Mean WD and human-only FPR under evidence discounting.

	WritingPrompts		CNN/DailyMail	
γ	Mean WD ↓	FPR ↓	Mean WD ↓	FPR ↓
1.00	0.3031	0.0778	0.2403	0.0444
0.98	**0.2987**	0.0889	0.2290	0.0444
0.95	0.2991	**0.0556**	0.2288	0.0444
0.90	0.3009	0.0667	**0.2271**	0.0444
0.80	0.3028	0.0667	0.2298	0.0444

where the last inequality follows from $k_\alpha = \lceil (N+1)(1-\alpha) \rceil$. If $k_\alpha = N+1$, the threshold is $+\infty$, so the alarm probability is zero and the bound holds trivially. □

E Discounted Evidence Accumulation

We explore an optional extension of CoMa-LD that discounts earlier detection evidence. An initial human-written segment can drive the log-wealth below zero, requiring subsequent evidence from LLM-generated text to offset this accumulated negative evidence before triggering an alarm. Discounting earlier evidence may help the detector respond sooner to later LLM-generated segments.

Let $M_t > 0$ denote the original CoMa-LD detection wealth after processing t sentences, with $M_0 = 1$. We introduce a discount factor $\gamma \in (0, 1]$ and define

$$S_0^{(\gamma)} = 0, \qquad S_t^{(\gamma)} = \gamma S_{t-1}^{(\gamma)} + \log \frac{M_t}{M_{t-1}}. \tag{17}$$

Equivalently, the discounted statistic $Z_t^{(\gamma)} = \exp(S_t^{(\gamma)})$ satisfies

$$Z_t^{(\gamma)} = \left(Z_{t-1}^{(\gamma)}\right)^\gamma \frac{M_t}{M_{t-1}}, \qquad S_t^{(\gamma)} = \sum_{i=1}^{t} \gamma^{t-i} \log \frac{M_i}{M_{i-1}}. \tag{18}$$

Thus, $\gamma = 1$ recovers the original wealth, whereas $\gamma < 1$ assigns less weight to earlier log-wealth increments. Discounting is applied to the accumulated detection evidence; the increments are computed from the original wealth process.

Negative log-wealth increments accumulated over an initial human-written segment can leave the detector with substantial negative evidence when LLM-generated text begins. Discounting reduces the contribution of these earlier increments, potentially allowing subsequent evidence to trigger an alarm sooner. However, it also downweights earlier positive evidence, so stronger discounting need not improve detection.

For $\gamma < 1$, the discounted statistic is generally not a nonnegative supermartingale. If M_t is a positive martingale under the human-written null hypothesis with respect to $\mathcal{F}_t$, then

$$\mathbb{E}_0\left[Z_t^{(\gamma)} \mid \mathcal{F}_{t-1}\right] = \left(Z_{t-1}^{(\gamma)}\right)^\gamma. \tag{19}$$

For $\gamma < 1$, this conditional expectation exceeds $Z_{t-1}^{(\gamma)}$ whenever $0 < Z_{t-1}^{(\gamma)} < 1$, violating the supermartingale condition. Consequently, the original Ville-based false-alarm guarantee does not directly extend to the discounted statistic.

All variants in Table 7 raise an alarm when $S_t^{(\gamma)} \geq \log(1/\alpha)$, using the same threshold with $\alpha = 0.05$. Discounting yields small changes in WD on WritingPrompts and larger reductions on CNN/DailyMail, where $\gamma = 0.90$ achieves the lowest WD. Human-only FPR varies across discount factors on WritingPrompts and remains 0.0444 on CNN/DailyMail.

F Experimental Details

F.1 Mixed-text Generation

We construct mixed documents from WritingPrompts and CNN/DailyMail using human-written source texts with more than 800 words, following an alternating human–LLM segment design similar to that of Li et al. (2026). Unlike a fixed human-first design, our evaluation includes documents beginning with either human-written or LLM-generated text. For each $K \in \{1, 2, 3, 5\}$, we retain up to $16(K + 1)$ sentences and divide them into $K + 1$ contiguous, nonempty blocks. We replace alternating blocks with Qwen3-8B continuations, preserving the sentence count of each replaced block. Each LLM block is conditioned on the immediately preceding human block. For LLM-first documents, the initial LLM block instead uses a two-sentence human prefix that is excluded from the evaluated document. Both starting configurations therefore yield alternating human-written and LLM-generated segments with exactly K ground-truth change points. Generation uses temperature 0.8, top-p 0.95, and at most 512 new tokens per attempt.

Generation prompt. The following messages are passed through the generator's chat template. The placeholders specify the requested sentence count and the preceding human block.

