When Words Change but Meaning Remains: Robust Semantic–Stylistic Watermarking for LLMs

Anonymous authors
Paper under double-blind review

Abstract

Watermarking can support provenance verification for text generated by large language models (LLMs), but many existing methods encode evidence in token-level statistics that degrade under meaning-preserving rewriting. We introduce SS-WM, a base-model-training-free watermarking framework that distributes key-conditioned evidence across a joint semantic–stylistic carrier. The semantic channel uses a projected representation derived from a frozen sentence encoder, while the stylistic channel uses distributional statistics of textual realization. A secret key selects carrier constraints, and inference-time reranking softly steers generation toward the resulting key-specific region while preserving prompt fidelity. Detection extracts the same carrier feature families from completed text and performs a keyed statistical hypothesis test, supporting candidate-key verification after paraphrasing, translation, summarization, stylistic rewriting, truncation, and partial reuse. SS-WM does not assume that individual semantic coordinates correspond to interpretable discourse structures; instead, robustness relies on empirical transformation stability and complementarity with stylistic evidence. On LLaMA-2-7B across fixed, composed, and adaptive transformations, with additional cross-model evaluation on LLaMA-2-13B, Mistral-7B, and OPT-2.7B, SS-WM improves watermark retention over the evaluated token-level and semantics-aware baselines while maintaining strong clean-text detection and limited quality degradation. Additional analyses characterize prefix-to-sequence surrogate alignment, stylistic feature contributions, wrong-key behavior, prompt compatibility, and the trade-off between logit-access and black-box deployment.

1 Introduction

Large language models (LLMs) are increasingly used for content creation, search, software assistance, education, and decision support, making reliable provenance of generated text an important deployment requirement (Kirchenbauer et al., 2023; Liu et al., 2024b). Text watermarking is attractive because it embeds evidence directly in generated content, without requiring external metadata to survive every downstream transformation (Guo et al., 2024b; Pang et al., 2024). However, many practical watermarks encode evidence in token selections, token ranks, or local probability statistics (Kirchenbauer et al., 2023; Giboulot & Furon, 2024; Huo et al., 2024). These signals can be weakened by paraphrasing, translation, summarization, style rewriting, truncation, and partial reuse even when the useful meaning of the text is retained (Krishna et al., 2023; Kirchenbauer et al., 2024; Liang et al., 2024).

Recent semantics-aware methods improve robustness by moving part of the signal away from exact lexical form (Liu et al., 2024a; Ren et al., 2024; Dabiriaghdam & Wang, 2025). This motivates a broader carrier question: can a watermark distribute key-dependent evidence across more than one transformation-stable textual representation so that degradation of one channel does not immediately remove the whole signal? We study this question using a joint carrier consisting of projected semantic embedding features and interpretable stylistic statistics. Importantly, we do not assume that projected semantic coordinates correspond one-to-one with human-interpretable discourse concepts. We use them operationally because we can extract them from original and rewritten text, steer them during generation, and empirically test their transformation stability.

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.

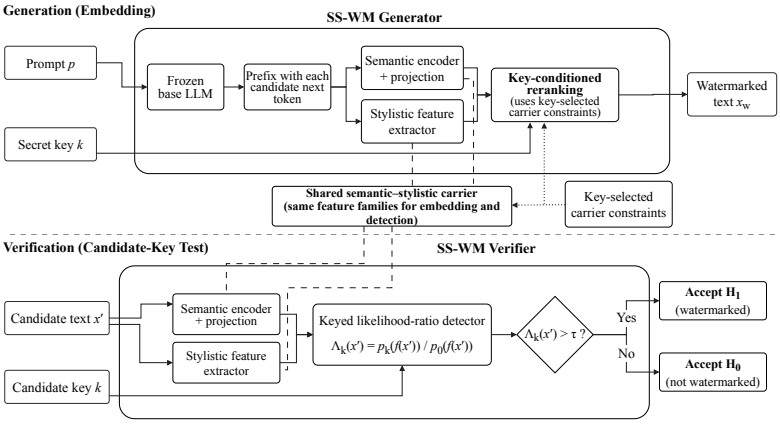


Figure 1: End-to-end view of SS-WM. A frozen LLM generates text under key-conditioned semantic–stylistic guidance; verification tests the extracted carrier under a supplied candidate key.

We propose SS-WM, a semantic–stylistic watermarking framework for candidate-key verification. Given a secret key, a public codebook deterministically selects coordinate-level targets in the semantic and stylistic carrier. The primary implementation uses logit-access reranking: candidate continuations are scored under the base model, a prompt-fidelity term, and key-conditioned semantic and stylistic surrogate penalties. A strict black-box compatibility mode instead selects among completed candidates. The base LLM is never fine-tuned or modified; we train an auxiliary semantic projection offline and then freeze it before watermark generation and detection. This distinction motivates the term *base-model-training-free* throughout the paper. Figure 1 summarizes the end-to-end embedding and candidate-key verification workflow.

The paper makes four contributions. First, we formulate robust watermarking as a transformation-stable carrier problem and introduce a joint projected-semantic/stylistic carrier with key-conditioned constraint selection. Second, we connect sequence-level carrier objectives to practical decoding through explicitly defined prefix-level surrogate penalties, and we evaluate their alignment with the completed-text carrier. Third, we formulate candidate-key verification through a keyed likelihood-ratio detector and distinguish calibrated null false-positive control from the stronger, unclaimed problem of registry-wide attribution. Fourth, we evaluate clean detection, transformation robustness, carrier drift, channel and feature-family ablations, wrong-key behavior, prompt compatibility, text quality, and deployment trade-offs against token-level and semantics-aware baselines.

2 Related Work

Token- and probability-based watermarks modify sampling through keyed token partitions, logit biases, token ranks, or related decoding statistics and detect the resulting lexical signal statistically (Kirchenbauer et al., 2023; Huo et al., 2024; Park et al., 2026; WONG et al., 2025; Wouters, 2024). They are efficient and often provide strong clean-text detection, but reliability studies show substantial degradation after meaning-preserving rewriting (Kirchenbauer et al., 2024; Krishna et al., 2023; Liang et al., 2024).

The closest line of work uses semantic information to improve robustness. SIR exploits semantic invariance (Liu et al., 2024a); SemaMark uses semantic structure for paraphrase robustness (Ren et al., 2024); SimMark uses sentence-level semantic similarity (Dabiriaghdam & Wang, 2025); and Waterfall studies scalable robust provenance (Lau et al., 2024). SemStamp/k-SemStamp and Unigram Watermark are also directly relevant comparison points because they target paraphrase robustness or strong statistical detection (Hou et al., 2024; Zhao et al., 2024). Unlike semantic-only designs, SS-WM distributes keyed evidence over two carrier families: a projected semantic embedding

representation and a vector of stylistic statistics. This contributes a carrier-combination question rather than a claim that semantic embedding coordinates themselves have direct discourse-level interpretations.

Stylometry, authorship attribution, and model fingerprinting provide complementary evidence that distributional writing regularities can remain informative after lexical changes (Tan et al., 2025; Nasery et al., 2025). These approaches are generally passive, whereas SS-WM actively steers selected carrier coordinates according to a secret key. Appendix B provides a more detailed positioning.

3 METHODOLOGY

3.1 PROBLEM SETTING AND DEPLOYMENT SCOPE

Let $\mathcal{M}$ denote a pre-trained LLM generating a sequence $x = (w_1, \ldots, w_T)$ conditioned on prompt p under

$$P_0(x \mid p) = \prod_{t=1}^{T} P_0(w_t \mid w_{<t}, p). \tag{1}$$

Throughout, "attribution" means verification under a supplied candidate key. Given a secret key k, SS-WM seeks a parameter-preserving inference procedure that produces useful text while inducing a detectable key-dependent carrier distribution. The primary verification task is *candidate-key verification*: given text x' and a specified key k, determine whether x' is compatible with the watermark distribution induced by k. We do not claim open-set identification over an arbitrarily large key registry; that stronger setting requires family-wise error control, key-capacity analysis, and ambiguity testing.

The primary interface exposes candidate-token scores and supports decoding-time reranking. A strict black-box compatibility extension samples complete candidates and applies the same carrier objective at sequence level. The latter is weaker and more expensive because it cannot influence generation incrementally.

3.2 CARRIER PRINCIPLE AND REPRESENTATION

We define a joint carrier

$$z(x) = \left[z^{\mathrm{sem}}(x), z^{\mathrm{sty}}(x)\right]. \tag{2}$$

where $z^{\mathrm{sem}}(x)$ is a projected semantic embedding representation and $z^{\mathrm{sty}}(x)$ is a vector of stylistic statistics. The semantic component is not assumed to provide a coordinate-wise linguistic decomposition of topic transitions, argument flow, or information ordering. Instead, it is selected operationally for three properties: extractability from transformed text, controllability through reranking, and low empirical drift under the transformation family of interest. The stylistic channel captures how content is realized through sentence-length, punctuation, discourse-marker, syntactic, part-of-speech, function-word, lexical-diversity, average-word-length, and coarse n-gram statistics.

For a transformation T, the carrier-stability experiment uses the following representation-specific metrics. Semantic drift is measured by cosine dissimilarity,

$$D_T^{(\mathrm{sem})}(x) = 1 - \cos(z^{\mathrm{sem}}(T(x)), z^{\mathrm{sem}}(x)). \tag{3}$$

while stylistic drift is measured by normalized Euclidean displacement,

$$D_T^{(\mathrm{sty})}(x) = \frac{\|z^{\mathrm{sty}}(T(x)) - z^{\mathrm{sty}}(x)\|_2}{\|z^{\mathrm{sty}}(x)\|_2}. \tag{4}$$

Token drift is computed from normalized lexical-feature divergence, and the joint carrier uses the normalized combined distance in the concatenated semantic–stylistic representation. These quantities are used only to compare relative transformation stability across representation spaces; they are distinct from the standardized coordinates used by the keyed likelihood-ratio detector.

Table 1: Auxiliary semantic-projection architecture and training configuration.

Item	Implementation
Input dimension	384 (all-MiniLM-L6-v2)
Output carrier dimension d_s	32
Hidden dimensions	[256, 128, 64, 32]
Residual topology	Skip connections from layer-1 input to layer-3 input and from layer-2 input to layer-4 input, each via a learned linear projection when dimensions differ
Initialization	Kaiming (He) uniform, gain = $\sqrt{2}$ for ReLU
Training objective	NT-Xent contrastive loss, temperature $\tau = 0.07$
Training targets	Positive pairs: sentences within the same paragraph at distance ≤ 3 sentences in WikiText-103; negatives: in-batch sentences from different documents
Training split	50,000 training pairs / 5,000 validation pairs from the WikiText-103 training split; disjoint from all test and calibration data
Optimizer / LR	AdamW, lr = 1×10^{-4}, $\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay = 1×10^{-5}
Batch size / epochs	256 pairs per batch; 20 epochs maximum; early stopping with patience 3 on validation NT-Xent loss
Sharing policy	Single projection shared across all keys and all generator models evaluated here (LLaMA-2-7B, LLaMA-2-13B, Mistral-7B, OPT-2.7B)
Frozen at inference	Yes

3.3 Auxiliary Semantic Projection and Stylistic Extractor

The semantic channel begins with frozen Sentence-BERT `all-MiniLM-L6-v2`, which produces a 384-dimensional embedding. An auxiliary four-layer fully connected residual projection g_{sem} maps this embedding into a d_s-dimensional semantic carrier and is frozen during watermark generation and detection. The base LLM is never updated. The projection is trained offline using disjoint WikiText-103 representations. Table 1 summarizes its architecture and training configuration.

The stylistic channel is a 128-dimensional vector. Table 2 makes the feature families explicit; the exact coordinate inventory must sum to 128.

3.4 Key-Conditioned Constraint Codebook

Let $\mathcal{C} = (c_1, \ldots, c_M)$ be a public indexed codebook. In the current implementation, $M = 10{,}000$, and deterministic SHA-256 indexing selects $1{,}000$ entries per key, divided evenly between semantic and stylistic channels. The codebook is an indexed multiset rather than a set of $10{,}000$ unique coordinate–bin tuples: repeated tuples are permitted, and repeated selections are resolved at the coordinate level as described below. Each semantic entry identifies a coordinate of the projected semantic carrier and a target interval; each stylistic entry analogously targets a coordinate of the 128-dimensional stylistic representation. The key therefore selects a statistically distinctive carrier region rather than a meaningful semantic message or discourse template.

