SEMASYNC: ROBUST SEMANTIC-STATE WATERMARKING FOR STRUCTURED LLM GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) increasingly generate structured outputs such as mathematical solutions and software code, creating a growing need for reliable provenance verification. Watermarking these outputs faces two challenges: rewriting can disrupt token-dependent watermark states and weaken detection, while indiscriminate sampling biases can compromise correctness. We introduce SemaSync, a watermarking framework that uses semantic and structural anchors to support synchronization after rewriting. Multi-anchor voting provides redundancy when individual anchors change, while distribution-aware selective embedding balances watermark detectability with output quality. We establish a margin-based partition stability guarantee and use ablations to clarify the roles of state construction, selective embedding, and multi-anchor voting. Across GSM8K and MBPP with Qwen3.5-9B and Llama-3.1-8B-Instruct, SemaSync achieves higher AUROC on unmodified outputs than all evaluated baselines in all settings. On Qwen GSM8K, SemaSync improves TPR at 1% FPR by 20.1% on unmodified outputs and 44.2% after Mistral rewriting relative to the strongest baseline in each condition with moderate accuracy loss.

1 INTRODUCTION

Large language models (LLMs) are increasingly used in scientific, educational, and software development workflows (Liu et al., 2023), including the generation of mathematical solutions (Oum, 2026; Gao, 2026; Pei et al., 2025; Liu et al., 2026) and software code (Cao et al., 2026; Zhuo et al., 2025; Qiu et al., 2025; Wang et al., 2026). Reliable provenance verification helps determine whether these artifacts originated from a designated generation process.

Watermarking provides an active approach to provenance verification: it embeds a hidden statistical signal during generation and tests for that signal in the observed output (Kirchenbauer et al., 2023; Kuditipudi et al., 2023). Subsequent work improves generation utility and robustness through distribution-preserving sampling, semantic representations, and selective embedding (Dathathri et al., 2024; Sander et al., 2026; Hou et al., 2024; Lee et al., 2024; Wang et al., 2025). Despite these advances, paraphrasing and rewriting can substantially weaken detection (Liu et al., 2024; Pan et al., 2025; Zhang et al., 2025). The problem is especially important for mathematical solutions and code: users can rephrase a derivation or refactor a program while preserving its intended result, yet even a small watermark-induced change to a number or operator may alter that result.

These settings expose two challenges. **First, the watermark state must remain reconstructable after rewriting.** Many token-level methods derive their states from previous token sequences, tying synchronization to the lexical generation trajectory. Rewriting tokens used to construct these states can disrupt alignment between generation and detection. Structured artifacts offer a more direct source of synchronization: numerical and structural events that remain observable when surrounding text changes. The difficulty is to turn these events into a deterministic watermark state that can tolerate the loss or substitution of individual events.

Second, embedding should account for the choices available under that state. Mathematical reasoning and code contain constrained positions where only a small set of continuations preserves correctness. Biasing such positions can disrupt the task, while biasing positions that already favor green tokens (the keyed subset favored by the watermark's sampling bias) may add little useful evidence. SWEET already selects high-entropy positions (Lee et al., 2024), and MorphMark adapts

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.

bias strength to green-list probability (Wang et al., 2025). The suitability of a generation step for watermarking depends on how its high-probability tokens are distributed across the current keyed partition.

We introduce **SemaSync**, a semantic-state watermarking framework for structured LLM generation. Its name reflects the central idea of *semantic synchronization*: using preserved content events to reconstruct watermark states after rewriting. SemaSync extracts canonical semantic and structural *anchors*, such as numerical events in a solution and literals or control-flow events in code. Recent anchors then vote through keyed masks to form the green partition, providing redundancy when some events are changed or removed. We give a margin-based guarantee that characterizes which partition decisions survive a bounded number of voter replacements.

SemaSync further uses a two-sided gate on the base model's green probability mass. If this mass is too low, a strong green preference would promote choices that the model considers atypical, potentially hurting correctness or producing evidence that rewriting removes. If it is too high, clean generation already favors green tokens, leaving little additional evidence for the bias to create. SemaSync therefore abstains at both extremes and applies a fixed bias at intermediate positions. This design gives each bound a clear role, which we evaluate by removing the bounds separately and together.

We evaluate SemaSync against the evaluated baselines on GSM8K and MBPP with Qwen3.5-9B and Llama-3.1-8B-Instruct. SemaSync attains the highest AUROC on unmodified (raw) outputs among these baselines in all settings. On Qwen GSM8K, its TPR at 1% FPR reaches 0.920, compared with 0.766 for SWEET, the strongest baseline in this condition. After Mistral rewriting, SemaSync achieves a TPR of 0.248, compared with 0.172 for the strongest baseline, KGW. These detection gains are achieved with 92.8% generation accuracy, compared with 94.4% for unwatermarked generation, jointly ranking third among the nine watermarking configurations in this setting. Experiments with a second rewriter further assess rewrite resilience. Under the same selective-embedding settings, semantic states yield higher AUROC and TPR after Mistral rewriting than previous-token states on both tasks (Section 4.4). We also examine strongly compressed rewrites and sparse-anchor settings, where limited surviving evidence makes detection more difficult. Our code is available for anonymous review.[1]

Our main contributions are:

- We propose an anchor-derived watermark-state construction using deterministic semantic and structural events and keyed multi-anchor voting. We establish a margin-based stability guarantee under bounded voter replacement.
- We develop a distribution-aware carrier rule that abstains at both extremes of green probability mass. We explain the role of each bound and test their effects on detection, rewrite resilience, and task utility.
- We evaluate SemaSync on mathematical reasoning and code generation with two generators and multiple rewriters, and use ablations to examine state construction, the carrier gate, and multi-anchor voting. An open-domain extension examines the limits of anchor coverage.

2 Preliminaries

2.1 Related Work

Watermark states and robustness. KGW (Kirchenbauer et al., 2023) derives a keyed green–red vocabulary partition from token history and biases sampling toward green tokens. Its state can desynchronize after lexical changes. Other work studies robustness to content modification (Kuditipudi et al., 2023). Semantic approaches reduce dependence on exact wording through content representations (Hou et al., 2024; Ren et al., 2024; Liu et al., 2024). SIR (Liu et al., 2024) derives watermark biases from semantic representations. SemStamp uses sentence embeddings, locality-sensitive hashing, and rejection sampling to enforce valid transitions, adding sentence-level computation. Unigram (Zhao et al., 2024) removes synchronization dependence through a fixed partition,

[1] https://anonymous.4open.science/r/SemaSync-AEF8/

but exposes a single green list that can be estimated from watermarked samples (Pan et al., 2025). SemaSync retains content-dependent states through deterministic anchors and uses voting to tolerate changes to individual events.

Generation utility. SynthID-Text (Dathathri et al., 2024), unbiased watermarking (Hu et al., 2024), DiPmark (Wu et al., 2023), and TextSeal (Sander et al., 2026) preserve the expected sampling distribution while retaining detectability. Permute-and-Flip (Zhao et al., 2025) develops a decoder with a quality–stability guarantee and a watermark that preserves its sampling distribution. These approaches address the utility cost of watermarking, which matters especially when small changes affect numerical or program correctness.

Selective embedding and detection. SWEET (Lee et al., 2024) selects high-entropy positions, MorphMark (Wang et al., 2025) increases its bias with green-list probability, and EWD (Lu et al., 2024) weights detection evidence by entropy. STONE (Kim et al., 2026) preserves syntax tokens and embeds in non-syntactic code tokens. SemaSync uses a fixed bias and skips both extremes of green mass: the lower bound avoids promoting unlikely choices, while the upper bound excludes positions where clean generation already favors green tokens. Section 3.3 explains the rule and Section 4.5 tests each bound. Main comparisons evaluate complete systems; the gate ablation isolates selection within SemaSync.

2.2 LLM Watermarking Background

Given prompt x and completion $y = (y_1, \ldots, y_T)$, an autoregressive model produces $p_t(v) = P(y_t = v \mid x, y_{<t})$ for token $v \in \mathcal{V}$. A watermark state $S_t = f(y_{<t})$ and secret key k determine a green set $G_t = g(S_t, k)$. KGW adds bias δ to its logits: $\ell'_t(v) = \ell_t(v) + \delta \mathbb{I}(v \in G_t)$. The detector reconstructs the partitions and counts green tokens N_G. With nominal green ratio γ, its score is $z = (N_G - \gamma T)/\sqrt{T\gamma(1-\gamma)}$.

2.3 Problem Formulation

We consider two objectives. **Utility-preserving embedding** aims to embed detectable evidence while maintaining the task utility of constrained outputs. **Transformation-resilient verification** retains evidence after a meaning-preserving transformation τ. Specifically, if $D_k(x, y)$ is the detection score, $D_k(x, \tau(y))$ should remain distinguishable from scores on matched clean text. Token-history-based states $S_t = f(y_{<t})$ can change when the tokens used to construct them are rewritten, disrupting alignment between generation and detection. SemaSync instead derives states from canonical content events that can be recovered from the transformed output.

3 Methodology

3.1 Overview

SemaSync constructs a watermark state from visible content events and evaluates whether the next position can carry useful evidence. Figure 1 shows three operations: extract canonical anchors from completed events (①); aggregate their keyed votes into a green partition (②); and apply a fixed bias when the partition's base probability mass lies inside the carrier interval (③).

During detection, SemaSync reconstructs the anchor states, green partitions, and carrier decisions from the observed completion using the original key, and frozen model–tokenizer pair. It then scores the selected tokens without a recorded generation-time trace. Preserved anchors support synchronization after rewriting; voting provides redundancy when individual events change.

3.2 Anchor-based Watermark State Construction

3.2.1 Anchor Extraction

A deterministic extractor $\mathcal{E}$ maps a completion to ordered anchor occurrences, $\mathcal{E}(y) = [(a_1, p_1), \ldots, (a_n, p_n)]$. Each identity $a_i = (\tau_i, \pi_i)$ contains an event type and canonical payload; p_i is the token position where the event is complete. Mathematical anchors represent numerical,

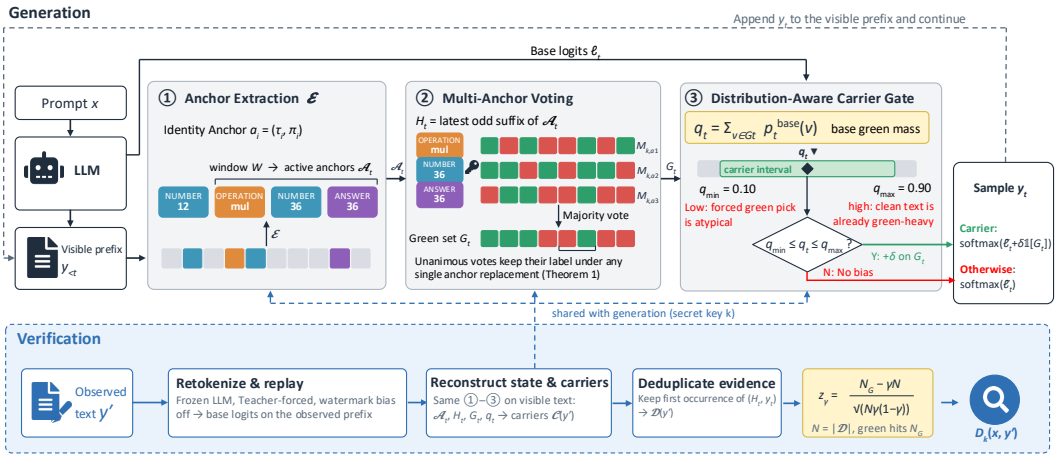

Figure 1: Overview of SemaSync. Generation (top) builds the watermark state from visible anchors (①), multi-anchor voting (②), and a distribution-aware carrier gate (③). Verification (bottom) reconstructs the same three modules from the observed text and requires no generation-side metadata.

operational, and unit-related events. Code anchors represent literals, operators, branches, loops, and returns. Canonicalization maps supported surface variants of an event to the same anchor identity. This does not require all anchors to survive. Appendix A gives the task-specific ontologies, held-out stability analysis, and frozen event sets; Appendix C studies named-entity and acronym anchors in prose. At position t, the active sequence is $\mathcal{A}_t = [a_i \mid p_i < t,\ t - p_i \le W]$, where W bounds the anchor's lifetime in completion tokens.

