CONSTITUTION-GUIDED WATERMARKING

Anonymous authors
Paper under double-blind review

ABSTRACT

Watermarking enables language model providers to identify text generated by their models. However, its desired properties can conflict (*i.e.* stronger watermark signals can degrade text quality), while designs that resist editing may also facilitate forgery. Providers address these trade-offs by choosing configurations that balance competing objectives or prioritize particular properties. Either approach imposes a shared operating point on requests with different requirements, potentially sacrificing quality where wording preservation matters or robustness where reliable attribution is essential. To allow flexible and adaptable designs, we introduce *Constitution-Guided Watermarking*, a framework that selects request-appropriate trade-offs from provider requirements, listed as natural-language principles. *Offline*, a pretrained reasoning agent examines constitutional rules alongside watermark implementations and iteratively refines rule-specific configurations using empirical feedback. *At deployment*, a separate monitor identifies applicable rules and retrieves the corresponding policy, including watermarking exemptions, without modifying the serving model. Furthermore, our framework supports offline parallel optimization and refinement of rule-specific configurations based on evolving provider requirements without affecting deployment, and binds each deployed configuration to its evaluation evidence, making deployment decisions auditable. In a proof-of-concept evaluation using KGW and a five-rule constitution, our framework selects configurations responsive to provider priorities and improves post-paraphrase detection on robustness-prioritized requests by up to 14 percentage points over fixed configurations, while matching or exceeding all baselines in aggregate quality and clean detection at a nominal 0.1% false-positive rate.

1 INTRODUCTION

Watermarking enables language model providers to identify generated text by embedding a secret-key-dependent signal during generation (Kirchenbauer et al., 2023; Zhao et al., 2024a; Dathathri et al., 2024). A useful watermark should remain detectable, preserve output quality, survive relevant transformations, and resist forgery. These objectives can conflict (Zhao et al., 2024b). For bias-based schemes such as KGW (Kirchenbauer et al., 2023), increasing watermark strength can improve detectability at the cost of distortion (Wouters, 2024). At the same time, a watermark that survives benign editing may also survive malicious changes to the text's meaning, enabling false attribution (Pang et al., 2024). Watermark deployment therefore requires deciding which properties to prioritize or trade off and which compromises are acceptable.

(a) Watermark Verification Agent (Offline)

Rule	Requirement	Certified Config
r1	creative writing <balance trade-offs>	δ=2.0, γ=0.5
r2	news text <strong detectability>	δ=4.0, γ=0.5
r3	code <preserve quality>	δ=2.0, γ=0.25
r4	medical advice <no watermark>	no watermark

(b) Deployment Monitor

```
Prompt:      write a short report on last night's storm.
Reasoning:   reportage of a real, dated event for public circulation,
             not fiction → news text, not creative writing; r2 is the
             narrower match.
Generation:  Winds of up to ninety kilometres an hour downed power
             lines across the eastern districts, and crews expect
             repairs to continue through the weekend.
```

Figure 1: Conceptual illustration: (a) offline, the verification agent assigns each rule a configuration; (b) at deployment, the monitor routes a prompt to its rule. Rules shown are illustrative.

Existing approaches address these trade-offs by selecting configurations that balance competing objectives or designing schemes that favor particular properties (Kuditipudi et al., 2023; Wouters,

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.

2024; Dathathri et al., 2024). However, a configuration that performs well on average need not meet the requirements of each request. For example, a provider may prioritize wording preservation when proofreading human-authored text, minimize interference with deterministic answers, and require detection after paraphrasing for misuse-prone generations. Applying the same compromise throughout can impose unnecessary distortion on quality-sensitive requests while providing insufficient robustness where attribution matters most. This motivates selecting operating points according to the requirements of each request.

These distinctions have practical consequences as watermarking enters deployment. Article 50 of the EU AI Act requires machine-readable marking of synthetic content, with exceptions including assistance with standard editing (EU AI Act, 2024). Anthropic's announcement of text watermarking for Claude illustrates a related attribution limitation: a detected watermark indicates model involvement but does not distinguish model-authored text from human text that the model heavily edited (Anthropic, 2026). Providers consequently need policies governing not only how reliably a watermark is detected, but also when it should be applied and what its presence is intended to convey.

Request-dependent requirements cannot be reduced to subject matter alone. Proofreading a medical explanation and generating medical advice concern the same topic, yet a provider may exempt the former and require low-distortion watermarking for the latter. Adaptive watermarking methods already adjust embedding according to token uncertainty or generation features (Liu & Bu, 2024; Wang et al., 2025b;a). Such adaptation addresses how to embed a signal under a chosen objective; our focus is how providers specify which objectives and exemptions apply to a request. Natural-language rules can express these distinctions directly, while quantitative acceptance criteria make the resulting policies testable. Connecting each configuration to its governing rule and evaluation evidence also makes deployment decisions inspectable. This motivates our central question:

Central Question

Can natural-language provider requirements guide the selection and empirical evaluation of request-dependent watermark policies?

Our Framework. We introduce *constitution-guided watermarking*, a framework in which providers specify permitted watermark algorithms, ordered natural-language rules, and quantitative acceptance gates. Together, these requirements form a *watermark constitution*, with rules such as "do not watermark low-risk editing" or "for misuse-prone requests, prioritize robustness over quality" (Figure 1). Inspired by Constitutional AI (Bai et al., 2022), the constitution governs the watermarking system rather than the serving model, which remains frozen. The framework separates two learning problems. An offline *watermark verification agent* learns to relate watermark mechanisms, theoretical assumptions, and empirical trade-offs to constitutional objectives. An online *deployment monitor* learns to identify the rules applicable to a user prompt and retrieve the corresponding policy. This separation keeps expensive algorithmic analysis offline and constrains deployment to previously evaluated configurations. Our proof-of-concept study instantiates both roles with pretrained models.

Before deployment, the verification agent examines each permitted watermark's implementation and configuration, then proposes a rule-specific configuration with a supporting rationale. A *deterministic verifier* rejects invalid proposals before the remaining candidates are evaluated on a held-out, rule-specific reference set. The resulting measurements and unmet requirements are returned to the agent, which revises its proposals over successive iterations. Each selected policy is stored with its empirical evidence i.e., configuration, evaluation evidence, and gate status. If no evaluated configuration satisfies a rule's gates within the search budget, the framework returns a best-effort alternative and explicitly records the unmet requirements. At deployment, the monitor identifies applicable rules and retrieves the corresponding policy which is enforced during generation.

Contributions. (i.) First, we introduce *Constitution-Guided Watermarking*, to our knowledge the first framework to operationalize provider-written natural-language constitutions as request-dependent text watermarking policies. Inspired by Constitutional AI (Bai et al., 2022), the constitution specifies when watermarking applies and which properties take priority, while separate agents perform offline configuration refinement and deployment monitoring. **(ii.)** Building on this formulation, we develop an iterative procedure in which the watermark verification agent reasons over the constitution rules and watermark algorithm, proposes rule-specific configurations, and revises them using empirical feedback, with deterministic checks enforcing configuration validity and policy execution. We support this procedure with constitution-specific datasets generated through proposal and critique,

and explicitly record unmet requirements when the selected configurations remain best-effort. **(iii.)** We present a proof-of-concept evaluation using KGW, one serving model, and a five-rule constitution, comparing against two fixed configurations and budget-matched random search. Our framework improves post-paraphrase detection by up to 14 percentage points on robustness-prioritized requests while matching the best clean detection and quality. We report routing failures, observed false-positive rates, and unmet requirements alongside these gains. We further examine configuration sensitivity to constitutional priorities and the effect of deployment monitoring on policy execution. Appendix A discusses the related work in detail.

2 Problem Formalization

Watermarking system. Let $\mathcal{M}$ be the frozen serving model, $\pi \in \Pi$ a prompt, and $x \in \mathcal{R}$ a response. Unwatermarked generation is written $\mathcal{M}(\pi) \to x$. For our framework, we additionally index each permitted algorithm $a \in \mathcal{W}$ by a legal configuration $\psi \in \Psi_a$:

$$\mathsf{Watermark}_{\mathsf{gk}}^{\mathcal{M}}(\pi; a, \psi) \to x. \tag{1}$$

Here, gk and dtk denote generation and detection keys; in practice they are mostly the same. At nominal FPR α, the configured detector is

$$\mathsf{Detect}_{\mathsf{dtk}}^{\alpha}(x; a, \psi) = \begin{cases} \mathsf{true}, & s_{a,\psi,\mathsf{dtk}}(x) \geq \eta_{a,\psi,\mathsf{dtk},\alpha}, \\ \mathsf{false}, & \text{otherwise}, \end{cases} \tag{2}$$

where s is the detection score and η is calibrated on null text. We fix the deployment key pair throughout the empirical evaluation. Detection must match generation in its key, tokenizer, vocabulary size, and scoring-relevant parameters. Generation-only parameters, such as KGW's logit bias δ, can differ without changing the detector.

Provider constitution. A constitution is

$$\mathcal{C} = (\mathcal{W}, \mathcal{S}, \prec, \boldsymbol{\tau}), \tag{3}$$

where $\mathcal{S} = \{r_1, \ldots, r_m\}$ is the rule set, $\prec$ specifies precedence, and $\boldsymbol{\tau}$ contains quantitative gates. We use $\mathcal{S}$ for rules to reserve $\mathcal{R}$ for responses. Each rule has a reviewable representation $r = (u_r, d_r, \mathbf{o}_r, \mathcal{G}_r)$: its authoritative natural-language instruction u_r, watermarking decision $d_r \in \{\mathsf{WM}, \mathsf{NoWM}\}$, ordered objectives $\mathbf{o}_r$, and active gates $\mathcal{G}_r$.

Let $\mathcal{S}(\pi)$ contain the rules applicable to π. Precedence resolves conflicting watermarking decisions. For compatible watermarked rules $\mathcal{S}_{\mathsf{WM}}(\pi)$, objective recurrence determines priority:

$$n_j(\pi) = \sum_{r \in \mathcal{S}_{\mathsf{WM}}(\pi)} \mathbf{1}[j \in \mathbf{o}_r]. \tag{4}$$

The combined profile orders objectives by decreasing $n_j(\pi)$, breaking ties by provider rule order and retaining the applicable gates. Thus, two requests for quality outweigh one request for robustness in the objective order without removing the robustness requirement.