Calibration means μ_j and standard deviations σ_j are estimated from the held-out calibration set, and each carrier coordinate is standardized as $\tilde{z}_j = (z_j - \mu_j)/\sigma_j$. Each standardized semantic coordinate ($d_s = 32$) and stylistic coordinate (128 dimensions) is partitioned into $B = 20$ equal-width bins over the fixed interval $[-4, 4]$. Each codebook entry c_i is a tuple $(j, b_j^{\text{lo}}, b_j^{\text{hi}}, \tau_j)$ specifying a coordinate index j, standardized bin boundaries, and standardized target value $\tau_j = (b_j^{\text{lo}} + b_j^{\text{hi}})/2$. When multiple selected constraints target the same coordinate j, their target values are averaged: $[\mu_k]_j = |C_{k,j}|^{-1} \sum_{c \in C_{k,j}} \tau_j(c)$, where $C_{k,j}$ is the subset of C_k with coordinate index j. Coordinates with no selected constraint are excluded from the watermark penalty. In the watermark objective below, z^{sem} and z^{sty} denote these calibrated standardized carrier coordinates. Let $m_k^{\text{sem}} \in \{0,1\}^{d_s}$ and $m_k^{\text{sty}} \in \{0,1\}^{128}$ denote binary masks whose active coordinates are the distinct indices selected by C_k^{sem} and C_k^{sty}, respectively. Unselected coordinates therefore contribute zero to the watermark objective regardless of their values.

Table 2: Stylistic feature inventory. The 128-dimensional carrier combines sentence-, lexical-, syntactic-, discourse-, and n-gram-level statistics. All frequency features are normalized by the corresponding token or n-gram count.

Family	Dims.	Statistics	Extraction / update
Sentence length / rhythm	10	Mean, std., min, max, and six percentiles (10, 25, 40, 60, 75, 90) of sentence token counts	spaCy sentence segmentation; sentence boundary
Punctuation	15	Relative frequencies of 15 punctuation categories (period, comma, semicolon, colon, !, ?, dashes, brackets, quotes, apostrophe, ellipsis, slash)	Regex; incremental
Discourse markers	20	Relative frequencies of 20 PDTB-style connective categories, including contrast, concession, cause, result, elaboration, condition, temporal relation, purpose, comparison, summary, example, restatement, addition, alternative, and related classes	Fixed lexicon; incremental
Syntactic depth	5	Mean, std., min, max, and median sentence-level dependency depth	spaCy dependency parser; sentence boundary
POS ratios	12	Relative frequencies of selected Universal POS categories (NOUN, VERB, ADJ, ADV, PRON, DET, ADP, CCONJ, NUM, PART, INTJ, PUNCT)	spaCy POS tagger; incremental
Function words	20	Relative frequencies of 20 fixed high-frequency function words	Fixed lexicon; incremental
Lexical diversity	3	Type–token ratio, distinct-1, and distinct-2	Rolling token sets; incremental
Average word length	1	Mean character length of non-punctuation tokens	Rolling average; incremental
Coarse n-grams	42	Self-BLEU-1–4 (4), n-gram entropy for $n = 1, \ldots, 4$ (4), and 34 coarse frequency-bin statistics over word and character n-grams	NLTK; prefix / sentence level
Total	**128**		

Implementation versions: spaCy v3.7.2 with `en_core_web_sm-3.7.1`, and NLTK v3.8.1. The fixed punctuation, discourse-marker, function-word, and coarse-frequency lexicons are included in the released implementation.

The resulting target vectors are denoted μ_k^{sem} and μ_k^{sty}. We define the masked sequence-level penalties

$$\Phi_{\mathrm{sem}}(x;k) = \left\| \left(z^{\mathrm{sem}}(x) - \mu_k^{\mathrm{sem}}\right) \odot m_k^{\mathrm{sem}} \right\|_2^2, \tag{5}$$

$$\Phi_{\mathrm{sty}}(x;k) = \left\| \left(z^{\mathrm{sty}}(x) - \mu_k^{\mathrm{sty}}\right) \odot m_k^{\mathrm{sty}} \right\|_2^2. \tag{6}$$

Equivalently, each sum is taken only over the coordinates selected by the key.

3.5 Embedding with Prefix-Level Surrogate Reranking

The ideal sequence-level target is

$$P_k(x \mid p) \propto P_0(x \mid p) \exp[-\lambda_{\mathrm{sem}} \Phi_{\mathrm{sem}}(x;k) - \lambda_{\mathrm{sty}} \Phi_{\mathrm{sty}}(x;k)]\,. \tag{7}$$

Exact sampling is intractable because the carrier penalties depend on an evolving sequence. In the logit-access implementation, SS-WM therefore evaluates local surrogate penalties on the candidate-appended prefix. At a reranking step t, candidate w is scored as

$$\widetilde{\ell}_t(w) = \ell_t(w) - \lambda_{\mathrm{fid}} s_{\mathrm{fid}}(w;t) - \lambda_{\mathrm{sem}} \widehat{\Phi}_{\mathrm{sem}}(x_{t-1} \oplus w; k) - \lambda_{\mathrm{sty}} \widehat{\Phi}_{\mathrm{sty}}(x_{t-1} \oplus w; k). \tag{8}$$

The semantic surrogate uses the projected embedding of the growing prefix. Stylistic features that can be updated incrementally are maintained online, whereas sentence- or parse-dependent quantities become informative mainly after clause or sentence completion. Reranking is performed over the top-$K = 50$ candidates every five decoding steps; between reranking steps, the base decoding distribution is used unchanged. Thus, the mechanism is approximate prefix-level steering toward the completed-text carrier, not exact token-level control of global discourse structure.

The prompt-fidelity term is distinct from the semantic watermark carrier:

$$s_{\text{fid}}(w;t) = 1 - \cos(E(x_{t-1} \oplus w), \bar{e}_p)\,. \tag{9}$$

where $\bar{e}_p$ is a prompt-conditioned semantic anchor. Current experiments use fixed $\lambda_{\text{fid}} = 1.0$, $\lambda_{\text{sem}} = 2.5$, and $\lambda_{\text{sty}} = 2.0$; there is no dynamic gate that disables the watermark when prompt constraints become narrow. We therefore evaluate prompt compatibility explicitly in Section 4.5.

3.6 Candidate-Key Detection

Given candidate text x' and specified key k, the detector extracts the completed-text semantic and stylistic carrier features, applies the same calibration normalization used during fitting, and evaluates

$$H_0 : f(x') \sim P_{\text{null}}, \qquad H_1 : f(x') \sim P_k. \tag{10}$$

We use factorized Gaussian density estimates on normalized features and compute

$$\Lambda_k(x') = \frac{p_k(f(x'))}{p_0(f(x'))}. \tag{11}$$

For each candidate key used in evaluation, the key-conditioned density p_k is fitted offline from 1,000 held-out watermarked feature vectors generated under that key; p_0 is fitted from a separate set of 1,000 held-out non-watermarked feature vectors. These detector-fitting samples are disjoint from the 5,000 non-watermarked C4 passages used for threshold calibration and from all reported test sets. A single global threshold τ is calibrated once from the pooled null-score distribution obtained by evaluating the 5,000 non-watermarked C4 passages under the evaluated candidate-key set; it targets the desired overall null false-positive rate and is then reused across candidate keys and attack conditions. Sliding windows of size 100 with stride 50 support partial-text verification; window log-likelihood ratios are summed to obtain the text-level statistic used for the decision. This is a candidate-key test: the current theory does not provide family-wise error control for searching an arbitrarily large key registry.

3.7 Statistical Properties and Scope

Appendix E gives the formal assumptions and proofs. The analysis establishes calibrated null false-positive control under the global threshold, a concentration-based characterization of detection power, local stability of the likelihood-ratio statistic under bounded carrier drift, and partial-text aggregation under weak dependence. These are statistical properties under explicit assumptions, not cryptographic unforgeability. Wrong-key separation between P_k and $P_{k'}$ is evaluated empirically rather than inferred from null calibration.

4 Experiments

4.1 Setup and Comparison Protocol

The main experiments use LLaMA-2-7B, with additional generator results for Mistral-7B and OPT-2.7B; LLaMA-2-13B is used as a perplexity evaluator and a cross-model test generator. Prompts include C4, Natural Questions, CNN/DailyMail, and additional task prompts described in Appendix F, with disjoint WikiText-103 text used for auxiliary carrier construction. Unless otherwise stated, decoding uses nucleus sampling with temperature 1.0, top-$p = 0.95$, length 200, and thresholds calibrated on 5,000 non-watermarked C4 samples. All detectors are compared at the same 1% clean-text FPR, and we report TPR, ROC-AUC, F1, quality, and deployment cost.

The baseline set includes KGW, SIR, EXP-Edit, SemStamp, k-SemStamp, SemaMark, and Unigram Watermark. SimMark is discussed as a closely related method but is excluded from the quantitative table because a directly comparable run is unavailable. SemStamp, k-SemStamp, SemaMark, and Unigram Watermark use the authors' released implementations and published default hyperparameters. The only evaluation-level modification is recalibration of each detector threshold on the same held-out 5,000 non-watermarked C4 samples to match the 1% clean-text FPR operating point used throughout this paper. Generation settings (temperature 1.0, top-$p = 0.95$, length 200) are matched where each method's decoding interface permits.

Table 3: Robustness at 1% clean-text FPR. Values are TPR (%). "Worst" is the lowest TPR among the representative fixed paraphrase, translation, summarization, and composed settings in this comparison; it excludes the more intensive settings of Table 11.

Method	Carrier	Clean	Paraphrase	Translation	Summarization	Worst
KGW-1	Token	98.4	59.0	63.2	42.5	28.5
EXP-Edit	Token	99.5	43.5	52.0	42.5	32.5
Unigram WM	Token/statistical	98.6	56.8	54.8	43.7	31.6
SIR	Semantic	98.7	74.0	80.5	68.5	68.5
SemStamp	Semantic	98.5	65.4	80.1	73.6	53.9
k-SemStamp	Semantic	98.7	71.1	83.9	79.1	64.6
SemaMark	Semantic	98.4	68.8	81.7	65.8	61.9
SS-WM	Semantic+style	**99.1**	**92.6**	**89.4**	**81.2**	**76.9**

Table 4: Carrier-drift diagnostics under the representation-specific metrics defined in Section 3.2. Lower values indicate greater stability. Since the four columns use different normalized measures, the table is intended for within-column comparison across attacks.

Attack	Token	Semantic	Stylistic	Joint Carrier
GPT-3.5 Paraphrase	0.852	0.352	0.408	0.218
DIPPER-2	0.808	0.325	0.385	0.195
Translation (1-hop)	0.745	0.298	0.362	0.182
Summarization (0.5)	0.688	0.275	0.340	0.165
Style Rewrite	0.712	0.312	0.355	0.188
Composed Attack	0.835	0.378	0.415	0.240

4.2 Main Robustness Results

Table 3 compares SS-WM against both token-level and semantics-aware methods at the same 1% clean-text FPR operating point.

The experiments cover GPT-3.5 and DIPPER (Krishna et al., 2023) paraphrasing, one- to three-hop translation, summarization at several retention ratios, style rewriting, partial reuse, composed laundering, constrained adaptive selection, and score-aware candidate selection.

4.3 Carrier and Embedding Validation

Carrier drift. We evaluate token, semantic, stylistic, and joint carrier drift using the carrier-stability protocol defined in Section 3.2: normalized lexical-feature divergence for token features, cosine dissimilarity for the semantic carrier, normalized Euclidean displacement for the stylistic carrier, and the corresponding normalized combined distance for the joint representation. The reported drift values use these representation-specific metrics. Table 4 reports representation-specific drift under each attack. Because each carrier uses its own normalized metric, we interpret the table primarily by comparing attack severity within each representation rather than by comparing absolute values across columns.

These empirical drift diagnostics are descriptive stability measures; they are not substituted numerically for the Euclidean feature-drift quantity $\Delta_{\mathcal{T}}$ used in Theorem 3.

Channel and stylistic-family ablation. The semantic-only and stylistic-only results in Table 17 show that both channels are individually detectable and that their combination is stronger. Table 5 removes one stylistic family at a time to identify which families contribute to robustness.

Prefix-to-sequence alignment. To validate the connection between decoding-time surrogates and completed-text detection, we record cumulative semantic and stylistic surrogate penalties during generation and compare them with the corresponding completed-text carrier penalties. For the detector comparison, we use the negative accumulated joint surrogate energy so that larger values indicate better key compatibility, matching the direction of the final $\log \Lambda_k$.

Table 5: Stylistic-family ablation: TPR (%) at 1% FPR.

