ROUTERMAS: VALUE-PRESERVING WATERMARKING FOR MULTI-AGENT ROUTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-agent systems derive capability from how they delegate work, yet embedding an ownership signal into delegation can alter every decision that follows. Existing behavioral and expert-routing watermarks provide signature carriers without characterizing the routing freedom left by a system's required continuation utility and cost. We propose RouterMAS, a value-preserving watermarking framework for multi-agent routing. Our key insight is that fixing task values need not fix the routing distribution: Bellman recursion converts prefix-wise preservation into local conservation equations, exposing an admissible space of probability redistributions. RouterMAS first constructs this space from continuation values, then encodes a secret key through a minimum-information-change routing law, and finally verifies the signature using evidence induced by the same law. An error budget connects value estimation and deployed-policy approximation to accumulated return drift. Controlled finite-state experiments validate conservation, sequential verification, and algebraically inert controls. Across reasoning, code generation, and environment interaction, RouterMAS preserves task performance and operating cost while maintaining reliable detection. Its usable signature space expands with routing alternatives and contracts as additional task values are protected, connecting watermark strength to the structure of multi-agent orchestration.

1 INTRODUCTION

Language-model agents increasingly solve tasks by proposing, checking, revising, and executing one another's work. Role-based collaboration structures these interactions (Hong et al., 2024; Chen et al., 2024), while learned routing selects collaboration modes, roles, and constituent models (Yue et al., 2025). Consequently, two systems built from the same models can embody different orchestration policies. The routing policy is a system-level asset, and communication-topology inference exposes a corresponding provenance problem (Wu et al., 2026).

A routing watermark must change something that a verifier can observe, but delegation is also a functional decision. Recent orchestration work models state-conditioned agent selection as a sequential decision process (Dang et al., 2025). A delegation changes the message and history on which later choices, retries, and costs depend. We study a deployment contract that preserves expected task utility and operating cost after each admissible execution prefix. This contract prevents gains on easy prefixes from compensating for losses on difficult ones. It also makes a fixed secret path an inadequate starting point: the path may be identifiable precisely because it sends work where the original system would not.

Text watermarks bias generated tokens (Kirchenbauer et al., 2023); behavioral watermarks encode agent decisions (Huang et al., 2026); and MoE watermarks use internal expert paths (Gao et al., 2026). Equivalent-action and history-conditioned watermarks also exploit execution structure (Wang et al., 2026; An et al., 2026). Carrier-first transfers to MAS routing share an implicit assumption that a secret preference can be fixed before its continuation-value admissibility is established. The missing object is the admissible carrier itself. Figure 1 contrasts the closest behavioral and expert-routing carriers with the system-level MAS routing carrier studied here. An immediately harmless action need not be harmless downstream, and a history with uniquely determined routing probabilities leaves no room for a value-preserving intervention.

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

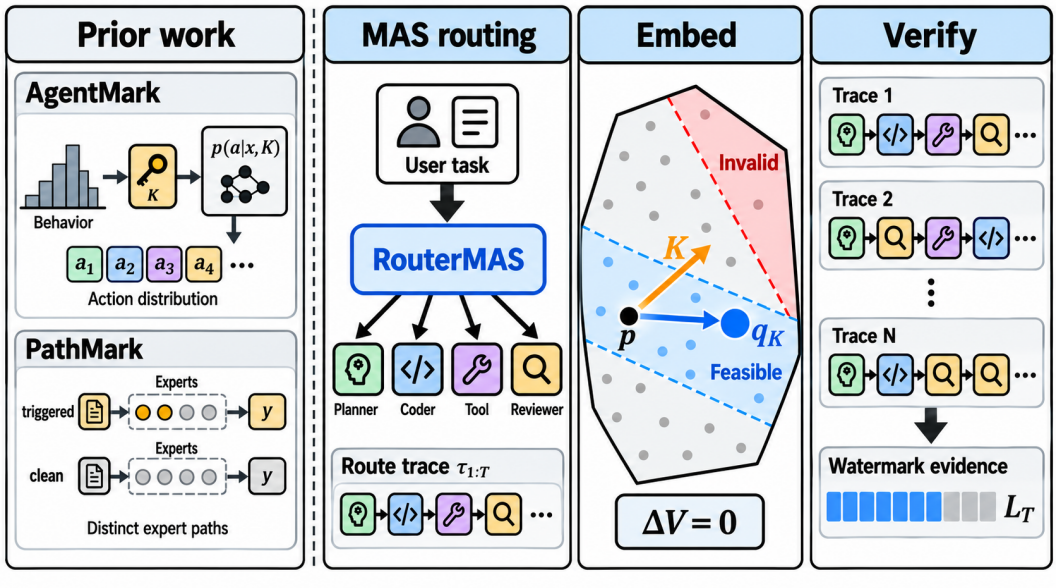

Figure 1: RouterMAS and prior carriers. AgentMark marks agent behavior; PathMark marks triggered MoE expert paths (Huang et al., 2026; Gao et al., 2026). RouterMAS marks the system-level MAS routing policy: a private key shifts p to q_K within the continuation-value-preserving region, and delegation traces accumulate likelihood-ratio evidence L_T.

Our preliminary audit (Figure 2) motivates continuation-aware conservation, a support-aware feasibility check, and key-directed allocation. Appendix E.3 reports its protocol and detailed observations.

The key is therefore not to choose a watermark route, but to identify which routing variations remain undetermined by the task values the system must preserve. Rather than enumerate variable-length execution paths, we optimize conditional routing probabilities. Bellman conservation reduces the whole-future preservation requirement to a local probability space. Within this space, the geometry selecting a least-change keyed update also determines its leading-order verification information. Figure 1 summarizes this progression from task requirements to a detectable routing signature.

We ask: *which multi-agent routing variations preserve required continuation values, and how can they carry a verifiable signature with controlled intervention and estimation cost?* RouterMAS has two stages: embedding constructs the value-preserving space and encodes the key within it; verification accumulates evidence from the resulting delegations. The formulation returns the original router when no admissible signal remains. An explicit error budget governs the estimated and deployed versions.

Our contributions are threefold: (1) we formulate system-level MAS router watermarking as probability redistribution under prefix-wise continuation-value preservation; (2) we derive the admissible carrier and connect its least-change encoder to sequential verification through the same local probability geometry; and (3) we demonstrate low-intervention detection across tasks and routing scales, and quantify signature persistence under router modification.

2 Related Work

LLM-based multi-agent systems. AgentVerse organizes specialized agents into task-solving collaborations (Chen et al., 2024). MetaGPT coordinates role-specialized agents through standard operating procedures (Hong et al., 2024). AutoGen provides programmable multi-agent conversations integrating language models, tools, and human input (Wu et al., 2024). GPTSwarm represents agent computation as graphs and optimizes their prompts and interconnections (Zhuge et al., 2024). Mixture-of-Agents aggregates responses across successive layers of language-model agents (Wang et al., 2025). AFlow searches executable agentic workflows through an automated optimization

process (Zhang et al., 2025b). These frameworks develop collaborative capability; identifying an independently verifiable signature of the orchestration policy is a separate system requirement.

Watermarking for language models and agents. KGW promotes keyed green-list tokens and detects their statistical excess (Kirchenbauer et al., 2023). Unbiased Watermark studies embedding schemes that preserve the relevant output distribution (Hu et al., 2024). GaussMark embeds structural signals in model weights and verifies them through Gaussian independence testing (Block et al., 2025). AgentMark embeds multi-bit behavioral signatures through distribution-preserving conditional sampling over planning decisions (Huang et al., 2026). AGENTWM biases distributions over semantically equivalent tool-execution segments to trace imitation (Wang et al., 2026). SeqWM encodes history-conditioned action transitions and uses position-agnostic, random-key-calibrated verification (An et al., 2026). ActHook inserts keyed hook actions into agent-trajectory data to detect downstream training reuse (Meng et al., 2026). PathMark embeds ownership signatures in triggered, cross-layer MoE expert routing (Gao et al., 2026). These methods establish behavioral and routing carriers; the remaining question here is which fixed-key orchestration changes preserve specified continuation values at every execution prefix.

Routing in multi-agent systems. RouteLLM learns cost-sensitive selection between language models from preference data (Ong et al., 2025). AgentPrune removes redundant spatial and temporal communication edges (Zhang et al., 2025a). MasRouter jointly routes collaboration modes, roles, and constituent language models (Yue et al., 2025). Evolving Orchestration formulates state-conditioned agent selection as sequential control and learns the orchestrator through reinforcement learning (Dang et al., 2025). Guided Topology Diffusion synthesizes task-conditioned communication graphs with reward-guided discrete diffusion (Jiang et al., 2026). CIA infers hidden communication topology from observable MAS behavior (Wu et al., 2026). Routing optimization and topology inference leave open how a protected router can carry a signature without changing its required downstream task values.