Prompt for generating an LLM segment

System message:

```
You are a creative writing assistant.  Continue the given text
naturally and fluently, matching the style and tone of what came
before.
```

User message:

```
Continue the following text by writing exactly {target_sentences}
new sentences.  Return ONLY the new continuation sentences, without
repeating the original text.

{context_text}
```

F.2 Calibration Data Construction

All methods use calibration and test data from disjoint source documents. For each dataset, we construct a human sentence-reference pool A containing 100 source documents and a document-calibration pool B containing 500 source documents. Both pools are randomly sampled from held-out human texts, with no shared sources or duplicate texts across A, B, and the test set.

CoMa-LD and CoMa-LD$_{\mathrm{DC}}$ use the same sentence-reference construction. We sample 100 sentences without replacement from the pooled sentences in A to form the initial reference. Reference and test sentences use the same scoring model and function. The reference pool resets to this initial set for each document and grows sequentially, adding each sentence's score only after that sentence is evaluated.

CoMa-LD$_{\mathrm{DC}}$ additionally processes each document in B using the same initial reference and sequential detector. The resulting maximum log-wealth statistics determine the document-level alarm threshold. For each dataset and scoring configuration, the initial reference and document-calibrated threshold are fixed and shared across evaluation seeds in $\{10, 20, 30\}$.

F.3 Human-Only Controls

To evaluate false alarms, we construct 100 human-only control documents per dataset and evaluation seed $\{10, 20, 30\}$. Control sources are disjoint from all reference, calibration, and mixed-text evaluation sources, with no duplicate texts across sets. Each control retains the original sentences without generation or rewriting. All sentences are labeled Human.

We report the false-positive rate as the proportion of controls flagged as containing LLM-generated text. CoMa-LD variants use their document-level alarm, while localization baselines flag a document if any sentence is predicted as LLM-generated. Each control set is evaluated once per seed.

F.4 Nonconformity Scores

We instantiate the sentence-level nonconformity function $\mathcal{A}$ using two model-based scores. Let $X_t = (x_{t,1}, \ldots, x_{t,m_t})$ denote a tokenized sentence and let q_θ be a fixed scoring language model. Both scores are oriented so that larger values provide stronger evidence of LLM generation.

Fast-DetectGPT. Fast-DetectGPT measures how much the observed tokens' log probabilities exceed those of alternative tokens under the same contexts (Bao et al., 2024). Let $\ell_{t,j}(v) = \log q_\theta(v \mid x_{t,<j})$ and let r_ϕ denote the reference language model. For each position, we compute

$$\begin{aligned} \mu_{t,j} &= \mathbb{E}_{v \sim r_\phi(\cdot \mid x_{t,<j})} \left[\ell_{t,j}(v) \right], \\ \sigma^2_{t,j} &= \mathbb{E}_{v \sim r_\phi(\cdot \mid x_{t,<j})} \left[(\ell_{t,j}(v) - \mu_{t,j})^2 \right]. \end{aligned} \tag{20}$$

The sentence-level statistic is

$$\mathcal{A}_{\mathrm{FD}}(X_t) = \frac{\sum_{j=2}^{m_t} \left[\ell_{t,j}(x_{t,j}) - \mu_{t,j} \right]}{\sqrt{\sum_{j=2}^{m_t} \sigma^2_{t,j}}}. \tag{21}$$

We evaluate the expectations analytically over the token vocabulary, without sampling alternative texts. A larger score indicates that the observed sentence has a higher log-likelihood relative to the reference distribution, after variance normalization. Our main experiments use Llama-3.2-3B-Instruct as both q_θ and r_ϕ.

Negative mean NLL. Likelihood-based detection exploits the tendency of LLM-generated text to contain higher-probability tokens under a language model (Gehrmann et al., 2019). We therefore use the negative mean negative log-likelihood as the nonconformity score:

$$\mathcal{A}_{\mathrm{NLL}}(X_t) = -\,\mathrm{NLL}(X_t) = \frac{1}{m_t - 1} \sum_{j=2}^{m_t} \log q_\theta(x_{t,j} \mid x_{t,<j}). \tag{22}$$

The sign reversal ensures that lower NLL corresponds to a larger nonconformity score. We use Fast-DetectGPT for the main results and negative mean NLL for supplementary experiments. In both cases, the same scoring function is applied to calibration and test sentences before conformal calibration.

G Additional Experiments

G.1 Detector–Localizer Cascades

We compare CoMa-LD and CoMa-LD$_{\mathrm{DC}}$ with six detector–localizer cascades, combining each of CTM, MCP, and Online Betting with either VCP or LepskiCP. Each cascade returns no change points unless its detector raises an alarm; following an alarm, it returns the localizer's prediction over the full document. We use the same models, scoring-function settings, and test and calibration data as in the main experiments, retaining each method's existing parameters and calibration procedure.

