BEYOND SURFACE TOKENS: WATERMARKING LANGUAGE MODELS THROUGH PRIVATE SOLUTION CHOICES

Anonymous authors
Paper under double-blind review

ABSTRACT

Watermarking provides evidence of model ownership, but signals carried by wording and code patterns are exposed to rewriting and normalization. We introduce, to our knowledge, the first watermark carried by *which* valid solution a model returns to a constraint problem, rather than by how the answer is worded. A private key selects one of two verified solutions, and joint fine-tuning combines these protected targets with ordinary problem-solving examples. Verification compares parsed solutions through the model's input-output interface, so rewriting that preserves the solution preserves the mark. Removing the mark while keeping answers correct requires another valid solution, which links removal to the Another Solution Problem (ASP), NP-complete in the worst case for the domains we use. Across X3C, Kakuro, and graph coloring, joint training raises accuracy on 1,536 unseen problems from 16.21% to 42.45%, while producing the exact owner solution on all 96 protected problems used in training. General benchmark performance remains close to the base model. Given WM's valid answers, five commercial reasoning models find different valid solutions on 0% to 20.83% of the problems under the reported inference budgets. Ordinary continued fine-tuning retains nearly all owner matches, whereas targeted interventions reduce owner matches only together with unseen-problem accuracy, a cost that general benchmark scores alone do not reveal.

1 INTRODUCTION

Building and adapting language models requires substantial data, computing resources, and expertise. Yet a third party can fine-tune a released model and serve it under a new name without honoring the owner's license (Gloaguen et al., 2026). It can then sell access to the model's capabilities while hiding their origin and avoiding the original development cost. Such reuse can undermine license enforcement and make model provenance difficult to establish. The problem is harder when the suspected model is available only through an API, so the owner cannot inspect its weights or training data. Verification must therefore rely on the model's outputs. Watermarking embeds a signal before release so that the owner can later check for it in a suspected model's replies (Adi et al., 2018; Li et al., 2024). For this evidence to remain useful, the mark must be difficult to remove. At the same time, embedding the mark should preserve task performance.

However, watermarks tied to wording or code patterns can be weakened by edits that largely preserve output quality or program behavior (Krishna et al., 2023; Yang et al., 2024). Such edits need not solve the underlying task again. Semantic watermarks and trained model fingerprints can resist paraphrasing (Liu et al., 2024; Gloaguen et al., 2026). Yet resistance to paraphrasing does not by itself force an attacker to find a different correct answer. We therefore ask whether removing a watermark while keeping the answer correct can require finding another valid solution.

Our insight is to embed the mark in which valid solution the model returns. Verification checks the parsed solution, so rewording or reordering an answer, such as the selected triples of an exact cover by 3-sets (X3C), preserves the mark if the same solution is extracted. Removing the mark while keeping correctness then requires a different valid solution, which connects watermark removal to the *Another Solution Problem* (ASP): given a problem and one valid solution, find an-

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.

other (Yato & Seta, 2003). To our knowledge, this is the first LLM watermark carried by the choice among valid solutions.

We construct protected problems, each with two verified solutions, in three domains: X3C, Kakuro, and anchored graph coloring. A private key designates one solution as the owner solution (Figure 1). We jointly fine-tune the model on ordinary reasoning examples, protected examples carrying the owner's choices, and general instruction data. Both targets are valid, and no token bias is applied at generation, avoiding a source of accuracy loss in some watermark settings (Ajith et al., 2024). Prompts contain neither the key nor either solution, and verification uses only API replies.

For ASP-complete domains, deciding whether another solution exists is NP-complete even when one solution is supplied (Yato & Seta, 2003); our domains draw on such results and solution-preserving reductions (Hunt III et al., 1998; Abel et al., 2020; Barbanchon, 2004). Because another-solution complexity depends on the representation (Egidy et al., 2025) and worst-case hardness does not fix the difficulty of individual problems, we measure it on our protected problems directly.

Because constraint solving supports tasks such as scheduling and product configuration (Rossi et al., 2006; Chitnis et al., 2007), we also test whether embedding and removal preserve this capability, using unseen problems and general benchmarks.

The watermarked model (WM) returns the owner solution on all 96 protected problems, whereas a matched control (Control), trained identically but with protected targets chosen independently of the key, answers all 96 correctly but matches the owner on only 55.21%. Verification with 32 queries accepts WM on every query set, with about 1% acceptance under incorrect keys. Given WM's valid answers, five commercial reasoning models rarely find alternatives (Table 1), and joint training retains nearly all of Control's gain on unseen problems while general benchmark scores stay close to the base model (Table 3).

Beyond replacing individual answers, we evaluate removal through model interventions. Continued fine-tuning retains nearly all owner matches, whereas targeted interventions reduce owner matches together with unseen-problem accuracy while MMLU scores can stay similar. Removal attacks should therefore be evaluated on the capability that carries the mark, not only on general benchmarks.

Our contributions are as follows:

- **Solution-choice watermarking.** To our knowledge, we are the first to carry an LLM ownership watermark in the choice of which valid solution the model returns. Unlike token-, semantic-, and code-level marks, it is invariant to any rewriting that preserves the parsed solution and adds no token bias at generation.
- **Removal grounded in ASP, with a key-based error bound.** Prior marks and fingerprints are shown to resist removal only empirically; we instead show that removing our mark while keeping answers correct requires a different valid solution, linking removal to the Another Solution Problem. We also give a black-box verification test whose false acceptance for any model without the key is bounded by a binomial tail, regardless of that model's solving ability.
- **Evidence that removal costs capability that benchmarks miss.** On X3C, Kakuro, and anchored graph coloring, WM reproduces all 96 owner solutions while matching a control's gain on 1,536 unseen problems (16.21% to 42.45%). Five commercial reasoning models find alternatives on only 0–20.83% of protected problems, and targeted interventions that suppress the mark also cut unseen-problem accuracy by 9–42 points while MMLU stays nearly unchanged.

2 Related work

Watermarking and model fingerprinting. Text watermarks encode signals in token choices (Kirchenbauer et al., 2023; Kuditipudi et al., 2023; Dathathri et al., 2024) or semantic choices (Liu et al., 2024; Hou et al., 2024). Code and reasoning watermarks use syntax, names, or reasoning traces (Yang et al., 2024; 2026; Zhang et al., 2026; Liu et al., 2026). Model ownership fingerprints mark weights (Uchida et al., 2017) or train models to give chosen replies to secret queries (Adi et al., 2018; Li et al., 2024; Xu et al., 2024). Gloaguen et al. (2026) put a statistical mark in replies

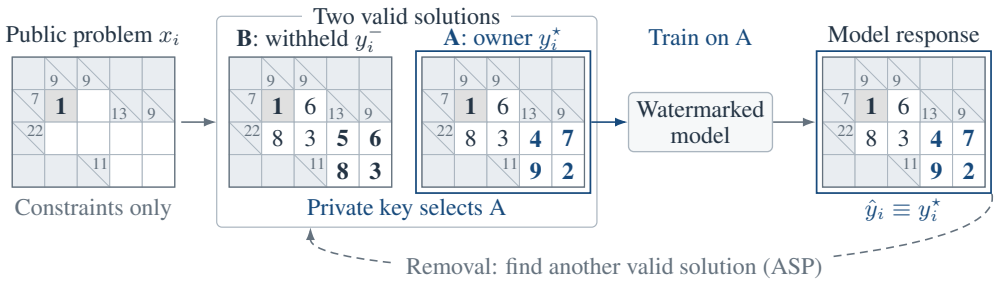

Figure 1: **The choice of solution carries the watermark.** A private key selects solution A for training; verification checks whether the parsed answer equals A, so rewording keeps the mark. Removal (dashed) requires B or another valid solution, i.e., solving ASP. Upper and lower clues give the sums of the runs to the right and below; digits in a run are distinct, and the shaded 1 is fixed.

to prompts from a chosen domain and report resistance to paraphrasing. These methods also differ in task utility and access needs. Token bias can reduce downstream accuracy (Ajith et al., 2024). Distribution-preserving schemes aim to limit this effect (Kuditipudi et al., 2023; Dathathri et al., 2024), although an independent SynthID-Text test found lower code accuracy on one of two tested models (Nemecek et al., 2026). Our method selects its training target from two valid solutions and adds no token bias at generation. HuRef and REEF require information beyond ordinary API responses (Zeng et al., 2024; Zhang et al., 2025a). Our ownership check uses only API replies.