Configuration	Clean	Paraphrase	Composed
Full SS-WM	99.1	92.6	76.9
− sentence length/rhythm	98.8	91.2	75.0
− syntax/POS	98.7	90.5	74.2
− discourse markers	98.4	88.8	72.1
− punctuation	99.0	92.1	76.1
− function words	98.8	90.8	73.5
− lexical diversity/n-grams	98.9	91.7	75.5

Table 6: Alignment between accumulated prefix surrogates and completed-text quantities.

Comparison	Pearson r	Spearman ρ
Semantic surrogate vs. final semantic penalty	0.76	0.73
Stylistic surrogate vs. final stylistic penalty	0.84	0.81
Negative joint surrogate energy vs. final $\log \Lambda_k$	0.79	0.76

The prefix-to-sequence correlation analysis in Table 6 is used as the primary validation of the local surrogate approximation.

4.4 Wrong-Key Behavior and Verification Scope

Table 7 compares correct-key detection and wrong-key acceptance, both expressed as percentages. Null false positives and wrong-key acceptance are different quantities. For each watermarked sample generated under k, we therefore test independently sampled mismatched keys $k' \neq k$ and report the empirical wrong-key acceptance distribution rather than repeating the calibrated 1% null FPR.

For each attack condition, we evaluate 1,000 watermarked texts. We sample 50 mismatched keys $k' \neq k$ uniformly at random once and hold this same mismatched-key set fixed across all five attack conditions. Wrong-key acceptance is the fraction of the resulting $1{,}000 \times 50 = 50{,}000$ (x', k') evaluations for which $\Lambda_{k'}(x') > \tau$. A single global threshold τ, calibrated from pooled null scores on 5,000 clean non-watermarked C4 passages, is used without attack-specific recalibration. The 95% confidence intervals are obtained by non-parametric bootstrap with 1,000 resamples over texts, treating the 50 wrong-key evaluations for each text as a within-sample average. The paper's deployment claim remains candidate-key verification.

4.5 Prompt Compatibility

Because SS-WM uses fixed watermark weights, we evaluate whether key-selected targets remain compatible with tasks that offer progressively less output freedom. These are direct task-generation tests before any post-generation laundering attack and are therefore distinct from the BART summarization attack in Table 10. We include open-ended explanation, summarization, factual QA, and an exact short-answer regime in which the prompt requires only the reference answer with no explanatory text. Table 8 summarizes detection and task quality in these regimes.

4.6 Quality, Efficiency, and Limitations

Automatic quality evaluation reports perplexity, ROUGE-L, QA EM/F1, self-BLEU, distinct-n, and Sentence-BERT semantic similarity. Because the same sentence-embedding family participates in the semantic carrier, we treat Sentence-BERT similarity as a descriptive quality measure rather than as independent semantic evidence. We do not include a human-subject annotation study in this submission; the absence of independent human judgments is therefore a limitation of the current evaluation.

For computational cost, we report the interface-level deployment trade-off in Table 16. Strict black-box reranking requires m complete candidates and therefore increases base-generation cost approximately linearly with m, whereas the logit-access mode performs feature-guided reranking

Table 7: Correct-key TPR and wrong-key acceptance (%) under the global threshold calibrated to 1% null FPR. Confidence intervals are non-parametric bootstrap intervals over texts.

Setting	Correct-key TPR	Wrong-key acceptance	95% CI
Clean	99.1	1.1	[0.7, 1.5]
Paraphrase	92.6	1.3	[0.9, 1.7]
Translation	89.4	1.2	[0.8, 1.6]
Summarization	81.2	1.4	[1.0, 1.9]
Composed	76.9	1.6	[1.2, 2.1]

Table 8: Prompt-compatibility evaluation. TPR values are percentages.

Task regime	Task quality	TPR@1%FPR	Quality change vs. no-WM
Open-ended generation	PPL 6.58 ± 1.45	99.1	$-0.76^{\dagger}$
Summarization	ROUGE-L 0.42 ± 0.05	97.8	-0.01
Factual QA	EM 0.68 ± 0.12	92.4	-0.01
Exact short-answer output	Exact match 0.61 ± 0.09	63.5	-0.04

†For perplexity, quality change is reported as $-(\mathrm{PPL}_{\mathrm{WM}} - \mathrm{PPL}_{\text{no-WM}})$ so that a negative value consistently indicates quality degradation. The absolute perplexity increase is $+0.76$ points (from 5.82 to 6.58).

every five decoding steps. No implementation-specific wall-clock latency measurement is reported for the all-MiniLM-L6-v2 carrier.

The principal limitations are the dependence on an external semantic encoder, approximate prefix-level surrogates, reduced control and increased sampling cost in strict black-box mode, potential degradation for highly constrained outputs, candidate-key rather than registry-wide attribution, the offline cost of fitting key-conditioned detector densities for candidate keys, the absence of an independent human evaluation, and the absence of cryptographic unforgeability. The evaluated attack suite also relies on GPT-3.5, DIPPER, NLLB-200, and BART-era rewriting systems and does not include a current frontier instruction-following rewriter; this remains an important direction for future stress testing. The score-aware detector attack is treated as an oracle-aided stress test beyond the primary threat model rather than as an ordinary black-box adversary.

5 Conclusion

We introduced SS-WM, a base-model-training-free watermarking framework that distributes key-dependent evidence across projected semantic embedding features and stylistic statistics. The method uses prefix-level reranking to steer generation toward a key-specific carrier region and candidate-key statistical verification to recover that evidence after transformation. The semantic-channel claim is limited to what is empirically established: the semantic channel is a transformation-stable embedding carrier, not a claim of direct coordinate-wise control over argument flow or discourse hierarchy. Across the evaluated transformations and baselines, the joint carrier improves robustness while retaining clean-text detectability and output quality. The evaluation includes direct semantic baselines, stylistic-family ablation, prefix-to-sequence alignment, empirical wrong-key analysis, and prompt compatibility.

Reproducibility Statement

The main paper specifies the carrier construction, inference interfaces, candidate-key detector, evaluation protocol, and the distinction between statistical guarantees and empirical wrong-key behavior. The appendices provide the complete feature inventory, auxiliary projection training details, attack configurations, baseline configurations, algorithms, proofs, and extended results. An anonymized implementation, configuration files, and reproduction scripts are available at `https://anonymous.4open.science/r/sswm`.

ETHICS STATEMENT

SS-WM is intended for provenance verification and accountability of machine-generated text. The same carrier-shaping mechanism could in principle be used to mark text covertly for tracking or surveillance purposes; responsible deployment therefore requires disclosure of watermarking to affected parties, auditable key management, calibrated false-positive thresholds, and accessible appeal mechanisms. Our evaluation reports false-positive operating points and scopes the method to candidate-key verification rather than unrestricted identity attribution. No human-subject annotation study is conducted in this submission.

REFERENCES

Amirhossein Dabiriaghdam and Lele Wang. SimMark: A robust sentence-level similarity-based watermarking algorithm for large language models. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 30785–30806, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1567. URL https://aclanthology.org/2025.emnlp-main.1567/.

Eva Giboulot and Teddy Furon. Watermax: breaking the llm watermark detectability-robustness-quality trade-off. *Advances in Neural Information Processing Systems*, 37:18848–18881, 2024.

Thibaud Gloaguen, Nikola Jovanović, Robin Staab, and Martin Vechev. Black-box detection of language model watermarks. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=E4LAVLXAHW.

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

Yuxuan Guo, Zhiliang Tian, Yiping Song, Tianlun Liu, Liang Ding, and Dongsheng Li. Context-aware watermark with semantic balanced green-red lists for large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 22633–22646, 2024b.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28, 2015.

Abe Bohan Hou, Jingyu Zhang, Tianxing He, Yichen Wang, Yung-Sung Chuang, Hongwei Wang, Lingfeng Gao, Matthew McDermott, and James Glass. SemStamp: A semantic watermark with paraphrastic robustness for text generation. In *Proceedings of NAACL*, 2024.

Baizhou Huang, Xiao Pu, and Xiaojun Wan. b4: A black-box scrubbing attack on llm watermarks. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 9113–9126, 2025.

Mingjia Huo, Sai Ashish Somayajula, Youwei Liang, Ruisi Zhang, Farinaz Koushanfar, and Pengtao Xie. Token-specific watermarking with enhanced detectability and semantic coherence for large language models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b. *ArXiv*, abs/2310.06825, 2023.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. *Proceedings of the 40th International Conference on Machine Learning (ICML)*, 2023. URL https://proceedings.mlr.press/v202/kirchenbauer23a.html. arXiv:2301.10226.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Manli Shu, Khalid Saifullah, Kezhi Kong, Kasun Fernando, Aniruddha Saha, Micah Goldblum, and Tom Goldstein. On the reliability of watermarks for large language models. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=DEJIDCmWOz. arXiv:2306.04634.

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 (NeurIPS)*, 36:27469–27500, 2023.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.

Gregory Kang Ruey Lau, Xinyuan Niu, Hieu Dao, Jiangwei Chen, Chuan-Sheng Foo, and Bryan Kian Hsiang Low. Waterfall: Scalable framework for robust text watermarking and provenance for LLMs. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 20432–20466, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.1138. URL https://aclanthology.org/2024.emnlp-main.1138/.

Jiacheng Liang, Zian Wang, Spencer Hong, Shouling Ji, and Ting Wang. Watermark under fire: A robustness evaluation of llm watermarking. In *Conference on Empirical Methods in Natural Language Processing*, 2024. URL https://api.semanticscholar.org/CorpusID:274150116.

Aiwei Liu, Leyi Pan, Xuming Hu, Shiao Meng, and Lijie Wen. A semantic invariant robust watermark for large language models. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 6499–6519, 2024a.

Aiwei Liu, Leyi Pan, Yijian Lu, Jingjing Li, Xuming Hu, Xi Zhang, Lijie Wen, Irwin King, Hui Xiong, and Philip S. Yu. A survey of text watermarking in the era of large language models. *ACM Computing Surveys*, 2024b. arXiv:2312.07913 (v6).

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

Anshul Nasery, Jonathan Hayase, Creston Brooks, Peiyao Sheng, Himanshu Tyagi, Pramod Viswanath, and Sewoong Oh. Scalable fingerprinting of large language models. In *Lock-LLM Workshop: Prevent Unauthorized Knowledge Use from Large Language Models*, 2025. URL https://openreview.net/forum?id=dCu5dh7h4v.

Qi Pang, Shengyuan Hu, Wenting Zheng, and Virginia Smith. No free lunch in llm watermarking: Trade-offs in watermarking design choices. *Advances in Neural Information Processing Systems*, 37:138756–138788, 2024.

Shinwoo Park, Hyejin Park, Hyeseon Ahn, and Yo-Sub Han. Watermod: Modular token-rank partitioning for probability-balanced llm watermarking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 32683–32691, 2026.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

Jie Ren, Han Xu, Yiding Liu, Yingqian Cui, Shuaiqiang Wang, Dawei Yin, and Jiliang Tang. A robust semantics-based watermark for large language models against paraphrasing. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 613–625, 2024. doi: 10.18653/v1/2024.findings-naacl.40. SemaMark; arXiv:2311.08721.

Tarun Suresh, Shubham Ugare, Gagandeep Singh, and Sasa Misailovic. Is the watermarking of llm-generated code robust? *arXiv preprint arXiv:2403.17983*, 2024.

Xinhao Tan, Songhua Liu, Xia Cong, Kunjun Li, and Xinchao Wang. Open-world authorship attribution. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 17744–17758, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.913. URL https://aclanthology.org/2025.findings-acl.913/.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Zhuoshang Wang, Yubing Ren, Yanan Cao, Fang Fang, Xiaoxue Li, and Li Guo. Rethinking llm watermark detection in black-box settings: A non-intrusive third-party framework. *arXiv preprint arXiv:2603.14968*, 2026.

KA HIM WONG, Jicheng Zhou, Jiantao Zhou, and Yain-Whar Si. An end-to-end model for logits-based large language models watermarking. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=9sNiCqi2RD.

Bram Wouters. Optimizing watermarks for large language models. In *International Conference on Machine Learning*, pp. 53251–53269. PMLR, 2024.

Xiaojun Xu, Jinghan Jia, Yuanshun Yao, Yang Liu, and Hang Li. Robust multi-bit text watermark with LLM-based paraphrasers. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=DVjkling5x.

Yifan Yan, Xudong Pan, Mi Zhang, and Min Yang. Rethinking White-Box watermarks on deep learning models under neural structural obfuscation. In *32nd USENIX Security Symposium (USENIX Security 23)*, pp. 2347–2364, Anaheim, CA, August 2023. USENIX Association. ISBN 978-1-939133-37-3. URL https://www.usenix.org/conference/usenixsecurity23/presentation/yan.