Constitutional objectives. For a rule profile c, let $\mathcal{D}_c$ be its prompt distribution and $\mathcal{A}_{\mathrm{FPR}}$ the provider's nominal FPR regimes. For each evaluated prompt π_i, generate $\mathsf{Watermark}_{\mathsf{gk}}^{\mathcal{M}}(\pi_i; a, \psi) \to x_i$. The *empirical detectability rate* is

$$\widehat{D}_c(a, \psi) = \min_{\alpha \in \mathcal{A}_{\mathrm{FPR}}} \frac{1}{n_c} \sum_{i=1}^{n_c} d_{a,\psi}^{\alpha}(x_i). \tag{5}$$

For the fixed detection key, empirical FPR on a null set $\mathcal{D}_0$ disjoint from threshold calibration is

$$\widehat{F}_{\alpha}(a, \psi) = \frac{1}{|\mathcal{D}_0|} \sum_{z \in \mathcal{D}_0} d_{a,\psi}^{\alpha}(z). \tag{6}$$

For a reference language model q and a global natural-text baseline b_q fixed before search, the *quality pass rate* is

$$\widehat{Q}_c(a, \psi) = \frac{1}{n_c} \sum_{i=1}^{n_c} \mathbf{1}[\mathrm{PPL}_q(x_i) \leq b_q]\,. \tag{7}$$

Low-distortion or exactness requirements additionally need task-specific measurements, such as $L_c(x_i, x_i^0)$ relative to $\mathcal{M}(\pi_i) \to x_i^0$ or an appropriate task reference. The PPL gate alone establishes neither task correctness nor distribution preservation.

Let $\mathcal{E}$ be an attack channel. Let $\mathcal{T}_c$ contain the channels relevant to profile c, and let $x'_{i,b} \leftarrow \mathcal{E}(x_i)$ be B independently randomized attacks conditional on each response. At $\alpha_* = \min \mathcal{A}_{\mathrm{FPR}}$, ***empirical robustness*** is

$$\widehat{R}_c(a, \psi) = \min_{\mathcal{E} \in \mathcal{T}_c} \frac{1}{n_c B} \sum_{i=1}^{n_c} \sum_{b=1}^{B} d_{a,\psi}^{\alpha_*}(x'_{i,b}). \tag{8}$$

We report attack validity and execution failures separately and cluster uncertainty estimates by source prompt. Robustness is evaluated only when activated by the rule. For NoWM decisions, responses instead follow $\mathcal{M}(\pi) \to x$ and are evaluated for action compliance and any applicable quality requirements. End-to-end evaluation retains generation and routing failures in its attempted-prompt denominators and reports their counts separately.

Unlike removal attacks, *forgery* seeks acceptance of content not legitimately issued by the provider. For profile c, let $\mathcal{H}_c$ denote such target content and Acc_c the provider's acceptance rule. An adversary $\mathcal{B}$ observing N watermarked responses produces $x^\star$, with success probability

$$\mathrm{Adv}_c^{\mathrm{forge}}(\mathcal{B}, N) = \Pr[x^\star \in \mathcal{H}_c \ \wedge \ \mathsf{Acc}_c(x^\star) = 1]\,. \tag{9}$$

A compatible extension randomly selects one of $K \geq 2$ secret keys within each profile and accepts text only when exactly one key detects it (Aremu et al., 2025). For blind attackers unable to separate samples by key, assuming key-symmetric, mutually independent detection outcomes on forged text, success is bounded by $(1 - 1/K)^{K-1}$ per attempt, irrespective of N while these assumptions hold. Appendix D.1 gives the construction, assumptions, proof, and FPR calibration.

Feasibility and best effort. For gate $g \in \mathcal{G}_c$, let $\widehat{m}_{c,g}(a, \psi)$ be its measured value, τ_g its threshold, and $\bowtie_g$ its comparison. Empirical acceptance requires

$$\mathrm{Pass}_c(a, \psi) = \bigwedge_{g \in \mathcal{G}_c} [\widehat{m}_{c,g}(a, \psi) \ \bowtie_g \ \tau_g]\,. \tag{10}$$

The provider's objective order $\mathbf{o}_c$ determines the incumbent among tested passing configurations. If none passes within the search budget, the system returns the best measured incumbent under that order, labels it BEST-EFFORT, and reports it explicitly. Note that failure to find a passing configuration does not prove that the constitution is infeasible.

3 METHODOLOGY

Constitution-guided watermarking separates offline policy construction search from online policy execution. A verification agent A_θ proposes rule-specific watermark policies using watermark implementations and empirical feedback, while a deployment monitor M_φ selects the applicable policy for each request. Our proof-of-concept uses pretrained agents without weight updates (Appendix D.2 gives objectives for training both roles); the serving model $\mathcal{M}$ remains frozen. Prompts $\pi \in \Pi$ follow Section 2; $\mathcal{P}$ denotes the evaluated deployment policy. Figure 2 illustrates this separation.

3.1 DATASET GENERATION

Evaluating a constitution requires prompts labeled with the rules they trigger, which existing watermarking benchmarks do not provide. We therefore construct constitution-specific data through a generate–critique–revise loop adapted from Constitutional AI (Bai et al., 2022). Given the constitution and a target scenario, a *generator* proposes a realistic user prompt. An independent *critic* then labels every applicable rule, the resolved watermark action, and the active gates, and rejects prompts that are unrealistic or do not match the target scenario. A *reviser* repairs rejected prompts, which are critiqued again; prompts that still fail are discarded. To test rule recognition beyond surface wording, we add paraphrases of accepted prompts and include prompts that trigger multiple rules, which test rule recognition under overlapping requirements. Before search begins, we fix a small rule-stratified reference set $\mathcal{D}_{\mathrm{ref}}$ and a disjoint test set $\mathcal{D}_{\mathrm{test}}$. Each source prompt and its paraphrases are assigned to the same split to prevent leakage between them. Only $\mathcal{D}_{\mathrm{ref}}$ enters the configuration loop; because it is reused across rounds, it serves as adaptation data, and all reported results use $\mathcal{D}_{\mathrm{test}}$.

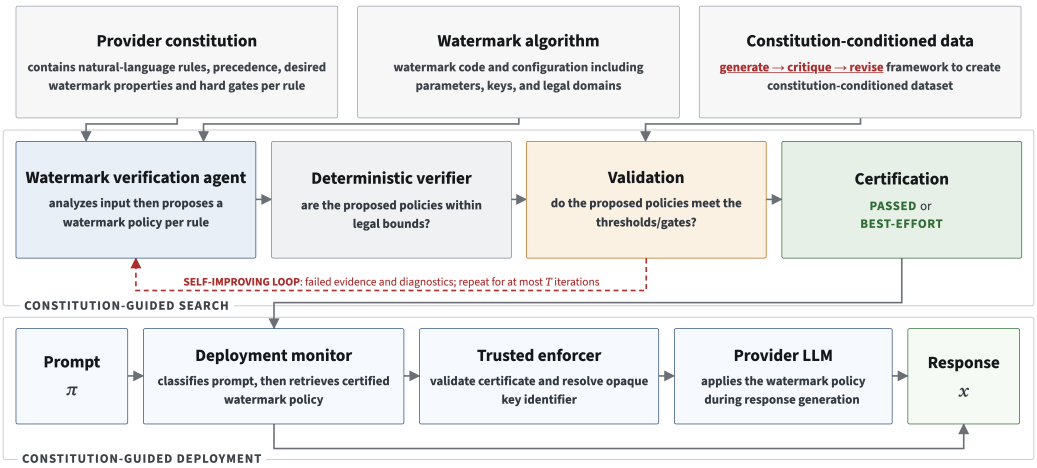


Figure 2: Constitution-Guided Watermarking framework.

3.2 Constitution-Guided Search

Algorithm 1 Source-grounded constitutional configuration search

Input: A_θ, constitution $\mathcal{C}$, source S, legal domains $\{\Psi_a\}$, frozen $\mathcal{M}$, $\mathcal{D}_{\text{ref}}$, budget T
Output: policy $\mathcal{P}$ with measured incumbents and gate status

$U \leftarrow$ watermarked profiles; $H_c \leftarrow \emptyset$ for every $c \in U$
▷ Refine configurations using reference-set feedback.
for $t = 1, \ldots, T$ **while** $U \neq \emptyset$ **do**
  $Z \leftarrow A_\theta(\mathcal{C}, S, U, \{H_c\}_{c \in U})$
  ▷ Reject illegal configurations before generation.
  $Z \leftarrow \textsc{ValidateOrRequestRepair}(Z, S, \{\Psi_a\})$
  $\widehat{\mathbf{m}} \leftarrow \textsc{Evaluate}(\mathcal{M}, \mathcal{D}_{\text{ref}}, Z)$
  for each $c \in U$ **do**
    $H_c \leftarrow H_c \cup \{(Z_c, \widehat{\mathbf{m}}_c)\}$
    ▷ Prefer passing candidates; otherwise retain the best measured trade-off.
    $z_c^* \leftarrow \textsc{BestMeasured}(H_c, \mathbf{o}_c, \mathcal{G}_c)$
    if $\text{Pass}_c(z_c^*)$ and freeze is enabled **then** $U \leftarrow U \setminus \{c\}$
  end for
end for
▷ Attach evidence, exemptions, and explicit unmet requirements.
$\mathcal{P} \leftarrow \textsc{BindEvidence}(\mathcal{C}, S, \{z_c^*, H_c\})$
return $\mathcal{P}$

At iteration t, our watermark verification agent A_θ receives the original natural-language rules, watermark algorithm, and previous reference measurements. The agent proposes one watermark policy $(a_c^{(t)}, \psi_c^{(t)})$ per unresolved profile and states its rationale, assumptions, and expected effect on each active gate. Deterministic checks enforce source provenance, parameter legality, detector binding, profile coverage, and proposal novelty. Reference answers follow $\mathsf{Watermark}_{\mathsf{gk}}^{\mathcal{M}}(\pi_i; a_c^{(t)}, \psi_c^{(t)}) \to x_i$ under each legal proposal. For reasoning models, watermark generation and detection operate on the visible answer, not the preceding reasoning. We then calibrate detector thresholds using null text and use a separate null reference set to measure FPR. Our evaluator checks only the profile's active gates, records failures and sample counts, and returns this evidence to the watermark verification agent. The agent retains the strongest measured incumbent under the profile's objective order and revises unsuccessful proposals. When freezing is enabled, profiles satisfying all active gates are removed from subsequent search. Unresolved profiles continue until the budget is exhausted, and their retained configurations are labeled best-effort. After at most T rounds, every profile has a passing record or an explicit *best-effort* alternative.