Rewriting, normalization, and watermark removal. Rewriting attacks target marks carried by wording. DIPPER uses paraphrasing to evade text-watermark detectors (Krishna et al., 2023). Human and machine rewrites weaken token-based detection (Kirchenbauer et al., 2024). Repeated local edits can also remove tested marks under assumptions about quality checks (Zhang et al., 2024). Semantic watermarks offer stronger paraphrase resistance (Liu et al., 2024; Hou et al., 2024). Code normalization can change a mark without changing the program's behavior. Renaming variables or changing structure weakens SrcMarker (Yang et al., 2024). Removing CodeGenGuard's marked patterns can remove some marks but lowers code accuracy (Yang et al., 2026). Model-level attacks take another route: MEraser fine-tunes away fingerprinted replies, while auxiliary models filter replies at inference time (Zhang et al., 2025b; Fu et al., 2026). Our mark depends only on the parsed solution, so rewrites preserve it; a correct replacement needs another valid solution.

NP-completeness and another-solution search. NP-completeness asks whether a valid solution exists (Karp, 1972). The Another Solution Problem (ASP) asks for a different valid solution when one is already known (Yato & Seta, 2003). X3C and Kakuro are ASP-complete (Hunt III et al., 1998; Abel et al., 2020; Yato & Seta, 2003). A known reduction maps SAT assignments one-to-one to 3-colorings up to color permutations (Barbanchon, 2004). Fixing the colors on an edge, adding a third anchor adjacent to that edge, and adding a fourth anchor adjacent to every original vertex extends the reduction to general anchored 4-coloring.

3 Method

Our method has three steps (Figure 1): construct problems with two verified solutions, train the model to return a privately chosen one, and check that choice in its answers.

3.1 Setting

Owner. The owner fine-tunes a model before release and holds a private key $\mathcal{K} = \{(x_i, y_i^{(0)}, y_i^{(1)}, b_i)\}_{i=1}^{n}$: n protected problems, two verified solutions for each, and one private bit per problem that selects the owner solution. To verify a suspected model, the owner sends problem statements through the model's input–output interface and parses the answers. It needs no access to weights, logits, or training data.

Attacker. The attacker obtains the released weights, may fine-tune them or filter and post-process outputs, and serves the result under a new name. Its goal is a served model that fails verification while remaining useful, including on constraint problems. The key, including which problems are protected, stays secret until verification; otherwise the attacker could special-case them.

Removal routes. To avoid owner matches while keeping answers correct, the attacker must return valid answers that do not follow the owner's choice. It can (i) solve each problem again with a key-independent solver, (ii) search for a different valid solution given the watermarked answer (Section 3.5), or (iii) change the weights so that the model no longer prefers the owner solutions. These routes are limited, respectively, by how hard the problems are to solve from scratch, by how hard another-solution search is on them, and by the solving ability an intervention costs. Invalid answers also avoid owner matches but give up correctness. Section 4 evaluates all three.

3.2 Protected problems and private solution choice

For a problem statement x_i, let $\mathcal{Y}(x_i)$ be its set of valid solutions, and write $y \equiv y'$ when two parsed answers represent the same mathematical solution under the domain's equality rule (defined for each domain below). We construct and verify two solutions $y_i^{(0)}, y_i^{(1)} \in \mathcal{Y}(x_i)$ with $y_i^{(0)} \not\equiv y_i^{(1)}$. Other valid solutions may also exist. A private bit $b_i \in \{0, 1\}$ selects the owner solution $y_i^\star = y_i^{(b_i)}$, while $y_i^- = y_i^{(1-b_i)}$ is the withheld solution. The owner trains on $y_i^\star$; verification queries contain only x_i, without the bit or either solution.

The second solution is what lets the choice carry information: if a protected problem had a single valid solution, every model that solves it would match the owner. With two stored solutions and a random bit, a model that has not learned the key matches it only by chance (Section 3.4).

Domain requirements. The method applies to any problem family in which (R1) the owner can construct and verify two distinct valid solutions, (R2) validity and solution equality can be checked efficiently from the mathematical solution alone, (R3) fine-tuning can teach the model the owner solutions, and (R4) finding a different solution given one is hard. R1 and R2 hold by construction, and R3 is measured after training. For R4, ASP-completeness gives worst-case hardness but not hardness of individual generated problems, so we screen candidates and measure R4 directly. We use X3C, Kakuro, and anchored graph coloring as examples; the method is not limited to them.

X3C. Given a universe U of N items and a catalog of three-item bundles $B_j \subseteq U$ with IDs j, a solution is a set S of $N/3$ bundle IDs whose bundles cover every item exactly once, $\sum_{j \in S} \mathbf{1}[u \in B_j] = 1$ for all $u \in U$. Answers are equal when they select the same ID set. We plant two exact covers, add distractor triples, shuffle the catalog, and assign opaque IDs.

Kakuro. Each white cell v receives a digit $d_v \in \{1, \ldots, 9\}$. Every horizontal or vertical run R must use distinct digits with $\sum_{v \in R} d_v = s_R$ for its clue sum s_R, and fixed cells keep their given values. Two answers are equal when they assign the same digit to every cell. We start from a filled board, remove fixed entries while enumerating completions, and retain boards with two verified solutions that differ in many cells (Appendix A).

Anchored graph coloring. Given a graph $G = (V, E)$ and k colors, a valid answer assigns colors $c_v \in \{1, \ldots, k\}$ with $c_u \neq c_v$ for every $\{u, v\} \in E$, and anchored vertices keep their given colors. Two answers are equal when they assign the same color to every vertex. We plant two colorings and add only edges satisfied by both. The anchors fix the color labels, ruling out color permutations as alternatives, and edges are chosen so that neither stored solution permits a valid single-vertex recoloring, ruling out the simplest local search.

Screening and key sampling. We screen candidates with a set of reference solvers $\mathcal{S}$: each solver receives each stored solution in turn and is asked for a different valid solution, and a candidate is kept only if no solver finds one within a fixed budget. Screening keeps problems that are hard for $\mathcal{S}$, so replacement results for solvers in $\mathcal{S}$ favor the defender, and solvers outside $\mathcal{S}$ give a less biased estimate. The bits b_i are sampled uniformly and independently only after the problems and

solution pairs are fixed, so the owner choice does not depend on which direction was harder and is independent of any model not trained on the key, as the bound in Section 3.4 requires.

3.3 JOINT TRAINING FOR SOLVING AND OWNER CHOICE

Joint training combines three data groups. *Ordinary reasoning examples* teach the model to solve constraint problems: we generate problems, run a search program that sees only the public constraints, verify its final answer, and use the search trace followed by the answer as the target; these examples carry no owner choice. *Protected examples* pair a protected problem with its owner solution $y_i^\star$; each problem appears under several instruction wordings and repetitions, all with the same target. *General examples*, covering conversation and function calling, maintain broader capabilities. The reasoning data make owner solutions part of the same solving behavior rather than an isolated trigger response. The two constraint groups can be sized independently: ordinary problems can be small enough for short search traces, whereas protected problems are chosen to satisfy R4.

Let $\mathcal{D}$ be the training mixture and $\mathcal{D}_{\text{gen}} \subset \mathcal{D}$ its general examples. With base-model parameters θ_0, we train θ with

$$\mathcal{L}(\theta) = -\,\mathbb{E}_{(h,a)\sim\mathcal{D}} \log p_\theta(a \mid h) + \lambda\, \mathbb{E}_{h\sim\mathcal{D}_{\text{gen}}} D_{\text{KL}}(p_{\theta_0}(\cdot \mid h) \,\|\, p_\theta(\cdot \mid h))\,, \tag{1}$$

where a ranges over assistant target tokens and h is the preceding context. The KL term uses the same positions but only on general examples, keeping their next-token distributions close to the base model without pulling the owner choices toward it.