Both CoMa-LD variants attain lower mean WD than all six cascades on both datasets and at every evaluated transition count (Table 8). Among the cascade detectors, CTM has the lowest mean miss rate on both datasets, but this does not consistently translate into lower WD across localizers and transition counts. The diagnostics reveal two sources of empty predictions in the cascades: missed detections prevent localization, and the localizer may return no change points even after an alarm (Table 9). Both CoMa-LD variants exhibit lower miss rates and lower overall no-change-point rates than all evaluated cascades. CoMa-LD$_{\mathrm{DC}}$ also achieves the lowest mean empirical FPR on both datasets, whereas the base variant has a higher FPR than all cascades on WritingPrompts. Together, these results support the effectiveness of CoMa-LD's joint detection and localization approach relative to the evaluated cascades.

G.2 Results with NLL-Based Scores

Using the same models as in the main experiments, we evaluate CoMa-LD with negative mean NLL as an alternative nonconformity score. As shown in Table 10, CoMa-LD and its document-calibrated

Table 8: Localization quality and document-level false alarms for detector–localizer cascades and CoMa-LD.

	Method	Mixed documents: WD ↓ $K=1$	$K=2$	$K=3$	$K=5$	Human-only: FPR ↓ $K=0$
WritingPrompts	MCP → VCP	$0.314_{\pm 0.005}$	$0.388_{\pm 0.007}$	$0.405_{\pm 0.006}$	$0.410_{\pm 0.004}$	$0.050_{\pm 0.017}$
	MCP → LepskiCP	$0.372_{\pm 0.008}$	$0.397_{\pm 0.003}$	$0.416_{\pm 0.002}$	$0.415_{\pm 0.005}$	$0.050_{\pm 0.017}$
	Online Betting → VCP	$0.320_{\pm 0.004}$	$0.382_{\pm 0.004}$	$0.394_{\pm 0.006}$	$0.400_{\pm 0.007}$	$0.060_{\pm 0.017}$
	Online Betting → LepskiCP	$0.353_{\pm 0.003}$	$0.407_{\pm 0.002}$	$0.438_{\pm 0.007}$	$0.422_{\pm 0.005}$	$0.060_{\pm 0.017}$
	CTM → VCP	$0.304_{\pm 0.008}$	$0.368_{\pm 0.004}$	$0.390_{\pm 0.006}$	$0.400_{\pm 0.004}$	$0.063_{\pm 0.015}$
	CTM → LepskiCP	$0.402_{\pm 0.016}$	$0.432_{\pm 0.008}$	$0.441_{\pm 0.016}$	$0.426_{\pm 0.008}$	$0.063_{\pm 0.015}$
	CoMa-LD	$0.263_{\pm 0.004}$	$\mathbf{0.282_{\pm 0.011}}$	$0.291_{\pm 0.015}$	$0.321_{\pm 0.004}$	$0.080_{\pm 0.017}$
	CoMa-LD$_{\mathrm{DC}}$	$\mathbf{0.262_{\pm 0.023}}$	$0.284_{\pm 0.011}$	$\mathbf{0.288_{\pm 0.013}}$	$\mathbf{0.320_{\pm 0.004}}$	$\mathbf{0.040_{\pm 0.017}}$
CNN/DailyMail	MCP → VCP	$0.233_{\pm 0.011}$	$0.308_{\pm 0.013}$	$0.327_{\pm 0.019}$	$0.368_{\pm 0.013}$	$0.177_{\pm 0.095}$
	MCP → LepskiCP	$0.453_{\pm 0.012}$	$0.449_{\pm 0.025}$	$0.470_{\pm 0.016}$	$0.421_{\pm 0.009}$	$0.177_{\pm 0.095}$
	Online Betting → VCP	$0.293_{\pm 0.004}$	$0.342_{\pm 0.007}$	$0.347_{\pm 0.005}$	$0.378_{\pm 0.012}$	$0.090_{\pm 0.020}$
	Online Betting → LepskiCP	$0.383_{\pm 0.011}$	$0.424_{\pm 0.006}$	$0.448_{\pm 0.007}$	$0.412_{\pm 0.002}$	$0.090_{\pm 0.020}$
	CTM → VCP	$0.253_{\pm 0.025}$	$0.293_{\pm 0.005}$	$0.335_{\pm 0.005}$	$0.370_{\pm 0.011}$	$0.120_{\pm 0.075}$
	CTM → LepskiCP	$0.459_{\pm 0.016}$	$0.467_{\pm 0.019}$	$0.467_{\pm 0.016}$	$0.422_{\pm 0.003}$	$0.120_{\pm 0.075}$
	CoMa-LD	$0.187_{\pm 0.012}$	$0.198_{\pm 0.013}$	$0.217_{\pm 0.011}$	$\mathbf{0.271_{\pm 0.033}}$	$0.043_{\pm 0.023}$
	CoMa-LD$_{\mathrm{DC}}$	$\mathbf{0.158_{\pm 0.011}}$	$\mathbf{0.187_{\pm 0.006}}$	$\mathbf{0.211_{\pm 0.006}}$	$0.284_{\pm 0.019}$	$\mathbf{0.017_{\pm 0.021}}$

Table 9: Detection and localization diagnostics for detector–localizer cascades and CoMa-LD.