3.2.2 Multi-Anchor Voting for Partition Stability

A single-anchor state depends entirely on one event. SemaSync instead selects the longest odd-length suffix of $\mathcal{A}_t$ with at most K anchors, where K is a positive odd integer. Write this sequence as H_t and its length as m_t. If no anchor is active, $H_t = \varnothing$ and semantic-state embedding is inactive.

For each anchor, a keyed pseudorandom function assigns token v a vote $M_{k,a_i}(v) \in \{0, 1\}$. We use green ratio $\gamma = 1/2$. When H_t is nonempty, the majority partition is

$$G_t = \left\{ v \in \mathcal{V} : \sum_{a_i \in H_t} M_{k,a_i}(v) \ge \frac{m_t + 1}{2} \right\}.$$

Define $b_i(v) = 2M_{k,a_i}(v) - 1$ and margin $\Delta_t(v) = |\sum_{a_i \in H_t} b_i(v)|$. A token is r-replacement stable if every replacement of at most r voters leaves its partition label unchanged, allowing arbitrary replacement votes.

Theorem 1 (Margin-based partition stability)**.** *For an odd number of voters, a token v is r-replacement stable if and only if $\Delta_t(v) > 2r$.*

Each replaced voter changes the signed sum by at most two, so a margin larger than $2r$ protects the label. The full proof is in Appendix B. Under the idealized independent-mask model, the expected fraction of protected labels is nondecreasing with the odd voter count for a fixed replacement budget. For example, three unanimous votes survive one replacement, while a single vote cannot.

Initialization and task scope. We use task-specific extractors for mathematical solutions and Python code. MBPP forms three-anchor states relatively late (Table 11), motivating an initializer for the prefix before structural anchors become available. Until the first structural anchor completes, a domain-separated pseudo-anchor from the previous ordinary token supplies the state, using the last ordinary prompt token at the first position. From the next position after that anchor completes, SemaSync permanently switches to semantic states; an empty anchor window disables embedding. Both phases use the same gate and bias, and detection follows the same transition (Appendix D.2).

3.3 Distribution-Aware Carrier Selection and Watermark Embedding

A recoverable partition does not make every position a useful carrier. We first compute the clean sampling distribution p_t^{base}: apply temperature ϑ to the base logits, retain the top-p nucleus at threshold η, and renormalize. The *base green probability mass*, hereafter *green mass*, is $q_t = \sum_{v \in G_t} p_t^{\text{base}}(v)$.

Why skip both extremes? With fixed sampling support, a green-token bias δ changes green probability to $q_t^{(\delta)} = \frac{e^\delta q_t}{1 - q_t + e^\delta q_t}$, $\quad \Delta q_t = \frac{(e^\delta - 1) q_t (1 - q_t)}{1 + (e^\delta - 1) q_t}$. At low q_t, obtaining substantial green evidence requires promoting choices the clean model considers unlikely. Such choices may hurt correctness or be removed by rewriting. At high q_t, clean text already favors green tokens, leaving little extra probability for the bias to add. Under the nominal-γ score, these positions can also contribute positive evidence on clean text. The two extremes thus give different reasons to abstain. This calculation motivates the gate; it does not assert that a particular interval is optimal. Section 4.5 tests the interval empirically.

For an available state, the carrier indicator is $c_t = \mathbb{I}(q_{\min} \le q_t \le q_{\max})$; without a state, $c_t = 0$. Here an available state includes the MBPP initializer described above. SemaSync applies $\ell'_t(v) = \ell_t(v)/\vartheta + c_t \delta \mathbb{I}(v \in G_t)$. It then recomputes the top-p nucleus and samples the next token. Thus the gate reads the unwatermarked distribution, while sampling uses the biased distribution. The default interval is $[0.10, 0.90]$.

3.4 Watermark Detection

Given the prompt x, visible completion y, key k, and frozen model/tokenizer pair, the detector returns $D_k(x, y)$; we omit the model and tokenizer from the notation. It retokenizes y and teacher-forces the original model on the observed prefix to reconstruct the states, green partitions, base green masses, and carrier decisions. Let $\mathcal{C}(y)$ contain eligible nonspecial completion positions where this rule is active, and let $X_t = \mathbb{I}(y_t \in G_t)$.

Repeated state–token pairs can overcount local patterns (Fernandez et al., 2023; Kirchenbauer et al., 2023; Sander et al., 2024). We define $\kappa_t = (H_t, y_t)$, retaining anchor order and multiplicity, and keep only its first occurrence among carriers: $\mathcal{D}(y) = \{t \in \mathcal{C}(y) : \kappa_t \notin \{\kappa_s : s < t,\ s \in \mathcal{C}(y)\}\}$. With $N = |\mathcal{D}(y)|$ and $N_G = \sum_{t \in \mathcal{D}(y)} X_t$, the score is $D_k(x, y) = \frac{N_G - \gamma N}{\sqrt{N\gamma(1-\gamma)}}$. Larger scores indicate stronger evidence of the keyed watermark. Evaluation thresholds use the matched clean population as described in Section 4.1.

4 Experiments

We test the overall detection–utility tradeoff, resilience across rewriters, the two main design choices, and sensitivity to controlled anchor corruption.

4.1 Experimental Setup

Tasks and generators. We evaluate GSM8K mathematical reasoning (final-answer accuracy) and MBPP Python synthesis (Pass@1) with Qwen3.5-9B and Llama-3.1-8B-Instruct, using 500 watermarked and 500 matched clean outputs per task and condition. Ablations use Qwen; C4 provides an open-domain extension (Appendix C).

Baselines and settings. We compare KGW (Kirchenbauer et al., 2023), SWEET (Lee et al., 2024), MorphMark (Wang et al., 2025), MorphMark+EWD (Wang et al., 2025; Lu et al., 2024), SemStamp (Hou et al., 2024), TextSeal (Sander et al., 2026), Unigram (Zhao et al., 2024), and SIR (Liu et al., 2024); MBPP also includes STONE (Kim et al., 2026). SemaSync uses $K{=}3$, $W{=}64$, $\gamma{=}1/2$, $\delta{=}3$, and $q_t \in [0.10, 0.90]$, with the MBPP initializer in Section 3.2.

Rewriting and metrics. Mistral-Small-3.2-24B-Instruct is the primary rewriter; GPT-5.6 Luna Max uses the same prompts and acceptance rules. We report AUROC and TPR at 1% FPR, calibrating thresholds on matched clean outputs in each condition, including rewritten clean text. Appendix D provides configurations, prompts, and fallback rules; Appendix D.5 reports paired bootstrap intervals (2000 replicates) for the main Qwen results.

4.2 Detectability, Rewrite Resilience, and Utility

Table 1: Main Qwen3.5-9B results. Detection uses raw outputs and Mistral rewrites. Additional TPR levels and confidence intervals are in Appendix D.5.

GSM8K

	Raw		Mistral rewrite		Utility
Method	AUROC	TPR@1%	AUROC	TPR@1%	Accuracy
Clean	–	–	–	–	.944
KGW	.897	.462	.709	.172	.910
SWEET	.977	.766	.729	.136	.932
MorphMark	.709	.126	.583	.078	.924
MorphMark+EWD	.763	.172	.613	.084	.924
SemStamp	.797	.190	.654	.080	.920
TextSeal	.764	.174	.628	.070	**.938**
Unigram	.890	.400	.702	.112	.926
SIR	.751	.108	.665	.056	.928
SemaSync	**.995**	**.920**	**.786**	**.248**	.928

MBPP

	Raw		Mistral rewrite		Pass@1	
Method	AUROC	TPR@1%	AUROC	TPR@1%	Raw	Rewritten
Clean	–	–	–	–	.560	.532
KGW	.823	**.368**	.637	.034	.300	.326
SWEET	.860	.322	.709	.094	.554	.546
MorphMark	.606	.056	.561	.028	.580	.552
MorphMark+EWD	.786	.160	.640	.042	.580	.552
SemStamp	.675	.046	.509	.002	.542	.524
TextSeal	.680	.120	.584	.036	**.584**	.566
Unigram	.777	.240	**.710**	.108	.300	.294
SIR	.807	.112	.592	.012	.554	**.610**
STONE	.625	.056	.576	.038	.528	.518
SemaSync	**.884**	.344	.694	**.110**	.538	.526

Raw detection. SemaSync has the highest AUROC in all four task–generator settings (Tables 1–2). On Qwen GSM8K, its TPR@1% is 0.920, compared with 0.766 for SWEET and 0.462 for KGW. On Llama MBPP, it reaches 0.320, compared with 0.136 for the strongest baseline.

Mistral rewriting. SemaSync has the highest rewritten AUROC in two of four settings. On Qwen GSM8K, its AUROC is 0.786 and TPR is 0.248, versus 0.729 and 0.172 for the strongest baseline in each metric. On Qwen MBPP, Unigram and SWEET have higher AUROC (0.710 and 0.709 vs. 0.694), while SemaSync has the highest TPR (0.110). On Llama GSM8K, Unigram and SIR achieve higher AUROC and TPR, but their generation accuracy is more than 10 percentage points lower than SemaSync's.

Task utility. On Qwen, SemaSync changes GSM8K accuracy from 0.944 to 0.928 and MBPP Pass@1 from 0.560 to 0.538. On Llama, GSM8K accuracy decreases from 0.850 to 0.776, while MBPP Pass@1 is 0.554, compared with 0.552 for clean generation. Utility costs therefore vary across tasks and generators.

4.3 Additional Rewriters and a Destructive Stress Test

Second rewriter. Luna compresses GSM8K solutions more strongly than Mistral (median length retention: 57% versus 95%). SemaSync's AUROC falls to 0.542 on GSM8K but remains 0.690 on MBPP, among the leading methods; paired intervals do not establish a difference between these leaders. These results support resilience on code while showing the limits of detection under strong mathematical-solution compression (Appendix D.7).

Destructive stress. DIPPER (Krishna et al., 2023) substantially degrades answer accuracy and code correctness. We therefore treat it as a destructive stress test; detection and utility results are in Appendix D.8.

Table 2: Main Llama-3.1-8B-Instruct results under the same protocol as Table 1. Additional TPR levels are in Appendix D.6.

GSM8K

	Raw		Mistral rewrite		Utility
Method	AUROC	TPR@1%	AUROC	TPR@1%	Accuracy
Clean (no watermark)	–	–	–	–	.850
KGW	.979	.788	.709	.216	.722
SWEET	.996	.908	.668	.150	.798
MorphMark	.830	.230	.615	.090	.826
MorphMark+EWD	.873	.350	.611	.076	.826
SemStamp	.864	.256	.624	.112	.800
TextSeal	.881	.444	.602	.046	**.850**
Unigram	.964	.676	**.779**	**.356**	.638
SIR	.909	.490	.754	.270	.670
SemaSync	**.999**	**.982**	.740	.186	.776

MBPP

	Raw		Mistral rewrite		Pass@1	
Method	AUROC	TPR@1%	AUROC	TPR@1%	Raw	Rewritten
Clean (no watermark)	–	–	–	–	.552	.520
KGW	.727	.080	.628	.052	.510	.500
SWEET	.845	.136	.645	.068	.526	.492
MorphMark	.598	.064	.552	.024	**.568**	.524
MorphMark+EWD	.713	.094	.579	.028	.568	.524
SemStamp	.584	.000	.536	.000	.546	.518
TextSeal	.638	.056	.557	.036	.552	.528
Unigram	.757	.100	.618	.036	.524	.508
SIR	.642	.050	.557	.024	.496	.480
STONE	.572	.054	.546	.030	.506	.480
SemaSync	**.855**	**.320**	**.669**	**.070**	.554	**.544**

4.4 Ablation: State Source

This ablation asks whether semantic states improve detection after rewriting beyond what the two-sided gate achieves with token-history states. We compare two settings: *Token + Gate*, which always constructs its state from the previous ordinary token, and *SemaSync*, which constructs states from semantic anchors with up to three voters. On MBPP, SemaSync initially uses the same token-based state, then switches to semantic states after the first anchor completes; Token + Gate continues using token states throughout. Under otherwise identical settings, we compare the two state constructions. We evaluate both raw outputs and Mistral rewrites to distinguish initial detectability from detection after rewriting, and report generation utility alongside detection (Table 3).