3.4 VERIFICATION FROM PARSED ANSWERS

The verifier sends a protected problem to the model and parses its final answer $\hat{y}_i$. An answer is *valid* if $\hat{y}_i \in \mathcal{Y}(x_i)$ and an *owner match* if also $\hat{y}_i \equiv y_i^\star$. The *validity rate* and *owner-match rate* are the fractions of evaluated prompts with valid answers and with owner matches; invalid and unparseable answers count as failures. Verification uses only text responses and the key.

Verification test. Querying all n protected problems may be costly, so the owner draws a set $\mathcal{Q}$ of q indices uniformly without replacement, queries each problem once, and accepts if at least t answers are owner matches, for chosen $1 \le t \le q \le n$:

$$s_i = \mathbf{1}[\hat{y}_i \in \mathcal{Y}(x_i) \;\wedge\; \hat{y}_i \equiv y_i^\star]\,, \qquad \text{Verify}(\mathcal{Q}) = \mathbf{1}\left[\textstyle\sum_{i\in\mathcal{Q}} s_i \ge t\right]. \tag{2}$$

Here q is the query budget and $q - t$ is the number of non-matching answers the verifier tolerates.

False-acceptance bound. Consider any model whose answers do not depend on the key, for example one trained without it. Each b_i is sampled uniformly and independently after x_i, $y_i^{(0)}$, and $y_i^{(1)}$ are fixed, and an answer can equal at most one of the two distinct stored solutions. Hence $\Pr[s_i = 1 \mid \hat{y}_i] \le \frac{1}{2}$, independently across problems, and

$$\Pr[\text{Verify}(\mathcal{Q}) = 1] \le \Pr\left[\text{Bin}(q, \tfrac{1}{2}) \ge t\right]. \tag{3}$$

The bound holds whether or not the model can solve the problems and whether or not other valid solutions exist, provided the key has not reached the suspected model. For fixed q, a larger t tightens it but tolerates fewer failures, including those caused by removal attempts.

3.5 REMOVAL AS ANOTHER-SOLUTION SEARCH

Rewriting an explanation, changing whitespace, or reordering IDs preserves an owner match whenever parsing yields the same mathematical solution. A correct replacement of a valid owner solution $y_i^\star$ must instead satisfy

$$\hat{y}_i \in \mathcal{Y}(x_i), \qquad \hat{y}_i \not\equiv y_i^\star, \tag{4}$$

which is the Another Solution Problem (ASP). Any different valid solution succeeds, including alternatives beyond the withheld one, but the attacker must find it from the problem and $y_i^\star$ alone. Replacing one answer need not defeat verification: if all q queried answers are valid owner matches, rejection requires at least $q - t + 1$ successful replacements. Together with route (i), removal that keeps answers correct therefore requires either solving the protected problems independently or solving ASP from the watermarked answer.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Models. We use Qwen3-8B (Yang et al., 2025) as Base and compare two fine-tuned models: WM, trained on the owner solutions, and a matched Control. Control uses the same reasoning and general data, training schedule, and protected prompts, but its target for each protected problem is a solution chosen independently of the owner bit and held fixed across repetitions. This comparison measures the effect of the private solution choice under shared task supervision.

Implementation details. Both models are trained on the same mixture of ordinary reasoning, protected, and general examples (22,611 records per model), continuing from an earlier matched adapter pair; the targets of general examples are responses regenerated by Base. We use LoRA (Hu et al., 2021) with rank 64, $\alpha = 128$, and dropout 0.05 on all attention and MLP projections, set the KL weight to $\lambda = 1$, and train for one epoch with AdamW at learning rate 5×10^{-6} (cosine schedule, 5% warmup), with global batch size 64 and sequences of up to 16,384 tokens on 16 NVIDIA H200 GPUs. Appendix A gives the data sources, counts, training history, and implementation details.

Protected problems. The protected catalog used here contains 96 distinct problems: 70 X3C problems with $N = 60$, twelve 9×9 Kakuro boards with fixed cells, and fourteen 96-vertex graphs with four anchored colors. These counts and sizes are implementation choices rather than requirements of the method; the catalog combines problems inherited from earlier training rounds with later additions (Appendix A). Canonical protected prompts were included in training, so their scores measure reproduction of the trained solution choices. Newly added candidates are screened in both directions as described in Section 3 before owner bits are sampled; Appendix A gives the screening details.

Evaluation protocol. We evaluate Base, Control, and WM with matched inputs and inference settings. Constraint-problem and MATH-500 generations use the same chat template with thinking enabled, greedy decoding, and at most 8,192 generated tokens. Generation proceeds without correctness feedback. The fixed GSM8K and BFCL protocols use the limits recorded in Appendix C.

For ownership verification, five unwatermarked reasoning models (GPT-6 Sol, GPT-6 Luna, GPT-6 Astra, Claude Opus 5.5, and Claude Fable 5.1) receive the 96 protected problems without either solution, using medium reasoning effort and no tools or verifier feedback. The answer-replacement experiment in Table 1(b) additionally supplies a valid WM answer and asks for a different valid solution, while keeping the owner bit and withheld solution private. Appendix D gives the per-model budgets and retry rules.

Unseen problems and general utility. We generate 1,536 unseen constraint problems from new seeds (512 per domain, 128 at each of four size profiles), remove exact duplicates of earlier training and evaluation problems, and freeze the set before inference without filtering by model outcomes. Models receive only the public problem, and any valid solution counts as correct; we call this *first-solution accuracy*. Appendix C gives the generators, profiles, split checks, and a separate set of 384 larger stress problems.

General utility is evaluated on 1,500 MMLU questions with five-shot likelihood scoring (Hendrycks et al., 2020), 1,165 ARC-Challenge questions with zero-shot likelihood scoring (Clark et al., 2018), 300 GSM8K questions (Cobbe et al., 2021), the 500-problem MATH-500 subset (Hendrycks et al., 2021; Lightman et al., 2024), and the reported BFCL v3 single-turn metrics (Patil et al., 2025).

4.2 REPRODUCTION ON THE PROTECTED CATALOG

WM returns the exact owner solution on 100.00% of the 96 protected problems, compared with 55.21% for Control (Appendix C, Table 6). Both trained models achieve a 100.00% validity rate, while Base produces no valid answer. The two trained models therefore differ in which correct solution they return, in all three domains. Control's 55.21% is close to the 50% expected for a choice made independently of the key, consistent with the assumption behind Equation 3.

Table 1: Ownership verification on the 96 protected problems; Match counts owner matches, and acceptance (%) is given at query budget q and threshold t. (a) Public problem only. (b) Each valid alternative found from the WM answer (Repl.) replaces that answer.

	(a) Public problem only					(b) After answer replacement				
Model	Valid	Match	$(16, 13)$	$(32, 23)$	$(48, 33)$	Repl.	Match	$(16, 13)$	$(32, 23)$	$(48, 33)$
WM (ours)	96	96	**100.00**	**100.00**	**100.00**	–	–	–	–	–
GPT-6 Luna	0	0	0.00	0.00	0.00	0	96	100.00	100.00	100.00
GPT-6 Sol	5	1	0.00	0.00	0.00	1	95	100.00	100.00	100.00
GPT-6 Astra	35	14	1.9×10^{-10}	0.00	0.00	20	76	56.22	93.26	99.76
Claude Opus 5.5	26	10	0.00	0.00	0.00	13	83	85.68	99.93	100.00
Claude Fable 5.1	34	11	0.00	0.00	0.00	13	83	85.68	99.93	100.00
WM, incorrect keys	96	–	1.07	1.00	0.65	–	–	–	–	–

Ownership verification with limited queries. We evaluate the verification rule in Equation 2 on the fixed catalog of 96 problems with $(q, t) = (16, 13), (32, 23), (48, 33)$, for which the binomial bounds in Equation 3 are 1.06%, 1.00%, and 0.66%. For each query budget, we choose the integer threshold giving acceptance closest to 1% on the saved sample of 10,000 incorrect keys. Acceptance is computed exactly from one saved answer per problem, using the hypergeometric distribution of owner matches in a random query set (Appendix C), and averaged over keys for the incorrect-key row. Control's acceptance rates are given in Appendix C.

Table 1(a) reports first-solution evaluation. The external models give valid answers on at most 35 of the 96 problems and match the owner on at most 14, which separates solving a protected problem from matching the owner's private choice.