3 Preliminary Analysis

Figure 2 motivates three requirements. Immediate-value preservation can leave downstream utility and cost drift, requiring continuation-aware constraints. More legal delegations can expose feasible redistributions, whereas more protected returns remove directions. At matched probability-deviation budgets, these directions differ in key alignment, requiring key-directed allocation as well as a support-aware feasibility check. The audit protocol and detailed observations are in Appendix E.3; exact finite-state checks independently isolate the sequential effect in Appendix F.

4 Method

4.1 Problem Setup

Following MAS routing and state-conditioned orchestration (Yue et al., 2025; Dang et al., 2025), we consider a finite-horizon MAS with frozen constituent agents, tools, and conditional executor. At history h, the legal delegation set is $\mathcal{A}_h$ and the reference routing law is $p_h = p(\cdot \mid h)$, positive on its legal support. A delegation a specifies an agent and its current role or communication destination. The executor produces a response and the next history. The horizon H includes an absorbing termination state. For each of m protected returns, indexed by j, $V_j^p(h)$ is the remaining expected return under p, and $Q_j^p(h, a)$ is that return after taking a and then following p. Task success and total remaining agent-call cost supply the two principal returns. Our contract is $V_j^q(h) = V_j^p(h)$ for every admissible prefix and protected return.

The owner controls routing and retains the reference policy. A precommitted secret key K determines a bounded delegation score $s_{K,h}$. The marked law $q_{K,h}$ is distributed with the router; a verifier reads delegation histories and reconstructs the reference and marked conditional laws from frozen snapshots. A single K is fixed before test evidence is observed. The verification level α specifies the reference-null false-positive budget, and L_T denotes accumulated log evidence over T auditable decisions.

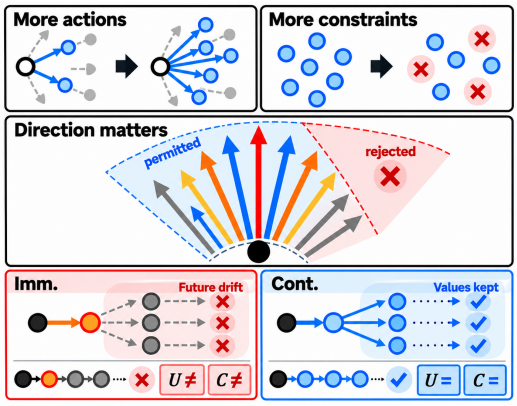

Figure 2: Continuation-aware routing structure. More legal delegations can expose routing freedom; more protected returns remove directions, and feasible directions provide unequal key-aligned signal. Immediate matching can change downstream utility or cost; continuation-aware constraints preserve the protected future values.

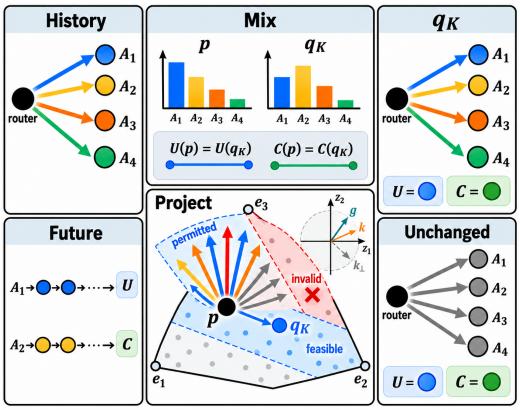

Figure 3: Value-preserving routing at a fixed history. The reference law p and marked law q_K mix legal delegations differently while preserving continuation utility and cost. Value-changing directions are excluded; q_K remains feasible, and histories without a usable direction retain the reference router.

4.2 Method Overview

Figure 1 summarizes embedding and verification. Algorithm 1 uses audit budget M to estimate continuation values $\widehat{Q}_h$ and their errors, then stacks normalization and value constraints into B_h and the admissible family $\mathcal{F}_h$. A keyed least-change law supplies the sampler; $\eta > 0$ controls its relative-entropy penalty and $\tau_{j,h}$ its local return-error allowance. Algorithm 2 accumulates evidence L_T from these conditional laws. Appendix A collects the notation.

4.3 Continuation Conservation Defines the Carrier

Lines 1–2 of Algorithm 1 establish the legal support and evaluate complete counterfactual continuations under the unchanged reference policy. Following a selected action to termination is essential: the quantity being preserved includes the future decisions that its response enables. Constrained policy optimization separates protected returns from the search objective (Achiam et al., 2017); requiring preservation at every prefix gives the following exact characterization.

Proposition 1 (Prefix-wise conservation)**.** *With a shared finite-horizon executor and terminal return convention, $V_j^q(h) = V_j^p(h)$ at every admissible history if and only if $(q_h - p_h)^\top Q_j^p(h) = 0$ at every history for each protected return.*

Backward induction proves sufficiency; equal successor values in the Bellman recursions prove necessity. Lines 3–4 of Algorithm 1 define the affine feasible family, whose exact-value local dimension is $|\mathcal{A}_h| - \operatorname{rank}(B_h)$. Lines 5–6 of Algorithm 1 construct its probability metric and projector. Figure 3 illustrates the permitted mixtures; Appendix B gives the proof and a four-action witness.

4.4 Least-Change Encoding Inside the Feasible Family

PathMark motivates routing as a signature channel (Gao et al., 2026); here the key acts inside the continuation-feasible family. For an attainable increase in expected key score, encoding selects the feasible law with minimum relative entropy to the baseline—an information projection (Csiszár, 1975). KL has a sequential interpretation: with a common initial-task law and shared executor, trajectory KL equals the marked-policy expectation of summed conditional routing KLs (Appendix C.1). The carrier itself is divergence-independent; another divergence can replace KL, while KL supplies this additive information accounting and an exponential-family encoder.

Algorithm 1 Value-preserving watermark embedding

Require: History h, reference router, frozen executor, protected returns, key K, audit budget M, weight $\eta > 0$, local budgets τ_j
History subscripts are suppressed; the audit follows the full reference router.
1: $(\mathcal{A}, p) \leftarrow \textsc{LegalRouting}(h, \text{reference router})$
2: $(\widehat{Q}, \epsilon) \leftarrow \textsc{ContinuationAudit}(h, \mathcal{A}, \text{reference router}, M)$
3: $B \leftarrow [\mathbf{1}^\top; (\widehat{Q}_1^p)^\top; \ldots; (\widehat{Q}_m^p)^\top]$
4: $\mathcal{F} \leftarrow \{q \geq 0 : B(q-p) = 0\}$
5: $D \leftarrow \mathrm{diag}(p)$
6: $P \leftarrow \textsc{OrthogonalProjector}(\ker(BD^{1/2}))$
7: $s_K \leftarrow \textsc{KeyScore}(K, h, \mathcal{A})$
8: $g \leftarrow D^{1/2} P D^{1/2} s_K$
9: **if** $g = 0$ or the audit error is unbounded **then return** $q_K = p$
10: $q_K^* \leftarrow \arg\max_{q \in \mathcal{F}} \{s_K^\top (q-p) - \eta D_{\mathrm{KL}}(q \| p)\}$
11: $\delta \leftarrow q_K^* - p$
12: $b_j \leftarrow |\delta^\top \widehat{Q}_j^p|$ for each protected return j
13: $\rho \leftarrow \min\{1, \min_j \tau_j / (b_j + \epsilon_j \|\delta\|_1)\}$
14: $q_K \leftarrow p + \rho\delta$
15: **return** q_K to the delegation sampler and evidence recorder

Lines 7–8 of Algorithm 1 form the key score and its probability-weighted feasible projection. The KL expansion equips probability changes with the Fisher metric underlying natural policy gradients (Kakade, 2001); its constrained projection yields the local direction $\eta^{-1} g_h$. Line 9 of Algorithm 1 returns the original router for full-rank constraints, row-space keys, or unbounded audit error. Fixed-history subscripts are suppressed inside this algorithm.

Line 10 of Algorithm 1 solves the scalarized local problem, consistent with relative-entropy policy updates (Schulman et al., 2015). Its unique optimum minimizes KL among feasible laws attaining at least its key-score gain; stationarity gives the exponential tilt derived in Appendix C. Lines 12–14 of Algorithm 1 then charge value-estimation and solver errors to the local budgets and shrink toward the baseline. This preserves feasibility and proportionally reduces the attained score gain; verification uses the final returned law.

For an implemented router $\widetilde{q}$, let J_j denote the initial-task expected return. With residual $b_{j,h} = |(\widetilde{q}_h - p_h)^\top \widehat{Q}_j^p(h)|$, suppose the continuation error is at most $\epsilon_{j,h}$ in maximum norm. The finite-horizon performance-difference identity gives

$$|J_j(\widetilde{q}) - J_j(p)| \leq \mathbb{E}_{\widetilde{q}} \sum_{t<H} \left[b_{j,h_t} + \epsilon_{j,h_t} \|\widetilde{q}_{h_t} - p_{h_t}\|_1 \right]. \tag{1}$$

The expectation uses the implemented policy's own visitation distribution, following the policy-comparison reasoning of Achiam et al. (2017). Equation (1) connects audit uncertainty, numerical residuals, and policy compilation to the same return budget. We compile the target routing law into a trainable controller and evaluate its realized residual separately from the exact sampler.