	Method	Human-only FPR ↓	Mixed documents Miss rate ↓	No-CP rate ↓	WD ↓
WritingPrompts	MCP → VCP	$0.050_{\pm 0.017}$	$0.839_{\pm 0.016}$	$0.917_{\pm 0.021}$	$0.379_{\pm 0.003}$
	MCP → LepskiCP	$0.050_{\pm 0.017}$	$0.839_{\pm 0.016}$	$0.910_{\pm 0.007}$	$0.400_{\pm 0.003}$
	Online Betting → VCP	$0.060_{\pm 0.017}$	$0.687_{\pm 0.028}$	$0.864_{\pm 0.029}$	$0.374_{\pm 0.004}$
	Online Betting → LepskiCP	$0.060_{\pm 0.017}$	$0.687_{\pm 0.028}$	$0.837_{\pm 0.019}$	$0.405_{\pm 0.003}$
	CTM → VCP	$0.063_{\pm 0.015}$	$0.561_{\pm 0.021}$	$0.815_{\pm 0.020}$	$0.366_{\pm 0.004}$
	CTM → LepskiCP	$0.063_{\pm 0.015}$	$0.561_{\pm 0.021}$	$0.765_{\pm 0.008}$	$0.425_{\pm 0.005}$
	CoMa-LD	$0.080_{\pm 0.017}$	$\mathbf{0.270_{\pm 0.021}}$	$\mathbf{0.276_{\pm 0.019}}$	$\mathbf{0.289_{\pm 0.006}}$
	CoMa-LD$_{\mathrm{DC}}$	$\mathbf{0.040_{\pm 0.017}}$	$0.303_{\pm 0.028}$	$0.309_{\pm 0.028}$	$\mathbf{0.289_{\pm 0.005}}$
CNN/DailyMail	MCP → VCP	$0.177_{\pm 0.095}$	$0.313_{\pm 0.117}$	$0.565_{\pm 0.022}$	$0.309_{\pm 0.007}$
	MCP → LepskiCP	$0.177_{\pm 0.095}$	$0.313_{\pm 0.117}$	$0.624_{\pm 0.076}$	$0.448_{\pm 0.015}$
	Online Betting → VCP	$0.090_{\pm 0.020}$	$0.613_{\pm 0.086}$	$0.754_{\pm 0.041}$	$0.340_{\pm 0.005}$
	Online Betting → LepskiCP	$0.090_{\pm 0.020}$	$0.613_{\pm 0.086}$	$0.785_{\pm 0.063}$	$0.417_{\pm 0.005}$
	CTM → VCP	$0.120_{\pm 0.075}$	$0.308_{\pm 0.146}$	$0.569_{\pm 0.060}$	$0.313_{\pm 0.005}$
	CTM → LepskiCP	$0.120_{\pm 0.075}$	$0.308_{\pm 0.146}$	$0.644_{\pm 0.103}$	$0.454_{\pm 0.013}$
	CoMa-LD	$0.043_{\pm 0.023}$	$\mathbf{0.141_{\pm 0.073}}$	$\mathbf{0.145_{\pm 0.070}}$	$0.218_{\pm 0.014}$
	CoMa-LD$_{\mathrm{DC}}$	$\mathbf{0.017_{\pm 0.021}}$	$0.229_{\pm 0.042}$	$0.229_{\pm 0.042}$	$\mathbf{0.210_{\pm 0.009}}$

variant achieve lower WD than all evaluated baselines except on WritingPrompts with $K = 1$, where GCP performs best. Both variants also yield lower human-only FPR than all evaluated baselines. Document-level calibration further reduces false alarms on both datasets, while improving WD on WritingPrompts but increasing it on CNN/DailyMail. These results show that CoMa-LD remains effective beyond Fast-DetectGPT scores.

G.3 Additional model combinations

We evaluate an additional black-box setting using Llama-3.2-3B-Instruct as the generator and a shared scoring model from the AdaDetectGPT-based NFT implementation (Li et al., 2026; Zhou et al., 2025). As shown in Table 11, CoMa-LD achieves the lowest WD on WritingPrompts for $K = 1, 2, 3$ and on CNN/DailyMail for $K = 2$, while both variants yield lower human-only FPRs than the evaluated baselines. CoMa-LD$_{\mathrm{DC}}$ further lowers FPR on both datasets but has higher WD than CoMa-LD across all transition settings, indicating a trade-off between false alarms and localization performance in this model combination.