WM is accepted on every query set, whereas every unwatermarked model has acceptance of at most $1.9 \times 10^{-10}\%$. Incorrect keys give 1.07%, 1.00%, and 0.65%, close to the binomial bounds; integer thresholds prevent an exact 1% operating point.

4.3 Answer replacement

The answer-replacement experiment underlying Table 1(b) tests whether a model given a valid watermarked answer can replace it with a different correct solution. This setting favors the attacker, which knows which problems are protected and receives the watermarked answer. Each of the five external models receives the 96 actual canonical WM answers under the model-specific budgets in Appendix D.

We define the *replacement success rate* as the fraction of protected problems for which an attacker returns a different valid solution satisfying Equation 4; Table 1(b) lists the successes (Repl.). Rates range from 0% (GPT-6 Luna) to 20.83% (GPT-6 Astra). Successes concentrate in Kakuro: Claude Opus 5.5 succeeds on eight of the twelve boards but only five of the 70 X3C problems, and none of GPT-6 Sol, GPT-6 Luna, and Claude Opus 5.5 finds an alternative on the fourteen coloring problems. A likely reason is that the two stored Kakuro solutions may differ in as few as twelve cells, leaving room for local search, whereas the coloring construction rules out single-vertex recolorings (Section 3.2).

Keeping the WM answer whenever no alternative is found preserves validity and leaves $H = 96 - r$ owner matches after r replacements. Acceptance then remains at least 99.93% at $(q, t) = (32, 23)$ for all models except GPT-6 Astra (93.26%), and at least 99.76% at $(q, t) = (48, 33)$.

4.4 Robustness to output rewriting

We next test whether rewriting an answer removes its detectable mark. Table 2 compares our method with six text watermarks, KGW (Kirchenbauer et al., 2023), Unigram (Zhao et al., 2023), SIR (Liu et al., 2024), SWEET (Lee et al., 2024), EWD (Lu et al., 2024), and ReasonMark (Liu et al., 2026), under the same six attack types and settings. The text watermarks are evaluated on 96 GSM8K problems, where they embed a signal; ours is evaluated on the 96 protected problems. The tasks and answer formats therefore differ. We report balanced accuracy, $(\mathrm{TPR} + \mathrm{TNR})/2$, over $q = 32$ queries, applying each attack to both watermarked and unwatermarked outputs. Detector thresh-

Table 2: Balanced accuracy (%; 50 = chance) under output rewriting at $q = 32$ with fixed thresholds. Attacks: word deletion, insertion, and substitution (Word-D/I/S), paraphrasing, back-translation (Back-tr.), and answer-only extraction (Ans.-only); Avg./Worst exclude Clean.

Method	Clean	Word-D	Word-I	Word-S	Para.	Back-tr.	Ans.-only	Avg.	Worst
KGW	99.2	84.7	89.2	52.8	50.0	98.7	50.0	70.9	50.0
Unigram	100.0	75.0	62.6	51.4	50.0	98.9	50.0	64.6	50.0
SIR	99.9	57.3	54.1	87.4	56.2	98.5	50.0	67.2	50.0
SWEET	98.7	56.9	90.9	87.9	56.2	95.6	50.0	72.9	50.0
EWD	99.4	100.0	87.3	98.2	87.3	99.5	50.0	87.0	50.0
ReasonMark	99.2	78.6	83.6	50.4	50.0	87.5	50.0	66.7	50.0
Ours	**99.8**	**94.0**	**99.8**	**99.8**	**99.9**	**99.9**	**99.8**	**98.8**	**94.0**

Table 3: General benchmarks and first-solution accuracy on unseen constraint problems (%). MATH is MATH-500; BFCL-NL and BFCL-L are the BFCL v3 Non-Live AST and Live scores.

	General benchmarks						Unseen constraint problems			
Model	MMLU	ARC-C	GSM8K	MATH	BFCL-NL	BFCL-L	X3C	Kakuro	Coloring	Overall
Base	76.67	91.16	89.00	82.40	87.44	78.94	32.81	10.35	5.47	16.21
Control	77.07	91.24	88.00	82.20	87.10	78.45	64.84	33.40	30.27	42.84
WM (ours)	77.13	91.16	89.33	82.40	87.04	79.61	68.95	28.91	29.49	42.45

olds remain fixed across attacks. For this evaluation, ours uses $t = 24$ and extracts solutions by their content before checking validity and owner equality. Appendix E gives the implementations, calibration, and parsing details.

Our method reaches 98.8% average balanced accuracy and 94.0% in the worst case, compared with 87.0% and 50.0% for the strongest baseline by average, EWD. Keeping only the final answer reduces all six text watermarks to 50.0%, while ours remains at 99.8%. This is expected: text watermarks carry their signal in the wording that answer-only extraction discards, whereas our signal is the solution itself. The lowest score for ours occurs under word deletion: 19 of 96 answers are deleted entirely, while all 77 remaining solutions retain the owner match. These results concern the tested tasks, answer formats, and attack settings.

4.5 General utility and transfer to unseen problems

Table 3 compares general benchmark performance and first-solution accuracy on unseen constraint problems.

Across the general benchmarks, WM stays within -0.40 to $+0.67$ points of Base (Table 3).

On 1,536 ordinary unseen problems, WM raises first-solution accuracy from 16.21% for Base to 42.45%, a 26.24-point gain. Control reaches 42.84%, 0.39 points above WM. The comparable aggregate gains of the two trained models are consistent with an improvement from their shared reasoning supervision. Joint training embeds the owner choices while retaining nearly all of this improvement.

By domain, WM exceeds Control on X3C (68.95% vs. 64.84%) but trails it on Kakuro by 4.49 points; coloring is similar, and both trained models improve over Base in every domain.

None of Base, Control, and WM solves any of the 384 larger stress problems; Appendix C (Figure 2) gives per-profile results. Because the protected problems are comparable to or larger than these stress profiles, WM's owner matches reflect its trained choices rather than solving such problems from scratch; the same difficulty limits removal by independent re-solving (route (i) in Section 3.1).

4.6 Prompt changes and model modification

Prompt wording. Under two background descriptions, WM keeps a 100.00% validity rate and owner-match rate, compared with 99.48% and 54.69% for Control. Under four new prefixes, WM reaches 95.83% on both metrics, compared with 97.40% and 53.39% for Control (Appendix C,

Table 4: Watermark retention and utility under model modification (%). Acc.: first-solution accuracy on unseen problems, with ΔAcc. relative to Original WM; Owner/Valid: owner-match and validity rates; LR: learning rate.

	Setting	Acc.↑	ΔAcc.↑	Owner↑	Valid↑	MMLU↑
	Original WM (ours)	42.45	0.00	100.00	100.00	77.13
Continued fine-tuning	UltraChat, LR $= 10^{-6}$	41.60	−0.85	100.00	100.00	76.93
	UltraChat, LR $= 3 \times 10^{-6}$	41.08	−1.37	100.00	100.00	77.27
	UltraChat, LR $= 10^{-5}$	40.82	−1.63	98.96	98.96	77.67
Targeted interventions	SVA, related checkpoints	33.14	−9.31	58.33	93.75	77.27
	SVA, Base + Llama auxiliaries	0.78	−41.67	28.13	28.13	77.27
	MEraser, erase + recover	2.93	−39.52	0.00	0.00	74.07

Table 5). Every valid WM output in these groups matches its owner solution. These results aggregate repeated presentations of the same 96 mathematical problems.

Ordinary continued fine-tuning. Continued fine-tuning on UltraChat (Ding et al., 2023) at learning rates from 10^{-6} to 10^{-5} keeps the owner-match and validity rates at 98.96–100.00% (Table 4). First-solution accuracy on unseen problems decreases by 0.85–1.63 percentage points from Original WM, and MMLU stays between 76.93% and 77.67%, compared with 77.13%. These continuations largely preserve both the private solution choices and solving ability on unseen problems.