The projection used by the encoder also controls its local evidence. At a fixed history, write D, P, and g as in Algorithm 1.

Corollary 2 (Local embedding and evidence)**.** *For positive reference support and fixed constraints and key score,*

$$s_K^\top g = g^\top D^{-1} g = \left\| P D^{1/2} s_K \right\|^2. \tag{2}$$

The exact, unscaled optimizer satisfies, as $\eta \to \infty$,

$$s_K^\top (q_K^* - p) = \eta^{-1} s_K^\top g + O(\eta^{-2}), \tag{3}$$

$$\mathbb{E}_{a \sim q_K^*} \log \frac{q_K^*(a)}{p(a)} = \tfrac{1}{2} \eta^{-2} s_K^\top g + O(\eta^{-3}). \tag{4}$$

The same projected component therefore selects the local keyed update and determines its leading-order verification information. Zero projection makes both inert. Appendix C.4 derives the relation and its scaling for the final returned law.

Algorithm 2 Sequential routing-signature verification

Require: Auditable trace $\{(h_t, a_t)\}_{t<T}$, frozen reference/marked-law snapshots, precommitted K, level α

1: $L_0 \leftarrow 0$
2: **for** $t = 0, \ldots, T-1$ **do**
3: $(h_t, a_t) \leftarrow \textsc{ReadDelegation}(t)$
4: $(p_{h_t}, q_{K,h_t}) \leftarrow \textsc{ReconstructLaws}(h_t, K)$
5: $L_{t+1} \leftarrow L_t + \log\{q_{K,h_t}(a_t)/p_{h_t}(a_t)\}$
6: **if** $L_{t+1} \geq \log(1/\alpha)$ **then**
7: **return** detected signature
8: **end if**
9: **end for**
10: **return** no detected signature at budget T

4.5 Verification from the Marked Routing Law

Behavioral watermarking uses decision traces for attribution (Huang et al., 2026); likelihood-ratio watermark detection compares the marked and reference conditional laws (Hu et al., 2024). Lines 3–4 of Algorithm 2 recover the conditional laws for each observed delegation. Line 5 of Algorithm 2 accumulates their likelihood ratio, and Line 6 of Algorithm 2 permits stopping as soon as the evidence crosses the fixed threshold. Under the conditional-action null p, the likelihood-ratio process is a nonnegative martingale; Ville's inequality controls its running-maximum crossing probability by α (Howard et al., 2020). Under the marked law, the conditional expected log increment is $D_{\mathrm{KL}}(q_{K,h} \| p_h)$. This exact identity supplies the verification side of Corollary 2. Appendix D specifies the reference null and observation protocol.

5 Experiments

5.1 Experimental Settings

Tasks and systems. We use GSM8K for mathematical reasoning, HumanEval and MBPP for code generation, and ALFWorld for environment interaction (Cobbe et al., 2021; Chen et al., 2021; Austin et al., 2021; Shridhar et al., 2021). The primary system uses six role-specialized Qwen2.5-7B-Instruct delegates (Qwen Team, 2024) and an eight-delegation horizon. We run inference on four NVIDIA A100-80GB GPUs and the routing solver on a CPU. Centralized, layered, and graph-routed configurations share the same frozen executor within each comparison. Llama-3.1-8B-Instruct (Grattafiori et al., 2024) supplies a second model family. Appendix E specifies the configurations.

Baselines. The matched comparison includes unmarked routing, delegation-level red/green bias (Kirchenbauer et al., 2023), an AgentMark-F delegation adaptation, a PathMark-inspired forced inter-agent path, and RouterMAS. AgentMark and PathMark supply the behavioral and routing mechanisms (Huang et al., 2026; Gao et al., 2026). The mechanism study additionally compares immediate-value projection, a global-average penalty, and random feasible redistribution. The original published MAS, workflow, MoE, and trajectory-data results remain separate source-consistent panels in Appendix G.

Metrics and implementation. Functionality uses task success or pass@1, all-episode modeled cost normalized to the unmarked system, and $D_{U,95}$, the 95th-percentile absolute continuation-utility drift in percentage points. Detection uses TPR at a calibrated FPR and records auditable delegation count. We use $T \in \{32, 64, 128, 256\}$ with primary $\alpha = 0.01$, three execution seeds, and ten precommitted keys. Main estimated-value results use 128 continuation rollouts per audited action; the pilot uses 32. Controller fitting, critic fitting, threshold calibration, and testing are disjoint by root task. All counterfactual branches remain with their root task. Constituent generation uses temperature 0.7; the encoder uses $\eta = 1.5$ and rank tolerance 10^{-8}. HumanEval and MBPP execute in isolated sandboxes with identical time limits. The value estimator predicts success and remaining normalized agent-call cost. Comparisons use the same action alphabet, executor, observation interface, and decision budget. Appendix D defines the calibration protocol.

Table 1: Mechanism attribution under a matched evidence budget.

Method	TPR (%)	$D_{U,95}$ (pp)	$\lvert\Delta C\rvert$ (%)
Unprojected	98.5	4.9	8.6
Global avg.	97.1	4.8	1.8
Immediate	96.5	3.6	2.7
Random feasible	67.6	0.8	0.5
Exact-Q	96.8	0.0	0.0
Estimated-Q	96.8	0.7	0.4
Compiled	95.2	1.0	0.9

GSM8K; $T = 256$, $\alpha = 0.01$. Cost is absolute relative all-episode cost change.

Table 2: Local embedding–verification check.

Audit	η	R_S	R_I	E_S	E_I
Finite-state	8	0.9979	0.9968	8.61	11.41
Finite-state	32	0.9998	0.9998	2.24	2.98
Finite-state	128	1.0000	1.0000	0.56	0.75
GSM8K	8	0.9954	0.9878	9.0	14.5
GSM8K	32	0.9983	0.9980	3.1	3.9
GSM8K	128	0.9995	0.9994	0.7	1.1

R_S, R_I: exact/leading-order mean ratios; E_S, E_I: worst relative errors (%).

5.2 Routing Freedom

Figures 5 and 6 show that additional legal delegations expand usable marking freedom, while additional protected returns restrict it. Active marking depends on more than algebraic nullity: support restrictions, key alignment, and value-estimation uncertainty determine whether a feasible direction carries a usable signature. Appendix B derives the dimension formula and the conditions under which keys become inert.

5.3 Mechanism Analysis

The executed finite-state check in Appendix F preserves full continuation values to numerical precision while changing the routing law. Immediate-value constraints leave downstream drift, whereas full-rank constraints and row-space keys are inert. These controls isolate the conservation mechanism with known transitions. Appendix Figure 7 gives the corresponding geometric view; the quantitative mechanism comparison remains in Table 1.

Table 1 shows why feasibility and key alignment are both necessary. Unprojected bias sacrifices continuation values, global-average penalties permit prefix-specific losses, and immediate-value constraints miss downstream effects. Random feasible directions preserve values but carry weak key evidence. Targeted feasible encoding addresses both requirements; the compiled controller's additional drift is consistent with the deployment residual in Equation (1).

Embedding–verification relation. We freeze all 20 time–state prefixes and the key in the same finite-state system, sweep η, and compare the exact encoder with Corollary 2. Table 2 shows both normalized coefficients approaching one, with decreasing worst-prefix error. Halving the final update halves its score gain and approximately quarters its mean log evidence; zero-projection controls remain inert. Thus the same projected component explains the local update and its verification information, rather than two independently tuned signals. Appendix F.1 reports the replay and scaling checks. The GSM8K comparison in Table 2 exhibits the same small-update pattern on frozen prefixes, supporting the local geometric relation under estimated values. Task utility and fixed-FPR detection require the separate matched evaluation below.

5.4 Watermark Effectiveness

Table 3 shows that RouterMAS retains detectable routing signatures while keeping task success, operating cost, and continuation values close to the unmarked system. Forced-path marking strengthens the signature at greater functional cost. The AgentMark delegation adaptation largely preserves aggregate utility but leaves more prefix drift, indicating that aggregate performance alone can hide changes to continuation behavior. Together with Table 1, this comparison supports explicit prefix-wise constraints when both signature evidence and functionality matter.

5.5 Estimation and Scalability

Figure 4 compares absolute expected-utility change with the budget in Equation (1), separately from the prefix quantile $D_{U,95}$. Table 4 shows high-budget detection gains leveling off and relatively stable solve times, which exclude counterfactual sampling. The structural trends in Figures 5 and 6 impose a separate limit: additional estimation effort cannot create feasible routing freedom when the protected returns already determine the routing law. Section 5.8 reports transfer across router layouts and model families.

Table 3: Task utility, cost, and routing-signature detection under the common delegation protocol.