Ruisi Zhang, Shehzeen Samarah Hussain, Paarth Neekhara, and Farinaz Koushanfar. {REMARK-LLM}: A robust and efficient watermarking framework for generative large language models. In *33rd USENIX Security Symposium (USENIX Security 24)*, pp. 1813–1830, 2024.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models, 2022.

Xuandong Zhao, Prabhanjan Vijendra Ananth, Lei Li, and Yu-Xiang Wang. Provable robust watermarking for AI-generated text. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=SsmT8aO45L.

AI USE STATEMENT

Generative AI tools were used solely to assist with language editing, sentence refinement, clarity, and organization of the manuscript. They were not used to develop the research methodology, design experiments, generate or analyze experimental results, or draw scientific conclusions. All technical content, literature selection, implementation, experiments, numerical results, and conclusions were developed and verified by the authors. The authors take full responsibility for the final manuscript.

A THREAT MODEL

This section formalizes the threat model considered in SS-WM. Our goal is to capture *realistic adversarial laundering and misuse scenarios* of LLM outputs, while avoiding overly idealized or cryptographic assumptions that do not reflect how LLM-generated text is transformed and reused in practice. Unlike classical watermarking settings that assume limited adversarial manipulation, we explicitly consider adversaries capable of applying powerful semantic-preserving transformations, reflecting the ease with which LLM outputs can be rewritten, translated, or repurposed without degrading utility.

A.1 System Participants

The system consists of three entities. (i) *Watermark Owner (Defender):* An entity that deploys a watermarked LLM and possesses a secret watermark key k. The defender generates text using the watermarked model and later performs attribution on suspicious text samples encountered in downstream settings (Huang et al., 2025). (ii) *Base LLM:* A pre-trained LLM M that generates text conditioned on user prompts. SS-WM does not retrain, fine-tune, or modify the parameters of M. We distinguish between two inference interfaces: a logit-access interface, where candidate-token scores are available for decoding-time reranking, and a strict black-box interface, where only completed text samples are returned, and watermarking is implemented through sequence-level candidate reranking (Ren et al., 2024). (iii) *Adversary:* An external party that obtains text generated by the watermarked LLM and attempts to evade attribution by removing, obscuring, or diluting the watermark signal while preserving the semantic content and practical usefulness of the text (Suresh et al., 2024).

A.2 Adversary Capabilities

We assume a *strong but realistic* adversary with the following capabilities.

Access Model. The adversary has black-box access to the outputs of the watermarked LLM but does not possess the secret watermark key k. We assume the adversary may be aware that semantic–stylistic watermarking is employed and may know the general detection methodology, but does not know the specific key-dependent constraint configuration (Wang et al., 2026).

Semantic-Preserving Transformations. The adversary may apply arbitrary sequences of transformations that preserve the underlying meaning of the text, including: (i) *Paraphrasing*, using other LLMs or manual rewriting; (ii) *Translation*, including single-hop or multi-hop translation across languages; (iii) *Summarization or Expansion*, altering length or granularity while retaining semantic content; (iv) *Stylistic Rewriting*, modifying tone, formality, or discourse style; and (v) *Partial Reuse and Composition*, truncating, quoting, or mixing segments of the text with other content. These operations are collectively denoted by a transformation operator

$$x' = \mathcal{T}(x),$$

where $\mathcal{T}$ preserves the semantic meaning of x up to acceptable task-level fidelity.

Adaptive Strategies. The primary adversary may adaptively choose among transformations based on public information and previous outputs, but does not observe the secret key or receive detector scores for arbitrary candidate texts. We separately report an *oracle-aided score-aware stress test* in which the attacker is deliberately granted the detector score $\Lambda_k(\cdot)$ to select among admissible rewrites. This stress test is stronger than the primary threat model and is not used as an assumption in the security analysis.

A.3 Adversary Limitations

Despite these capabilities, the adversary is subject to three important limitations (Huang et al., 2025). Most fundamentally, the watermark key remains secret, so the adversary cannot reconstruct or reliably imitate the key-specific constraint subset $\mathcal{C}_k$ selected from the public semantic–stylistic codebook. In addition, the adversary is constrained by utility preservation: transformations that substantially distort meaning, degrade readability, or undermine task performance are considered unsuccessful from the adversary's perspective. Finally, the adversary has no access to a perfect detection oracle that reveals watermark presence or identifies violated constraint components. These limitations define the evaluation setting: the relevant question is whether key-agnostic rewriting can reduce the watermark signal while still satisfying the stated semantic-fidelity and utility constraints.

A.4 Security Objectives

Under this threat model, SS-WM is designed to satisfy four security goals. First, it should support robust attribution under semantic-preserving transformations: given a suspicious text x' derived from

a transformed watermarked sample, the defender should be able to detect watermark presence with high confidence,

$$\Pr(\text{Detect}(x') = 1 \mid x' = \mathcal{T}(x)) \geq 1 - \delta.$$

Second, false positives should remain controlled on non-watermarked text,

$$\Pr(\text{Detect}(x) = 1 \mid x \sim P_{\text{null}}) \leq \alpha.$$

Third, watermarking should introduce limited quality degradation, evaluated empirically through perplexity, semantic-similarity, lexical-diversity, and task-specific quality metrics. We do not claim a measured KL-divergence bound or perceptual imperceptibility guarantee. Fourth, attribution should remain feasible even when only a fragment of the generated text is available, reflecting realistic leakage, quotation, and partial-reuse scenarios (Huang et al., 2025).

A.5 Out-of-Scope Attacks

We do not consider attacks that fundamentally destroy the semantics of the text, since such transformations trivially invalidate any content-based watermark and fall outside the intended use case (Suresh et al., 2024). We also exclude key compromise: if the secret watermark key k is leaked, attribution guarantees no longer hold. Finally, we do not consider attacks that modify the internal parameters or decoding logic of the base LLM, as SS-WM operates purely at inference time. These exclusions are standard in the watermarking literature and reflect realistic deployment boundaries (Wang et al., 2026).

A.6 Why Semantic–Stylistic Watermarking

Under the above threat model, token-level and logit-bias watermarking schemes are inherently fragile because surface lexical patterns can be easily erased by paraphrasing, translation, or stylistic rewriting (Ren et al., 2024). In contrast, projected semantic embedding features and stylistic statistics are more global and distributed properties of text, and therefore tend to persist more reliably under semantic-preserving transformations (Liu et al., 2024a). By embedding attribution signals into these empirically more stable carrier features, SS-WM better aligns the watermark carrier with the strongest realistic adversaries—those who seek to preserve meaning while laundering surface form—and thus targets settings in which the evaluated token-level schemes degrade substantially.

B Detailed Related Work

Work on provenance for LLM-generated text spans watermarking, attack evaluation, stylometric attribution, and external provenance mechanisms. Existing methods have made clear progress, but they also reveal a persistent weakness: most practical schemes still anchor attribution to representations that are too fragile under semantic-preserving rewriting. Below, we organize the literature around this failure mode and position SS-WM relative to prior directions.

Token- and Logit-Based Statistical Watermarking. The dominant line of work embeds watermarks by perturbing the token sampling process (Huo et al., 2024). Representative methods bias next-token probabilities toward a keyed subset of tokens—often implemented through green-list/red-list constructions, context hashing, logit biasing, or constrained sampling—and detect the watermark through statistical tests on token-partition frequencies under a null model (Huo et al., 2024; Park et al., 2026; WONG et al., 2025; Wouters, 2024). These methods are attractive because they are lightweight, base-model-training-free, and easy to integrate into existing decoding pipelines. They also expose explicit trade-offs among detectability, robustness, and output quality. However, their core signal remains lexical. This is a fundamental limitation rather than a minor implementation issue. Token-level statistics are precisely the part of text that paraphrasing, translation, summarization, truncation, and stylistic rewriting most readily modify. As a result, these methods often remain strong only when the generated text is evaluated close to its original surface form. Their practical weakness has therefore become increasingly clear: they are easiest to evade in the very settings where real-world attribution matters most.

Reliability Analyses and Attack-Driven Evaluations. A parallel line of work studies watermark reliability under adversarial or post-generation transformations, including paraphrasing, back-translation, summarization, editing, truncation, and composed laundering pipelines (Liang et al., 2024; Krishna et al., 2023; Yan et al., 2023). These studies consistently show the same pattern: performance that appears strong on clean generations can degrade sharply once the text is rewritten while preserving meaning. This literature is important because it shifts attention from clean-text detectability to deployment realism. In particular, it makes clear that robust watermarking cannot be judged solely by performance on unmodified outputs. This body of work also sharpens the main open problem. The challenge is not simply to make token biases stronger or to add redundancy. Stronger lexical signals often worsen quality or become easier to detect and imitate, while redundancy remains vulnerable if the carrier itself is unstable under rewriting. Thus, attack-driven evaluation has exposed a structural limitation of lexical watermarking rather than just a tuning problem (Kirchenbauer et al., 2024).

Semantics-Aware and Robust Watermarking. To overcome lexical fragility, recent work has incorporated semantic information into watermark design or detection (Dabiriaghdam & Wang, 2025; Xu et al., 2025; Lau et al., 2024). These methods move beyond raw token counts by exploiting semantic representations, paraphrase-consistent structures, or embedding-space invariants. This is a meaningful step forward: once attribution is tied to meaning-level structure, robustness under rewriting can improve substantially relative to purely token-level schemes. Yet existing semantics-aware approaches still leave an important gap. Many remain partially coupled to token-level decoding controls, or they treat semantic consistency as the primary robust signal. This narrows the carrier space and leaves robustness gains on the table. Semantic preservation alone does not exhaust what tends to survive rewriting. High-level stylistic regularities—such as discourse patterns, sentence rhythm, and syntactic realization—also persist under many semantic-preserving transformations, but they are not typically used as an explicitly keyed watermark channel. SS-WM advances this line of work by treating projected semantic embedding features and stylistic realization as a *joint*, key-conditioned carrier and by detecting watermark evidence through a unified statistical test over the combined feature space.

Stylometric Signals, Authorship Attribution, and Model Fingerprinting. Stylometry, authorship attribution, and model fingerprinting provide complementary evidence that higher-level textual regularities can remain informative even when lexical choices change. These approaches analyze distributional stylistic patterns, discourse preferences, or other model-specific output signatures to infer provenance or authorship. Their central lesson is highly relevant to watermarking: stable attribution signals need not reside at the token level (Tan et al., 2025; Nasery et al., 2025). However, these methods are typically *passive*. They infer provenance from naturally occurring output statistics rather than embedding a secret, key-dependent signal at generation time. This distinction matters. Passive fingerprints are not designed to provide keyed attribution guarantees, and they can degrade under model-agnostic rewriting, prompt changes, or distribution shift. SS-WM instead adopts an *active* design: it intentionally shapes projected semantic embedding features and stylistic realization according to a secret key, thereby combining the persistence of higher-level structure with the stronger attribution guarantees of keyed watermarking.

Alternative Provenance Mechanisms: Metadata and Classifiers. Watermarking is only one part of the broader provenance toolbox. Metadata-based mechanisms, such as content credentials, can provide strong guarantees when preserved end-to-end (Zhang et al., 2024). However, they are external to the text itself and can be removed through copying, reformatting, or cross-platform transfer. Classifier-based detectors require no generation-time modification, but they often degrade under domain shift, adversarial adaptation, and rapid model evolution (Guo et al., 2024a). These limitations motivate intrinsic, text-based attribution mechanisms that remain meaningful even after downstream transformation.

Positioning of the Proposed SS-WM. The literature now establishes three points clearly. First, token-level watermarking is practical but fundamentally fragile under semantic-preserving rewriting. Second, attack-driven evaluation has shown that clean-text performance is an insufficient proxy for deployment robustness. Third, semantics-aware methods and stylometric analyses both suggest that higher-level textual structure is a more promising attribution substrate than lexical form alone.

What remains missing is a practical watermarking framework that *jointly* exploits these higher-level signals in an explicitly keyed manner. SS-WM fills this gap. It differs from prior work in two respects. First, it treats projected semantic embedding features and stylistic realization as a *joint transformation-stable carrier* for keyed watermark evidence, rather than relying primarily on lexical statistics or semantic cues alone. Second, it performs attribution through a unified keyed statistical test in semantic–stylistic feature space, enabling robustness to laundering transformations and support for partial-text detection. In this sense, SS-WM is not simply a more robust lexical watermark or a semantic-only extension; it is a shift toward carrier-level watermarking for real-world LLM provenance.