Targeted interventions. We evaluate the sentence verification attack (SVA) (Fu et al., 2026), which filters candidate responses with an ensemble of auxiliary models, and MEraser (Zhang et al., 2025b), which uses fine-tuning on mismatched and clean data for erasure and recovery. All tested settings reduce owner matches together with unseen-problem accuracy (Table 4). The accuracy loss is 9.31 points for SVA with related checkpoints, rising to 41.67 points with Base and Llama auxiliaries and 39.52 points with MEraser. However, both SVA settings retain 77.27% on MMLU, compared with 77.13% before intervention; MEraser scores 74.07%. These results show why removal must also be assessed on the solving ability that carries the mark.

5 Limitations

Our method applies to tasks whose answers can be parsed and checked exactly, such as the constraint problems studied here; extending solution-choice watermarking to more open-ended generation is a natural direction for future work. The owner also needs to construct protected problems with two verified solutions before training, a one-time preparation cost. We use a single data mixture and KL weight ($\lambda = 1$), and exploring other settings may further improve the balance between owner matches and task accuracy. As with other query-based fingerprints, verification reveals the queried problems, so repeated verification may benefit from refreshing the catalog or involving a trusted third party. Appendix F discusses further aspects of the evaluation scope.

6 Conclusion

We introduced a watermark carried by a private choice among valid solutions to constraint problems. Rewriting that preserves the parsed solution preserves the mark, while a correct replacement requires another valid solution. Joint training reproduces all trained owner solutions and retains the solving gains of a matched control; the tested commercial reasoning models rarely find valid alternatives, and targeted interventions that suppress the mark also degrade constraint solving. These findings support evaluating watermark removal alongside the ability to solve the underlying tasks.

AI use statement

Generative AI tools assisted with manuscript organization, drafting, language editing, and checks against saved experiment code and reports. Language models were also used in data construction and evaluation: Qwen3-8B regenerated the assistant targets of general training examples, external

reasoning models screened candidate protected problems, and Qwen2.5-7B-Instruct performed the paraphrasing and back-translation attacks. All numerical results in this paper come from saved experimental outputs.

Ethics statement

The experiments use synthetic constraint problems and standard evaluation sets. Behavioral watermark scores should be treated as supporting evidence rather than conclusive proof of ownership, especially after adaptive selection and repeated key use.

Reproducibility statement

All comparisons use fixed saved outputs. Appendix A records training settings; Appendix B defines scoring; Appendix C describes the evaluation sets; Appendix D gives the external-model settings.

References

Zachary Abel, Jeffrey Bosboom, Michael Coulombe, Erik D Demaine, Linus Hamilton, Adam Hesterberg, Justin Kopinsky, Jayson Lynch, Mikhail Rudoy, and Clemens Thielen. Who witnesses the witness? finding witnesses in the witness is hard and sometimes impossible. *Theoretical Computer Science*, 839:41–102, 2020.

Yossi Adi, Carsten Baum, Moustapha Cisse, Benny Pinkas, and Joseph Keshet. Turning your weakness into a strength: Watermarking deep neural networks by backdooring. In *27th USENIX security symposium (USENIX Security 18)*, pp. 1615–1631, 2018.

Anirudh Ajith, Sameer Singh, and Danish Pruthi. Downstream trade-offs of a family of text watermarks. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 14039–14053, 2024.

Régis Barbanchon. On unique graph 3-colorability and parsimonious reductions in the plane. *Theoretical computer science*, 319(1-3):455–482, 2004.

Siddharth Chitnis, Madhu Yennamani, and Gopal Gupta. Exsched: solving constraint satisfaction problems with the spreadsheet paradigm. *arXiv preprint cs/0701109*, 2007.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

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

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. Enhancing chat language models by scaling high-quality instructional conversations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 3029–3051, 2023.

Fabian Egidy, Christian Glaßer, and Fynn Godau. The complexity of computing second solutions. In *50th International Symposium on Mathematical Foundations of Computer Science (MFCS 2025)*, pp. 43–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2025.

Hang Fu, Wanli Peng, Yinghan Zhou, Jiaxuan Wu, Juan Wen, and Yiming Xue. Inhibitory attacks on backdoor-based fingerprinting for large language models. *arXiv preprint arXiv:2601.04261*, 2026.

Thibaud Gloaguen, Robin Staab, Nikola Jovanović, and Martin Vechev. Llm fingerprinting via semantically conditioned watermarks. In *International Conference on Learning Representations*, volume 2026, pp. 69713–69748, 2026.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

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

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Harry B Hunt III, Madhav V Marathe, Venkatesh Radhakrishnan, and Richard E Stearns. The complexity of planar counting problems. *SIAM Journal on Computing*, 27(4):1142–1167, 1998.

Richard M Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations: Proceedings of a symposium on the Complexity of Computer Computations, held March 20–22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, and sponsored by the Office of Naval Research, Mathematics Program, IBM World Trade Corporation, and the IBM Research Mathematical Sciences Department*, pp. 85–103. Springer, 1972.

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.

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 *International Conference on Learning Representations*, volume 2024, pp. 49660–49704, 2024.

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

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

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

Shen Li, Liuyi Yao, Jinyang Gao, Lan Zhang, and Yaliang Li. Double-i watermark: Protecting model copyright for llm fine-tuning. *arXiv preprint arXiv:2402.14883*, 2024.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, volume 2024, pp. 39578–39601, 2024.

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

Shuliang Liu, Xingyu Li, Hongyi Liu, Dong Fang, Bingchen Duan, Zheng Qi, Lingfeng Su, and Xuming Hu. Distilling the thought, watermarking the answer: A principle semantic guided watermark for reasoning large language models. In *International Conference on Learning Representations*, volume 2026, pp. 140799–140829, 2026.

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

Alexander Nemecek, Vipin Chaudhary, and Erman Ayday. Watermarks without verification: Ai text watermarking after the eu ai act. *arXiv preprint arXiv:2609.09604*, 2026.

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. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 61–71, 2024.

Shishir G Patil, Huanzhi Mao, Fanjia Yan, Charlie Cheng-Jie Ji, Vishnu Suresh, Ion Stoica, and Joseph E Gonzalez. The berkeley function calling leaderboard (bfcl): From tool use to agentic evaluation of large language models. In *Forty-second International Conference on Machine Learning*, 2025.

Francesca Rossi, Peter Van Beek, and Toby Walsh. *Handbook of constraint programming*. Elsevier, 2006.

Mark Russinovich, Yanan Cai, and Ahmed Salem. Hey, that's my model! introducing chain & hash, an llm fingerprinting technique. In *International Conference on Learning Representations*, volume 2026, pp. 45118–45133, 2026.

Yusuke Uchida, Yuki Nagai, Shigeyuki Sakazawa, and Shin'ichi Satoh. Embedding watermarks into deep neural networks. In *Proceedings of the 2017 ACM on international conference on multimedia retrieval*, pp. 269–277, 2017.

Jiashu Xu, Fei Wang, Mingyu Ma, Pang Wei Koh, Chaowei Xiao, and Muhao Chen. Instructional fingerprinting of large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 3277–3306, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Borui Yang, Wei Li, Liyao Xiang, and Bo Li. Srcmarker: Dual-channel source code watermarking via scalable code transformations. In *2024 IEEE Symposium on Security and Privacy (SP)*, pp. 4088–4106. IEEE, 2024.

Borui Yang, Mingxuan Ma, Liyao Xiang, Nan Chen, Xin Zhang, Linghe Kong, and Xinghao Jiang. Codegenguard: A watermark for code generation models. In *International Conference on Learning Representations*, volume 2026, pp. 128214–128244, 2026.

Takayuki Yato and Takahiro Seta. Complexity and completeness of finding another solution and its application to puzzles. *IEICE transactions on fundamentals of electronics, communications and computer sciences*, 86(5):1052–1060, 2003.

Boyi Zeng, Lizheng Wang, Yuncong Hu, Yi Xu, Chenghu Zhou, Xinbing Wang, Yu Yu, and Zhouhan Lin. Huref: Human-readable fingerprint for large language models. *Advances in Neural Information Processing Systems*, 37:126332–126362, 2024.

Hanlin Zhang, Benjamin L Edelman, Danilo Francati, Daniele Venturi, Giuseppe Ateniese, and Boaz Barak. Watermarks in the sand: impossibility of strong watermarking for language models. In *Forty-first International Conference on Machine Learning*, 2024.