Table 10: Black-box results with NLL-based scores. DetectGPT-Seg and FastDetectGPT-Seg retain their original scores. Bold and underlined values indicate the best and second-best results within each dataset, respectively.

		Mixed documents: WD ↓				Human-only: FPR ↓
	Method	$K=1$	$K=2$	$K=3$	$K=5$	$K=0$
WritingPrompts	DetectGPT-Seg	$0.922_{\pm 0.016}$	$0.906_{\pm 0.010}$	$0.884_{\pm 0.007}$	$0.775_{\pm 0.028}$	$1.000_{\pm 0.000}$
	FastDetectGPT-Seg	$0.894_{\pm 0.007}$	$0.883_{\pm 0.008}$	$0.865_{\pm 0.009}$	$0.766_{\pm 0.028}$	$0.997_{\pm 0.006}$
	PaLD	$0.510_{\pm 0.012}$	$0.493_{\pm 0.004}$	$0.484_{\pm 0.006}$	$0.456_{\pm 0.008}$	$1.000_{\pm 0.000}$
	VCP	$0.299_{\pm 0.036}$	$0.336_{\pm 0.023}$	$0.379_{\pm 0.007}$	$0.404_{\pm 0.003}$	$0.593_{\pm 0.040}$
	WCP_{len}	$0.260_{\pm 0.002}$	$0.323_{\pm 0.007}$	$0.381_{\pm 0.010}$	$0.405_{\pm 0.003}$	$0.180_{\pm 0.052}$
	GCP	$\mathbf{0.222_{\pm 0.016}}$	$0.258_{\pm 0.006}$	$0.313_{\pm 0.005}$	$0.379_{\pm 0.001}$	$0.460_{\pm 0.044}$
	LepskiCP	$0.367_{\pm 0.008}$	$0.399_{\pm 0.012}$	$0.369_{\pm 0.009}$	$0.349_{\pm 0.007}$	$1.000_{\pm 0.000}$
	CoMa-LD	$0.242_{\pm 0.018}$	$\underline{0.248_{\pm 0.012}}$	$\underline{0.252_{\pm 0.005}}$	$\underline{0.271_{\pm 0.006}}$	$\underline{0.143_{\pm 0.021}}$
	CoMa-LD$_{\text{DC}}$	$\underline{0.227_{\pm 0.020}}$	$\mathbf{0.232_{\pm 0.005}}$	$\mathbf{0.251_{\pm 0.006}}$	$\mathbf{0.258_{\pm 0.006}}$	$\mathbf{0.073_{\pm 0.032}}$
CNN/DailyMail	DetectGPT-Seg	$0.939_{\pm 0.008}$	$0.910_{\pm 0.014}$	$0.852_{\pm 0.021}$	$0.723_{\pm 0.018}$	$1.000_{\pm 0.000}$
	FastDetectGPT-Seg	$0.833_{\pm 0.022}$	$0.801_{\pm 0.025}$	$0.739_{\pm 0.025}$	$0.637_{\pm 0.052}$	$1.000_{\pm 0.000}$
	PaLD	$0.504_{\pm 0.006}$	$0.488_{\pm 0.004}$	$0.468_{\pm 0.004}$	$0.439_{\pm 0.002}$	$1.000_{\pm 0.000}$
	VCP	$0.326_{\pm 0.027}$	$0.365_{\pm 0.011}$	$0.375_{\pm 0.016}$	$0.398_{\pm 0.007}$	$0.653_{\pm 0.085}$
	WCP_{len}	$0.289_{\pm 0.030}$	$0.351_{\pm 0.013}$	$0.372_{\pm 0.008}$	$0.389_{\pm 0.002}$	$0.247_{\pm 0.023}$
	GCP	$0.314_{\pm 0.011}$	$0.294_{\pm 0.022}$	$0.322_{\pm 0.017}$	$0.369_{\pm 0.017}$	$0.513_{\pm 0.078}$
	LepskiCP	$0.502_{\pm 0.026}$	$0.488_{\pm 0.023}$	$0.462_{\pm 0.017}$	$0.425_{\pm 0.014}$	$1.000_{\pm 0.000}$
	CoMa-LD	$\mathbf{0.221_{\pm 0.035}}$	$\mathbf{0.237_{\pm 0.034}}$	$\mathbf{0.250_{\pm 0.015}}$	$\mathbf{0.279_{\pm 0.020}}$	$\underline{0.140_{\pm 0.000}}$
	CoMa-LD$_{\text{DC}}$	$\underline{0.239_{\pm 0.025}}$	$\underline{0.271_{\pm 0.031}}$	$\underline{0.271_{\pm 0.001}}$	$\underline{0.325_{\pm 0.006}}$	$\mathbf{0.030_{\pm 0.026}}$

Table 11: WD and false-alarm performance with Llama-3.2-3B-Instruct as the generator and NFT as the scoring model in the black-box setting. Bold and underlined values indicate the best and second-best results within each dataset.