C Embedding and Detecting Algorithms

We now summarize the operational procedures for semantic–stylistic watermark embedding and detection. The goal of this subsection is to clarify the logical flow of the proposed framework and the interaction between watermark keys, carrier selection, generation, and attribution.

C.1 Watermark Embedding

SS-WM supports two embedding modes, depending on the inference interface. Algorithm 1 describes the main logit-access mode used in our experiments. Given a prompt p and a secret watermark key k, the watermark owner first derives a key-specific subset of semantic–stylistic constraints C_k from the public codebook. Text generation then proceeds token by token using the base LLM M. At each step, candidate tokens are obtained from the original model distribution and softly reranked using semantic and stylistic penalties induced by C_k. This mode requires access to candidate-token scores or logits, but it does not retrain, fine-tune, or modify the parameters of the base model. Algorithm 2 describes the strict black-box sequence-level variant, which requires only completed candidate outputs from the base model.

C.2 Semantic–Stylistic Scoring and Reranking

The token-level scores below are practical local surrogates for the sequence-level semantic and stylistic penalties in Eqs. (5)–(6), evaluated on partial continuations during decoding. Here, $g^{\mathrm{sem}}(E(x))$ and $\phi(x)$ define the semantic and stylistic components of the same carrier family later used by the detector through $f^{\mathrm{sem}}(x)$ and $f^{\mathrm{sty}}(x)$, with normalization and density fitting applied only at detection time. We now briefly specify how the semantic–stylistic penalties are instantiated in practice. The token-level scoring form is used in Algorithm 1, while the same penalties are aggregated at the sequence level in Algorithm 2.

At decoding step t, the base model produces logits $\ell_t \in \mathbb{R}^{|\mathcal{V}|}$ over the vocabulary $\mathcal{V}$, inducing the distribution

$$P_0(w \mid w_{<t}, p) = \mathrm{softmax}(\ell_t)_w.$$

We first restrict attention to a candidate set $\mathcal{S}_t \subset \mathcal{V}$ (e.g., top-K or nucleus sampling). Because many semantic and stylistic carrier features are naturally defined over sentences or longer text spans, the token-level reranking scores are computed using partial-state proxies on the current continuation $(x_{t-1} \oplus w)$ rather than full-document statistics. Concretely, semantic carrier features are estimated from the projected encoder representation of the growing prefix, while stylistic features such as sentence-length statistics, punctuation usage, discourse markers, and syntactic-depth tendencies are maintained as rolling or prefix-level statistics. Thus, the decoding-time scores provide local approximations to the same feature families later evaluated by the detector on the completed text.

Formally, in the logit-access setting, we use partial-state surrogate penalties

$$\widehat{\Phi}_{\mathrm{sem}}(x_{t-1} \oplus w; \mathcal{C}_k) \quad \text{and} \quad \widehat{\Phi}_{\mathrm{sty}}(x_{t-1} \oplus w; \mathcal{C}_k),$$

defined on the currently generated continuation. The token-level reranking scores are evaluations of these surrogate penalties on candidate continuations:

$$s_{\mathrm{sem}}(w; t, k) \equiv \widehat{\Phi}_{\mathrm{sem}}(x_{t-1} \oplus w; \mathcal{C}_k), \qquad s_{\mathrm{sty}}(w; t, k) \equiv \widehat{\Phi}_{\mathrm{sty}}(x_{t-1} \oplus w; \mathcal{C}_k).$$

These surrogate penalties share the same key-conditioned carrier targets as the full sequence-level penalties Φ_{sem} and Φ_{sty}, but are evaluated on partial continuations during decoding rather than

on completed sequences. The same key-specific binary masks m_k^{sem} and m_k^{sty} are applied at both decoding and completed-text evaluation.

Semantic fidelity and semantic watermark scoring. Let $E(\cdot)$ be a frozen sentence- or paragraph-level encoder, and let $\bar{e}_p$ denote a prompt-conditioned semantic anchor. We first use a fidelity score

$$s_{\mathrm{fid}}(w;t) = 1 - \cos(E(x_{t-1} \oplus w), \bar{e}_p)\,,$$

which is the same prompt-fidelity term defined in Eq. (9). This term preserves task meaning but does not itself constitute the watermark signal. To embed key-conditioned semantic evidence, SS-WM also uses a semantic carrier score. Let $g_{\mathrm{sem}}(\cdot)$ denote a projection from encoder space to the semantic carrier space. Each constraint $c \in C_k^{\mathrm{sem}}$ specifies one or more coordinate-level target values in the semantic carrier space. The semantic target vector μ_k^{sem} is then formed by coordinate-wise averaging of the selected target values induced by the subset C_k^{sem}. We define

$$s_{\mathrm{sem}}(w;t,k) = \|(g_{\mathrm{sem}}(E(x_{t-1} \oplus w)) - \mu_k^{\mathrm{sem}}) \odot m_k^{\mathrm{sem}}\|_2^2\,.$$

Unlike the fidelity score, $s_{\mathrm{sem}}(w;t,k)$ is key-conditioned and steers generation toward a secret-key-selected region of the projected semantic embedding carrier.

Stylistic scoring. Let $\phi(\cdot)$ denote a vector of stylistic features, including sentence-length statistics, discourse-marker usage, syntactic depth, punctuation patterns, and function-word statistics. Given the selected stylistic constraint subset C_k^{sty}, where each constraint specifies one or more coordinate-level target values of the stylistic feature vector, the stylistic target vector μ_k^{sty} is formed by coordinate-wise averaging of the selected target values induced by C_k^{sty}. The stylistic watermark score is

$$s_{\mathrm{sty}}(w;t,k) = \left\|\left(\phi(x_{t-1} \oplus w) - \mu_k^{\mathrm{sty}}\right) \odot m_k^{\mathrm{sty}}\right\|_2^2\,.$$

Reranking. Semantic and stylistic costs are combined with the base logits to form reweighted logits

$$\tilde{\ell}_t(w) = \ell_t(w) - \lambda_{\mathrm{fid}} s_{\mathrm{fid}}(w;t) - \lambda_{\mathrm{sem}} s_{\mathrm{sem}}(w;t,k) - \lambda_{\mathrm{sty}} s_{\mathrm{sty}}(w;t,k), \qquad w \in S_t.$$

This is the Appendix restatement of the reranking rule in Eq. (8). Here, s_{fid} preserves prompt-level semantic fidelity, while s_{sem} and s_{sty} provide the key-conditioned watermark signal.

C.3 Watermark Detection

Algorithm 3 outlines the detection procedure. Given a candidate text x' and the secret key k, the detector reconstructs the corresponding constraint subset $\mathcal{C}_k$ and extracts semantic and stylistic feature representations from x'. Detection is formulated as a keyed hypothesis test by computing a likelihood ratio $\Lambda(x')$ that compares the compatibility of the observed features with the watermark-induced distribution versus a null distribution. The candidate-key test accepts the watermark hypothesis if $\Lambda(x')$ exceeds a threshold τ, which is calibrated to achieve a desired false-positive rate. Because detection operates on semantic–stylistic representations rather than surface tokens, it remains effective even when x' has undergone paraphrasing, translation, summarization, or partial reuse.

C.4 Adaptive Robustness Evaluation

In addition to the embedding and detection procedures, SS-WM is evaluated against an adaptive semantic-preserving attacker. Given a generated text, the attacker selects from a family of paraphrasing, translation, summarization, and stylistic-rewriting operations the transformation with the lowest detector score while remaining within predefined semantic-fidelity and quality constraints. The evaluator grants score access solely for this stress test. The selection is oracle-aided because it uses detector scores; it provides a stress test that aligns with the carrier-stability principle introduced earlier.

C.5 Abstraction and Flexibility

We emphasize that Algorithms 1, 2, and 3 abstract away the exact choice of semantic encoders, stylistic feature extractors, carrier projections, and reranking strategies. These components may be

Algorithm 1 SS-WM Embedding: Logit-Access Mode

Require: Prompt p, secret key k, base LLM M, candidate size K, reranking interval $R = 5$
Ensure: Watermarked text x

$C_k \leftarrow \mathrm{Select}(C, k)$
Initialize $x \leftarrow \emptyset$
for $t = 1$ to T **do**
  Obtain candidate set S_t and logits $\ell_t(w)$ for $w \in S_t$ from M
  if $t \bmod R = 0$ **then**
    for each $w \in S_t$ **do**
      $s_{\mathrm{fid}} \leftarrow s_{\mathrm{fid}}(w; t)$
      $s_{\mathrm{sem}} \leftarrow s_{\mathrm{sem}}(w; t, k)$
      $s_{\mathrm{sty}} \leftarrow s_{\mathrm{sty}}(w; t, k)$
      $\tilde{\ell}_t(w) \leftarrow \ell_t(w) - \lambda_{\mathrm{fid}} s_{\mathrm{fid}} - \lambda_{\mathrm{sem}} s_{\mathrm{sem}} - \lambda_{\mathrm{sty}} s_{\mathrm{sty}}$
    end for
  else
    $\tilde{\ell}_t(w) \leftarrow \ell_t(w)$ for all $w \in S_t$
  end if
  Sample $w_t \sim \mathrm{softmax}(\tilde{\ell}_t)$
  $x \leftarrow x \oplus w_t$
end for

return x

Algorithm 2 SS-WM Embedding: Strict Black-Box Sequence Mode

Require: Prompt p, secret key k, base LLM M, number of candidates m
Ensure: Watermarked text x^*

$C_k \leftarrow \mathrm{Select}(C, k)$
Query M to generate $\mathcal{X}_m = \{x^{(1)}, \ldots, x^{(m)}\}$
for each $x^{(i)} \in \mathcal{X}_m$ **do**
  $E_k(x^{(i)}) \leftarrow \lambda_{\mathrm{fid}}[1 - \cos(E(x^{(i)}), \bar{e}_p)] + \lambda_{\mathrm{sem}} \Phi_{\mathrm{sem}}(x^{(i)}; C_k^{\mathrm{sem}}) + \lambda_{\mathrm{sty}} \Phi_{\mathrm{sty}}(x^{(i)}; C_k^{\mathrm{sty}})$
end for
$x^* \leftarrow \arg\min_{x^{(i)}} E_k(x^{(i)})$

return x^*

instantiated differently across deployment settings, provided that they preserve the core requirement of low carrier drift under semantic-preserving transformations. This abstraction allows SS-WM to be applied across different models, domains, and generation tasks without altering its underlying attribution logic.

The logit-access mode provides a stronger watermark signal because the carrier objective can influence generation incrementally at each decoding step. The strict black-box mode is more interface-compatible but weaker, since the carrier objective can only select among complete candidates already generated by the base model. We therefore use logit-access SS-WM for the main experiments and report the black-box sequence-reranking variant separately as an interface-compatibility analysis.

SS-WM embeds watermarks by shaping semantic and stylistic distributions rather than surface tokens. This design provides robustness to rewriting attacks, preserves text quality, and enables reliable attribution under realistic adversarial conditions.

D Feature Stability, Feature Inventory, and Generalizability

The robustness of SS-WM depends on empirically transformation-stable carrier features, not on a claim that individual semantic coordinates have direct discourse interpretations. The semantic channel is derived from a frozen sentence encoder followed by the auxiliary projection described in

Algorithm 3 Semantic–Stylistic Watermark Detection

Require: Text x', secret key k
Ensure: Watermark decision
1: $\mathcal{C}_k \leftarrow \mathsf{Select}(\mathcal{C}, k)$
2: Extract $f^{\mathrm{sem}}(x')$ and $f^{\mathrm{sty}}(x')$
3: Compute likelihood ratio $\Lambda(x')$
4: **if** $\Lambda(x') > \tau$ **then**
5: **Accept** H_1 (watermarked)
6: **else**
7: **Accept** H_0 (not watermarked)
8: **end if**

Section 3.3. The stylistic channel uses the 128-dimensional inventory summarized in Table 2. All coordinates are standardized on held-out calibration data.

Semantic carrier. Semantically similar text should remain relatively close in the selected projected embedding representation even when lexical form changes. We test this property directly through the carrier-drift analysis rather than assuming strict invariance. Carrier stability is evaluated using the primary frozen semantic encoder and the drift protocol defined in Section 3.2.

Stylistic carrier. The style vector aggregates coarse distributional quantities rather than brittle token identities. Family-level ablations in Table 5 determine which groups contribute to robustness. The complete coordinate inventory and extractor versions are specified in Table 2.