Token + Gate has higher raw AUROC on both tasks. After rewriting, SemaSync has higher AUROC and TPR: AUROC increases from 0.751 to 0.786 on GSM8K and from 0.682 to 0.694 on MBPP. MBPP Pass@1 is identical (0.538), while GSM8K accuracy is 0.928 for SemaSync versus 0.920 for Token + Gate. These results support a benefit from semantic-state construction after rewriting beyond the gate alone, with a smaller observed gain on MBPP. Appendix D.12 provides the full protocol and separately examines the MBPP initializer through removal and channel-wise rescoring.

4.5 Ablation: Distribution-Aware Carrier Gate

We ablate each bound separately and jointly, and evaluate a narrower interval while keeping the remaining design fixed (Table 4). The default gate gives the highest raw and rewritten AUROC on both tasks. On MBPP, admitting high-mass positions (lower bound only) yields similar Pass@1 (0.542 versus 0.538) but lowers raw AUROC to 0.588. Admitting low-mass positions (upper bound only) lowers Pass@1 to 0.464. These results are consistent with the motivation for the two bounds: the lower bound helps avoid costly choices, while the upper bound removes positions with limited discrimination. On GSM8K, removing either bound reduces both accuracy and detection.

Table 3: State-source ablation on Qwen3.5-9B. Utility is original-generation accuracy (GSM8K) or Pass@1 (MBPP); rewriting uses Mistral.

		Raw	Mistral rewrite		
Task	Configuration	AUROC	AUROC	TPR@1%	Utility
GSM8K	Token + Gate	**.997**	.751	.204	.920
	SemaSync	.995	**.786**	**.248**	**.928**
MBPP	Token + Gate	**.898**	.682	.102	**.538**
	SemaSync	.884	**.694**	**.110**	**.538**

Table 4: Carrier-gate ablation on Qwen3.5-9B.

		Raw		Mistral rewrite		
Dataset	Gate on q_t	AUROC	TPR@1%	AUROC	TPR@1%	Raw utility
MBPP	none	.734	.220	.637	**.114**	.466
	$q_t \geq 0.10$ only	.588	.044	.563	.044	.542
	$q_t \leq 0.90$ only	.793	.270	.626	.088	.464
	$[0.10, 0.90]$ (default)	**.884**	**.344**	**.694**	.110	.538
	$[0.20, 0.80]$	.851	.280	.666	.066	**.546**
GSM8K	none	.788	.186	.644	.086	.906
	$q_t \geq 0.10$ only	.680	.104	.572	.076	.916
	$q_t \leq 0.90$ only	.814	.192	.648	.056	.904
	$[0.10, 0.90]$ (default)	**.995**	**.920**	**.786**	**.248**	.928
	$[0.20, 0.80]$	.990	.850	.727	.158	**.934**

The gate does not maximize every metric: ungated MBPP has slightly higher rewritten TPR (0.114 vs. 0.110). A tighter $[0.20, 0.80]$ interval raises utility by less than one percentage point on each task but reduces detection. These ablations support using both bounds and the default interval among the tested choices. Full diagnostics are in Appendix D.9.

4.6 Ablation: Multi-Anchor Voting

Table 5 compares $K{=}1$ with $K{=}3$, keeping the other settings fixed. The margin result predicts protection for some partition decisions; under independent masks, three voters protect an expected quarter of labels from one arbitrary replacement. On raw GSM8K, both settings reach AUROC

Table 5: Voting ablation on Qwen3.5-9B: latest anchor ($K{=}1$) versus default ($K{=}3$).

		Raw		Mistral rewrite		Utility	
Dataset	K	AUROC	TPR@1%	AUROC	TPR@1%	Raw	Rewritten
GSM8K	1	.995	.882	.776	.196	.924	–
	3	.995	.920	.786	.236	.928	–
MBPP	1	.892	.402	.671	.060	.558	.526
	3	.884	.344	.694	.110	.538	.526

0.995. On raw MBPP, $K{=}1$ has higher AUROC (0.892 vs. 0.884) and TPR (0.402 vs. 0.344), with Pass@1 of 0.558 versus 0.538. After Mistral rewriting, $K{=}3$ yields higher AUROC and TPR on both tasks, reversing the raw detection ranking on MBPP.

4.7 Controlled Anchor-State Corruption

During detection, we independently replace each reconstructed anchor with probability $\rho \in \{10, 20, 50, 60\}\%$, drawing from a held-out donor pool disjoint from evaluation. Both watermarked and clean states are corrupted. Each arm's thresholds are calibrated on its own uncorrupted clean population and fixed across ρ; nonzero levels use three seeds. Detection declines with corruption

(Figure 2; full results in Table 25). At ρ=60%, K=3 TPR@1% on GSM8K falls from 0.894 to 0.194. On GSM8K, K=1 overtakes K=3 at 50%; on MBPP, it also has higher TPR@1% at 60%. Theorem 1 guarantees stability for partition labels with sufficiently large vote margins, but this local protection does not ensure an overall detection advantage under random anchor corruption.

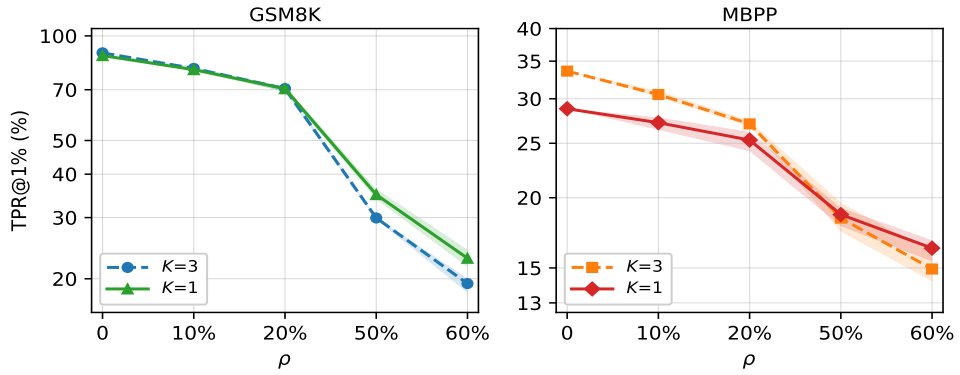

Figure 2: TPR under anchor-state corruption on Qwen3.5-9B, using fixed thresholds calibrated to 1% FPR on uncorrupted clean text; bands span three seeds.

4.8 Runtime Overhead

SemaSync adds no generation-time forward pass. On Llama-3.1-8B MBPP (50 prompts, one H100), generation takes a median of 0.535 s per sample (9.1% overhead), the fastest among the measured watermark-generation implementations. Median detection time is 0.587 s, the fastest of the three replay-based detectors. Several non-replay detectors run in milliseconds. Full timings are in Appendix D.11.

5 Limitations

Task-specific ontologies and sparse anchors. SemaSync requires task-specific ontologies selected on held-out rewrite data (Appendix A). New domains require ontology design and validation. Sparse anchors reduce usable states and weaken detection, as in the C4 extension. Richer ontologies may improve coverage, but their stability and diversity require validation.

Provider-side and third-party verification. Like SWEET and MorphMark+EWD, SemaSync requires the original key, frozen model, and tokenizer for detection replay. These resources fit provider-side verification, where the service holds the key and can retain model versions; Section 4.8 reports the measured overhead. Independent third-party verification is limited when these resources are unavailable.

Strong rewriting. Rewriting can preserve a final answer while removing reasoning and anchors, as compressed GSM8K rewrites illustrate (Table 20). The voting guarantee covers bounded voter replacement, not arbitrary deletion, compression, or restructuring.

6 Conclusion

SemaSync derives watermark states from canonical content events, uses keyed multi-anchor voting for partition stability, and selects embedding positions with a two-sided green-mass gate. Across two tasks and two generators, it achieves the highest raw AUROC among the compared baselines and retains useful evidence after rewriting. Ablations support both gate bounds and show a clearer voting gain on rewritten code, while utility costs and rewrite resilience remain task-dependent. These findings support content-derived synchronization and distribution-aware selection for structured-output watermarking. Extending anchor coverage and retaining evidence under strong compression are promising directions for future work.

Reproducibility Statement

Appendix A specifies the anchor ontologies and stability analysis; Appendix B gives the proof and probability derivation. Appendix D records model revisions, decoding settings, rewriting prompts, validation rules, and additional metrics. We provide the SemaSync implementation, experiment configurations, and evaluation scripts for anonymous review at `https://anonymous.4open.science/r/SemaSync-AEF8/`.

Statement on the Use of Large Language Models

We used large language models to assist with manuscript drafting and language editing, code implementation and debugging, and the planning and execution of experimental workflows. The authors retain full responsibility for the methodology, experimental results, interpretations, and final manuscript.

References

Ruisheng Cao, Mouxiang Chen, Jiawei Chen, Zeyu Cui, Yunlong Feng, Binyuan Hui, Yuheng Jing, Kaixin Li, Mingze Li, Junyang Lin, et al. Qwen3-coder-next technical report. *arXiv preprint arXiv:2603.00729*, 2026.

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

Pierre Fernandez, Antoine Chaffin, Karim Tit, Vivien Chappelier, and Teddy Furon. Three bricks to consolidate watermarks for large language models. In *2023 IEEE international workshop on information forensics and security (WIFS)*, pp. 1–6. IEEE, 2023.

Shuhong Gao. Counterexamples to the jacobian conjecture in dimensions greater than two. *arXiv preprint arXiv:2608.00222*, 2026.

Abe Hou, Jingyu Zhang, Tianxing He, Yichen Wang, Yung-Sung Chuang, Hongwei Wang, Lingfeng Shen, Benjamin Van Durme, Daniel Khashabi, and Yulia Tsvetkov. Semstamp: A semantic watermark with paraphrastic robustness for text generation. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 4067–4082, 2024.

Zhengmian Hu, Lichang Chen, Xidong Wu, Yihan Wu, Hongyang Zhang, and Heng Huang. Unbiased watermark for large language models. In *International Conference on Learning Representations*, volume 2024, pp. 45408–45436, 2024.

Jungin Kim, Shinwoo Park, and Yo-Sub Han. Marking code without breaking it: Code watermarking for detecting LLM-generated code. In *Findings of the Association for Computational Linguistics: EACL 2026*, pp. 3990–4002. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-eacl.207. URL `https://aclanthology.org/2026.findings-eacl.207/`.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. In *International conference on machine learning*, pp. 17061–17084. PMLR, 2023.

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense. *Advances in neural information processing systems*, 36:27469–27500, 2023.

Rohith Kuditipudi, John Thickstun, Tatsunori Hashimoto, and Percy Liang. Robust distortion-free watermarks for language models. *arXiv preprint arXiv:2307.15593*, 2023.

Taehyun Lee, Seokhee Hong, Jaewoo Ahn, Ilgee Hong, Hwaran Lee, Sangdoo Yun, Jamin Shin, and Gunhee Kim. Who wrote this code? watermarking for code generation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4890–4911, 2024.

Aiwei Liu, Leyi Pan, Xuming Hu, Shiao Meng, and Lijie Wen. A semantic invariant robust watermark for large language models. In *International Conference on Learning Representations*, volume 2024, pp. 6499–6519, 2024.

Fan Liu, Zherui Yang, Cancheng Liu, Tianrui Song, Xiaofeng Gao, and Hao Liu. Mm-agent: Llm as agents for real-world mathematical modeling problem. *Advances in Neural Information Processing Systems*, 38:20881–20934, 2026.

Yiheng Liu, Tianle Han, Siyuan Ma, Jiayue Zhang, Yuanyuan Yang, Jiaming Tian, Hao He, Antong Li, Mengshen He, Zhengliang Liu, et al. Summary of chatgpt-related research and perspective towards the future of large language models. *Meta-radiology*, 1(2):100017, 2023.