Task	Method	SR (%)	Cost / base	Detection (%)	$D_{U,95}$ (pp)
GSM8K	Unmarked router	94.2	1.000	0.9	0.0
	RG-action adaptation	94.0	1.086	98.5	4.9
	AgentMark-F (delegation)	94.3	1.011	99.0	1.6
	Forced path (PathMark-inspired)	90.6	1.225	99.0	9.3
	RouterMAS	94.2	1.004	96.8	0.7
HumanEval	Unmarked router	87.9	1.000	0.4	0.0
	RG-action adaptation	87.1	1.076	98.5	6.3
	AgentMark-F (delegation)	87.9	1.012	98.6	1.8
	Forced path (PathMark-inspired)	84.8	1.207	98.8	8.6
	RouterMAS	87.6	1.006	96.5	0.8
MBPP	Unmarked router	76.8	1.000	0.6	0.0
	RG-action adaptation	76.1	1.067	98.0	5.9
	AgentMark-F (delegation)	76.3	1.008	98.2	1.6
	Forced path (PathMark-inspired)	73.3	1.206	99.1	7.4
	RouterMAS	76.5	1.005	96.0	0.9
ALFWorld-OOD	Unmarked router	71.8	1.000	0.5	0.0
	RG-action adaptation	68.4	1.092	98.1	8.1
	AgentMark-F (delegation)	71.6	1.020	98.0	2.5
	Forced path (PathMark-inspired)	65.7	1.254	99.4	12.2
	RouterMAS	71.2	1.009	95.6	1.1

$T = 256$ and nominal $\alpha = 0.01$. Detection is the false-positive rate for unmarked rows and the true-positive rate for marked rows. Cost / base is normalized to the corresponding unmarked system and includes failed requests. RG, AgentMark-F, and forced-path rows use delegation-level adaptations.

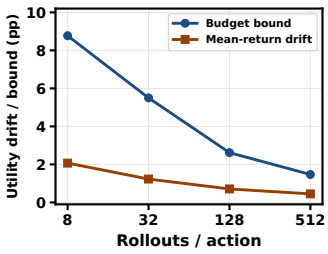

Figure 4: Continuation-audit scaling. The absolute expected-utility change and its accumulated error budget decrease as continuation rollouts per action increase.

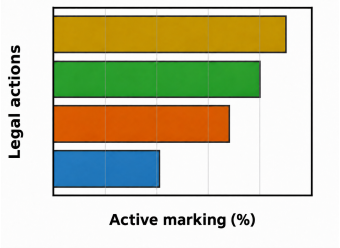

Figure 5: Usable routing freedom versus legal action count. With two protected returns fixed, active marking increases as additional legal delegations become available.

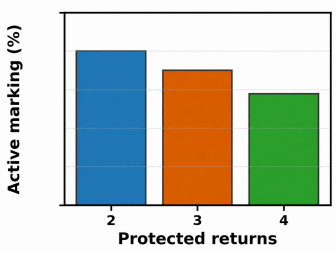

Figure 6: Usable routing freedom versus preservation constraints. With eight legal actions fixed, active marking decreases as additional continuation returns are protected.

5.6 Watermark Persistence

Table 5 shows that the signature survives copying and reversible relabeling when the verifier knows the inverse map. Deterministic decoding, fine-tuning, imitation, and value-preserving erasure weaken detection by changing the conditional routing law. Thus persistence depends on how much of that law survives modification, even when aggregate functionality is retained. Key-aware erasure additionally exposes the private statistic to the attacker, so its result characterizes a stronger access model.

5.7 Sequential Detection

Table 6 evaluates the running-maximum detector in the executed finite-state system. Longer traces increase detection power, while empirical false-positive rates remain below their nominal levels. Stricter thresholds reduce short-trace power, which additional evidence recovers. With exact conditional laws, this comparison isolates sequential evidence accumulation from value-estimation and controller errors. Together with Appendix F, it verifies that continuation conservation and detectable routing changes can coexist.

Table 4: Continuation-estimation budget, detection, and online solve time.

Rollouts	ϵ_U	TPR (%)	Solver ms
		92.4	2.89
8	0.137	± 0.11	± 0.10
		94.2	2.86
32	0.086	± 0.12	± 0.15
		96.8	3.18
128	0.041	± 0.12	± 0.10
		96.6	3.18
512	0.023	± 0.14	± 0.18

Bound and mean-return drift appear in Figure 4. Solver time excludes audit rollouts, LLM inference, and verification.

Table 5: Routing-artifact persistence after modification.

Condition	Det. (%)	SR (%)	Cost/base
Copy	95.2	94.1	1.009
Deterministic	58.7	93.6	0.962
FT-5k	72.0	94.0	1.018
Imitation-5k	48.3	93.8	1.023
Blind erase	32.2	94.0	1.010
Aware erase	7.8	93.9	1.011
Perm.+inv.	95.2	94.1	1.009

Compiled RouterMAS on GSM8K, with the same private detector and 256-decision budget. Key-aware erasure has stronger attacker access.

Independent trace groups repeat episodes under each frozen law and a precommitted key. Evidence accumulates across episode boundaries, with rejection at any threshold crossing before the delegation budget. Holding the encoder fixed isolates the effect of observation length.

Table 6: Executed sequential verification in the finite-state system.

Decisions	FPR $\alpha = .01$	TPR $\alpha = .01$	FPR $\alpha = .001$	TPR $\alpha = .001$
32	0.36	50.81	0.02	18.88
64	0.67	88.24	0.03	69.73
128	0.75	99.30	0.04	97.46
256	0.75	100.00	0.05	99.99

Percentages over 10,000 independent trace groups per law, one precommitted key, and repeated four-step episodes. Rejection uses the running maximum of log evidence.

5.8 Cross-System Generalization

Table 7 compares router layouts and constituent model families. Marking keeps success and cost close to each baseline while retaining detectable signatures. Layered routing restricts actions by stage; graph routing adds communication destinations to action labels. The consistent pattern supports applying the conservation construction to different local action spaces and model families.

Each comparison freezes its reference controller and executor. Preservation is therefore relative to that system's own baseline: the construction is reapplied to each system, without assuming that continuation estimates or a marked controller transfer unchanged between executors.

Table 7: Cross-router and cross-model functionality and detection on GSM8K.

Model	Router	Base SR	Marked SR	Cost/base	TPR
Qwen2.5-7B	Centralized	94.2	94.2	1.004	96.8
Qwen2.5-7B	Layered	93.9	94.0	1.008	95.9
Qwen2.5-7B	Graph-routed	94.7	94.4	1.009	97.1
Llama-3.1-8B	Centralized	91.0	90.8	1.005	95.9
Llama-3.1-8B	Layered	90.2	90.1	1.010	94.9
Llama-3.1-8B	Graph-routed	91.2	91.1	1.011	96.0

$T = 256$, nominal $\alpha = 0.01$; success and TPR are percentages. Each row pairs one frozen system with its marked version.

6 Conclusion

RouterMAS formulates multi-agent watermarking in the routing freedom left by required continuation values. Bellman conservation defines the carrier, relative entropy determines its keyed redistribution, and the resulting law provides sequential evidence. Experiments connect usable routing freedom to detection, functionality, and modification cost across multi-agent settings. This perspective makes the task-preserving degrees of freedom of an orchestration policy an explicit object of provenance design.

AI USE STATEMENT

Generative AI tools were used only to polish the presentation of this paper, namely to edit wording, grammar, and LaTeX layout and to refine figures. They were not used for research ideation, theoretical derivations, experiment design, implementation, data analysis, or the generation of results or scientific claims. All AI-assisted edits were reviewed by the authors, who take full responsibility for the content of this paper.

REFERENCES

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *International Conference on Machine Learning*, pp. 22–31, 2017.

Hyeseon An, Shinwoo Park, Dongsu Kim, and Yo-Sub Han. Sequential behavioral watermarking for LLM agents. arXiv:2605.11036, 2026.

Jacob Austin et al. Program synthesis with large language models. arXiv:2108.07732, 2021.

Adam Block, Alexander Rakhlin, and Ayush Sekhari. GaussMark: A practical approach for structural watermarking of language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 4507–4569. PMLR, 2025.

Mark Chen et al. Evaluating large language models trained on code. arXiv:2107.03374, 2021.

Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chi-Min Chan, Heyang Yu, Yaxi Lu, Yi-Hsin Hung, Chen Qian, Yujia Qin, Xin Cong, Ruobing Xie, Zhiyuan Liu, Maosong Sun, and Jie Zhou. AgentVerse: Facilitating multi-agent collaboration and exploring emergent behaviors. In *International Conference on Learning Representations*, 2024.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. arXiv:2110.14168, 2021.

Imre Csiszár. I-divergence geometry of probability distributions and minimization problems. *The Annals of Probability*, 3(1):146–158, 1975. doi: 10.1214/aop/1176996454.

Yufan Dang, Chen Qian, Xueheng Luo, Jingru Fan, Zihao Xie, Ruijie Shi, Weize Chen, Cheng Yang, Xiaoyin Che, Ye Tian, Xuantang Xiong, Lei Han, Zhiyuan Liu, and Maosong Sun. Multi-agent collaboration via evolving orchestration. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 165025–165059. Curran Associates, Inc., 2025. doi: 10.52202/085713-5502.