Generalizability. Because the detector operates on extracted representations rather than generator hidden states, the same frozen carrier extractor can be applied across base LLMs. Table 15 reports cross-model ROC-AUC point estimates. The in-distribution LLaMA-2-7B→LLaMA-2-7B value matches the clean-text ROC-AUC reported in Table 9; confidence intervals are unavailable because per-example cross-model predictions are unavailable.

E Statistical Analysis and Scope

This section analyzes statistical properties of SS-WM under the threat model in Section A. We focus on (i) attribution reliability as a keyed statistical test, (ii) resistance to evasion under semantic-preserving transformations, (iii) empirical wrong-key behavior and calibrated null false-positive control, and (iv) partial-text attribution. Our guarantees are statistical rather than cryptographic: we assume the watermarking mechanism and detector are public, and security relies on the secrecy of the key and the adversary's requirement to preserve semantic utility.

E.1 Security Model and Assumptions

Let $f(x) \in \mathbb{R}^d$ denote the semantic–stylistic feature representation extracted from text x. For a secret key k, the watermark induces a feature distribution P_k (equivalently, a density p_k), while non-watermarked text induces a null distribution P_{null} (density p_0). Detection is performed via a keyed test on $f(x)$.

We use the following assumptions, which are standard for statistical watermarking (Gloaguen et al., 2025) and are consistent with the deployment of SS-WM:

A1 (Density model and calibration). The detector uses a calibrated likelihood ratio statistic

$$\Lambda(x) \triangleq \frac{p_k(f(x))}{p_0(f(x))}, \qquad \ell(x) \triangleq \log \Lambda(x), \tag{12}$$

with a global threshold τ chosen from the pooled held-out null-score distribution over the evaluated candidate keys so that the overall calibrated null false-positive rate is at most α.

A2 (Empirical wrong-key separation). We do not assume a theorem-level separation guarantee between arbitrary P_k and $P_{k'}$. The same globally calibrated threshold is used for all evaluated candidate keys, while separation between correct-key and mismatched-key distributions is measured empirically using the protocol in Section 4.4. This avoids assuming the property that the wrong-key experiment is intended to test.

A3 (Transformation stability). For semantic-preserving transformations $\mathcal{T}$ in the threat model, feature drift is bounded:

$$\|f(\mathcal{T}(x)) - f(x)\| \leq \Delta_{\mathcal{T}}(x), \tag{13}$$

where $\Delta_{\mathcal{T}}(x)$ is small in expectation for meaning-preserving transformations and grows with transformation strength.

Assumption A3 is intentionally weak: it does not require invariance, only bounded drift. We do not estimate a global Lipschitz constant in practice; rather, Theorem 3 should be interpreted as a local stability characterization on the bounded normalized feature region induced by calibration and detector fitting.

E.2 Attribution as a Keyed Hypothesis Test

Watermark detection is a binary hypothesis test (Gloaguen et al., 2025):

$$H_0 : f(x) \sim P_{\text{null}}, \qquad H_1 : f(x) \sim P_k.$$

We characterize detection by false-positive rate α and false-negative rate β.

Theorem 1 (Calibrated false-positive control)**.** *Let τ be chosen such that $\Pr_{x\sim P_{\text{null}}}(\Lambda(x) > \tau) \leq \alpha$. Then the detector's false-positive rate is at most α.*

Proof. By definition of the decision rule, a false positive occurs iff $\Lambda(x) > \tau$ under H_0. Thus

$$\Pr(\text{FP}) = \Pr_{x\sim P_{\text{null}}}(\Lambda(x) > \tau) \leq \alpha,$$

which proves the claim. □

Theorem 2 (Detection power via log-likelihood concentration)**.** *Assume $\ell(x) = \log \Lambda(x)$ has finite variance under P_k and define*

$$\mu_k \triangleq \mathbb{E}_{x\sim P_k}[\ell(x)], \qquad \sigma_k^2 \triangleq \text{Var}_{x\sim P_k}[\ell(x)].$$

Then for any threshold $\log \tau < \mu_k$,

$$\beta = \Pr_{x\sim P_k}\big(\ell(x) \leq \log \tau\big) \leq \frac{\sigma_k^2}{(\mu_k - \log \tau)^2}. \tag{14}$$

Proof. By Chebyshev's inequality,

$$\Pr(\ell(x) - \mu_k \leq -(\mu_k - \log \tau)) \leq \frac{\sigma_k^2}{(\mu_k - \log \tau)^2},$$

and the left-hand side is exactly $\Pr(\ell(x) \leq \log \tau) = \beta$. □

Theorem 2 formalizes a key intuition: detection becomes reliable when the watermark induces a positive expected log-likelihood advantage μ_k over the null and the threshold is set below it.

E.3 Resistance to Evasion via Semantic-Preserving Transformations

An adversary outputs $x' = \mathcal{T}(x)$ for some semantic-preserving transformation $\mathcal{T}$. We analyze how the detection statistic changes under feature drift.

Theorem 3 (Stability of likelihood ratio under bounded feature drift)**.** *Assume the log-densities* $\log p_k(u)$ *and* $\log p_0(u)$ *are L-Lipschitz on a bounded feature region* $\mathcal{U} \subset \mathbb{R}^d$ *containing the normalized semantic–stylistic representations encountered in detection. Equivalently, the theorem may be read as a local Lipschitz statement on the practically relevant support of the fitted detector:*

$$\big|\log p_k(u) - \log p_k(v)\big| \le L\|u - v\|,$$
$$\big|\log p_0(u) - \log p_0(v)\big| \le L\|u - v\|.$$

Then for $x' = \mathcal{T}(x)$*,*

$$|\ell(x') - \ell(x)| \le 2L\|f(x') - f(x)\| \le 2L\,\Delta_{\mathcal{T}}(x), \tag{15}$$

and consequently

$$\ell(x') \ge \ell(x) - 2L\,\Delta_{\mathcal{T}}(x).$$

Proof. By definition $\ell(x) = \log p_k(f(x)) - \log p_0(f(x))$. Thus

$$\begin{aligned} |\ell(x') - \ell(x)| &\le |\log p_k(f(x')) - \log p_k(f(x))| \\ &\quad + |\log p_0(f(x')) - \log p_0(f(x))| \\ &\le L\|f(x') - f(x)\| + L\|f(x') - f(x)\| \\ &= 2L\|f(x') - f(x)\|. \end{aligned} \tag{16}$$

Applying Eq. (13) yields Eq. (15). □

Theorem 3 provides a formal robustness statement: as long as transformations preserve meaning and therefore induce bounded drift in semantic–stylistic feature space, the log-likelihood ratio cannot drop arbitrarily. In contrast, token-level watermark statistics can be substantially altered by transformations with small semantic cost. In our implementation, this bounded region corresponds to the normalized feature range induced by held-out calibration data and the detector's fitted support, so the theorem should be interpreted as a local stability characterization rather than a global property of Gaussian densities on all of $\mathbb{R}^d$.

E.4 Global Null Calibration and Empirical Wrong-Key Analysis

The detector uses the single global threshold τ defined in Assumption A1 and Theorem 1. This calibration controls the false-positive rate under the held-out null calibration distribution; it should not be interpreted as a family-wise guarantee over an arbitrarily large registry of candidate keys. In particular, null calibration does not imply that text generated under another watermark key k' will be accepted under key k with probability exactly α.

We therefore evaluate practical key isolation empirically through wrong-key testing in Table 7, where watermarked samples are tested against 50 randomly sampled mismatched keys under the same global threshold. Low wrong-key acceptance indicates that attribution depends on the correct key-specific constraint subset rather than generic semantic–stylistic similarity. If an adversary is allowed repeated adaptive queries to a detection oracle, or if deployment requires searching a large key registry, stronger multiple-testing and oracle-resistant analyses would be required and are outside the scope of this work.

E.5 Partial-Text and Composition Robustness

SS-WM performs detection over sliding semantic windows. Let a candidate text be partitioned into n windows, and let ℓ_i denote the log-likelihood ratio computed from window i. Define the aggregated statistic

$$S_n \triangleq \sum_{i=1}^{n} \ell_i. \tag{17}$$

Theorem 4 (Partial-text detection via aggregation)**.** *Assume* $\{\ell_i\}_{i=1}^n$ *are weakly dependent window-level log-likelihood ratios with* $\mathbb{E}_{H_1}[\ell_i] = \mu > 0$ *and* $\mathrm{Var}_{H_1}(\ell_i) \le \sigma^2$*, and suppose the dependence is sufficiently limited that* $\mathrm{Var}(S_n) \le n\sigma^2$ *for* $S_n = \sum_{i=1}^n \ell_i$*. Then under* H_1*,*

$$\Pr(S_n \le 0) \le \frac{\sigma^2}{n\mu^2}. \tag{18}$$

Proof. Under H_1, $\mathbb{E}[S_n] = n\mu$ and, by assumption, $\mathrm{Var}(S_n) \le n\sigma^2$. Applying Chebyshev's inequality to S_n gives

$$\Pr(S_n - n\mu \le -n\mu) \le \frac{n\sigma^2}{(n\mu)^2} = \frac{\sigma^2}{n\mu^2}.$$

□

Theorem 4 shows that partial attribution improves with the number of available windows: as long as each window carries a small positive expected watermark signal, aggregating windows yields a concentration effect. In our implementation, sliding windows overlap (window size 100, stride 50), so neighboring ℓ_i are not independent; Theorem 4 should therefore be interpreted under weak dependence, with overlap-induced correlations absorbed into the aggregate variance bound.

E.6 Limits of Statistical Guarantees

SS-WM does not claim cryptographic unforgeability. An adversary willing to substantially alter semantic structure (e.g., inject unrelated content, destroy discourse coherence, or re-author the text) can reduce feature compatibility with P_k and defeat attribution. Such attacks fall outside semantic-preserving transformations and eliminate much of the original content's utility. Similarly, compromise of the secret key invalidates attribution guarantees. As with any keyed watermarking scheme, secure key management and controlled key reuse policies are required in deployment. In summary, SS-WM provides statistically grounded attribution guarantees under realistic adversarial conditions. Theorem 1 ensures calibrated null false-positive control; Theorem 2 bounds false negatives in terms of log-likelihood separation; Theorem 3 characterizes robustness under bounded feature drift; Table 7 empirically evaluates wrong-key rejection; and Theorem 4 explains why partial-text attribution improves with aggregation. Together, these results justify SS-WM's robustness advantages over token-level watermarking while clearly delineating its security limits.

F Additional Experimental Details

F.1 Models and datasets

We evaluate three open-weight autoregressive LLMs that support decoding-time control: LLaMA-2-7B (Touvron et al., 2023), Mistral-7B-v0.1 (Jiang et al., 2023), and OPT-2.7B (Zhang et al., 2022). LLaMA-2-13B is additionally used as the perplexity evaluator and as a cross-model test target; it is not the primary watermark-generation model. All models are used strictly in inference mode: SS-WM does not fine-tune, retrain, or modify its parameters. The main experiments use the logit-access implementation, where candidate-token scores are available for semantic–stylistic reranking during decoding. We additionally evaluate a strict black-box sequence-reranking variant in which the model is queried only for completed candidate continuations. Following prior watermarking work (Kirchenbauer et al., 2023; Zhao et al., 2024), the primary benchmark is C4 (Raffel et al., 2020): we sample 10,000 passages, use the first 30 tokens as prompts, and generate 200-token continuations; the original human-written continuations are used for false-positive analysis. We additionally evaluate 2,000 task prompts spanning instruction following, question answering, summarization, argumentative writing, and technical explanation. The question answering prompts are drawn from Natural Questions (Kwiatkowski et al., 2019), and the summarization prompts from CNN/DailyMail (Hermann et al., 2015); the remaining task prompts cover instruction following, argumentative writing, and technical explanation. Semantic embeddings for watermark-model construction are derived from disjoint WikiText-103 (Merity et al., 2016).

F.2 Decoding, carrier extraction, projection training, and detector

Unless otherwise stated, decoding uses nucleus sampling with temperature $T = 1.0$, top-$p = 0.95$, maximum generation length 200, and no repetition penalty. Results are averaged over seeds

$\{42, 43, 44\}$, with 95% bootstrap confidence intervals where significance is claimed. The primary semantic encoder is frozen Sentence-BERT (`all-MiniLM-L6-v2`; `https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2`), producing 384-dimensional embeddings. The auxiliary semantic projection is the four-layer residual network summarized in Table 1: 384-dimensional MiniLM embeddings are mapped through hidden widths $[256, 128, 64, 32]$ to a 32-dimensional carrier, with projected residual connections, Kaiming initialization, and NT-Xent loss at temperature 0.07. Training uses AdamW with learning rate 10^{-4}, weight decay 10^{-5}, batch size 256, and up to 20 epochs with patience-3 early stopping on 50,000 training and 5,000 validation pairs from WikiText-103. A single trained projection is shared across keys and generator models and is frozen at inference.