Yijian Lu, Aiwei Liu, Dianzhi Yu, Jingjing Li, and Irwin King. An entropy-based text watermarking detection method. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11724–11735, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.630. URL https://aclanthology.org/2024.acl-long.630/.

Sang-il Oum. A proof of the cycle double cover conjecture by openai: An exposition. *arXiv preprint arXiv:2607.16356*, 2026.

Leyi Pan, Aiwei Liu, Shiyu Huang, Yijian Lu, Xuming Hu, Lijie Wen, Irwin King, and Philip S Yu. Can llm watermarks robustly prevent unauthorized knowledge distillation? In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13228–13251, 2025.

Qizhi Pei, Lijun Wu, Zhuoshi Pan, Yu Li, Honglin Lin, Chenlin Ming, Xin Gao, Conghui He, and Rui Yan. Mathfusion: Enhancing mathematical problem-solving of llm through instruction fusion. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7400–7420, 2025.

Ruizhong Qiu, Weiliang Zeng, James Ezick, Christopher Lott, and Hanghang Tong. How efficient is llm-generated code? a rigorous & high-standard benchmark. In *International Conference on Learning Representations*, volume 2025, pp. 2233–2261, 2025.

Jie Ren, Han Xu, Yiding Liu, Yingqian Cui, Shuaiqiang Wang, Dawei Yin, and Jiliang Tang. A robust semantics-based watermark for large language model against paraphrasing. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 613–625, 2024.

Tom Sander, Pierre Fernandez, Alain Durmus, Matthijs Douze, and Teddy Furon. Watermarking makes language models radioactive. *Advances in Neural Information Processing Systems*, 37: 21079–21113, 2024.

Tom Sander, Hongyan Chang, Tomáš Souček, Tuan Tran, Valeriu Lacatusu, Sylvestre-Alvise Rebuffi, Alexandre Mourachko, Surya Parimi, Christophe Ropers, Rashel Moritz, et al. Textseal: A localized llm watermark for provenance & distillation protection. *arXiv preprint arXiv:2605.12456*, 2026.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

Yinjie Wang, Ling Yang, Ye Tian, Ke Shen, and Mengdi Wang. Co-evolving llm coder and unit tester via reinforcement learning. *Advances in Neural Information Processing Systems*, 38:143630–143664, 2026.

Zongqi Wang, Tianle Gu, Baoyuan Wu, and Yujiu Yang. Morphmark: Flexible adaptive watermarking for large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4842–4860, 2025.

Yihan Wu, Zhengmian Hu, Hongyang Zhang, and Heng Huang. Dipmark: A stealthy, efficient and resilient watermark for large language models. 2023.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. BERTScore: Evaluating text generation with BERT. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SkeHuCVFDr.

Zhaoxi Zhang, Xiaomei Zhang, Yanjun Zhang, He Zhang, Shirui Pan, Bo Liu, Asif Qumer Gill, and Leo Yu Zhang. Character-level perturbations disrupt llm watermarks. *arXiv preprint arXiv:2509.09112*, 2025.

Xuandong Zhao, Prabhanjan Vijendra Ananth, Lei Li, and Yu-Xiang Wang. Provable robust watermarking for ai-generated text. In *International Conference on Learning Representations*, 2024.

Xuandong Zhao, Lei Li, and Yu-Xiang Wang. Permute-and-flip: An optimally stable and watermarkable decoder for llms. In *International Conference on Learning Representations*, volume 2025, pp. 32468–32496, 2025.

Terry Yue Zhuo, Minh Chien Vu, Jenny Chim, Han Hu, Wenhao Yu, Ratnadira Widyasari, Imam Nur Bani Yusuf, Haolan Zhan, Junda He, Indraneil Paul, et al. Bigcodebench: Benchmarking code generation with diverse function calls and complex instructions. In *International Conference on Learning Representations*, volume 2025, pp. 66602–66656, 2025.

A Anchor Ontology Design and Stability Evaluation

SemaSync instantiates its deterministic anchor extractor $\mathcal{E}$ using domain-specific semantic and structural ontologies. To evaluate whether candidate anchors provide sufficiently available, transformation-stable, and diverse synchronization signals, we analyze their behavior on held-out source–rewrite corpora. We report anchor density, sample-level K3 coverage, and ordered-state survival under rewriting. The ontology-selection criteria are described below.

A.1 Held-Out Rewrite Stability Protocol

For each domain, we construct a held-out development corpus containing 200 source generations and up to five independently seeded rewrites of each eligible source output. The development tasks are disjoint from the datasets used in the main experiments. All source generations and rewrites use Qwen3.5-9B, while the rewrite prompts and validity requirements are adapted to the semantic constraints of each domain (Appendices D.3 and D.4).

These development corpora are used only to evaluate anchor availability and transformation stability. Downstream watermark metrics, including AUROC, AUPRC, and TPR, are not used in the ontology analysis.

A.2 Exact Anchor-State Survival

Let

$$A(y) = [a_1, \ldots, a_n]$$

denote the ordered anchor sequence extracted from a source output y, and let $A(y')$ denote the corresponding sequence extracted from a meaning-preserving rewrite y'. Repeated canonical anchors remain distinct occurrences and are not deduplicated.

For dataset d, let

$$\mathsf{Match}_d(U, V)$$

denote the number of exact, order-preserving matches returned by the frozen sequence-alignment procedure for two sequences U and V. Exact matching requires equality of the complete canonical anchor identity

$$a = (\tau, \pi),$$

including both the event type τ and canonicalized payload π.

To measure preservation of the synchronization context, let

$$\mathcal{H}_K(y) = [H_1^{(K)}, H_2^{(K)}, \ldots, H_{L_y}^{(K)}]$$

denote the sequence of ordered K-anchor states reconstructed at the L_y eligible event transitions in y. Each state

$$H_\ell^{(K)} = [a_{\ell,1}, \ldots, a_{\ell,K}]$$

preserves anchor order and multiplicity. We define ordered-K exact survival as

$$s_{K,d}^{\text{exact}}(y, y') = \frac{\mathsf{Match}_d\left(\mathcal{H}_K(y), \mathcal{H}_K(y')\right)}{|\mathcal{H}_K(y)|}.$$

The dataset-level statistic aggregates these measurements over the valid source–rewrite pairs according to the corresponding frozen evaluation protocol. Exact state survival requires the same number of voters, the same anchor ordering, and equality of every canonical type–payload identity.

Anchor availability. Transformation stability alone is insufficient: an ontology containing only rare but highly stable events may provide little usable synchronization context. We therefore report two complementary coverage statistics.

First, the mean number of extracted anchor occurrences per source is

$$\text{AnchorsPerSource} = \frac{1}{n_{\text{src}}} \sum_{i=1}^{n_{\text{src}}} |A(y_i)|,$$

where n_{src} is the number of source outputs and repeated canonical anchors are counted as distinct occurrences.

Second, we define sample-level K3 coverage as

$$\text{K3Coverage} = \frac{1}{n_{\text{src}}} \sum_{i=1}^{n_{\text{src}}} \mathbb{I}\left[\exists\, t:\ |H_t(y_i)| = 3\right].$$

Unlike simply checking whether an output contains at least three anchors, this metric requires three anchors to be simultaneously active under the frozen context-window and event-availability rules. It therefore directly measures the fraction of source outputs that can instantiate at least one multi-anchor SemaSync state.

A.3 Task-Specific Candidate Events and Selection Outcomes

We instantiate the anchor extractor using task-specific candidate event families. For each task, we enumerate candidate semantic or structural events and evaluate representative candidate ontologies according to the same three high-level quantities: mean anchors per source, sample-level K3 coverage, and ordered-K3 exact survival.

A.3.1 GSM8K

Candidate events. For mathematical reasoning, the development extractor considers events associated with numerical computation, intermediate reasoning, and answer expressions. Table 6 summarizes the major event families considered during development.

Candidate-set performance. Table 7 compares two representative event-set configurations using the unified coverage and stability metrics.

Both configurations provide near-complete K3 coverage, indicating that anchor sparsity is not a limiting factor for GSM8K. The primary difference lies in transformation stability: the current semantic and structural event design achieves substantially higher ordered-K3 exact survival than surface lexical cues, indicating that arithmetic, quantity, operation, and unit events provide the dominant synchronization signal.

The reported SemaSync experiments retain the frozen six-family GSM8K extractor:

$$\mathcal{O}_{\text{GSM8K}} = \{\texttt{ANSWER}, \texttt{MONEY}, \texttt{PERCENT}, \texttt{OPERATION}, \texttt{UNIT}, \texttt{NUMBER}\}.$$

Table 6: Candidate GSM8K event families evaluated in the held-out rewrite-stability study.

Event family	Canonical representation and representative subtypes
Final result	`final_result:<value>`, including normalized integer, decimal, and fractional answer expressions
Quantities	`money:<value>`, `percent:<value>`, `fraction:<value>`, and `number:<value>`
Temporal quantity	`temporal_quantity:<unit>:<value>`, with `second`, `minute`, `hour`, `day`, `week`, `month`, and `year`
Operation	`operation:<operator>`, with `add`, `subtract`, `multiply`, `divide`, and `equal`
Relation	`relation:rate` and related rate expressions
Unit	`unit:<canonical-unit>`, covering temporal, distance, currency, object-count, volume, and weight units
Discourse	`discourse:conclusion`

Table 7: Held-out GSM8K candidate-set analysis. K3 coverage denotes the fraction of source completions containing at least one valid ordered three-anchor state.

Candidate set	Included event labels	Anchors/ source	K3 coverage	K3 exact
Current-extractor proxy	Number, money, percent, operation, unit, and final result	48.50	1.000	**0.625**
Surface lexical cues	Operation, relation, comparison, unit, and discourse events	27.11	0.990	0.545

A.3.2 MBPP

Candidate events and ontologies. The MBPP parser supports eight structural event families: `LITERAL`, `OPERATION`, `CALL`, `CONDITION`, `ITERATION`, `COLLECTION`, `INDEX_OR_SLICE`, and `RETURN`. Identifiers, variable spelling, comments, whitespace, indentation, and formatting are excluded from anchor identity.

We evaluate three candidate structural ontologies, as summarized in Table 8.

All three candidates provide high K3 coverage, while their rewrite stability differs substantially. Although $\mathcal{O}_C$ provides the largest anchor density and complete K3 coverage, its ordered-K3 exact survival is lower than that of the more compact $\mathcal{O}_A$. Similarly, $\mathcal{O}_B$ provides high coverage but exhibits the weakest state preservation. Therefore, among candidates satisfying the frozen coverage and diversity requirements, $\mathcal{O}_A$ achieves the best coverage–stability tradeoff and is selected as the final MBPP ontology:

$$\mathcal{O}_{\text{MBPP}} = \{\texttt{LITERAL}, \texttt{OPERATION}, \texttt{CONDITION}, \texttt{ITERATION}, \texttt{RETURN}\}.$$

A.3.3 C4 REALNEWSLIKE

Candidate events. For open-domain continuations, the development extractor considers a broader set of content-level events than those retained in the final ontology. Table 9 summarizes the principal candidate event families and their representative subtypes.

These event families are combined into several aggregate candidate ontologies. Table 10 reports five representative configurations using the same three metrics as GSM8K and MBPP.

Candidate-set performance. C4 exhibits a stronger coverage–stability tradeoff than GSM8K or MBPP. Broad candidate sets achieve complete K3 coverage but substantially reduce ordered-state preservation under rewriting. Named entities provide the highest ordered-K3 exact survival while still supporting at least one K3 state in 78.0% of source continuations.

The original preregistered event-level screen additionally required each candidate subtype to provide sufficient source coverage and occurrence frequency, high exact rewrite survival and precision,

Table 8: Held-out MBPP candidate-ontology analysis using the same coverage and stability metrics as GSM8K and C4.

Ontology	Included event types	Anchors/ source	K3 coverage	K3 exact
$\mathcal{O}_A$	LITERAL, OPERATION, CONDITION, ITERATION, RETURN	18.50	0.904	**0.621**
$\mathcal{O}_B$	OPERATION, CALL, CONDITION, ITERATION, RETURN	12.49	0.918	0.519
$\mathcal{O}_C$	LITERAL, OPERATION, CALL, CONDITION, ITERATION, COLLECTION, INDEX_OR_SLICE, RETURN	27.18	0.986	0.574

Table 9: Candidate C4 event families considered during anchor-ontology development.