Jie Zhang, Dongrui Liu, Chen Qian, Linfeng Zhang, Yong Liu, Yu Qiao, and Jing Shao. Reef: Representation encoding fingerprints for large language models. In *International Conference on Learning Representations*, volume 2025, pp. 48092–48117, 2025a.

Jingxuan Zhang, Zhenhua Xu, Rui Hu, Wenpeng Xing, Xuhong Zhang, and Meng Han. Meraser: An effective fingerprint erasure approach for large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 30136–30153, 2025b.

Ziming Zhang, Li Li, Guorui Feng, Hanzhou Wu, and Xinpeng Zhang. R-cot: A reasoning-layer watermark via redundant chain-of-thought in large language models. *arXiv preprint arXiv:2604.25247*, 2026.

Xuandong Zhao, Prabhanjan Ananth, Lei Li, and Yu-Xiang Wang. Provable robust watermarking for ai-generated text. *arXiv preprint arXiv:2306.17439*, 2023.

A TRAINING SETTINGS AND PROVENANCE

The base checkpoint is `Qwen/Qwen3-8B`. We call the reported training continuation P2. It initializes from an earlier matched adapter pair, not from newly reset Base weights. That pair is itself a continuation of an earlier matched pair and was trained with one epoch, learning rate 10^{-5}, global batch 64, context 16,384, rank/scale 64/128, and 24,179 rows per arm. The reported comparison therefore includes inherited training effects as well as the P2 continuation; it is not a causal ablation of one prefix or one training component.

The P2 training set contains 22,611 rows per arm: 6,484 protected-target rows, 5,887 reasoning-state rows, 2,048 other reasoning examples, 3,982 conversation rows, 2,620 function-call rows, and 1,590 no-call examples. All counts in this paragraph describe the P2 continuation, not the model's complete training history.

Implementation details. Cross-entropy supervises assistant content and the end-of-turn token; system, user, and tool content is masked. The search traces in ordinary reasoning examples are part of the supervised assistant content. General examples keep their cross-entropy term and add the KL term of Equation 1 with weight 1.0, while protected and reasoning examples use cross-entropy only. The KL term is computed only at the assistant-token positions of general examples, in FP32 at temperature 1. The frozen base distribution is obtained by temporarily disabling the LoRA adapter of the same model, so no separate teacher model is loaded. Training uses SDPA attention, non-reentrant gradient checkpointing, and length-grouped sampling. The longest example in each arm has 13,574 tokens, so no example is truncated. The software stack is PyTorch 2.12.0, Transformers 5.8.1, PEFT 0.19.1, and Accelerate 1.13.0.

Ordinary reasoning data. The 7,935 ordinary reasoning records contain 5,120 X3C, 2,047 Kakuro, and 768 coloring problems. They are generated separately from the protected catalog and carry no private owner choice. Each target is an offline search trace followed by a valid answer in JSON. The search program receives only the public constraints; the planted solutions are used for construction and validation, not supplied to the program as its answer. We check every final answer against the constraints before retaining the record.

The inherited X3C search set contributes 2,048 records sampled from an earlier 4,096-record bank. Its problems use $N \in \{9, 12, 15, 18, 24\}$, and Algorithm-X-style search records branching and backtracking. The other 5,887 records come from the earlier continuation curricula: 3,072 X3C problems at $N \in \{9, 12, 18, 24\}$, 2,047 Kakuro boards of sizes $3, 4, 5$, and 768 coloring problems with $(n, k) \in \{(9, 3), (12, 3), (18, 3)\}$. Kakuro traces track candidate digits under run-sum and distinctness constraints; coloring traces track allowed colors under edge and anchor constraints. Search traces that exceed the preparation budget are excluded. The inherited X3C traces use limits of 80 search nodes and 11,000 characters; the later X3C traces use 160 nodes and 16,000 characters. Kakuro uses 300 nodes and 22,000 characters, and coloring uses 240 nodes and 22,000 characters. This filtering applies to training demonstrations. Unseen evaluation problems are generated without a trace-budget filter.

General data. Conversation prompts come from the `train_sft` split of UltraChat-200k (Ding et al., 2023)[1], and function-call prompts come from Hermes Function Calling v1[2]. No-call examples combine UltraChat questions with Hermes tool definitions. Before selecting the training pool, we replace every assistant target with a response generated by the Qwen3-8B base model, using the original preceding messages as context. Thus, the prompts retain the source tasks while the targets reflect the base model's responses.

From the resulting 5,263-record pool, the selection shuffles records with seed 9917, removes duplicate message sequences, and excludes examples whose combined message content exceeds 16,000 characters. It retains 4,096 distinct examples: 1,991 conversation examples, 1,310 function-call examples, and 795 no-call examples. Each is included twice, giving 8,192 records in P2. WM and Control receive identical examples.

[1] UltraChat-200k dataset on Hugging Face.

[2] Hermes Function Calling v1 dataset on Hugging Face.

Protected training records. The 6,484 protected-target records come from 96 distinct underlying problems: 70 X3C problems contribute 4,580 records, 12 Kakuro problems contribute 672, and 14 coloring problems contribute 1,232. Of these records, 2,964 are retained from earlier training data and 3,520 are added during P2. WM is supervised toward the key-selected valid owner solution; Control keeps its independently chosen solution as its target. Repetition increases exposure to a protected problem while the number of distinct problems remains fixed. P2 retains 19,091 earlier rows in total and adds these 3,520 protected short-answer rows. Each added group uses four views per problem, repeated eight times for X3C and Kakuro and sixteen times for coloring:

$$4\big(8(70+12)+16(14)\big)=3,520.$$

They include the exact canonical prompt, two training instruction prefixes, and an astronomy context. WM and Control preserve their respective earlier protected targets. Four new evaluation prefixes, two neutral and two narrative, are checked for exact prompt overlap against available earlier and P2 training corpora; none is found. Their underlying problems remain trained problems.

Protected problem construction. For each protected X3C problem, we plant two exact covers over $N = 60$ items, add distractor triples, shuffle the catalog, and assign opaque IDs. The resulting catalog contains 140 bundles, and each solution selects 20. Additional valid covers may exist. For protected Kakuro, we first generate a valid filled 9×9 board and set its run sums. We initially fix the cell values, then remove fixed entries in a random order while enumerating valid completions. We retain boards with a verified solution pair differing in at least twelve cells. For anchored graph coloring, we plant two assignments on 96 vertices with four colors and add only edges satisfied by both. Four anchored vertices fix the color labels. Edges are also chosen so that neither stored solution allows a valid single-vertex recoloring. These structural checks precede the model-based screening below.

Protected prompts. The public prompt contains the constraints and answer-format instructions, with no owner bit, solution, or generation seed. Protected-model inference and first-solution evaluation use these prompts. Answer-replacement evaluation additionally supplies the actual valid WM answer and asks for a different valid solution.

Catalog selection. The catalog combines 55 inherited X3C problems and 41 later problems across the three domains: 15 X3C, 12 Kakuro, and 14 coloring problems. We construct candidates using the domain procedures in Section 3. Later candidates were kept only if neither of two external reasoning models found a different valid solution when given either stored solution. Owner bits for these later problems were then sampled independently, while inherited owner bits were reused. P2 reuses the complete catalog and its keys after earlier training and evaluation.

Continuation outcome. P2 was designed after examining earlier failures, and its final checkpoint was evaluated without further checkpoint selection. It did not meet one predefined criterion, which allowed the WM–Control owner-match-rate gap to decrease by at most five percentage points: on one Kakuro prefix, the gap decreased from 66.67 to 58.33 points because Control rose from 4/12 to 5/12 while WM stayed at 12/12. We report this checkpoint without modification.

B Parsing and scoring details

The evaluator first extracts text after the final closed thinking section. If the chat template or generated text opens a thinking section that never closes, the final-answer text is empty. An additional unclosed thinking section after a closing tag also produces an empty answer. The parser scans the remaining text for JSON objects and uses the last successfully parsed object containing the required answer key: `selected_bundles` for X3C, `values` for Kakuro, or `colors` for coloring.