Algorithm 2 Constrained constitutional deployment

Input: M_φ, prompt π, constitution $\mathcal{C}$, evidence index $\mathcal{I}$ for policy $\mathcal{P}$, frozen $\mathcal{M}$

▷ Identify applicable rules and resolve overlapping requirements.

$\widehat{\mathcal{S}}(\pi) \leftarrow M_\varphi(\pi, \mathcal{C}, \mathcal{I})$

$c \leftarrow \textsc{ResolveAndAggregate}(\widehat{\mathcal{S}}(\pi), \mathcal{C})$

$z \leftarrow \textsc{RetrieveRecord}(c, \mathcal{I})$

▷ Use the provider's fallback when no authorized policy can be applied.

if uncertain, missing, or unauthorized **then return** $\textsc{ProviderFallback}(\pi, c)$

▷ Enforce the retrieved action and its bound configuration.

$(d, a, \psi, \kappa) \leftarrow \textsc{EnforceAndBind}(z, \mathcal{I})$

if $d = \mathsf{NoWM}$ **then** $\mathcal{M}(\pi) \to x$

▷ Resolve secret keys only within the trusted runtime.

else $\mathsf{gk} \leftarrow \textsc{ResolveGenerationKey}(\kappa)$

$\mathsf{Watermark}^{\mathcal{M}}_{\mathsf{gk}}(\pi; a, \psi) \to x$

end if

▷ Record the executed policy without exposing secret keys.

$\textsc{Audit}(c, z, d, a, \psi, \kappa)$; **return** x

κ is an opaque key identifier. The trusted runtime resolves gk; the corresponding dtk is bound to the record for detection. Neither the monitor nor the log receives secret keys.

3.3 Constitution-Guided Deployment

After the iterative search process is complete, a versioned evidence record which holds the constitution rules, watermark information, evaluation records, and proposed watermark policy per rule is returned. We call this record a *certificate*, with an explicit scope that reveals the status of the pre-deployment search. Reference-Passed means that observed reference estimates satisfy the gates, while Best-Effort exposes every unmet gate. Note that exhausting a finite search budget only establishes that no passing configuration was found, and not that the constitution is mathematically infeasible. Ideally, deploying a best-effort record should depend on the provider approval. For prompt π, the deployment monitor predicts $\widehat{\mathcal{S}}(\pi)$ and retrieves the corresponding records from policy $\mathcal{P}$. For overlapping rules, provider precedence resolves the action and Equation 4 resolves compatible priorities. Our framework then validates the selected action and record and resolves gk through the key manager when watermarking is required. We evaluate known-configuration detection using the detector bound to the generation record. Appendix D.3 discusses this assumption and an unevaluated extension for prompt-free detector selection.

4 Experimental Setup

Constitutional-KGW setup. We instantiate our *watermark verification agent* A_θ with pretrained `gpt-5.6-luna` at high reasoning effort and *deployment monitor* M_φ with `gpt-5.4-nano` also at high effort. Table 1 summarizes the five rules reproduced in Appendix B.4. To generate our corpus, we use GPT-OSS-120B as generator and Qwen3-32B as critic. During the offline search process, evaluation is carried out on 10 reference prompts per rule (50 total) for at most five rounds. To test before deployment, we use 100 disjoint test prompts per rule (500 per method), never as feedback.

Table 1: **Provider constitution**, in first-applicable precedence order.

Scenario	Action	Constitutional priority
Misuse-prone requests	WM	Robustness over quality
Low-risk editing	NoWM	No watermark application
High-stakes advice	WM	Quality, low distortion, high detection
Deterministic answers	WM	Quality and detectability
General generation	WM	Minimize unnecessary distortion

We also leverage Qwen3-32B as our provider's LLM, allowing 8,192 total new tokens and at most 1,024 visible-answer tokens. We only watermark the generated response if the constitution allows and do not watermark the reasoning tokens. For our main experiment, we focus on the KGW watermark (Kirchenbauer et al., 2023), using its implementation within the MarkLLM repository (Pan

Table 2: **Main results.** Clean detection covers watermark-required rules ($n = 400$); quality measures PPL ≤ 18.25 on the same prompts. Robustness covers 100 robustness-required prompts (5 attacks/response). Brackets show 95% percentile-bootstrap intervals, clustered by prompt for robustness. All values are percentages; best results are **bolded**. Robustness at $\alpha = 1\%$ is in Appendix C.4.

	Clean Detection		**Quality**	**Robustness** ($\alpha = 0.1\%$)		**Compliance**
Method	$\alpha = 1.0\%$	$\alpha = 0.1\%$	PPL ≤ 18.25	DIAA	DIPPER	Exempt
Fixed KGW A	95.5 [93.5, 97.5]	92.5 [90.0, 95.0]	98.4 [97.2, 99.4]	2.8 [1.0, 4.8]	3.0 [1.6, 4.6]	0.0
Fixed KGW B	80.0 [76.3, 83.8]	61.8 [57.3, 66.3]	98.2 [97.0, 99.2]	1.4 [0.4, 2.4]	8.6 [6.0, 11.4]	0.0
Random search	**98.0** [96.5, 99.3]	**97.5** [96.0, 99.0]	97.2 [95.8, 98.6]	12.2 [8.0, 17.0]	11.6 [8.4, 15.0]	**100.0**
C-KGW (Ours)	**98.0** [96.5, 99.3]	**97.5** [96.0, 99.0]	**98.6** [97.4, 99.6]	**13.8** [10.8, 16.8]	**17.0** [12.8, 21.4]	**100.0**

et al., 2024), with integer $\delta \geq 1$, $0 < \gamma < 1$, integer context length $h \geq 1$, and $f \in \{\texttt{time}, \texttt{additive}, \texttt{skip}, \texttt{min}\}$ available to the verification agent.

Baselines and controls. Our experiments compare *Constitutional-KGW* with two fixed KGW settings: $(\gamma, \delta, h) = (0.25, 4, 3)$ and $(0.5, 2, 1)$, both with $f = \texttt{time}$. These are prespecified operating points by Piet et al. (2023) and Pan et al. (2024), not demonstrated constitution-specific optima. We also evaluate a budget-matched random search baseline that shares our deployment monitor (Appendix B.3).

Gates and attacks. Watermarked rules require **(i.)** detection ≥ 0.95, **(ii.)** quality pass rate ≥ 0.90, **(iii.)** robustness ≥ 0.50, and **(iv.)** empirical FPR ≤ 0.01. Detectors are calibrated at target FPRs $\alpha \in \{0.01, 0.001\}$ using 5,000 natural texts, with another 5,000 null texts used for reference FPR measurement. For quality, we use answer-only perplexity from Qwen2.5-7B-Instruct and a global threshold $b_q = 18.25$, the fixed 95th percentile of 200 natural-text samples. As for robustness, we leverage one non-adaptive and one adaptive paraphraser. DIAA's adaptive paraphraser and DIPPER (non-adaptive) each generate five attacks per answer, and we report the attack channels separately and gate their minimum (Diaa et al., 2025; Krishna et al., 2023). A separate semantic-validity audit is required to assess whether removal preserves the answer. Full decoding, budget and attack settings are in Appendix B.

Metrics. We report clean detection rates across the four watermark-required rules ($n = 400$). Quality measures fluency as the fraction of attempted watermark-required prompts ($n = 400$) yielding a response with PPL at or below the global threshold. For the robustness-required rule, we report post-attack detection separately for DIAA and DIPPER at nominal 0.1% FPR, using five attacks per response across 100 prompts. Editing compliance measures how often an editing request receives a response generated without the watermark processor. Routing and generation failures remain in the attempted-prompt denominators as unsuccessful outcomes. Attack execution failures likewise remain in the robustness denominators as non-detections. We give 95% bootstrap confidence intervals for generation metrics, resampling source prompts and keeping each response's attack repetitions together. At both FPR operating points (1% and 0.1%), we report observed false positives on unwatermarked texts with exact binomial confidence intervals (Appendix C.2). These null measurements were used as reference feedback and do not constitute an independent post-selection audit. The empirical FPR gate is $\leq 1\%$; passing it does not establish operation at 0.1% FPR.

5 Experimental Evaluation

We evaluate four questions: (i) whether Constitutional-KGW improves on fixed and searched configurations end to end (Table 2); (ii) how iterative feedback changes the agent's proposals (Table 3); (iii) whether selected policies respond to changes in constitutional priority (Table 4); and (iv) how deployment monitoring affects policy execution (Table 5). Observed false-positive rates of all deployed detectors are in Appendix C.2 (Table 7).

Main Results. Table 2 summarizes our evaluation across detection, quality, robustness, and compliance. On the 100 robustness-targeted prompts, Constitutional-KGW achieves post-paraphrase detection of 13.8% against DIAA and 17.0% against DIPPER, compared with 2.8% and 3.0% for

Fixed KGW A and 1.4% and 8.6% for Fixed KGW B, corresponding to gains of $+11.0$ and $+14.0$ percentage points over Fixed KGW A. These gains are measured on the robustness-active rule itself rather than diluted across unrelated requests. Relative to random search, Constitutional-KGW is more robust to both attacks (13.8% vs. 12.2% on DIAA; 17.0% vs. 11.6% on DIPPER), significantly for DIPPER (paired bootstrap; Appendix C.4). Both channels nonetheless remain well below the provider's 50% requirement, so the framework reports this rule as a best-effort policy rather than a validated one (Table 3). Across the four watermark-required rules ($n = 400$), clean detection at nominal $\alpha = 0.1\%$ is 97.5%, matching random search and exceeding Fixed KGW A (92.5%) and Fixed KGW B (61.8%); Appendix C.1 lists the configuration each method stores per rule. The quality pass rate (PPL ≤ 18.25) is 98.6%, comparable to Fixed KGW A (98.4%), Fixed KGW B (98.2%), and random search (97.2%). Finally, our method correctly exempts all 100 editing requests, whereas the fixed baselines watermark every request. ***(Key Takeaway.)*** In this proof-of-concept, Constitutional-KGW delivers targeted robustness gains under paraphrase attacks and full policy compliance while matching the best clean detection and quality, but paraphrase robustness remains far below deployment targets.