Yudong Gao, Qingyue Wang, Yuanyuan Yuan, Ruixuan Huang, Linghan Chen, Zimo Ji, and Shuai Wang. PathMark: Protecting intellectual property of mixture-of-expert LLMs via path watermarks. In *ACM Conference on Computer and Communications Security*, 2026.

Aaron Grattafiori et al. The llama 3 herd of models. arXiv:2407.21783, 2024.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. MetaGPT: Meta programming for a multi-agent collaborative framework. In *International Conference on Learning Representations*, 2024.

Steven R. Howard, Aaditya Ramdas, Jon McAuliffe, and Jasjeet Sekhon. Time-uniform chernoff bounds via nonnegative supermartingales. *Probability Surveys*, 17:257–317, 2020.

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

Kaibo Huang, Jin Tan, Yukun Wei, Wanling Li, Zipei Zhang, Hui Tian, Zhongliang Yang, and Linna Zhou. AgentMark: Utility-preserving behavioral watermarking for agents. In *Annual Meeting of the Association for Computational Linguistics*, 2026.

Eric Hanchen Jiang, Levina Li, Frank Wan, Xiao Liang, Sophia Yin, Yuchen Wu, Xinfeng Li, Yizhou Sun, Wei Wang, Kai-Wei Chang, and Ying Nian Wu. Dynamic generation of multi LLM agents communication topologies with graph diffusion models. In *Annual Meeting of the Association for Computational Linguistics*, pp. 38042–38060, 2026.

Sham M. Kakade. A natural policy gradient. In *Advances in Neural Information Processing Systems*, volume 14, 2001.

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, 2023.

Wenlong Meng, Chen Gong, Terry Yue Zhuo, Fan Zhang, Kecen Li, Zheng Liu, Zhou Yang, Chengkun Wei, and Wenzhi Chen. Watermarking LLM agent trajectories. In *International Conference on Machine Learning*, 2026.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M. Kadous, and Ion Stoica. RouteLLM: Learning to route LLMs from preference data. In *International Conference on Learning Representations*, 2025.

Qwen Team. Qwen2.5 technical report. arXiv:2412.15115, 2024.

John Schulman, Sergey Levine, Philipp Moritz, Michael I. Jordan, and Pieter Abbeel. Trust region policy optimization. In *International Conference on Machine Learning*, pp. 1889–1897, 2015.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. ALFWorld: Aligning text and embodied environments for interactive learning. In *International Conference on Learning Representations*, 2021.

Junlin Wang, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Zou. Mixture-of-agents enhances large language model capabilities. In *International Conference on Learning Representations*, 2025.

Liwen Wang, Zongjie Li, Yuchong Xie, Shuai Wang, Dongdong She, Wei Wang, and Juergen Rahmel. On protecting agentic systems' intellectual property via watermarking. arXiv:2602.08401, 2026.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W. White, Doug Burger, and Chi Wang. AutoGen: Enabling next-gen LLM applications via multi-agent conversation. In *Conference on Language Modeling*, 2024.

Yongxuan Wu, Xixun Lin, He Zhang, Nan Sun, Kun Wang, Chuan Zhou, Shirui Pan, and Yanan Cao. CIA: Inferring the communication topology from LLM-based multi-agent systems. In *Annual Meeting of the Association for Computational Linguistics*, pp. 17917–17933, 2026.

Yanwei Yue, Guibin Zhang, Boyang Liu, Guancheng Wan, Kun Wang, Dawei Cheng, and Yiyan Qi. MasRouter: Learning to route LLMs for multi-agent systems. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15549–15572, Vienna, Austria, July 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-long.757.

Guibin Zhang, Yanwei Yue, Zhixun Li, Sukwon Yun, Guancheng Wan, Kun Wang, Dawei Cheng, Jeffrey Yu, and Tianlong Chen. Cut the crap: An economical communication pipeline for LLM-based multi-agent systems. In *International Conference on Learning Representations*, 2025a.

Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xionghui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, Bingnan Zheng, Bang Liu, Yuyu Luo, and Chenglin Wu. AFlow: Automating agentic workflow generation. In *International Conference on Learning Representations*, 2025b.

Mingchen Zhuge, Wenyi Wang, Louis Kirsch, Francesco Faccio, Dmitrii Khizbullin, and Jürgen Schmidhuber. GPTSwarm: Language agents as optimizable graphs. In *International Conference on Machine Learning*, pp. 62743–62767, 2024.

SUMMARY OF APPENDIX

Section	Supplementary material
A	Notation and execution interface
B	Value conservation, feasible freedom, and inert controls
C	Exponential-family solver and deployment error
D	Sequential verification and evaluation metrics
E	Model, task, audit, and attack settings
F	Executed finite-state mechanism and embedding checks
G	Reproduced List
H	AI-use statement

A NOTATION AND EXECUTION INTERFACE

At a fixed history, we suppress history subscripts: p, q_K, and the encoder intermediates denote their values at that history. True continuation values and their estimates retain distinct notation. The LEGALROUTING operation returns the legal positive-probability support and the reference conditional law on that support; CONTINUATIONAUDIT retains the full reference policy for subsequent decisions.

Symbol	Meaning
$h, \mathcal{A}_h, a$	History, legal baseline-supported delegation set, and selected delegation.
n, m	Number of legal actions at a fixed history and number of protected returns.
$p_h, q_{K,h}, \widetilde{q}_h$	Reference law, returned marked law, and implemented law.
$q^*_{K,h}$	Keyed optimizer before safe scaling.
H, T	Per-request execution horizon and verification decision budget across requests.
$Q^p_j, V^p_j, J_j(p)$	Reference action- and history-continuation values, and initial-task expected return.
$\widehat{Q}_h, \epsilon_{j,h}, M$	Continuation estimates, maximum-norm error bound for return j, and continuation rollouts per audited action.
$B_h, \mathcal{F}_h$	Normalization-augmented conservation matrix and feasible family built from the estimated continuation values.
D_h, P_h	$D_h = \mathrm{diag}(p_h)$ and the orthogonal projector onto $\ker(B_h D_h^{1/2})$; the Fisher metric is D_h^{-1}.
$K, s_{K,h}, \eta$	Precommitted key, bounded action score, and relative-entropy penalty weight.
δ_h, g_h, ρ_h	Proposed change $q^*_{K,h} - p_h$, local direction $D_h^{1/2} P_h D_h^{1/2} s_{K,h}$, and safe-scaling factor.
$\tau_{j,h}, b_{j,h}$	Local return-error allowance and implemented-law conservation residual.
L_T, α	Accumulated log-likelihood evidence and reference-null test level.

For exact-value results, B_h and $\mathcal{F}_h$ use Q^p_j in place of $\widehat{Q}^p_j$. In Algorithm 1, $b_j = |\delta^\top \widehat{Q}^p_j|$ is the proposal residual before scaling; in Equation (1), $b_{j,h}$ measures the implemented law. The returned law is $q_{K,h} = p_h + \rho_h \delta_h$.

An executor response includes the selected agent output, tool result, completion status, and consumed cost. The history retains these outputs and all prior delegation labels. Agent prompts, tools, and their response kernels are fixed within a preservation experiment. Early termination enters an absorbing state with zero subsequent reward. A terminal success reward is charged to the last nonabsorbing transition. The legal support contains exactly actions with positive baseline probability and satisfied execution prerequisites.

The owner retains the baseline controller, key, value-estimation snapshot, and calibration manifest. The auditable interface records the history needed to evaluate both conditional laws. Key scores are fixed independently of verification traces. The implementation records raw delegation counts separately from the number of independent root tasks. Numerical rank uses a documented singular-value tolerance; changing that tolerance is a sensitivity experiment rather than changing the protected-return contract.

B Value Conservation and Feasible Freedom

B.1 Proof of Proposition 1

Let $r_j(h, a, o)$ be the one-step contribution to protected return j after executor response o. The baseline Bellman equations are

$$Q_j^p(h, a) = \mathbb{E}[r_j(h, a, o) + V_j^p(h') \mid h, a], \tag{5}$$

$$V_j^p(h) = \sum_{a \in \mathcal{A}_h} p_h(a) Q_j^p(h, a). \tag{6}$$

Suppose local conservation holds at every admissible history. At depth $H - 1$, the shared terminal convention implies $Q_j^q = Q_j^p$, so conservation gives $V_j^q = V_j^p$. Assume the equality holds at all histories at depth $t + 1$. The same conditional executor implies equality of Q_j^q and Q_j^p at depth t. Local conservation again yields equality of history values. Backward induction proves sufficiency.

Conversely, equality of continuation values at every history includes every successor history. The shared executor therefore yields $Q_j^q(h, a) = Q_j^p(h, a)$. Subtracting the two Bellman recursions gives the local conservation equation. Requiring every admissible prefix is the condition that makes this characterization necessary as well as sufficient.