Event family	Candidate event types and representative subtypes
Quantitative events	Number, ordinal, money, percent, date, time, duration, age, and measurement
Named entities	PERSON, GPE, NORP, ORG, LOC, EVENT, PRODUCT, WORK_OF_ART, LAW, and LANGUAGE
Acronym and title events	Acronym, acronym-title, document-title, and quoted-title events
Relations	Acquired-by, caused-by, compared-with, located-in, member-of, and related relational events
Predicates and concepts	Predicate, verb lemma, concept, generic verb, generic adjective, noun-chunk head, and proper-noun head
Other lexical/control events	Discourse cues, pronouns, and negative-control lexical events

consistent preservation across multiple rewrites, and adequate canonical-value diversity. No individual C4 event subtype simultaneously satisfied all of these requirements. In particular, broadly occurring events tended to provide high coverage but weaker state stability, whereas more stable semantic events were generally sparser.

We therefore use the held-out aggregate analysis to guide a relaxed entity-oriented ontology. The final frozen C4 ontology is

$$\mathcal{O}_{\mathrm{C4}} = \{\texttt{PERSON}, \texttt{GPE}, \texttt{NORP}, \texttt{ACRONYM}\}.$$

This four-class ontology is a relaxed post-hoc choice informed by the held-out development analysis after the original preregistered screen selected no eligible ontology. It is frozen before construction and evaluation of the independent C4 evaluation dataset, and no detection or utility result from that dataset is used to modify the anchor design.

A.4 Cross-Domain Anchor Availability

K3 onset. In addition to whether a task eventually forms a multi-anchor state, we measure how early the first valid ordered-K3 state becomes available. For a completion y that forms at least one K3 state, we define

$$\mathrm{FirstK3}(y) = \frac{\min\{t : |H_t(y)| = 3\}}{|y|} \times 100\%.$$

The statistic is computed only over K3-covered samples; outputs that never form a K3 state are represented separately by the K3 coverage metric. Table 11 summarizes the normalized onset of the first valid ordered-K3 state across the three tasks.

Discussion. Across all three tasks, anchor density alone is not a sufficient criterion for synchronization-state design. Increasing the number of extracted events generally improves K3 availability, but overly broad or surface-oriented ontologies can substantially reduce ordered-state preservation under rewriting. SemaSync therefore favors anchor designs that maintain sufficient K3 coverage while providing stronger exact ordered-state survival.

Table 10: Held-out C4 candidate-event analysis. K3 coverage is the sample-level fraction of source completions containing at least one valid ordered three-anchor state.

Candidate set	Anchors/source	K3 coverage	K3 exact
Quantitative events	6.24	0.595	0.422
Named entities	9.45	0.780	**0.439**
Named entities + quantities	15.69	0.900	0.401
All candidate events	66.87	1.000	0.226
Surface lexical cues	46.72	1.000	0.203

Table 11: Normalized onset of the first valid ordered-K3 state across tasks. First-K3 position is computed only over source completions that eventually form a K3 state.

Dataset	K3-covered sources	First-K3 position
GSM8K	200/200	10.91%
MBPP	132/146	76.78%
C4 RealNewsLike	114/200	33.88%

The tasks differ substantially in how early three-anchor voting becomes available. GSM8K establishes its first ordered-K3 state after only 10.91% of the completion on average, whereas C4 reaches its first K3 at 33.88%. In contrast, although most MBPP samples eventually form a K3 state, the first valid K3 appears only after 76.78% of the completion among covered samples. This late onset limits the portion of each completion that can use three-anchor voting.

B Partition Stability: Proof and Independent-Mask Analysis

For independent Bernoulli$(1/2)$ votes, the expected fraction of vocabulary labels stable under r replacements is $\rho_{m,r} = \Pr(|2B - m| > 2r)$, where $B \sim \text{Binomial}(m, 1/2)$. For fixed r, this fraction is nondecreasing along odd voter counts and tends to one. This probability model concerns independent masks; the margin criterion holds for any realized votes. We prove both statements below.

Proof. Fix a token v and define

$$S_t(v) = \sum_{a_i \in H_t} b_i(v), \qquad \Delta_t(v) = |S_t(v)|.$$

Since m is odd and each $b_i(v) \in \{-1, +1\}$, $S_t(v) \neq 0$, and its sign determines the majority-vote label of v.

Replacing one anchor voter can change its signed vote by at most 2. Therefore, replacing at most r anchor voters changes $S_t(v)$ by at most $2r$ in absolute value. Hence, if

$$\Delta_t(v) > 2r,$$

the sign of $S_t(v)$ cannot change under any such replacement, and the majority label of v is invariant.

Conversely, suppose $\Delta_t(v) \leq 2r$. Without loss of generality, assume $S_t(v) > 0$. Because $S_t(v)$ is odd,

$$\Delta_t(v) \leq 2r - 1.$$

Let

$$\nu = \frac{\Delta_t(v) + 1}{2}.$$

Then $\nu \leq r$. Replacing ν majority voters with anchors inducing the opposite vote changes the signed sum to

$$S'_t(v) = \Delta_t(v) - 2\nu = -1,$$

and therefore flips the majority label. Thus,

$$v \text{ is } r\text{-replacement stable} \quad \Longleftrightarrow \quad \Delta_t(v) > 2r.$$

Under the idealized independent Bernoulli$(1/2)$ mask model, let

$$B_v = \sum_{a_i \in H_t} M_{k,a_i}(v) \sim \text{Binomial}(m, 1/2)$$

be the number of green votes for token v. Since

$$\Delta_t(v) = |2B_v - m|,$$

the probability that v satisfies the deterministic stability condition is

$$\rho_{m,r} = \Pr(|2B - m| > 2r), \qquad B \sim \text{Binomial}(m, 1/2).$$

Consequently, by linearity of expectation,

$$\mathbb{E}\left[\frac{1}{|\mathcal{V}|}\sum_{v\in\mathcal{V}} \mathbb{I}(\Delta_t(v) > 2r)\right] = \rho_{m,r}.$$

For

$$0 \le r \le \frac{m-1}{2},$$

binomial symmetry further gives

$$\rho_{m,r} = 2^{1-m} \sum_{j=0}^{\frac{m-2r-1}{2}} \binom{m}{j}.$$

It remains to show that $\rho_{m,r}$ is nondecreasing with the odd voter count m. Let

$$X_1, \ldots, X_m \overset{\text{i.i.d.}}{\sim} \text{Unif}\{-1, +1\}, \qquad S_m = \sum_{i=1}^{m} X_i,$$

so that

$$\rho_{m,r} = \Pr(|S_m| > 2r).$$

Define

$$u_{m,r} = 1 - \rho_{m,r} = \Pr(|S_m| \le 2r).$$

For $r = 0$, $\rho_{m,0} = 1$ for every odd m. For $r \ge 1$, let $\alpha = 2r - 1$. Since S_m is odd,

$$|S_m| \le 2r \quad \Longleftrightarrow \quad |S_m| \le \alpha.$$

Writing $p_m(s) = \Pr(S_m = s)$ and using

$$S_{m+2} = S_m + X_{m+1} + X_{m+2},$$

we obtain

$$p_{m+2}(s) = \frac{1}{4}p_m(s-2) + \frac{1}{2}p_m(s) + \frac{1}{4}p_m(s+2).$$

Summing over $|s| \le \alpha$ yields

$$u_{m+2,r} = u_{m,r} + \frac{1}{2}\left[p_m(\alpha+2) - p_m(\alpha)\right].$$

For $s \ge 0$,

$$\frac{p_m(s+2)}{p_m(s)} = \frac{m-s}{m+s+2} \le 1,$$

and hence $p_m(\alpha+2) \le p_m(\alpha)$. Therefore

$$u_{m+2,r} \le u_{m,r},$$

or equivalently,

$$\rho_{m+2,r} \ge \rho_{m,r}.$$

Thus $\rho_{m,r}$ is nondecreasing along the odd voter counts.

Finally,

$$1 - \rho_{m,r} = \Pr(|S_m| \le 2r).$$

For fixed r, this event contains only a constant number of possible values of S_m. Hence,

$$1 - \rho_{m,r} \le (2r+1) \max_s p_m(s).$$

By the standard central-binomial-coefficient bound, there exists a constant $C > 0$ such that

$$\max_s p_m(s) \le \frac{C}{\sqrt{m}}.$$

Therefore,

$$1 - \rho_{m,r} \le \frac{(2r+1)C}{\sqrt{m}} \longrightarrow 0$$

for every fixed r as $m \to \infty$ through odd values. It follows that

$$\lim_{\substack{m\to\infty \\ m \text{ odd}}} \rho_{m,r} = 1.$$

□

C Open-Domain Extension: C4 RealNewsLike

We additionally evaluate SemaSync on open-domain news-like continuation (C4 RealNewsLike, N=500 prompts, IDs 0–499), using the four-class ontology described in Appendix A.3. Anchors are substantially sparser in this domain than in mathematical reasoning or code: only 57% of held-out continuations ever form a three-anchor state, and the first such state appears after 34% of the completion on average (Table 11). We report C4 as an extension rather than as a primary claim. The experimental protocol follows Section 4.1; utility is measured by perplexity under `gemma-3-12b-pt` (Team et al., 2025) and by BERTScore (Zhang et al., 2020) against the human continuation of each document.

Table 12: C4 RealNewsLike results. Detection: AUROC and TPR at 1% FPR under Raw and Mistral rewrite.

	Raw		Mistral		Utility	
Method	AUROC	TPR@1%	AUROC	TPR@1%	PPL ↓	BERTScore ↑
Clean	–	–	–	–	11.87	.8175
KGW	.99998	1.000	.8707	.454	15.30	.8069
SWEET	.99976	.998	.8729	.546	16.38	.8110
MorphMark	.9970	.974	.7856	.230	11.36	.8145
MorphMark+EWD	.9997	.994	.7992	.232	11.36	.8145
SemStamp	.9862	.896	.6947	.114	6.97	.8203
TextSeal	.9995	.996	.8259	.304	11.59	.8203
SemaSync	.9779	.948	.7791	.312	13.68	.8141

On raw outputs SemaSync remains clearly detectable (AUROC 0.978, TPR@1% 0.948) but has the lowest AUROC among the compared methods, reflecting the smaller number of carrier positions available when anchors are sparse. Under Mistral rewriting its AUROC (0.779) is mid-range, below KGW, SWEET, and TextSeal and comparable to MorphMark, while its TPR@1% (0.312) is third-highest. SemaSync's perplexity is higher than that of MorphMark and TextSeal but lower than that of KGW and SWEET, with BERTScore within 0.004 of clean text. Open-domain prose thus exposes the main cost of anchor-derived states—coverage—rather than a failure of synchronization, and motivates richer open-domain ontologies as future work.

D Experimental Details and Full Results

D.1 Models, Decoding, and Frozen Hyperparameters

Table 13 pins the exact model revisions and decoding settings of every generator, rewriter and scorer, and Table 14 summarizes the evaluation datasets, their output contracts, utility metrics and rewrite conditions.

Table 13: Pinned model identities and decoding settings.

Role	Model	Revision / setting
Generator (primary)	`Qwen/Qwen3.5-9B`	`c202236235762e1c871ad0ccb60c8ee5ba337b9a`; bf16; thinking off
Generator (second)	`meta-llama/Llama-3.1-8B-Instruct`	`0e9e39f249a16976918f6564b8830bc894c89659`; bf16
Decoding (both)	–	temperature 1.0, top-p 0.95, 1024 new tokens; GSM8K `max_seq_len` 2048
Rewriter (primary)	`mistralai/Mistral-Small-3.2-24B-Instruct-2506`	`95a6d26c4bfb886c58daf9d3f7332c857cb27b43`; T=0.15, top-p 0.95, 1024 tokens
Rewriter (second)	`gpt-5.6-luna` (OpenAI Responses API)	reasoning effort `max`; no sampling controls; `store=false`
Stress rewriter	DIPPER `kalpeshk2011/dipper-paraphraser-xxl`	`c1fbf7a9`; lex. div. 60, order div. 0, sentence interval 3, top-p 0.75
C4 utility scorer	`google/gemma-3-12b-pt` / RoBERTa-large	`295efb63`; PPL and BERTScore F1

Table 14: Evaluation datasets. Each watermark method is generated once per item. Rewritten clean corpora are generated once per dataset and shared across detectors.