Impact of Iterative Refinement. To show what the iterative watermark verification agent changes, we compare the initial and final proposals for our robustness-prioritized rule (Table 3). The initial proposal ($\delta = 3$) failed every robustness check on the 10-prompt reference set. In response to this feedback, the agent doubled the bias to $\delta = 6$ and left γ, context length, and scoring function unchanged. The final revision raised clean detection at nominal 0.1% FPR from 90% to 100% and worst-channel robustness from 0% to 10%, with no loss in the observed quality pass rate. Because robustness still fell well short of the 50% requirement, the framework exported the policy as *best-effort* rather than validated. It reported the unmet constraint explicitly instead of silently accepting it. These figures come from 10 reference prompts, so they illustrate the refinement mechanism rather than establish generalization.

Table 3: **Impact of Iterative Refinement.** $\gamma = 0.5$, $h = 4$, and $f = \texttt{min}$ are unchanged. Detection uses nominal $\alpha = 0.1\%$. Rates are percentages; best results are **bolded**. Full trace in Appendix C.5.

Reference-set measure	Initial	Final
KGW bias, δ	3	6
Clean detection ($\alpha = 0.1\%$)	90.0	**100.0**
Quality pass rate	**100.0**	**100.0**
Worst-channel robustness	0.0	**10.0**
Policy status	–	Best-effort

Table 4: **Constitution sensitivity on 100 held-out prompts.** Each variant runs the same pipeline under a different constitution priority; all selected $\gamma = 0.5$. Configurations are (δ, h, f). Quality measures PPL ≤ 18.25; robustness uses five attacks per response at nominal $\alpha = 0.1\%$. Quality and clean-detection brackets are 95% Clopper–Pearson intervals; robustness brackets are 95% bootstrap intervals clustered by prompt. All values are percentages; higher is better. Best results are **bolded**.

		Clean Detection		Quality	Robustness ($\alpha = 0.1\%$)	
Constitution	(δ, h, f)	$\alpha = 1.0\%$	$\alpha = 0.1\%$	PPL ≤ 18.25	DIAA	DIPPER
Quality-first	$(2, 1, \texttt{time})$	$93.0_{\,[86.1, 97.1]}$	$91.0_{\,[83.6, 95.8]}$	$\mathbf{98.0}_{\,[93.0, 99.8]}$	$4.6_{\,[2.4, 7.6]}$	$5.6_{\,[3.0, 8.4]}$
Robustness-first	$(6, 2, \texttt{additive})$	$\mathbf{100.0}_{\,[96.4, 100.0]}$	$\mathbf{98.0}_{\,[93.0, 99.8]}$	$91.0_{\,[83.6, 95.8]}$	$\mathbf{15.6}_{\,[12.5, 19.1]}$	$\mathbf{18.6}_{\,[15.3, 22.3]}$
Balanced	$(4, 1, \texttt{time})$	$98.0_{\,[93.0, 99.8]}$	$97.0_{\,[91.5, 99.4]}$	$94.0_{\,[87.4, 97.8]}$	$13.2_{\,[8.6, 18.0]}$	$17.0_{\,[13.2, 21.4]}$

Sensitivity to Constitutional Priorities. To test whether the agent responds to stated priorities, we run the pipeline under three constitutions that differ only in their trade-off preference (Table 4; exact wording in Appendix C.3). All three variants retain $\gamma = 0.5$ while selecting different bias strengths and, for robustness-first, different context and seeding settings. The quality-first variant selects the weakest bias ($\delta = 2$) and attains the highest quality pass rate (98.0%), at the cost of the lowest robustness (4.6% DIAA, 5.6% DIPPER). The robustness-first variant selects the strongest bias ($\delta = 6$) and attains the highest detection and robustness (15.6% DIAA, 18.6% DIPPER), while quality falls to 91.0%. The balanced variant ($\delta = 4$) falls between the two on every metric. However, its intervals overlap with robustness-first throughout, so adjacent priorities are not reliably distinguished at this sample size. Moreover, no variant exceeds 19% paraphrase robustness. These results therefore provide limited evidence of configuration sensitivity: the constitution shifts the trade-off in the intended direction, but reliably separating neighboring priorities and meeting stringent robustness targets remain open challenges.

Table 5: **Impact of deployment monitor.** Clean detection is averaged over watermark-required rules ($n = 400$); robustness covers 100 prompts under paraphrase attack at $\alpha = 0.1\%$; quality measures PPL ≤ 18.25 across advice and deterministic requests ($n = 200$). Rates are percentages; best results are **bolded**. Evasion probes and a routing rerun are in Appendix C.6.

	Rule	**Clean Detection**		**Quality**	**Robustness** ($\alpha = 0.1\%$)		**Policy Action**	**Routing**
Router Policy	**Assignment**	$\alpha = 1.0\%$	$\alpha = 0.1\%$	PPL ≤ 18.25	DIAA	DIPPER	Compliance	Failures (/500)
`gpt-5-nano` (low reasoning)	70.6	87.5	84.5	91.0	10.2	16.4	75.0	36
`gpt-5.4-nano` (high reasoning)	**100.0**	**98.0**	**97.5**	**98.0**	**13.8**	**17.0**	**100.0**	**0**

Impact of Deployment Monitor. To separate runtime rule enforcement from the quality of the selected watermarking configurations, we compare a lightweight monitor (`gpt-5-nano`, low reasoning) against a stronger monitor (`gpt-5.4-nano`, high reasoning) on all 500 prompts, both executing the Constitutional-KGW policy with frozen Qwen3-32B generation (Table 5). The stronger monitor assigns the correct rule on every request (100.0%) with no routing failures, whereas `gpt-5-nano` reaches only 70.6% and fails on 36 requests (7.2%), all due to invalid policy decisions rather than infrastructure errors. This deficit propagates downstream: clean detection falls from 98.0% to 87.5% at $\alpha = 1.0\%$ and from 97.5% to 84.5% at $\alpha = 0.1\%$, the quality pass rate on quality-prioritized requests drops from 98.0% to 91.0%, and policy-action compliance falls from 100.0% to 75.0%, driven mainly by editing requests that should have been exempt from watermarking. Robustness is similar (13.8% vs. 10.2% under DIAA; 17.0% vs. 16.4% under DIPPER), indicating that paraphrase vulnerability is a property of the watermark itself rather than of routing. Because outputs were regenerated for each arm, these differences are a deployment-level diagnostic rather than a counterfactual audit of identical generations; nonetheless, they show that constitutional execution depends as much on reliable policy interpretation as on configuration optimization.

6 Discussion and Conclusion

Key Takeaways. ***(i.)*** The framework's advantage comes from assigning different operating points to different rules: robustness gains appear only where required, while other requests keep low-distortion configurations and editing remains unwatermarked, a pattern no fixed configuration can reproduce. ***(ii.)*** No evaluated method meets the 50% robustness requirement. This establishes a limitation of the tested policies and search budget, not a robustness ceiling for KGW. The framework reports this as a best-effort policy with explicit unmet gates. ***(iii.)*** Policy execution depends on the monitor as much as on configuration selection, since a weaker monitor degrades every metric even with unchanged configurations. ***(iv.)*** Priority wording shifts selected configurations in the intended direction, but adjacent priorities remain statistically indistinguishable.

Limitations. ***(i.)*** Our evaluation uses one watermark family, serving model, key pair, and search seed, with one five-rule constitution and a limited priority-sensitivity experiment. ***(ii.)*** The watermark policies for each rule are selected on small reference sets that are reused during search, so results are finite-sample evidence rather than guarantees, but an unsuccessful search **does not** prove infeasibility. ***(iii.)*** Detection assumes access to the generation configuration, and we did not evaluate prompt-free detector selection. Perplexity does not establish task correctness or low distortion, and attacked outputs are not independently audited for semantic preservation. However, our experiments show the potential of our proposed framework. We also do not measure forgery resistance, but our framework is compatible with randomized-key defenses (Aremu et al., 2025) applied within each rule (see Appendix D.1). ***(iv.)*** While we made comparisons with fixed baselines and random search, comparisons with Bayesian optimization and simpler hand-designed policies can also isolate the contribution of the reasoning agent. Appendix E.2 details these limitations.

Conclusion. We introduced constitution-guided watermarking, which translates provider-written natural-language rules into request-dependent watermark policies through offline, constitution-guided search and deployment routing, while keeping the serving model frozen. The KGW demonstration supports the feasibility of this workflow, improving paraphrase robustness on robustness-prioritized requests, matching the best clean detection and quality among the evaluated baselines, and exempting editing requests as specified. Our framework also reports gates it is unable to reach within the given search budget. While these results demonstrate the potential of our framework, future work includes task-specific quality measurements, independent calibration audits, and evaluation across watermark families and constitutions.

Ethics Statement

Watermark detection is statistical evidence, not conclusive proof of authorship or misconduct. Deployment must account for mixed human–AI authorship, false attribution, prompt privacy, and key security. Evaluating misuse-prone requests does not justify bypassing the serving model's safety constraints; provenance control is not a substitute for content safety.

AI Use Statement

In this work, we used generative AI tools to generate the synthetic, constitution-labeled prompt datasets (Section 3.1; GPT-OSS-120B and Qwen3-32B) and to implement parts of the experimental code. We have not used generative AI tools for formulating mathematical claims or writing proofs. Additionally, we used generative AI tools to draft and edit parts of the manuscript. We have reviewed all AI-assisted work; for example, generated prompts were filtered by an independent critic and spot-checked by the authors; AI-assisted code was tested and verified by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

Reproducibility Statement

Appendix B gives all decoding, budget, attack, and baseline settings, and Appendix B.4 reproduces the evaluated constitution verbatim. Table 6 lists every selected configuration, and Table 7 every deployed detector with its calibrated thresholds. Section 3.1 describes how the constitution-labeled prompts were generated and split. Code, prompts, and run artifacts, including source hashes, agent proposals, rejections, and retained incumbents, will be released upon publication.

References

Scott Aaronson. Watermarking of large language models. https://www.youtube.com/watch?v=2Kx9jbSMZqA, 2023. Simons Institute, YouTube video.

Anastasios N. Angelopoulos and Stephen Bates. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *arXiv preprint arXiv:2107.07511*, 2021. URL https://arxiv.org/abs/2107.07511.

Anthropic. How Claude's text watermark works. Anthropic News, August 2026. URL https://www.anthropic.com/news/claude-text-watermark.

Toluwani Aremu, Noor Hussein, Munachiso Nwadike, Samuele Poppi, Jie Zhang, Karthik Nandakumar, Neil Gong, and Nils Lukas. Mitigating watermark forgery in generative models via randomized key selection. *arXiv preprint arXiv:2507.07871*, 2025. URL https://arxiv.org/abs/2507.07871.