B.2 Local dimension and inert keys

On a legal support of size n with strictly positive baseline probabilities, take a nonzero $d \in \ker B$. Both $p + td$ and $p - td$ remain nonnegative for sufficiently small t. Consequently the local feasible dimension is $n - \operatorname{rank} B$. Full column rank leaves only p. If a key score satisfies $s = B^\top v$, then $s^\top(q - p) = v^\top B(q - p) = 0$ for every feasible q. The keyed objective reduces to $-\eta D_{\mathrm{KL}}(q\|p)$ and has the unique maximizer p. These two cases justify the exact no-marking branch in Algorithm 1.

B.3 Four-action constructive witness

Let $p = (0.25, 0.25, 0.25, 0.25)$, $Q_U = (1, 0.8, 0.6, 0.4)$, and $Q_C = (1, 2, 2, 1)$. The direction $d = (1, -3, 3, -1)$ conserves normalization and both values. The law $q = p + 0.02d = (0.27, 0.19, 0.31, 0.23)$ has unchanged utility 0.7 and cost 1.5, while the score $s = (1, -1, 1, -1)$ gains 0.16 in expectation. Its divergence is $D_{\mathrm{KL}}(q\|p) = 0.01614323804869158$ nats. This is an exact constructed example of the probability freedom used by the method.

C Optimization and Deployment Error

C.1 Sequential information and local information projection

Let $\mathbb{P}^q$ and $\mathbb{P}^p$ be complete-trajectory laws with the same initial-task distribution and frozen conditional executor. A trajectory z records its histories, delegations, and executor outputs, with absorbing termination up to horizon H. Since executor factors cancel in the likelihood ratio,

$$\log \frac{d\mathbb{P}^q}{d\mathbb{P}^p}(z) = \sum_{t<H} \log \frac{q_{K,h_t}(a_t)}{p_{h_t}(a_t)}, \tag{7}$$

$$D_{\mathrm{KL}}(\mathbb{P}^q\|\mathbb{P}^p) = \mathbb{E}_q \sum_{t<H} D_{\mathrm{KL}}(q_{K,h_t}\|p_{h_t}) = \mathbb{E}_q L_H. \tag{8}$$

Here L_H is the log evidence along one request, padded to horizon H by absorbing transitions. Positive reference support makes the ratio well-defined on marked trajectories. The chain rule accounts for the histories visited by the marked policy. It motivates a conditional KL penalty; Algorithm 1 solves a local distribution-selection problem at each history, with its visitation distribution induced by execution.

At a fixed history, suppress history subscripts. For an attainable key-score increment $\gamma \geq 0$, the information projection (Csiszár, 1975) is

$$q_\gamma = \operatorname*{arg\,min}_{q \in \mathcal{F}:\ s_K^\top(q-p) \geq \gamma} D_{\mathrm{KL}}(q\|p). \tag{9}$$

Here γ constrains a linear score moment, not a detection probability. The feasible set is compact and convex on finite positive reference support; KL is continuous and strictly convex, so a feasible target has a unique minimizer. A target above the maximal feasible score gain is unattainable. At zero gain the minimizer is p; for a row-space key or full-column-rank constraints, zero is the only attainable gain.

The scalarized encoder has the same interpretation without adding a new runtime parameter. Let q_K^* be the optimizer in Line 10 of Algorithm 1 and set $\gamma = s_K^\top(q_K^* - p)$. For every feasible q attaining at least this gain, scalarized optimality gives

$$\eta\big[D_{\mathrm{KL}}(q\|p) - D_{\mathrm{KL}}(q_K^*\|p)\big] \geq s_K^\top(q-p) - \gamma \geq 0. \tag{10}$$

Thus q_K^* is precisely the information projection for its attained gain. Conversely, an interior positive target with relative strict feasibility has an active score multiplier $\beta > 0$; dividing its Lagrangian by β gives the scalarized encoder with $\eta = 1/\beta$. At the maximal target, a finite multiplier need not exist. The subsequent exponential-family derivation applies directly to the implemented finite-η scalarization.

Safe scaling is applied after that local optimum. If its factor is ρ, the returned law has score gain $\rho\gamma$ and KL at most $\rho D_{\mathrm{KL}}(q_K^*\|p)$ by convexity. Both preservation accounting and verification use this returned law. Equation (8) links it to expected log evidence; finite-budget detection power remains a property of the evidence distribution. Other divergences leave the Bellman-defined carrier unchanged, while changing the selection geometry and solver.

C.2 Exponential-family solution

For fixed h, suppress history subscripts. The Lagrangian of the keyed objective, with normalization multiplier μ and value multipliers λ_j, gives

$$s_a - \eta[\log(q_a/p_a) + 1] - \mu - \sum_j \lambda_j \widehat{Q}_{j,a} = 0. \tag{11}$$

Collecting scalar terms produces

$$q_a(\lambda) = \frac{p_a}{Z(\lambda)} \exp\left(\frac{s_a - \sum_j \lambda_j \widehat{Q}_{j,a}}{\eta}\right). \tag{12}$$

The strictly concave primal objective has a unique feasible optimum. A convex dual is

$$\Phi(\lambda) = \eta \log \sum_a p_a \exp\left(\frac{s_a - \lambda^\top \widehat{Q}_a}{\eta}\right) + \lambda^\top \widehat{Q} p. \tag{13}$$

Its gradient is $\widehat{Q}p - \widehat{Q}q(\lambda)$; its Hessian is $\eta^{-1}\,\mathrm{Cov}_{q(\lambda)}(\widehat{Q}_a)$. The implementation removes dependent constraint rows, optimizes the remaining dual variables by BFGS, and checks the primal conservation residual. A small residual is corrected in the feasible direction basis with a nonnegativity check. The final returned law, rather than the intended target alone, is used in residual accounting.

C.3 Probability-weighted local projection

Write $q = p + \delta$ and $D = \mathrm{diag}(p)$. On positive support,

$$D_{\mathrm{KL}}(p + \delta\|p) = \tfrac{1}{2}\delta^\top D^{-1}\delta + O(\|\delta\|^3). \tag{14}$$

Set $u = D^{-1/2}\delta$. The quadratic problem maximizes $(D^{1/2}s)^\top u - \eta\,\|u\|^2/2$ subject to $BD^{1/2}u = 0$. Orthogonal projection onto this kernel gives the local direction used in Algorithm 1. The finite-state numerical solver uses Equation (12) for finite updates; the projected direction identifies inert keys and the local evidence geometry.

C.4 Proof of Corollary 2

Fix a history and suppress its subscript. Since P is an orthogonal projector, $P^\top = P = P^2$. Substituting the encoder's g into the three expressions in Equation (2) gives the same quadratic form

$s_K^\top D^{1/2} P D^{1/2} s_K$. Also $Bg = 0$, so this direction conserves normalization and the chosen value constraints. With estimated values, these are the estimated constraints; true-value error remains governed by Equation (1).

On positive support the categorical Fisher metric is D^{-1} restricted to zero-sum probability changes, the distribution-space geometry used in natural policy gradients (Kakade, 2001). As $\eta \to \infty$, the exact scalarized optimizer approaches p. Restricted to the affine feasible tangent space, KL has a positive-definite Hessian there. Expanding its stationarity condition in the whitened coordinates of the preceding subsection gives

$$q_K^* - p = \eta^{-1} g + O(\eta^{-2}). \tag{15}$$

Multiplying by the key score proves Equation (3). Substituting into the KL expansion and using the shared quadratic form proves Equation (4), because expected log likelihood ratio equals KL. All expansions hold for fixed positive p, B, and s_K; constants can depend on these quantities.

For the final update $q_K = p + \rho(q_K^* - p)$, with $0 \le \rho \le 1$, the leading score term is multiplied by ρ and the leading expected log-evidence term by ρ^2. The remainders are respectively $O(\rho\eta^{-2})$ and $O(\rho^2\eta^{-3})$. The exact finite update and its returned-law likelihood are still used by Algorithms 1 and 2. The relation describes local mean information; finite-budget detection also depends on its fluctuations and the visited histories.

C.5 Accumulated error and safe scaling

With $A_j^p(h,a) = Q_j^p(h,a) - V_j^p(h)$, telescoping baseline values along the trajectory of any implemented law $\widetilde{q}$ with the same executor gives

$$J_j(\widetilde{q}) - J_j(p) = \mathbb{E}_{\widetilde{q}} \sum_{t<H} A_j^p(h_t, a_t) \tag{16}$$

$$= \mathbb{E}_{\widetilde{q}} \sum_{t<H} (\widetilde{q}_{h_t} - p_{h_t})^\top Q_j^p(h_t). \tag{17}$$

Adding and subtracting $\widehat{Q}_j^p$ inside each term and applying the ℓ_1–ℓ_∞ bound proves Equation (1). If the per-history error allowance bounds each bracketed term, their sum bounds the expected return difference. For a feasible proposal $p + \delta$, the segment $p + \rho\delta$ remains nonnegative and conserves estimated values for $0 \le \rho \le 1$. For a numerical proposal with conservation residual, that residual and its uncertainty charge both scale by ρ, which yields Line 13 of Algorithm 1. Terms with zero denominator are assigned an infinite permissible ratio; if all terms are zero, $\rho = 1$.