	GSM8K	MBPP	C4 RealNewsLike (extension)
Domain	grade-school math word problems	Python function synthesis	news-like text continuation
Items (N)	500 (IDs 0–499)	500 (tasks 11–510)	500 (IDs 0–499)
Output contract	free-form reasoning ending in a final numeric answer	raw Python source only	continuation of a document prefix
Utility	exact final-answer accuracy	Pass@1	perplexity under `gemma-3-12b-pt`; BERTScore F1 against the human continuation
Rewrite conditions	Mistral, GPT-5.6 Luna Max, DIPPER (stress)	Mistral, GPT-5.6 Luna Max, DIPPER (stress)	Mistral
Anchor ontology	ANSWER, MONEY, PERCENT, OPERATION, UNIT, NUMBER	LITERAL, OPERATION, CONDITION, ITERATION, RETURN	PERSON, GPE, NORP, ACRONYM
Generators	Qwen3.5-9B, Llama-3.1-8B	Qwen3.5-9B, Llama-3.1-8B	Qwen3.5-9B

Baseline hyperparameters. Table 15 lists the method configurations. KGW, SWEET, MorphMark, MorphMark+EWD, SemStamp, and TextSeal use the same respective hyperparameters across GSM8K, MBPP, and C4 RealNewsLike. Unigram and SIR use the configurations below on GSM8K and MBPP; STONE is evaluated on MBPP only. The configurations are shared across both generators. All methods use temperature 1.0, top-p 0.95, and a budget of 1024 new tokens. Where a logit bias is used, it is applied before top-p truncation. KGW and SemaSync use the string `"35317"` as the hash key; SWEET, MorphMark, and SemStamp use the integer 15485863. MorphMark+EWD shares MorphMark's generated texts and partition, and replays the model at detection time to compute entropy weights.

SIR uses $\delta = 1.5$, fixed before evaluation to give an approximately 3.0 green–red bias gap. Its published threshold 0.2 and STONE's default z threshold 10 remain in their configurations, but our comparisons calibrate thresholds on matched clean scores. STONE uses the official vocabulary sizes, with no extra vocabulary mask.

Scoring conventions. The primary SemaSync score is z_γ over deduplicated evidence with γ=0.5; a sample with $N(y) = 0$ receives $z_\gamma = 0$. Baselines use their official scores (KGW/SWEET z-score, MorphMark z-score, EWD z_{ewd}, SemStamp official transition score, TextSeal fused $-\log_{10} p$, Unigram original deduplicated z-score, and native SIR and STONE scores). TPR at a target FPR uses the threshold at which the matched clean population of the same condition attains that FPR.

D.2 The MBPP Pre-Anchor Initializer

Before the first semantic anchor commits, the MBPP controller operates on a single pseudo-anchor a_t^{P1} derived from the previous ordinary token y_{t-1} (for t=1, the last ordinary token of the rendered prompt). The pseudo-anchor is encoded as a domain-separated string that begins with a NUL byte followed by a fixed domain tag and the decimal token ID; frozen semantic anchors never contain NUL, so the two families cannot collide under the shared keyed mask. The same mask primitive, γ, δ, and gate $[0.10, 0.90]$ (with q_t on the post-temperature top-p nucleus) are applied. When the extractor commits the first semantic anchor at position t^*, the controller emits that token in initializer mode

Table 15: Method configurations. A dash indicates that the token-level green ratio or fixed logit bias is not used.

Method	Green ratio / bias	Configuration
KGW	0.5 / $\delta = 3$	Fixed bias at every eligible position; `kgw_seed`=0.
SWEET	0.5 / $\delta = 3$	Bias only when entropy exceeds $\tau = 0.5$ nats, computed from logits with padding masked.
MorphMark	0.5 / adaptive	`exp` variant; $p_0 = 0.15$, $k_{\text{exp}} = 1.30$.
MorphMark+EWD	Same generation	Detection-only entropy weighting with `ewd_delta`=1.25.
SemStamp	— / —	Sentence-level rejection sampling; `sp_dim`=3, `valid_region_ratio`=0.25, `rejection_margin`=0.02, `margin_relax_after`=16, `max_attempts`=100.
TextSeal	— / —	Keys $[42, 12387]$; `mixing_alpha`=0.5, `ngram`=1; scoring version `v2`.
Unigram	0.5 / $\delta = 3$	Fixed, context-free green list; `hash_key`=15485863; original deduplicated z-score.
SIR	— / $\delta = 1.5$	`chunk_length`=10; `scale_dimension`=300; native chunk retokenization and completion-only scoring.
STONE	0.5 / $\delta = 3$	Python; `prefix_length`=1; `hash_key`=15485863; `skipping_rule=all_pl`; `watermark_on_pl` is the upstream string `False`.
SemaSync	0.5 / $\delta = 3$	Bias only for $q_t \in [0.1, 0.9]$;

and switches to semantic mode from t^*+1 onward; the switch is irreversible even if the window later contains no anchors. Detection replays the identical state machine from the rendered prompt and the visible completion, and deduplicates evidence with channel-tagged keys: $(\mathrm{P1}, y_{t-1}, y_t)$ before the switch and (SEM, H_t, y_t) after it. Generation applies no deduplication.

On raw MBPP outputs, 7.0% of watermarked completions (35 of 500) contain no carrier under either mode and receive $z_\gamma = 0$.

D.3 Rewrite Prompts

Mistral and Luna use the same dataset-specific rewrite prompts and acceptance rules, with model-specific generation settings specified in this appendix. Within a dataset the same template is used for the clean corpus and for all watermarked corpora. Braces denote runtime substitutions; the blocks below reproduce the prompt text.

GSM8K (Mistral and Luna).

```
Paraphrase the following solution.
Preserve the key reasoning, numerical values, mathematical
relationships, and final answer.  You may reorganize wording
and sentence structure, but do not change the meaning of the
solution.
Keep the rewritten solution reasonably close in detail and
length to the original.
Output only the rewritten solution.
Problem:
{question}
Solution:
{solution}
End with exactly one final line:
Final Answer:  <same answer>
```

The original problem statement is supplied alongside the text being rewritten, so the rewriter is conditioned on the question rather than paraphrasing the solution in isolation. Answer preservation is requested in the prompt and enforced independently after generation: a candidate is accepted only if its final answer is strictly numerically equivalent to the source's and its whitespace-token

length ratio lies in $[0.30, 1.50]$; up to three attempts are made, after which the source text is retained (Appendix D.4).

MBPP Code-Only (Mistral and Luna).

```
You are given a Python programming problem and an existing
Python solution.
Rewrite the implementation while preserving its behavior
exactly.
Strict requirements:
1.  Preserve every required public function name and
function signature exactly.
2.  Preserve the expected return values and observable
behavior exactly.
3.  Preserve required imports and externally visible
interfaces.
4.  Keep the same algorithmic intent; use local
semantics-preserving rewrites such as renaming local
variables, simplifying or expanding equivalent expressions,
or making equivalent local control-flow changes.
5.  Do not replace the solution with a different algorithm.
6.  Do not add tests, examples, assertions, demonstrations,
execution output, comments about the rewrite, or prose.
7.  Return only complete executable Python source code.
8.  Do not use Markdown fences.
Problem:
{problem}
Existing solution:
{source_code}
```

A single candidate is requested per task.

C4 RealNewsLike (Mistral and Luna).

```
Rewrite the continuation so it remains coherent with the
preceding passage and conveys the same main meaning and
factual content.  Use substantially different wording and
sentence structure.  Return only the rewritten continuation.
Preceding passage:
{source_prompt}
Continuation:
{continuation}
```

Luna request settings. Luna is called through the Responses API with `model=gpt-5.6-luna` and `reasoning.effort=max`; `temperature`, `top_p`, `max_output_tokens`, and a seed are omitted from the request rather than set.

DIPPER. DIPPER is not driven by a natural-language instruction. Following the original implementation, the model consumes an explicit control string and a sentence window, `lexical = 40, order = 100` for our setting of lexical diversity 60 and order diversity 0.

D.4 Rewrite Validity and Fallback

A GSM8K row falls back to its source text when the source answer cannot be parsed or when all three rewrite attempts fail to preserve it; fallback rows stay in the 500-row detection population. Table 16 reports fallback counts by generator. Under Luna 99.2–100% of directly rewritten rows preserve the final answer. MBPP and C4 rewrites make a single attempt with no source fallback and no answer, length, or score gate. On MBPP under Luna every rewritten program whose source passed still passes, so conditional functional preservation is 1.00 for every corpus.

Table 16: GSM8K source-text fallbacks out of $N=500$ rows per corpus.

Corpus	Mistral	Luna
Qwen3.5-9B		
Clean	16	17
KGW	37	40
SWEET	20	21
MorphMark	24	25
SemStamp	28	28
TextSeal	17	17
Unigram	25	–
SIR	28	–
SemaSync	21	22

D.5 Confidence Intervals for the Main Qwen Results

Table 17 gives interval estimates for every cell of the main Qwen comparison (Table 1) and for the C4 extension (Appendix C), in six blocks: GSM8K, MBPP and C4 RealNewsLike, each before and after the Mistral rewrite. Every interval is a 95% paired sample-level bootstrap over 2000 replicates, in which the same resampled ID set is applied to a method and to the clean population of its own condition and the statistic is recomputed inside each replicate; AUPRC and the TPR at 5% and 10% FPR are point estimates. The point estimates repeat the main tables, which quote intervals only where a comparison hinges on them (Section 4.2).

Table 17: Qwen3.5-9B: AUROC, AUPRC, and TPR at 1/5/10% FPR with 95% paired sample-level bootstrap intervals (2000 replicates).

Method	AUROC [95% CI]	AUPRC	TPR@1% [95% CI]	TPR@5%	TPR@10%
GSM8K — Raw					
KGW	.897 [.881, .914]	.913	.462 [.334, .542]	.624	.734
SWEET	.977 [.969, .984]	.980	.766 [.572, .848]	.914	.936
MorphMark	.709 [.686, .732]	.732	.126 [.064, .194]	.262	.350
MorphMark+EWD	.763 [.740, .786]	.780	.172 [.130, .214]	.320	.428
SemStamp	.797 [.776, .819]	.809	.190 [.074, .262]	.382	.482
TextSeal	.764 [.741, .788]	.790	.174 [.086, .242]	.374	.474
Unigram	.890 [.873, .906]	.898	.400 [.294, .452]	.542	.660
SIR	.751 [.731, .771]	.753	.108 [.038, .164]	.256	.382
SemaSync	**.995** [.992, .998]	**.995**	**.920** [.836, .956]	**.980**	**.992**
GSM8K — Mistral rewrite					
KGW	.709 [.684, .733]	.748	.172 [.132, .216]	.320	.392
SWEET	.729 [.700, .759]	.750	.136 [.096, .218]	.302	.364
MorphMark	.583 [.555, .613]	.617	.078 [.044, .102]	.134	.238
MorphMark+EWD	.613 [.584, .644]	.643	.084 [.050, .108]	.154	.244
SemStamp	.654 [.626, .680]	.674	.080 [.050, .126]	.200	.292
TextSeal	.628 [.601, .655]	.646	.070 [.046, .094]	.156	.250
Unigram	.702 [.675, .728]	.725	.112 [.070, .162]	.248	.354
SIR	.665 [.645, .688]	.664	.056 [.022, .092]	.124	.268
SemaSync	**.786** [.759, .813]	**.817**	**.248** [.204, .324]	**.414**	**.518**

Continued on next page

Table 17 (continued)