Toluwani Aremu, Nils Lukas, and Jie Zhang. Watermarking should be treated as a monitoring primitive. *arXiv preprint arXiv:2605.13095*, 2026. URL https://arxiv.org/abs/2605.13095.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022.

James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(10):281–305, 2012. URL https://jmlr.org/papers/v13/bergstra12a.html.

Yixin Cheng, Hongcheng Guo, Yangming Li, and Leonid Sigal. Revealing weaknesses in text watermarking through self-information rewrite attacks. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 9982–10009. PMLR, 2025. URL https://proceedings.mlr.press/v267/cheng25c.html.

Miranda Christ, Sam Gunn, and Or Zamir. Undetectable watermarks for language models. In *The Thirty Seventh Annual Conference on Learning Theory*, pp. 1125–1139. PMLR, 2024.

C. J. Clopper and E. S. Pearson. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26(4):404–413, 1934. doi: 10.1093/biomet/26.4.404.

Sumanth Dathathri, Abigail See, Sumedh Ghaisas, Po-Sen Huang, Rob McAdam, Johannes Welbl, Vandana Bachani, Alex Kaskasoli, Robert Stanforth, Tatiana Matejovicova, Jamie Hayes, Nidhi Vyas, Majd Al Merey, Jonah Brown-Cohen, Rudy Bunel, Borja Balle, A. Taylan Cemgil, Zahra Ahmed, Kitty Stacpoole, Ilia Shumailov, Ciprian Baetu, Sven Gowal, Demis Hassabis, and Pushmeet Kohli. Scalable watermarking for identifying large language model outputs. *Nat.*, 634(8035): 818–823, October 2024. URL https://doi.org/10.1038/s41586-024-08025-4.

Abdulrahman Diaa, Toluwani Aremu, and Nils Lukas. Optimizing adaptive attacks against watermarks for language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 13546–13569. PMLR, 2025. URL https://proceedings.mlr.press/v267/diaa25a.html.

EU AI Act. Artificial intelligence act. Official Journal of the European Union, July 2024. URL http://data.europa.eu/eli/reg/2024/1689/oj. Adopted 13 June 2024; OJ L, 12 July 2024.

Thibaud Gloaguen, Nikola Jovanović, Robin Staab, and Martin Vechev. Discovering clues of spoofed lm watermarks. *arXiv preprint arXiv:2410.02693*, 2024.

Donald R. Jones, Matthias Schonlau, and William J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13:455–492, 1998. doi: 10.1023/A:1008306431147.

Nikola Jovanović, Robin Staab, and Martin Vechev. Watermark stealing in large language models. *ICML*, 2024.

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

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

Rohith Kuditipudi, John Thickstun, Tatsunori Hashimoto, and Percy Liang. Robust distortion-free watermarks for language models. *Trans. Mach. Learn. Res.*, 2024, 2023. URL https://api.semanticscholar.org/CorpusID:260315804.

Yepeng Liu and Yuheng Bu. Adaptive text watermark for large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 30718–30737. PMLR, 2024. URL https://proceedings.mlr.press/v235/liu24e.html.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2303.17651.

Leyi Pan, Aiwei Liu, Zhiwei He, Zitian Gao, Xuandong Zhao, Yijian Lu, Binglin Zhou, Shuliang Liu, Xuming Hu, Lijie Wen, et al. Markllm: An open-source toolkit for llm watermarking. *arXiv preprint arXiv:2405.10051*, 2024.

Qi Pang, Shengyuan Hu, Wenting Zheng, and Virginia Smith. No free lunch in llm watermarking: Trade-offs in watermarking design choices. In *Neural Information Processing Systems*, 2024. URL https://api.semanticscholar.org/CorpusID:267938448.

Julien Piet, Chawin Sitawarin, Vivian Fang, Norman Mu, and David Wagner. Mark my words: Analyzing and evaluating language model watermarks. *ArXiv*, abs/2312.00273, 2023. URL https://api.semanticscholar.org/CorpusID:265552122.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2303.11366.

Chenrui Wang, Junyi Shu, Billy Chiu, Yu Li, Saleh Alharbi, Min Zhang, and Jing Li. Learning to watermark: A selective watermarking framework for large language models via multi-objective optimization. In *Advances in Neural Information Processing Systems*, 2025a. URL https://openreview.net/forum?id=nJq5z21eUk.

Zongqi Wang, Tianle Gu, Baoyuan Wu, and Yujiu Yang. MorphMark: Flexible adaptive watermarking for large language models. *arXiv preprint arXiv:2505.11541*, 2025b. URL https://arxiv.org/abs/2505.11541.

Bram Wouters. Optimizing watermarks for large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 53251–53269. PMLR, 2024. URL https://proceedings.mlr.press/v235/wouters24a.html.

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*, 2024a. URL https://openreview.net/forum?id=SsmT8aO45L.

Xuandong Zhao, Sam Gunn, Miranda Christ, Jaiden Fairoze, Andres Fabrega, Nicholas Carlini, Sanjam Garg, Sanghyun Hong, Milad Nasr, Florian Tramer, et al. Sok: Watermarking for ai-generated content. *arXiv preprint arXiv:2411.18479*, 2024b.

A Related Work

Statistical text watermarking. Token-level watermarks alter or select tokens during decoding and detect the resulting statistical structure with a secret key. KGW partitions the vocabulary into context-dependent green and red lists and biases green-token logits (Kirchenbauer et al., 2023); Unigram uses a fixed keyed partition to improve robustness (Zhao et al., 2024a); EXP applies exponential sampling with keyed randomness (Aaronson, 2023); and SynthID aggregates tournament-based sampling scores (Dathathri et al., 2024). Distribution-preserving and cryptographic constructions provide stronger undetectability notions under explicit assumptions (Christ et al., 2024; Kuditipudi et al., 2023). Toolkits such as MarkLLM standardize implementations and evaluation, but do not determine which operating point a provider should deploy for a particular request (Pan et al., 2024).

Watermark trade-offs and attacks. Watermarking has no universally optimal configuration i.e., detectability, quality, robustness, and security depend on the watermark mechanism, model, decoding policy, and output distribution (Pang et al., 2024; Wouters, 2024; Piet et al., 2023). Paraphrasing and rewriting can weaken token-level signals, particularly when the attacker adapts to the known watermark algorithm (Krishna et al., 2023; Diaa et al., 2025; Cheng et al., 2025). Other attacks estimate the watermark or induce false attribution (Jovanović et al., 2024; Gloaguen et al., 2024). Randomized key selection can bound forgery under a stated multi-key threat model, while persistent key-dependent structure can itself enable monitoring across outputs (Aremu et al., 2025; 2026). We treat these properties as conditional requirements rather than assuming that every request requires the same security-utility trade-off.

Adaptive and selective watermarking. Prior adaptive methods change where or how a watermark is embedded using token entropy, semantics, or learned generation features (Liu & Bu, 2024; Wang et al., 2025b;a). These methods adapt within an output to improve a fixed design objective. Our problem is complementary and dynamic i.e., providers give their rules in natural language, and the system selects among algorithms and configurations using rule-specific empirical evidence. Thus, the adaptation target is the provider's deployment policy, not only the next-token distribution.

Configuration optimization. Watermark-specific optimization studies formalize trade-offs between detection and generation quality (Wouters, 2024). Recommended fixed settings are useful operating points, not constitution-specific optima. Random search (Bergstra & Bengio, 2012) and Bayesian optimization (Jones et al., 1998) provide additional baselines for testing whether source-grounded reasoning improves decisions under the same empirical evaluation budget.

Constitutions and feedback-driven agents. Constitutional AI uses written principles to supervise model behavior, while feedback-driven agents refine decisions over repeated trials without necessarily updating model weights (Bai et al., 2022; Shinn et al., 2023; Madaan et al., 2023). We use these ideas at a different system boundary by using the constitution to supervise watermark configuration selection and rule recognition, without altering the serving model.

B Experimental Details

B.1 Implementation

The main run uses seed 20260908, paired per-prompt seeds, and answer-only KGW: watermarking and detection act on the visible answer, not on reasoning tokens. Reasoning uses temperature 0.6, top-$p = 0.95$, and top-$k = 20$. The verification agent may set only the fields exposed by MarkLLM's KGW configuration (`KGW.json`); the provider key, left-context window policy, and calibrated detector operating points are held fixed. The verification agent has a $32{,}768$-token completion budget and the monitor a $4{,}096$-token budget. Model identifiers are reported exactly as served by the API. Source hashes, exact requests, proposal rejections, retained incumbents, and decoding settings are retained in the run artifacts.

B.2 Attack Settings

The adaptive paraphraser uses the `DDiaa/WM-Removal-KGW-Llama-3.1-8B` adapter on Llama-3.1-8B-Instruct, temperature 1, and at most $1{,}024$ new tokens. DIPPER-XXL uses lexical diversity 60, order diversity 0, sentence interval 1, top-$p = 0.75$, and at most 100 new tokens per generation call. Both use bfloat16 and five repetitions per attacked answer. Because generation caps can truncate paraphrases, post-attack detection measures raw signal survival; we do not audit whether attacked outputs preserve the answer's meaning (Appendix E.2).

B.3 Baselines

Fixed configurations. Fixed KGW A and B use $(\gamma, \delta, h, f) = (0.25, 4, 3, \texttt{time})$ and $(0.5, 2, 1, \texttt{time})$, prespecified operating points from Piet et al. (2023) and Pan et al. (2024). Both watermark every request, including editing.

Budget-matched random search. Random search (Bergstra & Bengio, 2012) receives the same empirical budget as the verification agent: at most five legal candidates per watermarked rule, each evaluated on the same 10 reference prompts per rule against the same active gates. It samples $\gamma \sim \mathrm{Beta}(2, 2)$, $\delta \sim \mathrm{Geom}(0.25)$, and $h \sim \mathrm{Geom}(0.5)$ on the positive integers, and f uniformly from its legal values; this is an explicit search prior, not a bounded grid. Lacking natural-language judgment, random search ranks candidates by passing all active gates first, then by weighted normalized gate shortfall, with attainment breaking ties. The weights favor robustness for misuse, quality for advice and deterministic answers, and equal priorities for general generation. The agent instead retains its incumbent by natural-language judgment under each rule's stated priorities. Test results never select a prior, weight, candidate, or incumbent.