The stylistic extractor produces exactly 128 dimensions using the inventory in Table 2. Sentence segmentation uses spaCy v3.7.2; dependency depth and POS features use `en_core_web_sm-3.7.1`; punctuation, discourse-marker, and function-word features use deterministic regex/lexicon matching; and coarse n-gram features use NLTK v3.8.1 tokenization. Before detector fitting, every semantic and stylistic coordinate is standardized using the held-out calibration mean and standard deviation. For each evaluated candidate key, the watermarked density is fitted from 1,000 held-out normalized watermarked samples generated under that key, and the null density is fitted from a separate set of 1,000 held-out normalized non-watermarked samples. Detection uses sliding windows of size 100 with stride 50 and sums the window log-likelihood ratios to obtain the text-level score. A separate set of 5,000 non-watermarked C4 samples is used once to form the pooled null-score distribution over the evaluated candidate keys and calibrate global thresholds targeting 1% and 10% overall null FPR; the same thresholds are reused across keys and attack conditions.

The public codebook has $M = 10{,}000$ entries, of which 1,000 are selected per key by deterministic SHA-256 indexing and divided evenly between semantic and stylistic channels. Calibration statistics are estimated from the 5,000-sample calibration set, after which each coordinate is standardized and partitioned into 20 equal-width bins over $[-4, 4]$; the standardized bin midpoint is the target value. The 10,000-entry codebook is an indexed multiset, so repeated coordinate–bin tuples are allowed. Duplicate selections for the same coordinate are averaged, and key-specific binary masks exclude unselected coordinates from the watermark penalty. The default watermark strengths are $\lambda_{\text{fid}} = 1.0$, $\lambda_{\text{sem}} = 2.5$, and $\lambda_{\text{sty}} = 2.0$. Logit-access reranking uses top-$K = 50$ candidates every five decoding steps.

F.3 Baselines

The baselines are KGW, SIR, EXP-Edit, SemStamp, k-SemStamp, SemaMark, and Unigram Watermark. SemStamp, k-SemStamp, SemaMark, and Unigram Watermark use the authors' released implementations with their published default hyperparameters; the only evaluation-level modification is recalibration of the detector threshold on our held-out 5,000 non-watermarked C4 samples to match the common 1% clean-text FPR operating point. Generation settings use temperature 1.0, top-$p = 0.95$, and length 200 where each method's decoding interface permits. SimMark is discussed as closely related work but is omitted from the quantitative table because a directly comparable run is unavailable.

F.4 Attack protocols

The attack suite includes GPT-3.5-Turbo (`gpt-3.5-turbo-0613`; `https://openai.com/index/function-calling-and-other-api-updates/`) paraphrasing at four intensities, DIPPER (Krishna et al., 2023) with lexical/order variation, one- to three-hop NLLB-200 back-translation (`https://arxiv.org/abs/2207.04672`), BART-large-CNN summarization (`https://huggingface.co/facebook/bart-large-cnn`) at 0.75/0.50/0.25 retention, three style-transfer settings, partial-text reuse, composed pipelines, copy-paste mixing, emoji insertion, and synonym substitution. These systems specify the rewriting models used in the attack suite.

Table 9: Attribution performance on clean (untransformed) text.

Method	TPR (%)	FPR (%)	AUC	F1
KGW-1	98.4	1.0	0.995	0.984
KGW-2	98.9	1.0	0.997	0.989
KGW-4	99.0	1.0	0.997	0.990
SIR	98.7	1.0	0.996	0.987
EXP-Edit	99.5	1.0	0.998	0.995
SS-WM	**99.1**	**1.0**	**0.998**	**0.991**

F.5 Oracle-aided score-aware candidate selection stress test

As an oracle-aided stress test, we consider a score-aware candidate selection attacker. For each watermarked sample x, the attacker generates a candidate set $\{x'_1, \ldots, x'_m\}$ using a strong paraphrasing model and retains only candidates satisfying the semantic-similarity and quality admissibility constraints used for the constrained adaptive attack. Among the admissible candidates, the attacker selects

$$x^*(x) = \arg\min_{x'_j} \Lambda(x'_j),$$

that is, the candidate with the lowest detector score. This stress test deliberately grants access to the detector score and is therefore stronger than the primary black-box threat model. It measures robustness under score exposure but is not assumed by the main security model.

F.6 Evaluation metrics

We report TPR at globally calibrated null FPR operating points of 1% and 10%, ROC-AUC, and F1. Robustness is measured as TPR degradation across attacks and intensities. Quality is evaluated using perplexity, ROUGE-L for summarization, exact match/F1 for QA, self-BLEU, distinct-n, and Sentence-BERT semantic similarity. Because the same embedding family participates in the semantic carrier, semantic-similarity scores are interpreted as descriptive rather than independent semantic validation. We do not conduct a human-subject annotation study in this submission. Deployment cost is characterized through the logit-access versus strict black-box comparison, including the candidate-count scaling of sequence-level reranking.

G Additional Results

G.1 Clean-Text Attribution

This experiment verifies that SS-WM preserves standard clean-text detectability before any post-generation transformation is applied. Since all methods are calibrated to the same false-positive regime, this result checks whether shifting the watermark carrier from token statistics to semantic–stylistic structure weakens basic attribution. Table 9 verifies that moving from token-level statistics to a semantic–stylistic carrier does not weaken clean-text attribution. All methods are calibrated to the same 1% false-positive rate, and all achieve above 98% TPR. SS-WM obtains 99.1% TPR, 0.998 AUC, and 0.991 F1, matching the strongest baselines while remaining slightly below EXP-Edit in clean TPR. This confirms that the main contribution of SS-WM is not clean-text detectability alone, where most modern watermarking methods already perform well, but robustness after semantic-preserving transformation.

G.2 Prefix-to-Sequence Surrogate Validation

We directly test whether the local reranking surrogates used during generation align with the completed-text carrier. For every generated sequence, we accumulate the semantic, stylistic, and joint surrogate penalties at reranking steps and compare them with the completed-text carrier penalties and final detector score. Table 6 in the main paper reports the prefix-to-sequence correlations.

G.3 Stylistic Feature-Family Ablation

We remove each stylistic family from the full carrier while keeping all other hyperparameters fixed. This identifies whether robustness is distributed across several style families or dominated by one extractor. Table 5 reports the family-level ablation under clean, paraphrase, and composed conditions, which are the representative settings used for this analysis.

G.4 Prompt-Constraint Compatibility

We evaluate open-ended generation, summarization, factual QA, and tightly constrained outputs under identical watermark strengths. The primary task-quality metrics, TPR values, and no-watermark quality deltas are reported in Table 8; Natural Questions and CNN/DailyMail provide the factual-QA and summarization settings described in Section 4.1.

G.5 Empirical Wrong-Key Protocol

For each of 1,000 watermarked texts generated under key k, we evaluate 50 independently sampled mismatched keys $k' \neq k$. The same set of 50 mismatched keys is held fixed across attack conditions. We report empirical mean wrong-key acceptance and non-parametric bootstrap 95% confidence intervals using 1,000 resamples over texts, with the 50 wrong-key evaluations for each text treated as a within-sample average. A single global threshold τ calibrated from pooled null scores on 5,000 clean non-watermarked C4 passages is used for all conditions. These measurements are distinct from the calibrated null-text FPR. The corresponding rates and intervals are reported in Table 7.

G.6 Full Robustness Results

Table 10 reports the complete per-attack robustness results behind the compact summary in the main paper. The key trend is that token-level methods degrade sharply once lexical form changes. For example, KGW-1 drops from 98.4% TPR on clean text to 59.0% under GPT-3.5 paraphrasing, 42.5% under summarization, and 28.5% under composed attacks. Larger-context KGW variants degrade even more severely, with KGW-4 reaching only 9.5% TPR under composed laundering. SIR is substantially stronger, reaching 74.0% under GPT-3.5 paraphrasing and 68.5% under composed attacks, confirming the value of semantic invariance. SS-WM remains the strongest across all attack families, achieving 92.6% under GPT-3.5 paraphrasing, 89.4% under translation, 81.2% under summarization, 88.7% under style rewriting, and 76.9% under composed attacks. These results show that the improvement is not isolated to one attack type but persists across paraphrasing, translation, summarization, style rewriting, and multi-stage laundering.

G.7 Attack Intensity and Partial-Text Figures

This subsection expands the main paper's analysis of attack strength and partial reuse. Table 11 reports how detection changes as paraphrasing, translation, and summarization become stronger. In particular, Table 11 gives a more granular view of robustness as attack strength increases. Under paraphrasing, SS-WM decreases gradually from 96.8% TPR under light rewriting to 82.5% under very strong rewriting, whereas KGW-4 falls from 48.5% to 15.2%. The same pattern appears for translation, where SS-WM remains at 81.4% even after 3-hop back-translation, and for summarization, where it retains 68.5% TPR even when only 25% of the content is retained. Figure 2 complements the partial-text analysis by showing that attribution improves smoothly as more text becomes available. This supports the sliding-window detector: even short fragments carry useful evidence, while longer fragments allow aggregation to recover stronger attribution.

G.8 Adaptive Attack Selection

Table 12 evaluates two oracle-aided selection settings. The first column reports the worst fixed attack observed in Table 10, where "worst" means the attack that yields the lowest TPR at 1% FPR for each method. The second column reports an evaluator that uses the detector score to select, for each sample, the lowest-scoring admissible transformation from a predefined pool consisting of GPT-3.5 paraphrasing, DIPPER-2, 1-hop translation, summarization at 0.5 retention, and style rewriting,

Table 10: Detection performance under semantic-preserving attacks (LLaMA-2-7B, sampling). TPR values are percentages; F1 values are proportions.

Setting	Method	1% FPR		10% FPR		Best
		TPR (%)	F1	TPR (%)	F1	F1
No Attack	KGW-1	98.4	0.987	100.0	0.952	0.995
	KGW-2	98.9	0.991	100.0	0.952	0.997
	KGW-4	99.0	0.992	100.0	0.952	0.997
	SIR	98.7	0.989	100.0	0.952	0.996
	EXP-Edit	99.5	0.996	100.0	0.952	0.998
	SS-WM	**99.1**	**0.993**	**100.0**	**0.952**	**0.998**
GPT-3.5 Paraphrase	KGW-1	59.0	0.738	88.5	0.891	0.905
	KGW-2	53.5	0.693	76.0	0.817	0.823
	KGW-4	22.5	0.364	49.0	0.614	0.705
	SIR	74.0	0.856	86.5	0.880	0.900
	EXP-Edit	43.5	0.602	64.5	0.739	0.775
	SS-WM	**92.6**	**0.948**	**96.5**	**0.935**	**0.952**
DIPPER-1	KGW-1	71.5	0.829	94.0	0.922	0.930
	KGW-2	45.0	0.616	71.0	0.785	0.815
	KGW-4	22.0	0.358	54.5	0.627	0.728
	SIR	76.5	0.862	90.5	0.903	0.920
	EXP-Edit	63.0	0.768	74.0	0.804	0.830
	SS-WM	**89.4**	**0.921**	**94.5**	**0.918**	**0.935**
DIPPER-2	KGW-1	76.5	0.862	93.5	0.918	0.925
	KGW-2	47.0	0.635	68.5	0.768	0.803
	KGW-4	15.0	0.259	47.5	0.603	0.718
	SIR	87.5	0.928	92.0	0.911	0.931
	EXP-Edit	48.5	0.649	63.5	0.732	0.775
	SS-WM	**91.2**	**0.940**	**95.8**	**0.930**	**0.948**
Translation (1-hop)	KGW-1	63.2	0.772	85.8	0.878	0.892
	KGW-2	54.8	0.705	72.5	0.802	0.828
	KGW-4	28.5	0.442	54.5	0.678	0.735
	SIR	80.5	0.887	94.5	0.924	0.938
	EXP-Edit	52.0	0.682	68.5	0.775	0.805
	SS-WM	**89.4**	**0.921**	**94.5**	**0.918**	**0.935**
Summarization (0.5)	KGW-1	42.5	0.595	73.5	0.815	0.792
	KGW-2	36.8	0.538	62.5	0.745	0.715
	KGW-4	19.5	0.325	48.5	0.625	0.672
	SIR	68.5	0.812	81.5	0.862	0.875
	EXP-Edit	42.5	0.595	58.5	0.725	0.748
	SS-WM	**81.2**	**0.865**	**89.8**	**0.882**	**0.905**
Style Rewrite	KGW-1	61.5	0.758	82.5	0.868	0.885
	KGW-2	54.2	0.702	71.8	0.795	0.822
	KGW-4	31.8	0.482	57.2	0.705	0.752
	SIR	79.2	0.880	91.8	0.912	0.928
	EXP-Edit	49.5	0.658	64.5	0.758	0.788
	SS-WM	**88.7**	**0.918**	**93.8**	**0.912**	**0.928**
Composed Attack	KGW-1	28.5	0.442	61.5	0.735	0.715
	KGW-2	24.5	0.394	49.5	0.658	0.635
	KGW-4	9.5	0.172	32.5	0.485	0.558
	SIR	68.5	0.812	79.5	0.845	0.865
	EXP-Edit	32.5	0.488	48.5	0.648	0.675
	SS-WM	**76.9**	**0.835**	**85.8**	**0.848**	**0.872**

Table 11: TPR (%) at 1% FPR under varying attack intensities.