Method	AUROC [95% CI]	AUPRC	TPR@1% [95% CI]	TPR@5%	TPR@10%
MBPP — Raw					
KGW	.823 [.801, .845]	.852	**.368** [.320, .442]	.488	.568
SWEET	.860 [.839, .880]	.869	.322 [.248, .416]	.498	**.624**
MorphMark	.606 [.580, .632]	.619	.056 [.020, .100]	.132	.210
MorphMark+EWD	.786 [.761, .810]	.797	.160 [.104, .250]	.344	.442
SemStamp	.675 [.651, .699]	.647	.046 [.024, .064]	.046	.046
TextSeal	.680 [.658, .705]	.708	.120 [.084, .152]	.220	.336
Unigram	.777 [.759, .796]	.803	.240 [.164, .330]	.366	.464
SIR	.807 [.781, .831]	.808	.112 [.000, .292]	.430	.524
STONE	.625 [.598, .652]	.649	.056 [.030, .108]	.168	.246
SemaSync	**.884** [.865, .903]	**.882**	.344 [.226, .412]	**.548**	.566
MBPP — Mistral rewrite					
KGW	.637 [.612, .661]	.642	.034 [.016, .100]	.154	.226
SWEET	.709 [.683, .734]	.710	.094 [.038, .128]	.210	**.334**
MorphMark	.561 [.536, .586]	.561	.028 [.010, .042]	.082	.144
MorphMark+EWD	.640 [.612, .668]	.639	.042 [.012, .070]	.134	.226
SemStamp	.509 [.490, .528]	.504	.002 [.000, .006]	.002	.002
TextSeal	.584 [.559, .609]	.612	.036 [.018, .074]	.162	.228
Unigram	**.710** [.689, .732]	**.712**	.108 [.020, .136]	.220	.308
SIR	.592 [.564, .619]	.573	.012 [.002, .028]	.074	.162
STONE	.576 [.552, .600]	.590	.038 [.010, .066]	.110	.188
SemaSync	.694 [.666, .723]	.704	**.110** [.040, .146]	**.266**	.300
C4 RealNewsLike — Raw					
KGW	**1.000** [1.000, 1.000]	**1.000**	**1.000** [.996, 1.000]	**1.000**	**1.000**
SWEET	**1.000** [.999, 1.000]	**1.000**	.998 [.994, 1.000]	.998	.998
MorphMark	.997 [.995, .999]	.998	.974 [.958, .988]	.986	.988
MorphMark+EWD	**1.000** [.999, 1.000]	**1.000**	.994 [.980, 1.000]	.998	**1.000**
SemStamp	.986 [.979, .992]	.989	.896 [.622, .952]	.950	.966
TextSeal	**1.000** [.999, 1.000]	**1.000**	.996 [.984, 1.000]	.998	.998
SemaSync	.978 [.968, .986]	.984	.948 [.928, .966]	.952	.954
C4 RealNewsLike — Mistral rewrite					
KGW	.871 [.847, .892]	.896	.454 [.320, .558]	.600	.686
SWEET	**.873** [.850, .895]	**.900**	**.546** [.316, .582]	**.638**	**.706**
MorphMark	.786 [.757, .813]	.816	.230 [.082, .334]	.434	.522
MorphMark+EWD	.799 [.771, .827]	.834	.232 [.120, .334]	.490	.576
SemStamp	.695 [.660, .726]	.718	.114 [.026, .160]	.286	.348
TextSeal	.826 [.801, .849]	.854	.304 [.256, .418]	.490	.586
SemaSync	.779 [.751, .806]	.810	.312 [.218, .360]	.402	.500

D.6 Second-Generator Details

This appendix provides additional results for Llama-3.1-8B-Instruct (Table 2). Table 18 gives raw GSM8K detection results and generation accuracy, including the clean reference. Table 19 reports GSM8K and MBPP detection after Mistral rewriting.

D.7 Second-Rewriter Detection Results

The second rewriter repeats the Mistral protocol with GPT-5.6 Luna Max at maximum reasoning effort: identical task prompts per dataset (Appendix D.3), the same acceptance rules (Appendix D.4) and, in every cell, the Luna-rewritten clean corpus of the same condition as the null. Every contrast quoted below is a paired bootstrap over the matched ID set: the same resampled IDs are applied to both methods and to their clean references, each AUROC is recomputed inside the replicate, and the reported bracket is the 95% interval of the resulting differences over $10{,}000$ draws.

Rewrite length and detection across rewriters. Table 20 puts the two rewriters side by side on SemaSync. Mistral leaves a GSM8K solution at 95% of its source length; Luna compresses it to 57%, and 96% of its rewrites are shorter than the source. Both preserve the final answer on every accepted rewrite and fall back to the source at almost the same rate (4.2% vs. 4.4%), so the difference

Table 18: Llama-3.1-8B-Instruct, GSM8K, raw outputs (N=500 watermarked vs. 500 clean).

Method	AUROC	AUPRC	TPR@1%	TPR@5%	TPR@10%	Accuracy
Clean	–	–	–	–	–	.850
KGW	.979	.983	.788	.910	.942	.722
SWEET	.996	.996	.908	.978	.994	.798
MorphMark	.830	.842	.230	.422	.544	.826
MorphMark+EWD	.873	.888	.350	.564	.666	.826
SemStamp	.864	.872	.256	.482	.684	.800
TextSeal	.881	.899	.444	.574	.698	**.850**
Unigram	.964	.968	.676	.848	.894	.638
SIR	.909	.922	.490	.680	.760	.670
SemaSync	**.999**	**.999**	**.982**	**.998**	**1.000**	.776

Table 19: Llama-3.1-8B-Instruct under Mistral rewriting (N=500 watermarked vs. 500 rewritten clean per method).

Method	AUROC	AUPRC	TPR@1%	TPR@5%	TPR@10%
GSM8K — Mistral rewrite					
KGW	.7086	.7359	.2160	.2580	.3240
SWEET	.6682	.7017	.1500	.2160	.2960
MorphMark	.6145	.6466	.0900	.1700	.2680
MorphMark+EWD	.6112	.6425	.0760	.1580	.2580
SemStamp	.6239	.6542	.1120	.1840	.2340
TextSeal	.6024	.6191	.0460	.1440	.2120
Unigram	**.7792**	**.8146**	**.3560**	**.4080**	**.4840**
SIR	.7537	.7858	.2700	.3380	.4300
SemaSync	.7395	.7643	.1860	.2760	.4120
MBPP — Mistral rewrite					
KGW	.6275	.6365	.0520	.1620	.2320
SWEET	.6445	.6452	.0680	.1640	.2220
MorphMark	.5517	.5591	.0240	.0740	.1520
MorphMark+EWD	.5789	.5803	.0280	.0840	.1600
SemStamp	.5358	.5244	.0000	.0000	.1680
TextSeal	.5572	.5682	.0360	.0900	.1460
Unigram	.6183	.6267	.0360	.1260	.2480
SIR	.5571	.5583	.0240	.0800	.1440
STONE	.5456	.5386	.0300	.0540	.1120
SemaSync	**.6689**	**.6735**	**.0700**	**.1740**	.2740

is not in what survives but in how much text is left to carry it: SemaSync's GSM8K AUROC falls from 0.786 under Mistral to 0.542 under Luna. On MBPP the same rewriter costs SemaSync almost nothing (0.694 to 0.690), because a rewritten program must remain syntactically valid.

Table 20: Mistral-Small-3.2-24B vs. GPT-5.6 Luna Max for **SemaSync** (Qwen3.5-9B, N=500 per task). AUROC is reported for GSM8K and MBPP; length and fallback statistics refer to GSM8K.

	Mistral	Luna
GSM8K AUROC	.786	.542
MBPP AUROC	.694	.690
GSM8K rewritten length / source length (median)	.949	.573
GSM8K rewrites shorter than source	78%	96%
GSM8K fallback to source	4.2%	4.4%

GSM8K. Four methods remain above chance—KGW (0.573), SemStamp (0.557), SemaSync (0.542, 95% interval $[0.509,\ 0.575]$) and TextSeal (0.536)—and none of the other three differs from SemaSync by an interval that excludes zero (SemaSync minus KGW -0.031 $[-0.073,\ 0.011]$, minus SemStamp -0.015 $[-0.057,\ 0.028]$, minus TextSeal $+0.006$ $[-0.037,\ 0.049]$). SWEET, Mor-

Table 21: Detection under the GPT-5.6 Luna Max rewrite (Qwen3.5-9B). Each cell scores the watermarked rewrites against the Luna-rewritten clean corpus of the same condition.

Method	AUROC	AUPRC	TPR@1%	TPR@5%	TPR@10%
GSM8K					
KGW	**.573**	**.616**	**.090**	**.148**	**.182**
SWEET	.492	.496	.010	.044	.104
MorphMark	.476	.487	.004	.046	.098
MorphMark+EWD	.485	.490	.004	.036	.088
SemStamp	.557	.586	.066	.100	.162
TextSeal	.536	.567	.058	.108	.146
SemaSync	.542	.578	.068	.112	.162
MBPP					
KGW	.667	.672	.062	.168	**.282**
SWEET	**.695**	**.689**	.064	.172	.259
MorphMark	.507	.499	.006	.036	.078
MorphMark+EWD	.500	.506	.010	.060	.110
SemStamp	.499	.500	.010	.010	.010
TextSeal	.559	.575	.028	.136	.174
SemaSync	.690	.686	**.072**	**.176**	**.282**

phMark+EWD and MorphMark (0.476–0.492) are not distinguishable from chance and are significantly below SemaSync ($+0.051$ $[0.007, 0.093]$, $+0.057$ $[0.015, 0.099]$, $+0.066$ $[0.026, 0.106]$).

MBPP. SWEET (0.695), SemaSync (0.690) and KGW (0.667) form a first tier in which neither baseline differs from SemaSync by an interval that excludes zero (SemaSync minus SWEET -0.006 $[-0.048, 0.034]$ on the common 499 tasks; minus KGW $+0.023$ $[-0.018, 0.065]$). Compared with SWEET and KGW, SemaSync has the highest or joint-highest TPR at every operating point ($0.072/0.176/0.282$). TextSeal (0.559) is above chance but significantly below SemaSync ($+0.130$ $[0.090, 0.171]$); MorphMark, MorphMark+EWD and SemStamp (0.499–0.507) are not distinguishable from chance. Paired AUROC-difference intervals do not distinguish SemaSync from KGW, SemStamp, or TextSeal on GSM8K, or from SWEET or KGW on MBPP.

Why the two datasets differ. SemaSync's evidence is bound to the anchor sequence of the visible text. A GSM8K solution can be shortened to roughly half its length while keeping the final answer, and the anchors are removed with the prose, so the detector reconstructs a different state than the one the watermark used.

D.8 Destructive Stress Test: DIPPER

GSM8K. Table 22 reports detection and utility after DIPPER. The rewrite is destructive rather than meaning-preserving: the exact-answer accuracy of every watermarked corpus falls to 0.376–0.448, the rewritten clean text is no better at 0.432, and fewer than half of the final answers survive in any corpus (0.392–0.456). Detection collapses with the content: the seven methods span AUROC 0.507–0.607, with SemaSync at 0.576 and KGW highest at 0.607. Since the rewriter has not preserved the task, no ordering in this table should be read as evidence about robustness to paraphrasing.

Table 22: GSM8K under DIPPER (Qwen3.5-9B). Detection on rewritten text; "Attacked acc." is the exact-answer accuracy of the rewritten corpus and "Answer preserved" the fraction of rewritten rows whose final answer equals the source's.

Method	AUROC	AUPRC	TPR@1%	TPR@5%	TPR@10%	Attacked acc.	Answer preserved
Clean	–	–	–	–	–	.432	.440
KGW	.607	.612	.056	.112	.194	.448	.456
SWEET	.549	.532	.006	.046	.118	.400	.408
MorphMark	.507	.496	.010	.036	.080	.376	.392
MorphMark+EWD	.533	.522	.004	.062	.114	.376	.392
SemStamp	.574	.574	.036	.074	.166	.446	.454
TextSeal	.547	.538	.004	.066	.124	.436	.446
SemaSync	.576	.564	.018	.086	.146	.438	.448

MBPP. On MBPP, DIPPER destroys program structure rather than paraphrasing it: after rewriting, only 2.4% of programs are syntactically valid and Pass@1 is 0.012 for the clean and watermarked corpora. SemaSync obtains AUROC 0.536 (95% CI $[0.500, 0.574]$). We therefore regard DIPPER on MBPP as a check that the pipeline degrades gracefully when its input is no longer code, not as an attack evaluation.