At deployment, random search and Constitutional-KGW use the same deployment monitor (`gpt-5.4-nano`, high reasoning) and exempt editing as the constitution specifies, so the two arms differ only in the per-rule configurations they store (Table 6). Answers are regenerated for each arm on the same 500 test prompts. The arms share an empirical evaluation budget, not a compute budget, and use one search seed.

B.4 PROVIDER CONSTITUTION

The evaluated constitution contains the following natural-language rules, reproduced verbatim in precedence order. Precedence is first-applicable: when several rules apply to a prompt, the monitor selects the first in the order below, and only that rule's action and stated priorities are enforced. Our experiments therefore do not exercise the objective recurrence of Equation 4, which applies to providers that combine compatible rules. Prompts that trigger several rules instead test whether the monitor applies precedence correctly. The sensitivity constitutions (Appendix C.3) each contain a single rule, so overlap resolution has no effect there.

SYSTEM POLICY RULES (first-applicable)

Misuse.	For misuse-prone, malicious, or abusive requests, watermark the response and prioritize robustness over quality.
Editing.	Do not watermark grammar correction, proofreading, or low-risk text editing.
Advice.	For medical and financial advice, watermark the response and prioritize quality and low distortion while keeping detectability high.
Deterministic.	For deterministic answers, watermark the response and prioritize quality and detectability.
General.	Watermark all other responses while minimizing unnecessary distortion.

C ADDITIONAL RESULTS

C.1 SELECTED PER-RULE CONFIGURATIONS

Table 6 lists the configuration each search procedure stored for every rule. The agent assigns its strongest bias and a longer context only to the robustness-prioritized misuse rule, and uses a single configuration, $(0.5, 4, 1, \texttt{additive})$, for all three quality-prioritized rules. The agent thus separates robustness-prioritized from quality-prioritized requests rather than tuning each quality rule individually. Random search stores legal configurations that are not motivated by the rules' priorities, such as $\delta = 27$ for deterministic answers, yet matches the agent's aggregate clean detection (Table 2); the agent's advantage lies mainly in paraphrase robustness on the misuse rule.

Table 6: **Configurations stored per rule.** Tuples are (γ, δ, h, f): green-list fraction, logit bias, context length, and seeding function. Rules are listed in precedence order. Agent configurations for advice, deterministic answers, and general generation are REFERENCE-PASSED; misuse is BEST-EFFORT (Table 3). Random-search γ is rounded for display only.

Rule (priority)	C-KGW agent	Random search
Misuse (robustness over quality)	$(0.5, 6, 4, \texttt{min})$	$(0.4109, 4, 2, \texttt{min})$
Editing (exempt)	NOWM	NOWM
Advice (quality, low distortion)	$(0.5, 4, 1, \texttt{additive})$	$(0.5603, 5, 6, \texttt{time})$
Deterministic (quality, detection)	$(0.5, 4, 1, \texttt{additive})$	$(0.6056, 27, 1, \texttt{skip})$
General (minimize distortion)	$(0.5, 4, 1, \texttt{additive})$	$(0.5763, 6, 6, \texttt{min})$

C.2 NULL-REFERENCE FALSE-POSITIVE RATES

We evaluate each deployed detector on 5,000 unwatermarked texts held out from threshold calibration (Table 7). Observed FPRs range from 0.88% to 1.14% at nominal 1% and from 0.08% to 0.18% at nominal 0.1%. Several detectors slightly exceed their nominal targets, although every 95% confidence interval includes the corresponding nominal rate. Because these texts also informed configuration selection, we interpret the intervals descriptively; an independent assessment of the selected policies would require a fresh unwatermarked evaluation set.

Table 7: **Measured null-reference FPRs, not an independent final audit.** Each row tests 5,000 unwatermarked texts disjoint from threshold calibration; their feedback was reused during configuration selection. η_α is the calibrated detection threshold at nominal FPR α. Brackets give two-sided 95% Clopper–Pearson intervals; these are descriptive, not guarantees after adaptive selection. The provider's looser 1% cap is a separate requirement. Detector tuples are (γ, h, f) with fixed key and tokenizer; generation bias δ does not affect the score. Random-search γ is rounded for display only. The agent's three non-misuse rules share one detector, which coincides with Fixed KGW B's because for $h = 1$ every seeding function reduces to the preceding token. Rates are percentages.

		Nominal $\alpha = 1.0\%$		**Nominal** $\alpha = 0.1\%$	
Detector / Use	(γ, h, f)	η_α	Observed FPR	η_α	Observed FPR
Fixed KGW A	$(0.25, 3, \texttt{time})$	2.531	0.940 $_{[0.691,\ 1.248]}$	3.408	0.080 $_{[0.022,\ 0.205]}$
Fixed KGW B	$(0.5, 1, \texttt{time})$	2.776	1.060 $_{[0.795,\ 1.384]}$	3.842	0.160 $_{[0.069,\ 0.315]}$
Agent: non-misuse	$(0.5, 1, \texttt{additive})$	2.776	1.060 $_{[0.795,\ 1.384]}$	3.842	0.160 $_{[0.069,\ 0.315]}$
Agent: misuse	$(0.5, 4, \texttt{min})$	2.371	1.040 $_{[0.778,\ 1.362]}$	3.208	0.140 $_{[0.056,\ 0.288]}$
Random: deterministic	$(0.6056, 1, \texttt{skip})$	2.967	1.000 $_{[0.743,\ 1.316]}$	4.111	0.140 $_{[0.056,\ 0.288]}$
Random: general	$(0.5763, 6, \texttt{min})$	2.529	1.020 $_{[0.760,\ 1.339]}$	3.475	0.080 $_{[0.022,\ 0.205]}$
Random: advice	$(0.5603, 6, \texttt{time})$	2.402	0.880 $_{[0.640,\ 1.180]}$	2.986	0.180 $_{[0.082,\ 0.341]}$
Random: misuse	$(0.4109, 2, \texttt{min})$	2.383	1.140 $_{[0.865,\ 1.475]}$	3.448	0.080 $_{[0.022,\ 0.205]}$

C.3 Constitution-Sensitivity Setup

The sensitivity study (Table 4) reuses one general-generation slice: the same 10 reference and 100 test prompts, serving model, KGW implementation, attack channels, seeds, and detectability, quality, and robustness gates. Only the natural-language priority instruction changes:

Sensitivity Policy Rules

Quality first. Watermark these general-generation responses. Require detectability, quality, and robustness; prioritize quality over robustness.

Robustness first. Watermark these general-generation responses. Require detectability, quality, and robustness; prioritize robustness over quality.

Balanced. Watermark these general-generation responses. Require detectability, quality, and robustness; balance these three objectives equally.

Each condition runs the same agent for at most five reference rounds and retains a passing or best-effort policy.

Priority-following. All three variants retained $\gamma = 0.5$ and differed mainly in bias strength. This suggests that the agent navigates a narrow region of KGW's configuration space rather than exploiting its full flexibility, which may explain why adjacent priorities are not reliably distinguished. Parameter changes alone do not establish constitutional control; they must be accompanied by the intended shift in measured outcomes, as Table 4 reports.

C.4 Robustness at Nominal 1% FPR and Paired Comparison

Table 8 repeats the robustness comparison of Table 2 at nominal $\alpha = 1\%$. The ranking is unchanged: Constitutional-KGW attains the highest post-paraphrase detection under both attacks (27.8% DIAA, 30.8% DIPPER), ahead of random search and the fixed configurations, with gains of up to 19.2 percentage points over Fixed KGW A. Even at this looser operating point, robustness remains below the provider's 50% requirement, which Equation 8 defines at the strictest regime, $\alpha_* = 0.1\%$.

To test the difference between Constitutional-KGW and random search directly, Table 9 reports paired differences on the same prompts. Every difference favors Constitutional-KGW, but only the DIPPER gain at nominal 0.1% FPR (+5.4 percentage points) excludes zero; the remaining differences lie within sampling uncertainty at this sample size.

Table 8: **Robustness at nominal $\alpha = 1\%$.** Post-paraphrase detection on the 100 robustness-required prompts (5 attacks/response), under the same generations, attacks, and routing as Table 2. Brackets show 95% percentile-bootstrap intervals, clustered by prompt. All values are percentages; higher is better. Best results are **bolded**.

	Robustness ($\alpha = 1.0\%$)	
Method	DIAA	DIPPER
Fixed KGW A	8.6 [5.6, 12.2]	11.8 [8.6, 15.4]
Fixed KGW B	8.8 [5.8, 12.0]	19.0 [14.8, 23.4]
Random search	26.6 [20.4, 33.0]	28.2 [23.0, 33.6]
C-KGW (Ours)	**27.8** [21.8, 34.4]	**30.8** [25.6, 36.2]

Table 9: **Paired robustness differences, Constitutional-KGW minus random search.** Both methods are evaluated on the same 100 robustness-required prompts (5 attacks/response). Brackets show 95% paired percentile-bootstrap intervals over source prompts (10,000 replicates), keeping each prompt's attack repetitions together; attack failures count as misses. Rates are percentages; differences are in percentage points.

Nominal FPR	**Attack**	**C-KGW**	**Random search**	**Difference**
0.1%	DIAA	13.8	12.2	+1.6 [−2.6, 3.8]
	DIPPER	17.0	11.6	**+5.4** [0.6, 10.2]
1.0%	DIAA	27.8	26.6	+1.2 [−7.8, 5.2]
	DIPPER	30.8	28.2	+2.6 [−3.8, 9.2]

C.5 Qualitative Examples

We show one verification-agent trace and three deployment-monitor decisions from the run artifacts. Agent and monitor rationales are quoted verbatim; configurations are written in the paper's notation, and rules by the names used in Appendix B.4. We trace the misuse rule because it is the only Best-Effort rule, and chose the routing examples to contrast editing and advice requests on the same subject.

Configuration refinement. Figure 3 follows the misuse rule from its first proposal to its exported record (cf. Table 3). The first proposal grounds its choices in the KGW implementation, selecting a four-token `min` aggregation for robustness, while stating that this rationale is a hypothesis requiring attack validation. After the reference set rejected it on detection and robustness, the agent retained the `min`/long-context structure and raised the bias. The final configuration passes detection and quality but reaches only 10% robustness, so the agent exports it as Best-Effort and states that no measured candidate satisfies the robustness gate, rather than reporting the rule as passed.