An error budget is evaluated on histories reached by the implemented policy. Simultaneous statistical coverage over those histories is specified together with any confidence statement. Tail latency, peak memory, and audit expense are recorded as distinct quantities; a protected expected agent-call cost is exactly the modeled return appearing in the conservation matrix.

D Sequential Verification and Evaluation Metrics

D.1 Reference-null calibration

Under the conditional-action null p, for a normalized, history-measurable marked law on legal support,

$$\mathbb{E}_p\left[\frac{q_{K,h_t}(a_t)}{p_{h_t}(a_t)}\middle| h_t\right] = 1. \tag{18}$$

The product of these factors is a nonnegative martingale initialized at one. Ville's inequality therefore bounds the chance that the running maximum reaches $1/\alpha$ by α. The log-space implementation is Algorithm 2. Across episodes, the filtration includes previous evidence and the frozen key; a fresh request continues the same test.

The reference null is the retained routing law p. Verification uses delegation labels and recorded histories sufficient to reconstruct both conditional laws. Key selection and threshold calibration precede test-set inspection. Testing multiple keys requires family-wise or predeclared aggregate calibration.

D.2 FUNCTIONALITY AND EVIDENCE

Let $\mathcal{H}_{\text{audit}}$ be a held-out, root-task-clustered prefix set. Independent audit continuations estimate the two policies' continuation values. We define

$$D_{U,95} = \text{Quantile}_{0.95}\{100|\widehat{V}_U^q(h) - \widehat{V}_U^p(h)| : h \in \mathcal{H}_{\text{audit}}\}. \tag{19}$$

Aggregate task utility uses success rate for reasoning/environment interaction and pass@1 for code. Cost is averaged over successful, failed, and terminated requests. Detection reports TPR at an independently validated FPR and the number of audited routing decisions.

The functionality tolerances are one percentage point of aggregate success and three percent of expected modeled cost. Source-native WSR, bit recovery, AUC, and successful-task Steps remain in their source-specific panels.

E DETAILED EXPERIMENTAL SETTINGS

E.1 FROZEN SYSTEMS AND TASK PARTITIONS

Each primary system contains six delegates: planner, solver, critic, verifier, tool specialist, and synthesizer. Qwen2.5-7B-Instruct is the primary constituent model and Llama-3.1-8B-Instruct is the model-family transfer setting. A probabilistic two-layer controller selects legal delegate/role labels from the full recorded history representation. The eight-delegation budget is shared across methods. Temperature is 0.7 for constituent generation; the reference controller and its sampling temperature remain fixed during each comparison.

The centralized configuration routes one delegation at a time. The layered configuration routes among the same six candidate delegates within successive planning, solution, and checking stages, with a fixed inter-stage aggregation operator. The graph-routed configuration additionally specifies a legal communication destination in each action label. The candidate set and executor are frozen separately for each configuration.

Controller fitting and value fitting use training roots, calibration uses a disjoint validation partition, and all reported benchmark results use held-out roots. HumanEval and MBPP tests execute in isolated sandboxes with identical time limits across methods. ALFWorld ID and OOD tasks use separate environment split identifiers. The evaluation manifest records model revisions, tokenizer revisions, prompts, task identifiers, execution limits, and cached tool responses.

E.2 CONTINUATION AUDITING AND COMPILATION

The pilot budget is $64 \times 6 \times 32 = 12{,}288$ counterfactual continuations before counting the LLM calls inside each continuation. The expanded audit uses 256 prefixes and compares 8, 32, 128, and 512 continuations per action. Return estimates are obtained from rollouts following the baseline after the first counterfactual delegation. A separate audit set evaluates implemented-policy drift.

A two-output value estimator predicts terminal success and remaining normalized agent-call cost. We fit it on root-disjoint counterfactual data and recalibrate maximum-error envelopes on an independent audit split. The main controller uses 128 samples per audited action, $\eta = 1.5$, and a singular-value rank tolerance of 10^{-8}. The compiled variant minimizes KL to the target conditional law over a fixed key/probe training set and uses three independent training seeds. We record conservation residuals and detection under the compiled law on held-out tasks and precommitted test keys.

E.3 PRELIMINARY ROUTING AUDIT

Scope. The protected object is the orchestration policy. The owner changes routing probabilities while agents, tools, and the executor remain frozen; verification uses auditable delegation histories and retained policy snapshots with a precommitted key. The base controller and constituent models are specified in Appendix E; copying, imitation, and removal are evaluated separately in Section 5.6.

We audit 256 held-out execution prefixes across GSM8K, HumanEval, MBPP, and ALFWorld, using six role-specialized delegates and independent counterfactual continuations. We compare immediate-value and continuation-value interventions, then vary the number of legal delegations and protected

returns. Figure 2 summarizes the three structural effects examined by this audit before a watermark direction is selected.

Sequential effects. Immediate-value preservation leaves downstream drift because subsequent delegations and retries depend on the selected agent's output. **Constrained freedom.** Additional legal alternatives expose probability redistributions that preserve the audited values, while additional protected returns remove directions. **Nonuniform evidence.** At matched probability-deviation budgets, feasible directions yield different key alignment; algebraic freedom alone does not determine usable signal.

These effects are visualized together in Figure 2: the upper panels describe how the carrier changes with routing alternatives and constraints, while the lower comparison isolates the immediate-versus-continuation distinction.

These findings identify three requirements: continuation-aware conservation, a support-aware feasibility check, and key-directed allocation. A finite-state audit with exact transitions independently tests the same causal distinction in Appendix F.

E.4 Baseline alignment and attacks

The RG adaptation biases delegation labels rather than tokens. AgentMark-F is applied to the same normalized delegation alphabet. The PathMark-inspired adaptation enforces a keyed inter-agent route; the original PathMark operates inside an MoE model. Each method receives identical histories and candidate actions for the matched study. Exact-Q is an oracle mechanism control; estimated-Q and compiled policies are evaluated separately.

Bias strengths are selected on calibration data for equal intervention budgets and for equal detection targets. The persistence comparison starts from the compiled RouterMAS controller; fine-tuning uses 5000 clean tasks and imitation uses 5000 task traces. Key-blind erasure reoptimizes on clean task utility and modeled cost without the key; key-aware erasure additionally minimizes the private signature statistic. Deterministic decoding changes the sampler without retraining. Label-permutation tests supply the inverse label map to the verifier.

F Executed Finite-State Checks

The finite-state system has four time steps, five states per layer, six legal actions, and two additive protected returns. A fixed seed, 20260923, determines transition kernels, rewards, baseline probabilities, and the key. Backward dynamic programming computes exact continuation values. The local optimizer uses $\eta = 1.5$ and an explicit numerical feasibility correction. A separate three-action instance yields full-rank constraints.

Table 8: Conservation and inert controls in the executed finite-state system.

Policy	$\max \|B\delta\|_\infty$	$\max \|\Delta V_U\|$	$\max \|\Delta V_C\|$	Mean KL
Unmarked	0	0	0	0
Exact-Q constrained	$< 10^{-15}$	$< 10^{-15}$	$< 10^{-15}$	0.0929
Unprojected bias	0.2037	0.2432	0.2304	0.1666
Immediate-Q constrained	0.0369	0.0156	0.0352	0.0898
Row-space key	$< 10^{-15}$	$< 10^{-15}$	1.78×10^{-15}	$< 10^{-15}$
Random feasible	$< 10^{-15}$	$< 10^{-15}$	1.78×10^{-15}	0.0065
Full-rank three-action control	0	0	0	0

Seed 20260923; horizon 4; five states per time layer; six legal actions; two protected returns; $\eta = 1.5$. The full-rank control uses three actions. Values below 10^{-15} are rounded for display.

Table 8 shows that continuation-value constraints preserve both protected values while changing the routing law, whereas immediate-value constraints preserve a weaker local quantity and leave downstream drift. Table 6 evaluates the running-maximum detector on the same controlled system. Figure 7 provides the geometric interpretation of these controls. Unconstrained and weaker local interventions can move toward value-changing regions, while full continuation conservation restricts the admissible modification to the value-preserving carrier. A row-space key has no component inside that carrier, and a full-rank constraint system leaves no nonzero feasible direction; both therefore return the reference law.

The figure is deliberately schematic: it explains why the controls behave differently, while Tables 1, 6, and 8 supply the numerical evidence. This separation keeps conceptual geometry distinct from measured detection and preservation results.