		Mixed documents: WD ↓				Human-only: FPR ↓
	Method	$K=1$	$K=2$	$K=3$	$K=5$	$K=0$
WritingPrompts	DetectGPT-Seg	$0.428_{\pm 0.013}$	$0.470_{\pm 0.004}$	$0.477_{\pm 0.004}$	$0.454_{\pm 0.004}$	$1.000_{\pm 0.000}$
	FastDetectGPT-Seg	$0.579_{\pm 0.050}$	$0.540_{\pm 0.033}$	$0.505_{\pm 0.013}$	$0.398_{\pm 0.026}$	$0.890_{\pm 0.036}$
	PaLD	$0.768_{\pm 0.019}$	$0.709_{\pm 0.010}$	$0.679_{\pm 0.001}$	$0.615_{\pm 0.016}$	$1.000_{\pm 0.000}$
	VCP	$0.217_{\pm 0.036}$	$0.209_{\pm 0.013}$	$0.174_{\pm 0.005}$	$\underline{0.141_{\pm 0.008}}$	$0.847_{\pm 0.038}$
	WCP_{len}	$0.246_{\pm 0.050}$	$\underline{0.203_{\pm 0.012}}$	$\underline{0.160_{\pm 0.013}}$	$\mathbf{0.137_{\pm 0.014}}$	$0.903_{\pm 0.031}$
	GCP	$\underline{0.179_{\pm 0.028}}$	$0.206_{\pm 0.015}$	$0.200_{\pm 0.012}$	$0.209_{\pm 0.005}$	$0.853_{\pm 0.021}$
	LepskiCP	$0.454_{\pm 0.017}$	$0.445_{\pm 0.006}$	$0.471_{\pm 0.013}$	$0.447_{\pm 0.014}$	$0.330_{\pm 0.061}$
	CoMa-LD	$\mathbf{0.163_{\pm 0.014}}$	$\mathbf{0.154_{\pm 0.006}}$	$\mathbf{0.154_{\pm 0.005}}$	$0.150_{\pm 0.007}$	$\underline{0.120_{\pm 0.010}}$
	CoMa-LD$_{\text{DC}}$	$0.256_{\pm 0.008}$	$0.234_{\pm 0.012}$	$0.217_{\pm 0.002}$	$0.210_{\pm 0.003}$	$\mathbf{0.070_{\pm 0.010}}$
CNN/DailyMail	DetectGPT-Seg	$0.813_{\pm 0.004}$	$0.789_{\pm 0.001}$	$0.738_{\pm 0.002}$	$0.649_{\pm 0.001}$	$1.000_{\pm 0.000}$
	FastDetectGPT-Seg	$0.604_{\pm 0.011}$	$0.553_{\pm 0.002}$	$0.454_{\pm 0.019}$	$0.330_{\pm 0.015}$	$0.960_{\pm 0.020}$
	PaLD	$0.764_{\pm 0.007}$	$0.701_{\pm 0.019}$	$0.698_{\pm 0.001}$	$0.603_{\pm 0.012}$	$1.000_{\pm 0.000}$
	VCP	$0.281_{\pm 0.025}$	$0.255_{\pm 0.007}$	$0.199_{\pm 0.010}$	$\underline{0.170_{\pm 0.007}}$	$0.930_{\pm 0.044}$
	WCP_{len}	$0.274_{\pm 0.017}$	$0.226_{\pm 0.014}$	$\mathbf{0.171_{\pm 0.009}}$	$\mathbf{0.141_{\pm 0.008}}$	$0.920_{\pm 0.010}$
	GCP	$\mathbf{0.175_{\pm 0.007}}$	$0.216_{\pm 0.021}$	$0.193_{\pm 0.002}$	$0.213_{\pm 0.005}$	$0.867_{\pm 0.006}$
	LepskiCP	$0.475_{\pm 0.021}$	$0.446_{\pm 0.012}$	$0.454_{\pm 0.014}$	$0.426_{\pm 0.007}$	$0.290_{\pm 0.030}$
	CoMa-LD	$\underline{0.219_{\pm 0.023}}$	$\mathbf{0.201_{\pm 0.008}}$	$\underline{0.182_{\pm 0.010}}$	$0.175_{\pm 0.003}$	$\underline{0.037_{\pm 0.015}}$
	CoMa-LD$_{\text{DC}}$	$0.235_{\pm 0.017}$	$\underline{0.209_{\pm 0.009}}$	$0.202_{\pm 0.005}$	$0.189_{\pm 0.011}$	$\mathbf{0.033_{\pm 0.006}}$