Deployment routing. Figure 4 shows three decisions by the lightweight `gpt-5-nano` monitor (Table 5). The monitor correctly distinguished editing a medical disclaimer from requesting medical advice: it assigned the former to the editing exemption and selected watermarking for the latter, showing how the requested operation, rather than the subject area, determines the policy. For a grammar-correction request annotated with both the editing and deterministic-answer rules, however, it identified only the deterministic-answer rule and selected watermarking. Under first-applicable precedence the editing exemption should have prevailed; omitting an applicable rule thus led the monitor to apply the wrong policy, the failure mode behind this monitor's lower compliance in Table 5.

C.6 Routing Robustness

Exemption evasion. A misuse-prone user could frame a request as editing to obtain the editing exemption and an unwatermarked response. We probe this with the `gpt-5.4-nano` monitor under the evaluated constitution, where first-applicable precedence places misuse before editing, so the

VERIFICATION-AGENT TRACE (misuse rule)

RULE	Instruction	*"For misuse-prone, malicious, or abusive requests, watermark the response and prioritize robustness over quality."*
ROUND 1	Proposal	$(\gamma, \delta, h, f) = (0.5,\ 3,\ 4,\ \texttt{min})$
	Rationale	*"Robustness is the rule's top objective, so a stronger bias and a four-token minimum aggregation are selected. The minimum f-scheme aggregation can preserve the selected minimum when some recent-token inputs change, but this robustness rationale is a hypothesis requiring attack validation."*
	Result	Detection **0.90** ✗ • Quality **1.00** ✓ • Robustness **0.00** ✗
		intermediate rounds omitted
FINAL	Proposal	$(\gamma, \delta, h, f) = (0.5,\ \mathbf{6},\ 4,\ \texttt{min})$ δ raised; γ, h, f retained
	Rationale	*"This is the specified unevaluated next configuration and follows the rule's robustness-first priority. It retains the best measured min/prefix structure and increases direct bias by one step as a robustness hypothesis, while all quality, detection, FPR, and robustness gates remain mandatory."*
	Result	Detection **1.00** ✓ • Quality **1.00** ✓ • Robustness **0.10** ✗
EXPORT	Status	**BEST-EFFORT**
	Agent review	*"Strongest robustness-first incumbent: it has the highest measured robustness rate, 0.10 (5/50), while detection and quality both passed and detector-FPR regimes passed. Its higher PPL is secondary because robustness has priority. No measured candidate satisfies the robustness gate."*

Figure 3: **Configuration refinement for the misuse rule.** Results are measured on the 10-prompt reference set at nominal $\alpha = 0.1\%$ against the gates detection ≥ 0.95, quality ≥ 0.90, and robustness ≥ 0.50; ✓/✗ mark whether each is met. Robustness counts detections over 50 attacked responses (10 prompts $\times$ 5 attacks), taking the worse of DIAA and DIPPER.

correct route for such requests is misuse with watermarking. The probes comprise 8 accepted corpus prompts that combine misuse and editing and 42 additional probes generated with an LLM from the same template; the generated probes were not passed through the critic but were manually checked by the authors. Controls are 50 distinct editing-only corpus prompts. Table 10 shows that the monitor routed every probe to the misuse rule and exempted 49 of 50 controls; the single control error watermarked an editing request, the conservative direction. Because generated probes may carry more explicit misuse cues than real evasion attempts, we report the subsets separately and treat this as a pilot; adaptive evasion remains future work.

Table 10: **Exemption-evasion pilot** (`gpt-5.4-nano`, high reasoning). Under first-applicable precedence, misuse precedes editing, so misuse requests framed as editing should be routed to the misuse rule and watermarked. Brackets give two-sided 95% Clopper–Pearson intervals. No routing failures occurred.

Probe set	**Expected route**	n	**Correct route (%)**
Misuse framed as editing: corpus prompts	Misuse → WM	8	100.0 [63.1, 100.0]
Misuse framed as editing: generated probes	Misuse → WM	42	100.0 [91.6, 100.0]
Misuse framed as editing: all	Misuse → WM	50	**100.0** [92.9, 100.0]
Editing-only controls (corpus)	Editing → NoWM	50	98.0 [89.4, 99.9]

Overlapping rules. Table 11 separates the 97 test prompts annotated with more than one applicable rule from single-rule prompts. For both monitors, accuracy on overlapping prompts is close to

DEPLOYMENT-MONITOR DECISIONS (`gpt-5-nano`, low reasoning)

EXAMPLE 1 Editing a medical text — ✓ Correct

Prompt	"Rewrite the following medical disclaimer to improve readability: 'This medication should only be taken as prescribed by a qualified healthcare professional.' "
Expected rule	EDITING → NoWM
Monitor decision	EDITING → NoWM
Rationale	*"The request is a low-risk proofreading/editing task; policy specifies no watermarking for low-risk edits."*

EXAMPLE 2 Requesting medical advice — ✓ Correct

Prompt	"I have a persistent cough and shortness of breath; what should I do?"
Expected rule	ADVICE → WM
Monitor decision	ADVICE → WM
Rationale	*"Medical advice with high-stakes implications; choose watermarking and prioritize quality/low distortion per policy."*

EXAMPLE 3 Overlapping rules — ✗ Misrouted

Prompt	"Can you correct the grammar in this sentence: 'She dont like the movie'?"
Annotated rules	EDITING DETERMINISTIC
Expected rule	EDITING → NoWM first-applicable precedence
Monitor decision	DETERMINISTIC → WM editing rule not identified
Rationale	*"Deterministic grammar correction task; watermark applied per KGW with priority on quality and detectability."*

Figure 4: **Deployment routing by the lightweight monitor.** Examples 1 and 2 share a medical subject but differ in the requested operation; Example 3 shows an omitted applicable rule. Prompts and rationales are verbatim; rule identifiers are mapped to the names used in Appendix B.4.

Table 11: **Routing on overlapping and single-rule test prompts.** Overlapping prompts are annotated with more than one applicable rule. `gpt-5-nano` results come from the end-to-end run in Table 5; `gpt-5.4-nano` results come from a separate routing-only rerun with the same instruction, frozen policy, and enforcer. Decision accuracy is the resulting WM/NoWM decision over all prompts, distinct from the editing-only compliance in Table 5. Routing failures count as incorrect; brackets give two-sided 95% Clopper–Pearson intervals for rule assignment.

Monitor	Prompts	n	Correct rule (%)	Correct decision (%)	Failures
`gpt-5-nano` (low)	Overlapping	97	72.2 $_{[62.1, 80.8]}$	93.8	4
	Single-rule	403	70.2 $_{[65.5, 74.6]}$	88.6	32
	All	500	70.6 $_{[66.4, 74.6]}$	89.6	36
`gpt-5.4-nano` (high)	Overlapping	97	95.9 $_{[89.8, 98.9]}$	97.9	0
	Single-rule	403	97.3 $_{[95.2, 98.6]}$	99.0	0
	All	500	97.0 $_{[95.1, 98.3]}$	98.8	0

single-rule accuracy, so the weaker monitor's errors are not concentrated on overlaps. In the separate routing-only rerun, `gpt-5.4-nano` assigned 97.0% of rules correctly, compared with 100.0% in the end-to-end run of Table 5, indicating run-to-run variation in LLM routing that providers should monitor in deployment.

D FRAMEWORK EXTENSIONS

This appendix describes three extensions that the framework supports but our experiments do not evaluate: a randomized-key forgery defense, specialist training of both agents, and detection without the original prompt.

D.1 CONDITIONAL FORGERY RESISTANCE

We describe an optional composition with the randomized-key defense of Aremu et al. (2025), not a new security theorem or an evaluated component of Constitutional-KGW. The argument applies to compatible statistical watermarks under the assumptions below.

Profile-conditioned construction. Fix a watermarked profile c, its policy version, and the selected algorithm and configuration (a_c, ψ_c). The provider holds $K \geq 2$ secret, compatible generation–detection key pairs $\{(\mathsf{gk}_{c,j}, \mathsf{dtk}_{c,j})\}_{j=1}^{K}$. For every request assigned to c, independently sample $J \sim \mathrm{Uniform}(\{1, \ldots, K\})$ and generate

$$\mathsf{Watermark}^{\mathcal{M}}_{\mathsf{gk}_{c,J}}(\pi; a_c, \psi_c) \to x.$$

Only one key is embedded in each response. Detection tests all K keys:

$$\begin{aligned} Z_{c,j}(x) &= \mathbf{1}\left[s_{a_c,\psi_c,\mathsf{dtk}_{c,j}}(x) \geq \eta_{c,j}\right], \\ S_c(x) &= \sum_{j=1}^{K} Z_{c,j}(x), \qquad \mathsf{Acc}_c(x) = \mathbf{1}[S_c(x) = 1]. \end{aligned} \tag{11}$$

Zero or multiple detections are rejected; multiple detections alone are not proof of forgery. Exempt profiles remain unwatermarked and receive no watermark-based provenance claim.

Threat model and assumptions. The attacker knows the algorithm, configuration, and defense, may adaptively collect N provider responses, but knows neither the secret keys nor their assignments and has no detection-oracle access. Target content $\mathcal{H}_c$ lies outside legitimate provider output; the cited work uses harmful content that the provider refuses to generate. Replaying a legitimate oracle response is not a successful forgery. The following conditions delimit the claim.

Fixed detection context. The profile and policy version are known through trusted metadata, or fixed for the security analysis. Their algorithm, detector parameters, key set, and thresholds are bound before testing the candidate text.

Key blindness. The attacker cannot reliably distinguish or partition collected responses by their generating key, including through side information. Withholding key labels alone does not establish this condition.

Adversarial-output symmetry and independence. For each attacker covered by the claim, its output $X^\star$ induces mutually independent indicators $Z_{c,j}(X^\star) \sim \mathrm{Bernoulli}(p)$ with the same marginal $p \in [0, 1]$ across keys. Here p is attacker-induced, not the null FPR. Key blindness does not imply this joint distribution, and independence on benign null text does not establish independence on adversarial outputs.