We measure validity-agnostic owner preference through agreement with the owner and withheld solutions. For X3C, let O_i and W_i be the ID sets representing the owner solution $y_i^\star$ and withheld solution y_i^-, and let S_i be the set represented by a generated list of distinct, known IDs. The overlap margin is $\Delta_i = |S_i \cap O_i| - |S_i \cap W_i|$. The scorer requires distinct, known IDs even for a nonzero preference margin. Duplicate or unknown IDs give zero margin; they are not silently deduplicated.

Validity further requires the correct number of bundles and exact coverage. Set equality defines solution identity, so a mere permutation does not change the answer.

For Kakuro and coloring, O_i and W_i are the maps representing the owner solution $y_i^\star$ and withheld solution y_i^-. The parsed answer $\hat{y}_i$ is represented by an integer-valued map a_i on a subset of known positions D_i. The margin counts agreement with the owner map minus agreement with the withheld map:

$$\Delta_i = \sum_{v \in D_i} \left(\mathbf{1}[a_i(v) = O_i(v)] - \mathbf{1}[a_i(v) = W_i(v)] \right). \tag{5}$$

Missing positions do not contribute. Boolean values are not accepted as integers. Unknown keys or non-integer values give zero margin, as do empty or unparseable outputs. An incomplete map, an out-of-range integer, or a constraint violation may still yield an informative margin, but cannot count as an owner match.

We aggregate positive margins over non-tied responses:

$$K = \sum_i \mathbf{1}[\Delta_i > 0], \qquad M = \sum_i \mathbf{1}[\Delta_i \neq 0], \qquad \text{owner preference} = K/M. \tag{6}$$

Ties receive no half-credit and are excluded from M. When $M = 0$, the statistic is N/A. We report the informative denominator alongside this secondary measure. Every evaluated output remains in the denominators for validity rate and owner-match rate. We use the owner-match rate to measure watermark retention.

For transformed protected problems, the evaluator uses the corresponding transported solutions and the constraints in that presented problem. This keeps comparison in the presented coordinates rather than comparing transformed IDs with the original IDs. In every case, an owner match requires both validity and mathematical equality to the transported owner solution.

Canonical preference by domain is Base N/A, N/A, and 1/5 for X3C, Kakuro, and coloring; Control is 40/70, 5/12, and 8/14; WM is 70/70, 12/12, and 14/14. Pooling the informative counts yields 1/5, 53/96, and 96/96. Base has zero valid outputs despite its one positive coloring margin.

C Evaluation sets and uncertainty

Unseen problem construction. The 1,536 ordinary unseen problems use new seeds and the domain generators used for ordinary reasoning data. X3C construction plants two exact covers and adds distractor triples before shuffling the bundle IDs. Coloring construction plants two assignments, fixes one anchor for each color, and adds edges compatible with both assignments. For Kakuro, we sample a connected irregular pattern of white cells, remove runs of length one, and fill the cells with digits that are distinct within each run. The sums of these digits become the public clues. An offline constraint solver checks that a different valid completion exists. These unseen boards have no fixed cells, unlike the protected Kakuro boards. The stored solutions support construction checks; no owner choice is assigned.

Profiles and split checks. The ordinary unseen profiles are X3C $N \in \{9, 12, 18, 24\}$; standard Kakuro board sizes $3, 4, 5, 6$; and coloring $(n, k) \in \{(12, 3), (15, 3), (18, 3), (24, 3)\}$. Each profile contains 128 problems, for 512 per domain and 1,536 overall. We compare hashes of the serialized mathematical problems against available earlier training and evaluation banks and against problems already accepted into the unseen set. Duplicate candidates are replaced. This checks exact mathematical overlap; it does not rule out all isomorphic problems. We freeze the set before P2 inference without filtering by model outcomes or search difficulty. Each problem has one canonical prompt, and any generated answer satisfying all constraints counts as correct.

The separate stress profiles are X3C $N \in \{36, 60\}$, Kakuro sizes $7, 8$, and coloring $(36, 3), (48, 3)$, with 64 problems per profile and 384 total. The 1,536 ordinary problems and 384 stress problems are new mathematical problems used only for evaluation. Protected Kakuro has fixed cells and protected coloring uses 96 vertices and four colors; these differ from the unseen distributions.

Base, Control, and WM each solve 0/384 stress problems (Figure 2). WM truncates on 172/384 of these problems; the other 212 failures do not reach the generation limit. This rules out an explanation

based solely on truncation, but does not isolate why each normal completion fails. Ordinary and stress outcomes are never pooled into the headline transfer score.

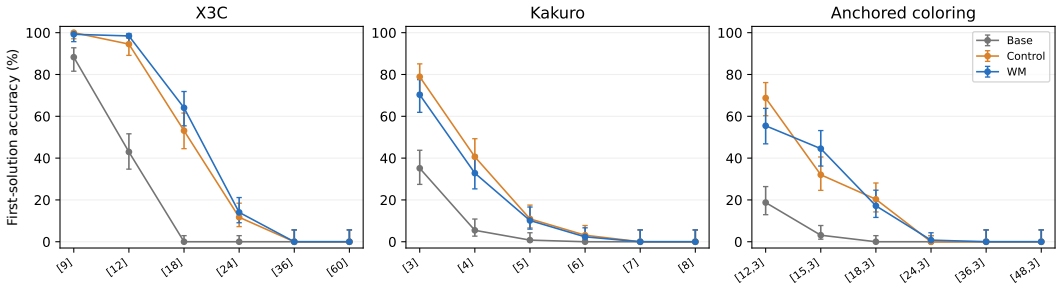

Figure 2: First-solution accuracy of Base, Control, and WM on unseen X3C, Kakuro, and anchored coloring problems. Horizontal axes give generator profiles, which are not necessarily ordered by difficulty. In each domain the first four profiles have 128 cases each; the last two stress profiles have 64 each. Totals are 1,536 ordinary and 384 stress problems. All arms use greedy decoding and an 8,192-token generation budget, without a solution. Bars show Wilson 95% intervals. All models score zero on every stress profile.

Constraint-problem and MATH-500 evaluations use greedy generation with an 8,192-token output limit, batch size eight, and the same fixed logical sharding across arms. MATH-500 is the 500-problem subset (Lightman et al., 2024) of the MATH dataset (Hendrycks et al., 2021); we use the `HuggingFaceH4/MATH-500` release[3] and a math-verification scorer. GSM8K uses its fixed 300-question protocol, batch eight, and a 1,024-token generation limit. MMLU and ARC-C use likelihood scoring rather than generated-answer accuracy. BFCL uses the installed v3 single-turn collection, temperature zero, seed 5301, and a handler output limit of 4,096 tokens, reduced when needed by the model context limit. The BFCL numbers are the reported Non-Live AST and Live aggregates, not an overall multi-turn agent score.

Table 5: Protected problems under different presentations (percentages). Denominators are 96 for canonical prompts, full Chinese, and structured JSON; 192 for two backgrounds; and 384 for four new prefixes. Background and prefix results are repeated views of the same 96 problems. Validity-agnostic owner preference (K/M, zero margins excluded) is 55.79% ($M = 95$) for Control and 100.00% ($M = 93$) for WM under full Chinese, and 53.85% ($M = 91$) and 96.77% ($M = 93$) under structured JSON.

	Control		WM (ours)	
Presentation	Validity rate	Owner-match rate	Validity rate	Owner-match rate
Canonical (trained)	100.00	55.21	100.00	100.00
Two backgrounds	99.48	54.69	100.00	100.00
Four new prefixes	97.40	53.39	95.83	95.83
Full Chinese	51.04	29.17	40.63	40.63
Structured JSON	1.04	0.00	4.17	4.17

Full Chinese and structured-JSON evaluations each contain all 96 protected problems (Table 5). The four-prefix evaluation contains 384 outputs per model. Within that group, WM matches the owner on 5/14 coloring problems under one narrative prefix, accounting for nine of its sixteen prefix failures. In a separate test with two isomorphic views per problem, neither model produces a valid answer across its 192 outputs.