D.9 Carrier-Gate Diagnostics

Tables 23 and 24 give the full detection metrics behind the carrier-gate ablation of Table 4, for MBPP and GSM8K respectively: each of the five gate settings is scored on raw and on Mistral-rewritten outputs against the matched clean population of the same condition.

Table 23: Full detection metrics for the MBPP carrier-gate ablation of Table 4

Gate on q_t	Condition	AUROC	AUPRC	TPR@1%	TPR@5%	TPR@10%
none	Raw	.7345	.7711	.220	.344	.414
none	Mistral	.6374	.6721	**.114**	.190	.284
$q_t \geq 0.10$ only	Raw	.5881	.5938	.044	.108	.180
$q_t \geq 0.10$ only	Mistral	.5625	.5724	.044	.100	.152
$q_t \leq 0.90$ only	Raw	.7933	.8105	.270	.374	.448
$q_t \leq 0.90$ only	Mistral	.6261	.6442	.088	.170	.236
$[0.10, 0.90]$ (default)	Raw	**.8843**	**.8822**	**.344**	**.548**	.566
$[0.10, 0.90]$ (default)	Mistral	**.6941**	**.7041**	.110	**.266**	**.300**
$[0.20, 0.80]$	Raw	.8511	.8461	.280	.444	**.610**
$[0.20, 0.80]$	Mistral	.6663	.6547	.066	.114	.188

Table 24: Full detection metrics for the GSM8K carrier-gate ablation of Table 4

Gate on q_t	Condition	AUROC	AUPRC	TPR@1%	TPR@5%	TPR@10%
none	Raw	.7876	.8034	.186	.354	.502
none	Mistral	.6443	.6675	.086	.174	.254
$q_t \geq 0.10$ only	Raw	.6800	.7001	.104	.206	.302
$q_t \geq 0.10$ only	Mistral	.5723	.6069	.076	.118	.208
$q_t \leq 0.90$ only	Raw	.8137	.8207	.192	.344	.506
$q_t \leq 0.90$ only	Mistral	.6475	.6463	.056	.118	.220
$[0.10, 0.90]$ (default)	Raw	**.9950**	**.9954**	**.920**	**.980**	**.992**
$[0.10, 0.90]$ (default)	Mistral	**.7861**	**.8169**	**.248**	**.414**	**.518**
$[0.20, 0.80]$	Raw	.9902	.9906	.850	.954	.972
$[0.20, 0.80]$	Mistral	.7270	.7476	.158	.282	.356

D.10 Anchor-State Corruption: Full Grid

The GSM8K corruption grid uses the 499 shared samples because sample 197 is absent from the $K{=}1$ corpus; MBPP uses 500 samples. The uncorrupted GSM8K reference and the main results use different score populations, so this grid is interpreted relative to its own reference.

Table 25 gives the full grid behind Section 4.7 and Figure 2. In every state the detector reconstructs, each anchor observation is independently replaced with probability $\rho \in \{10, 20, 50, 60\}\%$ by a held-out donor identity (GSM8K: 669 identities from 200 held-out solutions; MBPP: 17 identities from the discovery split; the donor need not match the type of the anchor it replaces). The visible text is unchanged, the same corruption is applied to the clean corpus, each arm's thresholds are calibrated once on its own $\rho{=}0$ clean population and held fixed across all corruption levels, and each non-zero ρ is run with three seeds. On GSM8K the $K{=}1$-minus-$K{=}3$ difference in AUROC degradation, by paired bootstrap over the 499 shared IDs, is $+0.030$ $[0.012,\ 0.049]$, $+0.016$ $[-0.004,\ 0.036]$ and $+0.023$ $[0.002,\ 0.043]$ for the three $\rho{=}50\%$ seeds.

Table 25: Anchor-state corruption on Qwen3.5-9B. GSM8K uses the paired $N{=}499$ population (sample 197 is absent from the $K{=}1$ corpus); MBPP uses $N{=}500$. Each arm's thresholds are calibrated on its own $\rho{=}0$ clean population and held fixed across ρ.

Arm	ρ	AUROC	TPR@1%	TPR@5%	TPR@10%
GSM8K $K{=}3$	0	.996	.894	.982	.994
	10%	.9907	.806	.949	.983
	20%	.9798	.705	.887	.947
	50%	.8948	.299	.579	.716
	60%	.836	.194	.432	.581
GSM8K $K{=}1$	0	.9953	.878	.972	.990
	10%	.9902	.800	.945	.976
	20%	.980	.705	.894	.945
	50%	.916	.349	.628	.755
	60%	.869	.229	.502	.651
MBPP $K{=}3$	0	.883	.336	.546	.562
	10%	.871	.305	.517	.537
	20%	.859	.271	.481	.504
	50%	.830	.184	.383	.414
	60%	.821	.149	.344	.375
MBPP $K{=}1$	0	.862	.288	.464	.638
	10%	.854	.272	.444	.619
	20%	.844	.253	.429	.603
	50%	.811	.187	.335	.527
	60%	.799	.163	.309	.486

D.11 Runtime Measurements

Detection. Table 26 reports detection times on one H100 with Llama-3.1-8B on the same 50 MBPP task IDs (11–60). Models and assets are loaded before timing. Each method runs three warm-up samples, and each task is timed three times; we report the median of the per-task medians. Model loading is excluded. KGW, TextSeal, STONE, SemStamp, Unigram, and SIR run in milliseconds. SemaSync, SWEET, and MorphMark+EWD use model replay, with SemaSync having the lowest median among these three in this benchmark. MorphMark detection does not require the base LLM or perform model forward passes; its 0.622 s median reflects full-vocabulary partition construction in the current GPU implementation. These timings reflect both scoring requirements and implementation costs.

Generation. Table 27 reports generation times under a shared protocol. Generation uses batch size 1 and frozen per-task seeds; the median is taken over the same 50 prompts, each measured three times. SemaSync is the fastest among the nine measured watermark-generation implementations, with a median of 0.535 s per sample versus 0.490 s without watermarking, an overhead of 9.1% per sample and 7.1% per token. SemStamp takes 6.905 s per sample with sentence-level rejection sampling. MorphMark+EWD uses MorphMark's generated texts and has no separate generation arm.

Table 26: Detection time on Llama MBPP, in seconds per sample and per token.

Method	Median	s/token
KGW	.0020	.00006
TextSeal	.0035	.00010
STONE	.0056	.00017
SemStamp	.0086	.00025
Unigram	.0249	.00073
SIR	.0345	.00106
SemaSync	.5870	.01716
MorphMark	.6219	.01816
SWEET	1.8066	.05283
MorphMark+EWD	2.3695	.06971

Table 27: Generation time under the same timing protocol.

Method	Median	s/token
No watermark	*.490*	*.0159*
SemaSync	.535	.0170
STONE	.556	.0162
TextSeal	.576	.0160
MorphMark	1.108	.0346
Unigram	1.156	.0393
KGW	1.257	.0354
SIR	1.405	.0333
SWEET	1.706	.0526
SemStamp	6.905	.2050

D.12 State Source and Initialization Ablations

Configurations and protocol. We evaluate Qwen3.5-9B on the same 500 tasks per dataset as the main experiments, using matched prompts and per-task seeds. All arms retain $\gamma{=}0.5$, $\delta{=}3$, the gate $q_t \in [0.1, 0.9]$, and the existing decoding and deduplicated scoring rules. *Token + Gate* reuses the MBPP initializer's domain-separated partition construction throughout generation and detection: its state uses the previous ordinary token, and never switches to semantic anchors. Full SemaSync uses semantic states on GSM8K and the initializer-to-semantic transition on MBPP. The MBPP *without-initializer* arm neither embeds nor scores before the first semantic anchor completes. It then uses the same semantic-state rule as the full method: one voter when one or two anchors are available, and three when at least three are available. It does not wait for three anchors to start embedding.

We use the main Mistral rewriting protocol and calibrate thresholds on each arm's matched clean population for the corresponding raw or rewritten condition. The full and without-initializer MBPP arms reuse existing corpora, with all 500 task IDs aligned in both conditions. Utility below measures original-generation correctness. Table 28 reports the detection and utility results.

Table 28: Full state-source and initialization results on Qwen3.5-9B ($N{=}500$ per task). TPR is reported at the indicated FPR. Utility denotes original-generation accuracy (GSM8K) or Pass@1 (MBPP). The no-watermark utility references are 0.944 and 0.560, respectively.

	Raw		Mistral rewrite			
Configuration	AUROC	TPR@1%	AUROC	TPR@1%	TPR@5%	Utility
GSM8K						
Token + Gate	.997350	.940	.750812	.204	.316	.920
SemaSync	.994972	.920	.786122	.248	.414	.928
MBPP						
Token + Gate	.897508	.440	.682472	.102	.224	.538
SemaSync	.884316	.344	.694080	.110	.266	.538
SemaSync w/o initializer	.740998	.250	.583042	.064	.140	.564

State source. Token + Gate achieves higher raw AUROC and TPR on both tasks. After rewriting, SemaSync improves AUROC by 0.0353 on GSM8K and 0.0116 on MBPP; TPR at 1% FPR increases by 4.4 and 0.8 percentage points, respectively. At 5% FPR, the corresponding gains are 9.8 and 4.2 percentage points. The two MBPP arms have the same observed Pass@1 (0.538), while GSM8K accuracy is 0.928 for SemaSync and 0.920 for Token + Gate. These comparisons support improved detection after rewriting under the tested protocol, with a modest gain on MBPP. They evaluate the complete state construction; the separate $K{=}1$ versus $K{=}3$ ablation in Section 4.6 examines voting. The Token + Gate comparisons are point estimates, without paired confidence intervals.

Removing the MBPP initializer. Without initialization, raw AUROC falls from 0.8843 to 0.7410 and rewritten AUROC from 0.6941 to 0.5830. Paired task-level bootstrap estimates using $10{,}000$ shared resamples give differences (without initializer minus full SemaSync) of -0.1433 with a 95% interval $[-0.1658, -0.1212]$ on raw text, and -0.1110 with interval $[-0.1402, -0.0824]$ after rewriting. Pass@1 rises from 0.538 to 0.564, compared with 0.560 for clean generation. Thus initialization contributes substantially to detection in this setting, alongside a generation-utility tradeoff; this comparison is not utility matched.

Initializer and semantic evidence on the same texts. We additionally rescore the full MBPP arm using only initializer positions (BOOT), only semantic positions (SEM), or their combined evidence (Table 29). These are detection-side analyses of the same generated texts, with zero-evidence samples retained.

The initializer supplies 26.1% of scored positions on raw text and 29.6% after rewriting. Its single-channel AUROC is higher than the semantic channel's, whereas the semantic channel has higher TPR at 1% FPR in both conditions. Combined evidence yields the highest AUROC and TPR among these three scoring choices. On raw watermarked text, combining the channels reduces the zero-

Table 29: Channel-wise rescoring of the full MBPP arm (500 watermarked and 500 matched clean outputs per condition). Counts are mean deduplicated scored positions per sample. Zero-evidence columns count samples with no scored positions. Combined denotes the full detector.

				Scored	Zero evidence	
Condition	Channel	AUROC	TPR@1%	positions	WM	Clean
Raw	BOOT	.813168	.128	2.384	63/500	65/500
	SEM	.765930	.266	6.760	172/500	201/500
	Combined	.884316	.344	9.144	35/500	35/500
Mistral rewrite	BOOT	.677578	.048	2.178	71/500	69/500
	SEM	.615204	.068	5.174	156/500	176/500
	Combined	.694080	.110	7.352	32/500	30/500

evidence fraction from 34.4% for SEM alone to 7.0%. These results show complementary coverage and useful evidence from both phases. The position shares describe coverage, not an additive decomposition of AUROC or detection gains.