Conditional bound. Under these assumptions, the result of Aremu et al. (2025) yields

$$\begin{aligned} \mathrm{Adv}_c^{\mathrm{forge}}(\mathcal{B}, N) &\leq \Pr[S_c(X^\star) = 1] \\ &= Kp(1-p)^{K-1} \leq \left(1 - \frac{1}{K}\right)^{K-1}. \end{aligned} \tag{12}$$

Proof. The independent, equal-marginal indicators give $S_c(X^\star) \sim \mathrm{Binomial}(K, p)$. For $0 < p < 1$, the derivative of $Kp(1-p)^{K-1}$ is $K(1-p)^{K-2}(1-Kp)$, so its maximum occurs at $p = 1/K$; both endpoints give zero. Requiring $X^\star \in \mathcal{H}_c$ can only decrease acceptance probability. □

The bound does not depend on the number of collected samples N, provided the assumptions continue to hold. It is per forgery attempt, not a bound on success over unlimited attempts. Its value is $1/2$ for $K = 2$ and tends to $1/e$ as K grows: it is not negligible, does not scale as $1/K$, and does not prove improvement over every single-key attack success rate. It instead excludes near-certain success within

the stated attack model. The experiments in the cited work are evidence for that work's evaluated settings, not for our constitutional policies.

False-positive calibration. Let α_{fw} be the provider's overall null error budget and α_{key} a per-key level. For independent null indicators with this common level, the Šidák choice used by the defense is

$$\alpha_{\mathrm{key}} = 1 - (1 - \alpha_{\mathrm{fw}})^{1/K}, \qquad \Pr_0[S_c \geq 1] = \alpha_{\mathrm{fw}}. \tag{13}$$

The exactly-one acceptance FPR is at most this family-wise rate. Without null independence, valid per-key levels at most α_{fw}/K instead give $\Pr_0[\mathsf{Acc}_c = 1] \leq \Pr_0[S_c \geq 1] \leq \alpha_{\mathrm{fw}}$ by the union bound. These statements assume valid null levels; estimated thresholds still require independent auditing and uncertainty reporting. Neither calibration establishes the adversarial-output assumptions above. Testing multiple profiles or choosing one from the candidate text requires calibrating the complete selection-and-detection procedure, the fixed-profile theorem cannot simply be transferred to an adaptive selector.

Constitutional scope and limitations. A provider may request this defense for a forgery-sensitive rule, with its assumptions recorded alongside the policy. Randomization must occur within that profile; predictable selection between rules does not supply key blindness. Mixing algorithms or configurations, including different algorithms with the same key, does not automatically preserve equal attack marginals or independence and requires a separate argument. Likewise, distortion-freeness or public undetectability alone does not establish the full set of assumptions. Although each response embeds only one watermark, stricter thresholds and rejection of multiple detections can reduce genuine acceptance, and testing K keys increases detection cost.

Our experiments keep one key pair fixed and do not evaluate this wrapper, its assumptions, or forgery attacks. We therefore record this extension as *conditional theoretical support*, not an empirically passed gate or a security certificate for the deployed configurations.

D.2 Specialist Training Objectives

Hypothesis H6 (Appendix E.1) posits that training both agent roles for their tasks would improve performance. We give objectives for this training; our experiments instantiate both roles with pretrained models and do not optimize these objectives. Verifier supervision spans algorithm implementations, configurations, proof statements, checked derivations, and empirical failure cases across watermark families. An example couples $v = (\mathcal{C}, S, c, H_c)$ with a target z^*: a legal algorithm and configuration, source-supported claims, explicit assumptions, and predicted trade-offs. Here S is the implementation bundle and H_c is the profile's measurement history. The supervised objective is

$$\mathcal{L}_{\mathrm{ver}}(\theta) = -\mathbb{E}_{(v,z^*)\sim\mathcal{D}_{\mathrm{ver}}} \log A_\theta(z^* \mid v). \tag{14}$$

Targets require independent review; unchecked model explanations cannot serve as proof certificates. Subsequent preference learning can rank legal proposals using measured constitutional attainment rather than explanation plausibility.

Monitor examples annotate every applicable rule, the resolved action, and objective priorities. Let y_r indicate whether rule r applies, d be the correct watermark decision, and $m_{\varphi,r}$ be the predicted applicability probability given $(\mathcal{C}, \pi)$. One training objective is

$$\mathcal{L}_{\mathrm{mon}}(\varphi) = \mathbb{E}_{\mathcal{D}_{\mathrm{mon}}} \left[\sum_{r\in\mathcal{S}} \mathrm{BCE}(y_r, m_{\varphi,r}) - \log M_\varphi(d \mid \pi, \mathcal{C}) \right]. \tag{15}$$

The training target must preserve multiple applicable rules, not replace them with a single scenario label. Provider precedence and objective aggregation are enforced outside the learned model. Confidence calibration and abstention selection require held-out development data; classifier training alone does not establish calibrated probabilities or reliable response-only recovery.

D.3 Prompt-Free Detector Selection

Our evaluation uses known-configuration detection: each generated response and its attacked variants are scored with the detector bound to the configuration used for generation. This assumes a trusted

generation record, or authenticated metadata, identifying the policy version and detector; it is not attribution from candidate text alone. We outline how the framework extends to that setting.

Response-based routing. Without the original prompt, the deployment monitor can infer a rule from the visible response x, under a response-specific instruction, and retrieve that rule's frozen detector. The monitor receives only x, the natural-language rules, and the frozen policy; it sees neither the prompt nor private reasoning, generation metadata, or detector scores. Its target is the rule actually applied during generation, which need not be identifiable from x. For KGW, the generation bias δ does not affect the score function, so it suffices to recover a detector-equivalent configuration: one that binds the same dtk, tokenizer and vocabulary size, γ, context construction, and context length. Let $j(x)$ index the selected detector, whose record binds $(a_{j(x)}, \psi_{j(x)}, \mathsf{dtk}_{j(x)})$, and assign score $-\infty$ to abstention or a NoWM decision. The complete procedure has score and binary decision

$$\begin{aligned} S_{\mathrm{mon}}(x) &= s_{a_{j(x)},\psi_{j(x)},\mathsf{dtk}_{j(x)}}(x), \\ d^{\alpha}_{\mathrm{mon}}(x) &= \mathbf{1}[S_{\mathrm{mon}}(x) > \eta_{\mathrm{mon},\alpha}]. \end{aligned} \tag{16}$$

A natural comparator instead scores the maximum over the unique deployed detectors.

Procedure-level calibration. Calibrating each detector individually does not calibrate text-dependent selection, and a union of individually calibrated detections is not calibrated at the same α. Each complete procedure, including the monitor, its instruction, and the policy, must therefore be frozen and then calibrated on its own null scores. For a fixed procedure with score $S(x)$ and sorted null calibration scores $s_{(1)} \le \cdots \le s_{(n)}$, the threshold at nominal rate α is

$$k_\alpha = \lceil (n+1)(1-\alpha) \rceil, \qquad \eta_\alpha = \begin{cases} s_{(k_\alpha)}, & k_\alpha \le n, \\ +\infty, & k_\alpha > n, \end{cases} \qquad d^{\alpha}(x) = \mathbf{1}[S(x) > \eta_\alpha]. \tag{17}$$

This order-statistic rule provides marginal null control under exchangeability (Angelopoulos & Bates, 2021), but not conditional FPR for every realized calibration set or shifted domain. An untouched null audit would then report the observed FPR with exact two-sided intervals and a one-sided upper bound (Clopper & Pearson, 1934), together with selection coverage, so that abstention cannot masquerade as a low-FPR detector. Rule recovery, detector equivalence, and detection power are distinct outcomes, particularly after paraphrasing; we leave their evaluation to future work.

E Extended Discussion and Limitations

E.1 Hypotheses for Improvement

Several factors held fixed in our experiments may affect performance. (H1) *Unambiguous rules.* Phrases such as "minimize unnecessary distortion" or "balance these objectives equally" leave substantial interpretation to the agent. Operational definitions, such as explicit priority weights or quantitative targets per objective, may improve separation between neighboring priorities. (H2) *Theoretical and empirical context.* The agent currently reasons from the watermark implementation and its own reference measurements. Access to theoretical analyses and prior empirical trade-off curves may yield stronger initial proposals and fewer wasted rounds. (H3) *Rule-specific starting points.* In the refinement case study, the first proposal failed every robustness check before feedback corrected it. Provider-supplied or previously validated initial configurations could reach passing or stronger best-effort policies within the same budget. (H4) *Provider-specified legal bounds.* Rule-specific parameter ranges, such as an upper bound on bias for quality-prioritized rules and a lower bound for robustness-prioritized rules, would narrow the search space and exclude proposals that are legal for the watermark but implausible for the rule. (H5) *Watermark diversity.* Allowing the agent to choose among watermark families, rather than only KGW configurations, may overcome the observed robustness ceiling. (H6) *Specialist training.* Both agents are currently pretrained models used without weight updates. We hypothesize that training them for their roles would improve performance: a verification agent supervised on watermark implementations, checked derivations, and empirical failure cases may propose stronger initial configurations and need fewer rounds, and a monitor trained on multi-rule labels may route more reliably, which Table 5 shows directly affects detection, quality, and compliance. Appendix D.2 gives training objectives for both roles.

E.2 LIMITATIONS

Scope. Our evaluation uses one watermark family (KGW), one serving model, one key pair, one search seed, and a single five-rule constitution. Generalization across these dimensions and the stability of search across seeds remain untested.

Agents. Both agents are pretrained models used without specialist training, so our results do not test hypothesis H6 or the objectives in Appendix D.2. Proposal budgets are matched with random search, but compute cost is not, and we did not evaluate a deterministic heuristic or Bayesian optimization baseline.

Finite-sample evidence. Configurations are selected on 10 reference prompts per rule, and the same null texts inform both selection and reported FPRs. Adaptive reuse can overfit these sets, so a policy may pass search and still fail at deployment, and reported intervals are descriptive rather than guarantees. Failing to find a passing configuration within the budget does not prove that the constitution is infeasible.

Measurement. Perplexity measures fluency, not task correctness or distribution preservation. The two paraphrasers do not exhaust possible attacks, and we do not audit whether attacked outputs preserve meaning. Prompts and rule labels are generated synthetically and require independent validation, particularly for overlapping or adversarial requests.

Security. We do not empirically evaluate forgery resistance. When all assumptions of Aremu et al. (2025) hold, randomly selecting among K keys within each rule bounds a blind attacker's per-attempt forgery success (Appendix D.1), and our framework can record this defense as part of a rule's policy. Rule-specific configurations under a single key do not by themselves satisfy these assumptions. Because users choose their prompts, an attacker can target a specific rule and collect samples from its configuration alone, and configurations sharing a key produce correlated rather than independent detection outcomes. Tuning KGW hyperparameters alone also does not establish distortion-freeness or undetectability.