Attack	Intensity	KGW-1	KGW-2	KGW-4	SIR	SS-WM
Paraphrase	Light (20%)	82.5	74.2	48.5	89.2	**96.8**
	Medium (40%)	68.4	58.7	35.8	82.5	**94.2**
	Strong (60%)	52.3	42.5	22.8	74.8	**89.7**
	V.Strong (80%)	38.2	28.5	15.2	65.2	**82.5**
Translation	1-hop	63.2	54.8	28.5	80.5	**89.4**
	2-hop	45.8	38.2	18.5	72.3	**87.8**
	3-hop	32.5	25.8	12.4	63.8	**81.4**
Summarization	0.75 retention	62.8	52.4	28.5	78.5	**93.5**
	0.50 retention	42.5	36.8	19.5	68.5	**81.2**
	0.25 retention	22.8	18.9	9.5	48.2	**68.5**

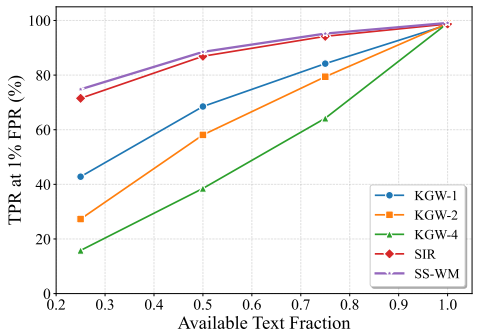


Figure 2: Partial-text attribution at 1% FPR. SS-WM remains more detectable than prior methods when only a short fragment of the generated text is available.

subject to semantic-similarity and quality constraints. Because this adaptive pool excludes multi-stage composed attacks and enforces admissibility constraints per sample, its TPR is not necessarily lower than the worst fixed composed attack. This oracle-aided adaptive setting tests per-sample selection beyond the primary score-blind threat model. SS-WM remains the strongest method under both worst fixed and constrained adaptive attacks, indicating that the joint semantic–stylistic carrier retains useful keyed evidence beyond a single attack template.

Table 12: TPR (%) under the worst fixed and constrained adaptive attack selection at 1% FPR. Lower TPR indicates a stronger attack. The worst fixed attack is selected from Table 10; the constrained adaptive attack uses detector scores to select per sample from the admissible single-transformation pool; both columns are oracle-aided stress tests.

Method	Worst Fixed Attack	Constrained Adaptive Attack
KGW-1	28.5 (Composed)	48.5
KGW-2	24.5 (Composed)	34.2
KGW-4	9.5 (Composed)	19.5
SIR	68.5 (Summ./Composed)	68.2
SS-WM	76.9 (Composed)	81.5

G.9 Oracle-Aided Score-Aware Candidate Selection Stress Test

As an oracle-aided stress test, we consider a score-aware candidate selection attacker. For each watermarked sample, the attacker generates m admissible paraphrase candidates and selects the one with the lowest detector score, subject to the same semantic-similarity and quality admissibility constraints used for the constrained adaptive attack. This yields a stronger attack than a single fixed paraphrase because the attacker explicitly searches over multiple semantically valid rewrites for one that most weakens attribution, while remaining weaker than unrestricted iterative detector-oracle optimization.

Table 13 reports TPR at 1% FPR under this attack. As expected, detection becomes harder as the candidate budget increases, since the attacker has more opportunities to find a low-score admissible rewrite. KGW-1 degrades sharply, dropping from 41.2 at $m = 5$ to 26.8 at $m = 20$, which reflects the fragility of token-level watermark evidence under rewrite search. SIR is more stable, decreasing from 72.4 to 63.7, suggesting that semantically similar paraphrase candidates tend to remain relatively close under its invariant representation, which limits the benefit of candidate selection. SS-WM remains strongest across all candidate budgets, decreasing from 84.6 at $m = 5$ to 78.3 at $m = 20$. This indicates that the semantic–stylistic carrier retains useful key-dependent evidence even when the attacker explicitly searches over multiple admissible rewrites. For brevity, this stress test reports KGW-1 as a token-level representative together with SIR and SS-WM; we do not extrapolate these results to baselines that were not evaluated under this oracle-aided setting.

Table 13: Detection under score-aware candidate selection at 1% FPR. For each sample, the attacker generates m admissible paraphrase candidates and selects the one with the lowest detector score. For brevity, we report KGW-1 as the strongest token-level baseline representative.

Method	$m = 5$	$m = 10$	$m = 20$
KGW-1	41.2	33.5	26.8
SIR	72.4	68.1	63.7
SS-WM	84.6	81.9	78.3

G.10 Quality Evaluation

The quality analysis reports automatic metrics: perplexity, ROUGE-L, QA EM/F1, self-BLEU, distinct-n, and Sentence-BERT semantic similarity. Sentence-BERT cosine similarity is reported as a descriptive metric because the same encoder family participates in the semantic carrier; we therefore do not treat it as independent semantic validation. The LLaMA-2-13B perplexity rises from 5.82 ± 1.24 without watermarking to 6.58 ± 1.45 with SS-WM, while the task-specific summarization and QA results remain close to the no-watermark values. No human-subject annotation study is reported.

G.11 Computational Cost

Table 16 reports interface-level generation costs, without an implementation-specific wall-clock latency estimate for the all-MiniLM-L6-v2 carrier: strict black-box reranking with m candidates requires approximately m times the base generation cost, while logit-access SS-WM performs feature-guided reranking every five decoding steps.

G.12 Edge-Case Attacks and Cross-Model Generalization

Table 14 tests whether SS-WM is overly specialized to the main attack families. It remains strong under synonym substitution, reaching 92.3% TPR under 30% substitution and 94.7% when context filtering is used. Copy-paste mixing is less damaging for all methods when the inserted watermarked segment remains sufficiently long, and SS-WM remains competitive in both same-topic and different-topic settings. Emoji insertion has little effect on SS-WM and SIR, but strongly disrupts KGW-2, suggesting that some token-level variants are sensitive to local token perturbations. Table 15 evaluates whether the learned detector generalizes across related base models. SS-WM achieves 0.981 when transferring from LLaMA-2-7B to LLaMA-2-13B, 0.958 to Mistral-7B, and 0.947 to OPT-2.7B, outperforming KGW-1 and SIR in all cross-model settings. This suggests that the semantic–stylistic carrier is not narrowly tied to one generator family.

Table 14: Additional robustness under edge-case attacks.

Attack Type	**KGW-1**	**KGW-2**	**SIR**	**SS-WM**
Copy-Paste (Same Topic)	92.0	91.2	91.8	**92.4**
Copy-Paste (Diff Topic)	**91.8**	91.1	88.1	89.5
Emoji Insertion	**99.0**	53.2	98.1	98.6
Synonym Substitution (30%)	78.5	68.2	88.5	**92.3**
Synonym + Context (30%)	82.3	72.8	91.2	**94.7**

G.13 Deployment Modes

Table 16 compares the main logit-access implementation with strict black-box sequence-level reranking. Logit-access reranking gives the strongest signal because the carrier objective can influence generation token by token, yielding 99.1% clean TPR, 92.6% under paraphrasing, and 76.9% under composed attacks. Black-box reranking is weaker because it can only select among completed samples from the base model, but it remains usable: with $m = 10$ candidates, it achieves 98.0% clean TPR, 89.1% under paraphrasing, and 72.8% under composed attacks. The candidate-generation cost

Table 15: Cross-model generalization (ROC-AUC point estimates). Per-example predictions are unavailable, so bootstrap confidence intervals cannot be computed. The in-distribution LLaMA-2-7B→LLaMA-2-7B values match the clean-text ROC-AUC in Table 9.

Fit source	Test generator	KGW-1	SIR	SS-WM
LLaMA-2-7B	LLaMA-2-7B	0.995	0.996	**0.998**
LLaMA-2-7B	LLaMA-2-13B	0.912	0.973	**0.981**
LLaMA-2-7B	Mistral-7B	0.865	0.945	**0.958**
LLaMA-2-7B	OPT-2.7B	0.848	0.932	**0.947**

increases approximately linearly with the number of completed candidates, from 5.0× for $m = 5$ to 10.0× for $m = 10$; these are relative sampling costs rather than measured wall-clock slowdowns. This clarifies the deployment tradeoff: logit access is preferable when available, while black-box reranking provides interface compatibility at a higher sampling cost.

Table 16: Comparison of SS-WM deployment modes at 1% FPR. Logit-access reranking provides the strongest signal, while strict black-box sequence reranking remains compatible with APIs that expose only completed text.

Mode	Clean	Paraphrase	Composed	Relative Sampling Cost
Logit-access reranking	99.1	92.6	76.9	1.0×
Black-box rerank ($m = 5$)	96.8	86.4	69.2	5.0×
Black-box rerank ($m = 10$)	98.0	89.1	72.8	10.0×

G.14 Hyperparameter Sensitivity

We vary the semantic and stylistic watermark strengths λ_{sem} and λ_{sty} to examine whether SS-WM depends on a narrow parameter setting. All other hyperparameters are held at their defaults ($\lambda_{\text{fid}} = 1.0$, $K = 50$, reranking every five steps). Results are reported on LLaMA-2-7B with C4 prompts, averaged over seeds $\{42, 43, 44\}$; attacked TPR uses GPT-3.5 Strong paraphrasing as a representative single attack at 1% FPR.

The results show the expected tradeoff: stronger weights improve attacked detection but may slightly increase perplexity or reduce semantic similarity. The default setting ($\lambda_{\text{sem}} = 2.5$, $\lambda_{\text{sty}} = 2.0$, marked with †) balances robustness and quality and is used throughout the main experiments. Encouragingly, the method is not sensitive to small perturbations around the default: varying either weight by ± 0.5 changes attacked TPR by at most 2.1 percentage points and perplexity by at most 0.18.

Table 17 also reports semantic-only ($\lambda_{\text{sty}} = 0$) and stylistic-only ($\lambda_{\text{sem}} = 0$) configurations. Both are weaker than the joint default, motivating the coupled design of SS-WM.

Table 17: Hyperparameter sensitivity on LLaMA-2-7B (C4, GPT-3.5 Strong paraphrase attack). TPR values are at 1% FPR. † marks the default configuration used in all main experiments. Perplexity is evaluated with LLaMA-2-13B as oracle. Semantic similarity is cosine similarity between Sentence-BERT embeddings.

λ_{sem}	λ_{sty}	**Clean TPR** (%)	**Para. TPR** (%)	**Sem. Sim.**	**PPL** $\downarrow$
0.5	0.5	96.8	78.2	0.924	6.35
1.0	1.0	97.5	83.5	0.921	6.41
1.5	1.5	98.2	87.1	0.918	6.47
2.0	1.5	98.6	89.4	0.916	6.51
2.0	2.0	98.8	90.8	0.915	6.54
2.5	1.5	98.9	91.2	0.914	6.55
$2.5^{\dagger}$	$2.0^{\dagger}$	**99.1**	**92.6**	0.912	6.58
2.5	2.5	99.1	92.8	0.910	6.63
3.0	2.0	99.2	93.4	0.908	6.69
3.0	2.5	99.2	93.7	0.906	6.74
3.5	2.5	99.3	94.1	0.902	6.83
3.5	3.0	99.3	94.2	0.899	6.91
4.0	3.0	99.3	94.5	0.895	7.08
Semantic only ($\lambda_{\text{sty}} = 0$)		98.2	85.7	0.917	6.49
Stylistic only ($\lambda_{\text{sem}} = 0$)		97.8	78.5	0.919	6.44