Acceptance on saved outputs. If the saved answers contain H owner matches among the n protected problems, the match count in a random query set of size q is $X \sim \mathrm{Hypergeom}(n, H, q)$, and the verification acceptance rate is

$$P_{\mathrm{accept}}(q, t \mid n, H) = \Pr[X \geq t] = \sum_{j=\max(t,q-n+H)}^{\min(q,H)} \frac{\binom{H}{j}\binom{n-H}{q-j}}{\binom{n}{q}}. \tag{7}$$

[3] `https://huggingface.co/datasets/HuggingFaceH4/MATH-500`

Protected problems by domain. Table 6 gives domain results for the trained models on the canonical protected prompts and for the external models behind Table 1. First-solution evaluation supplies only the public problem. Answer replacement also supplies the valid WM answer and asks for a different valid solution. Control's 53 owner matches give verification acceptance rates of 1.9385%, 1.6857%, and 0.6675% at $(q, t) = (16, 13)$, $(32, 23)$, and $(48, 33)$, respectively.

Table 6: Protected-problem outcomes by domain. (a) Valid answers / owner matches when models receive only the public problem. Base, Control, and WM answer the canonical protected prompts; the external models answer without a supplied WM solution. An owner match requires a valid answer equal to the owner solution. (b) Valid alternatives found when the actual WM answer is supplied; totals also give replacement success rates.

(a) Public problem only: valid answers / owner matches

	Trained models			External models		
Domain (problems)	Base	Control	WM (ours)	GPT-6 Sol	GPT-6 Luna	Claude Opus 5.5
X3C, $N = 60$ (70)	0 / 0	70 / 40	70 / 70	1 / 0	0 / 0	12 / 5
9×9 Kakuro (12)	0 / 0	12 / 5	12 / 12	4 / 1	0 / 0	10 / 4
96-vertex coloring (14)	0 / 0	14 / 8	14 / 14	0 / 0	0 / 0	4 / 1
Total (96)	0 / 0	96 / 53	96 / 96	5 / 1	0 / 0	26 / 10

(b) Different valid solutions from a supplied WM answer

Domain (problems)	GPT-6 Sol	GPT-6 Luna	Claude Opus 5.5
X3C, $N = 60$ (70)	0	0	5
9×9 Kakuro (12)	1	0	8
96-vertex coloring (14)	0	0	0
Total (96)	1 (1.04%)	0 (0%)	13 (13.54%)

D EXTERNAL-MODEL EVALUATION SETTINGS

The five external models are GPT-6 Sol, GPT-6 Luna, GPT-6 Astra, Claude Opus 5.5, and Claude Fable 5.1. All use medium reasoning effort, with no tools or verifier feedback. In first-solution evaluation, each receives the 96 public protected problems without either solution. In answer replacement, each additionally receives the actual valid canonical WM answer and is asked for a different valid solution; the owner bit and withheld solution stay private. GPT-6 Sol, GPT-6 Luna, and Claude Opus 5.5 use a 1,800-second timeout per problem and no explicit client-side output-token limit. GPT-6 Astra and Claude Fable 5.1 use a 32,000-token output limit and a 1,800-second scoring deadline. Interface failures are rerun and are not counted as mathematical failures.

GPT-6 Astra uses one mathematical attempt per problem. Claude Fable 5.1 first receives one attempt per problem; unsuccessful problems may receive one further attempt with the same prompt in a fresh session, without tools or feedback. First-solution evaluation retains the first valid answer, and answer replacement succeeds if either attempt finds a different valid solution. The reported Fable results are 34 valid first solutions, eleven owner matches, and thirteen successful replacements; four first-solution problems and five answer-replacement problems were evaluated with a single attempt. With one attempt only, Fable gives 20 valid first solutions with eight owner matches and eight successful replacements.

The verification thresholds in Table 1 are selected retrospectively to put the saved incorrect-key acceptance rate near 1% at each query budget. We reuse 10,000 fixed incorrect keys, compute the exact hypergeometric acceptance probability for each key, and average the probabilities. The thresholds are not calibrated on an independent key sample. The binomial bounds in Equation 3 average over independent random keys; the empirical Control and incorrect-key results describe their respective saved outputs and keys.

E OUTPUT-REWRITING PROTOCOL

Models and implementations. Table 2 uses KGW (Kirchenbauer et al., 2023), Unigram (Zhao et al., 2023), SIR (Liu et al., 2024), SWEET (Lee et al., 2024), EWD (Lu et al., 2024), and ReasonMark (Liu et al., 2026). The first five use the corresponding MarkLLM implementations (Pan et al., 2024); SIR's vocabulary mapping is regenerated for Qwen with a fixed seed. ReasonMark uses its released processor and detector. The text watermarks use Base Qwen3-8B on the first 96 GSM8K test problems, with temperature 0.6, top-p 0.95, top-k 20, seed 42, and an 8,192-token generation limit. Detection uses the answer after the closed thinking section. Ours uses the saved WM and matched Control answers on the 96 protected problems. In a native-prompt check on these protected problems, adding the five MarkLLM watermarks to Control leaves all 96 outputs unchanged; ReasonMark's processor does not activate because the expected thinking delimiter is absent. We therefore use GSM8K for the text-watermark rewriting comparison. This is a comparison across tasks, rather than a reproduction of the baseline papers' original evaluations.

Attacks. Word-D and Word-I delete and insert 30% of whitespace-separated words, respectively. Word-S applies WordNet synonym substitution at a rate of 50%. Paraphrasing and round-trip back-translation use the prompts from the ReasonMark implementation, executed with local Qwen2.5-7B-Instruct at temperature 0. This changes the language-model backend from the original implementation. Answer-only keeps the final answer in canonical form. The same transformations and settings are applied to positive and negative outputs. For ours, the compact JSON answer contains no whitespace, so Word-D treats the whole answer as one word and deletes it with probability 0.3. In the saved run, this leaves 19 of 96 WM outputs empty.

Scores and fixed thresholds. A trial samples 32 distinct problems from the relevant 96-problem set. Each text-watermark detector averages its per-answer scores over scoreable responses in that subset; empty or unscoreable responses contribute no score, and a subset with none is rejected. A small number of GSM8K generations never close their thinking section and yield no answer. For each text detector, the threshold is the 99th percentile of pooled scores on 96 separate, clean Base outputs from GSM8K test problems 97–192. It is then held fixed for all attacks. The achieved false-positive rate on evaluation negatives need not equal 1%. Ours uses the fixed rule of at least 24 owner matches among 32 outputs; invalid and empty outputs count as non-matches. Its negatives are the matched Control outputs, not incorrect keys. Thus, this table uses a different threshold and negative population from Table 1. Balanced accuracy gives equal weight to detecting watermarked subsets and rejecting negative subsets after the same attack. Values are rounded to one decimal place, and Avg. and Worst summarize the six attacked conditions.

Extracting solutions after rewriting. For this experiment, the parser first tries the structured answer and then extracts bundle IDs, vertex–color pairs, or cell–digit pairs directly from the text. It then checks all task constraints and equality to the owner solution. The same extractor is applied to WM and Control.

Separate fingerprint check. We also apply the rewrites to ten registered Chain&Hash replies (Russinovich et al., 2026) and check exact matches after whitespace normalization. All ten match before rewriting; none match after word insertion, synonym substitution, or paraphrasing, and four match after round-trip back-translation. These reply-level counts are separate from the $q = 32$ balanced-accuracy comparison.

F ADDITIONAL LIMITATIONS

Base model, keys, and training history. Our experiments use one base model, Qwen3-8B. The same keys were used across all training rounds, and later rounds were adjusted after inspecting earlier evaluation results, so the reported results do not use a key held out from these decisions. Results across other base models, independent keys, and training seeds remain to be measured.

Model coverage. False acceptance on unrelated models has been evaluated for GPT-6 Sol, GPT-6 Luna, GPT-6 Astra, Claude Opus 5.5, and Claude Fable 5.1; broader model coverage remains to be measured.

Presentation and transfer. The owner-match rate depends on the input presentation. Transfer also varies by domain and difficulty: WM scores below Control on unseen Kakuro, and Base, Control, and WM score zero on the stress profiles. Section 4 and Appendices A and C report these results.

Answer replacement. Answer-replacement results describe the screened problems, tested models, and inference budgets, and the average-case difficulty of our generated problems remains unproved. Larger search budgets remain to be evaluated, as do distillation and selective filtering of protected prompts.