G.4 Additional sentence-level classification results

We evaluate sentence-level authorship classification for $K = 3$ and $K = 5$ using the same models and scoring function as in the main experiments (Tables 12 and 13). Both CoMa-LD variants achieve higher precision than the evaluated baselines across both datasets and transition settings. On CNN/DailyMail, CoMa-LD achieves the highest F1 in both settings, while CoMa-LD$_{\text{DC}}$ attains higher precision but lower recall and F1. On WritingPrompts, LepskiCP achieves the highest recall in both settings, while the highest F1 is attained by CoMa-LD$_{\text{DC}}$ for $K = 3$ and LepskiCP for $K = 5$.

Table 12: Sentence-level precision, recall, and F1 for the LLM class with $K = 3$.

	WritingPrompts			CNN/DailyMail		
Method	Precision ↑	Recall ↑	F1 ↑	Precision ↑	Recall ↑	F1 ↑
VCP	$0.623_{\pm 0.004}$	$0.346_{\pm 0.054}$	$0.443_{\pm 0.046}$	$0.692_{\pm 0.033}$	$0.543_{\pm 0.112}$	$0.606_{\pm 0.081}$
WCP_{len}	$0.619_{\pm 0.004}$	$0.557_{\pm 0.020}$	$0.586_{\pm 0.011}$	$0.694_{\pm 0.040}$	$0.657_{\pm 0.088}$	$0.674_{\pm 0.065}$
LepskiCP	$0.642_{\pm 0.021}$	$\mathbf{0.798_{\pm 0.024}}$	$0.711_{\pm 0.003}$	$0.691_{\pm 0.006}$	$0.695_{\pm 0.059}$	$0.692_{\pm 0.027}$
CoMa-LD	$\mathbf{0.829_{\pm 0.015}}$	$0.671_{\pm 0.060}$	$\underline{0.741_{\pm 0.041}}$	$\underline{0.853_{\pm 0.011}}$	$\mathbf{0.840_{\pm 0.067}}$	$\mathbf{0.845_{\pm 0.030}}$
CoMa-LD$_{\text{DC}}$	$\underline{0.814_{\pm 0.001}}$	$\underline{0.692_{\pm 0.055}}$	$\mathbf{0.747_{\pm 0.033}}$	$\mathbf{0.913_{\pm 0.032}}$	$\underline{0.733_{\pm 0.038}}$	$\underline{0.812_{\pm 0.011}}$

Table 13: Sentence-level precision, recall, and F1 for the LLM class with $K = 5$.

	WritingPrompts			CNN/DailyMail		
Method	Precision ↑	Recall ↑	F1 ↑	Precision ↑	Recall ↑	F1 ↑
VCP	$0.566_{\pm 0.012}$	$0.327_{\pm 0.023}$	$0.415_{\pm 0.022}$	$0.604_{\pm 0.017}$	$0.396_{\pm 0.135}$	$0.472_{\pm 0.099}$
WCP_{len}	$0.573_{\pm 0.006}$	$0.514_{\pm 0.027}$	$0.541_{\pm 0.013}$	$0.594_{\pm 0.037}$	$0.560_{\pm 0.058}$	$0.576_{\pm 0.048}$
LepskiCP	$0.643_{\pm 0.012}$	$\mathbf{0.793_{\pm 0.021}}$	$\mathbf{0.710_{\pm 0.003}}$	$0.672_{\pm 0.039}$	$\underline{0.633_{\pm 0.071}}$	$0.649_{\pm 0.029}$
CoMa-LD	$\mathbf{0.801_{\pm 0.003}}$	$0.615_{\pm 0.019}$	$0.696_{\pm 0.013}$	$\underline{0.782_{\pm 0.012}}$	$\mathbf{0.708_{\pm 0.129}}$	$\mathbf{0.740_{\pm 0.072}}$
CoMa-LD$_{\text{DC}}$	$\underline{0.785_{\pm 0.013}}$	$\underline{0.641_{\pm 0.035}}$	$\underline{0.705_{\pm 0.017}}$	$\mathbf{0.851_{\pm 0.061}}$	$0.546_{\pm 0.035}$	$\underline{0.664_{\pm 0.018}}$

G.5 Case study

Figure 5 presents eight mixed-document examples from WritingPrompts and CNN/DailyMail. FastDetectGPT-Seg and PaLD often produce fragmented predictions, while several other baselines miss entire segments or merge neighboring segments. CoMa-LD more closely follows the ground-truth segmentation in several examples, including the CNN/DailyMail document with two transitions. In the CNN/DailyMail example with three transitions, CoMa-LD misses a transition and places the remaining boundaries away from their true positions. The bottom-right WritingPrompts example also contains an extra predicted transition.

H Limitations

Like MCP (Zhu et al., 2025) and Online Betting (Chen & Wang, 2025), CoMa-LD relies on additional human-written reference data, which our framework uses for calibration. Its localization operates at the sentence level, as in PaLD (Lei et al., 2025), and therefore cannot resolve authorship changes within a sentence. Future work could reduce calibration-data requirements and extend localization to finer units, such as clauses or tokens, while preserving document-level false-alarm control.

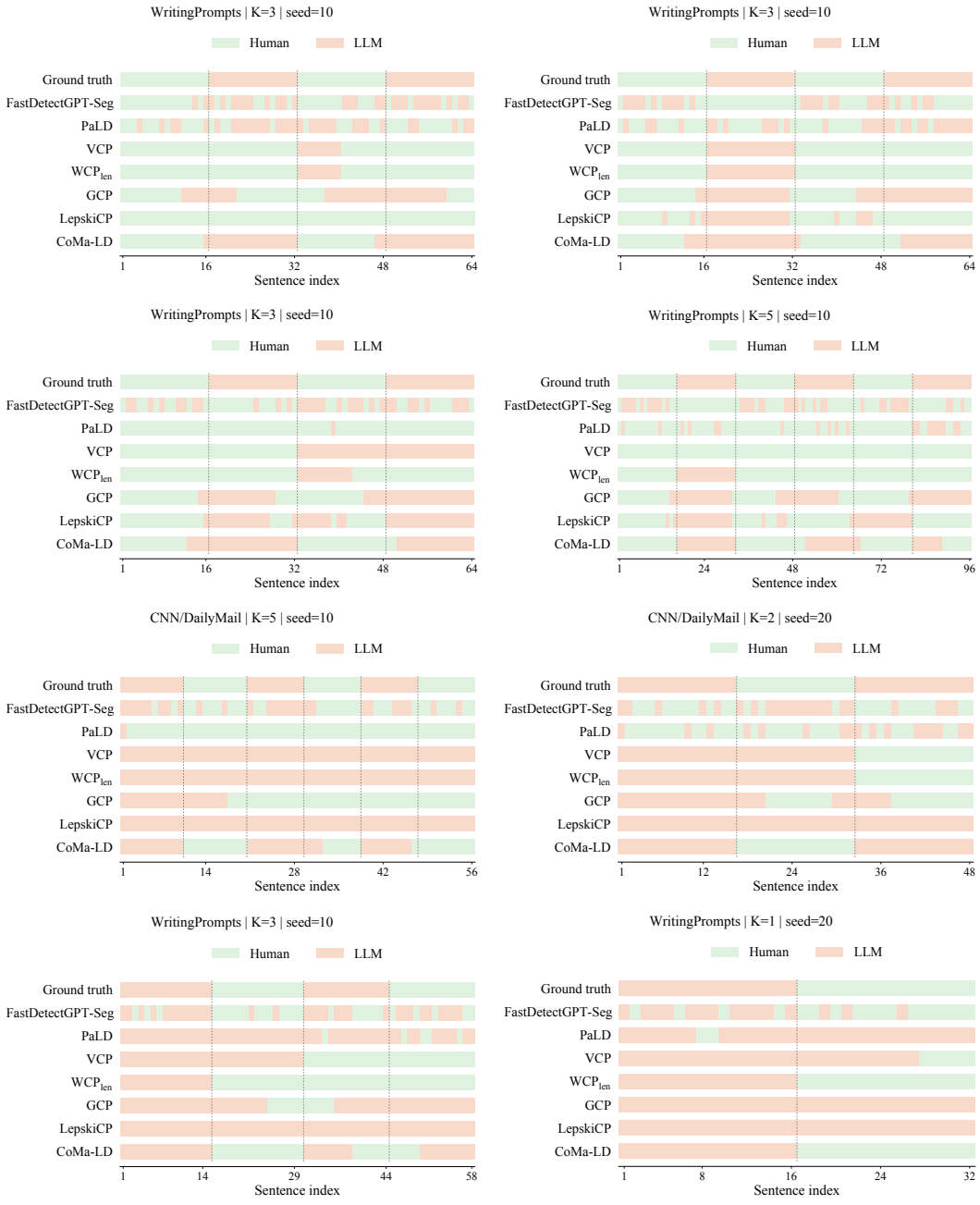


Figure 5: Sentence-level localization examples for eight mixed documents from WritingPrompts and CNN/DailyMail, comparing CoMa-LD with the baselines. In the ground-truth rows, green and orange denote Human and LLM labels, respectively. Dashed lines mark ground-truth boundaries. CoMa-LD uses the Ville threshold.