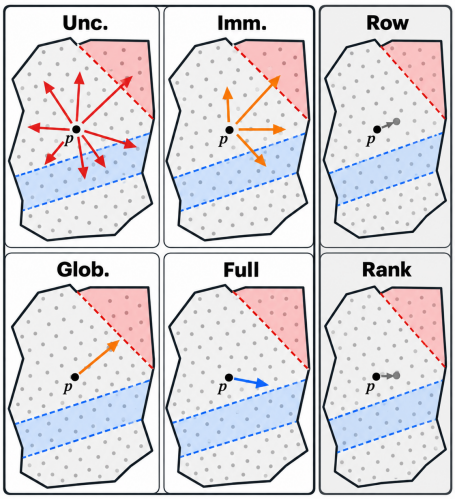

Figure 7: Intervention geometry and inert controls. Unconstrained, immediate-value, global-average, and full continuation-value interventions differ in whether their routing changes remain inside the continuation-value-preserving region. Row-space keys and full-rank constraints yield inert controls. The drawing is schematic; quantitative evidence is reported in Table 1 and the executed finite-state checks.

F.1 Local embedding–verification check

The audit reuses the fixed finite-state generator and exact backward values, holding each prefix's p, B, s_K fixed while varying $\eta \in \{2, 4, 8, 16, 32, 64, 128, 256\}$. The dual stationarity equations yield the exact finite-update law; each primal residual is checked independently. We average per-prefix normalized coefficients uniformly. Table 2 summarizes the normalized coefficients and worst-prefix errors.

The shared-form residual is below 10^{-15}, and the maximum conservation residual is below 2×10^{-15}. At $\eta = 128$, setting $\rho = 0.5$ gives a mean score ratio of 0.500000 and an information ratio of 0.249999; the largest departure from one quarter is 4.71×10^{-4}. Row-space keys and the full-rank three-action control both return exactly the baseline.

An independent categorical replay uses 10^6 draws at each of the 20 prefixes (seed 20260925). Its mean log increment is 1.2657×10^{-5} nats, with standard error 1.194×10^{-6}; direct integration gives 1.4251×10^{-5} nats. Their difference is 1.34 standard errors. This separates the sampled evidence check from the exact likelihood/KL identity.

The GSM8K configuration uses 256 held-out prefixes, six legal delegations, two estimated protected returns, and a precommitted key. The local comparison uses the unscaled optimizer; the separate ρ check applies the final-law correction. It evaluates conditional information, not a finite-budget TPR ordering at equal KL.

G REPRODUCED LIST

The following panels retain the source-specific tasks, models, access conditions, and metrics of the cited papers. MasRouter evaluates MAS capability and cost; AgentMark evaluates behavioral signatures; PathMark evaluates internal MoE signatures; ActHook evaluates trajectory-data reuse; and AFlow evaluates optimized workflows. Each caption identifies the paper and source table or figure.

G.1 MULTI-AGENT ROUTING AND WORKFLOW CAPABILITY

Table 9: Source-paired MAS capability and cost from MasRouter Tables 10 and 11 (Yue et al., 2025).

Method	Backbone	MBPP	Cost ($)	HumanEval	Cost ($)
IO	gpt-4o-mini	72.20	0.143	85.71	0.025
IO	claude-3.5-haiku	73.40	0.146	86.33	0.025
IO	gemini-1.5-flash	73.00	0.157	82.61	0.032
IO	llama-3.1-70b	68.20	0.105	80.75	0.013
SC(CoT)	gpt-4o-mini	73.00	0.449	87.58	0.218
SC(CoT)	gemini-1.5-flash	72.00	0.548	80.75	0.306
SC(ComplexCoT)	gpt-4o-mini	75.60	0.487	88.19	0.241
SC(ComplexCoT)	gemini-1.5-flash	73.60	0.633	81.99	0.335
LLM Debate	gpt-4o-mini	73.60	4.427	84.38	0.624
LLM Debate	gemini-1.5-flash	73.40	4.529	79.38	0.693
MacNet complete	gpt-4o-mini	75.20	2.932	85.00	0.488
MacNet complete	gemini-1.5-flash	74.20	3.088	83.75	0.568
AgentPrune	gpt-4o-mini	75.00	1.215	86.80	0.254
AgentPrune	gemini-1.5-flash	75.60	1.352	82.55	0.271
AFlow	gpt-4o-mini	82.20	1.723	90.15	0.363
AFlow	gemini-1.5-flash	76.00	1.832	85.69	0.386
FrugalGPT	LLM pool	74.40	0.139	87.31	0.026
RouterDC	LLM pool	75.20	0.145	87.75	0.023
MasRouter	LLM pool	84.00	1.039	90.52	0.185

Scores are percentages. Dollar costs retain the original source protocol and historical model identifiers. Each task score and its cost are paired within the same source table.

Table 10: MAS task scores reported in MasRouter Table 1 (Yue et al., 2025).

Method	Backbone	MMLU	GSM8K	MATH	HumanEval	MBPP	Avg.
Vanilla	gpt-4o-mini	77.81	93.17	66.09	85.71	72.20	79.00
GPTSwarm	gpt-4o-mini	82.80	94.66	68.85	86.28	75.40	81.60
GPTSwarm	gemini-1.5-flash	83.22	93.98	73.35	82.36	74.80	81.54
AgentPrune	gpt-4o-mini	83.02	94.89	68.45	86.80	75.40	81.71
AgentPrune	gemini-1.5-flash	83.10	93.88	73.54	82.55	75.80	81.77
AFlow	gpt-4o-mini	83.10	92.30	73.35	90.06	82.20	84.20
AFlow	gemini-1.5-flash	82.35	94.91	72.70	85.69	76.00	82.33
PromptLLM	LLM pool	78.43	93.92	73.03	86.33	73.60	81.06
RouteLLM	LLM pool	81.04	93.42	71.29	83.85	72.60	80.44
FrugalGPT	LLM pool	76.24	90.76	67.05	87.31	74.40	79.15
RouterDC	LLM pool	82.01	93.68	73.46	87.75	75.20	82.42
MasRouter	LLM pool	84.25	95.45	75.42	90.62	84.00	85.93

This panel retains the main-table reporting protocol; the score–cost panel uses the separately identified appendix tables.

Table 11: Workflow scores reported in AFlow Table 1 (Zhang et al., 2025b).

Method	HotpotQA	DROP	HumanEval	MBPP	GSM8K	MATH-lv5	Avg.
IO	68.1	68.3	87.0	71.8	92.7	48.6	72.8
CoT	67.9	78.5	88.6	71.8	92.4	48.8	74.7
CoT SC	68.9	78.8	91.6	73.6	92.7	50.4	76.0
MedPrompt	68.3	78.0	91.6	73.6	90.0	50.0	75.3
MultiPersona	69.2	74.4	89.3	73.6	92.8	50.8	75.1
Self Refine	60.8	70.2	87.8	69.8	89.6	46.1	70.7
ADAS	64.5	76.6	82.4	53.4	90.8	35.4	67.2
AFlow	73.5	80.6	94.7	83.4	93.5	56.2	80.3

All scores retain the original workflow, backbone, and benchmark split. AFlow source scores are separate from MasRouter re-evaluations of AFlow.

Table 12: AFlow source cost and task-performance comparison (Zhang et al., 2025b).

Method	Raw score	Score $\times 100$ (%)	Cost ($)
IO	0.8702	87.02	0.0223
CoT	0.8860	88.60	0.0277
CoT SC	0.9160	91.60	0.1794
MedPrompt	0.9160	91.60	0.2200
LLM Debate	0.8930	89.30	0.2278
Self Refine	0.8780	87.80	0.1232
AFlow(gpt-4o-mini)	0.9470	94.70	0.0513
AFlow(deepseek)	0.9084	90.84	0.0669

Cost definitions and historical model prices follow the original AFlow appendix protocol.

MasRouter's score–cost panel uses paired entries from its appendix Tables 10 and 11; its main Table 1 is retained separately. AFlow's original source panel and MasRouter's AFlow re-evaluation remain distinct. Monetary costs preserve historical source accounting. The source registry in the accompanying data records model identity, split, metric, location, and protocol for every block.

G.2 Behavioral watermarking

Table 13: AgentMark task utility and behavioral capacity from its Table 1 (Huang et al., 2026).

Setting	Base SR	RG SR	AM-F SR	Base steps	RG steps	AM-F steps	bps	bpt
ALFWorld-ID	89.5	78.8	89.3	19.7	26.1	19.4	1.19	25.50
ALFWorld-OOD	96.8	94.5	97.5	15.9	15.4	14.1	1.34	28.40
ToolBench	59.9	58.5	59.7	6.7	7.0	7.2	0.49	4.93

DeepSeek-Chat. Steps count successful tasks; bps counts bits per step over all decision steps, including inactive ones, and bpt counts bits per task. Capacity columns describe AgentMark-F.

Table 14: Cross-model AgentMark results from Table 6 (Huang et al., 2026).

Setting	Base SR	AM-F SR	Base steps	AM-F steps	bps	bpt	Δs/step
ALFWorld-ID	70.60	67.80	25.30	28.00	2.28	80.60	-0.18
ALFWorld-OOD	71.70	72.80	26.50	25.50	1.96	63.40	-0.01
ToolBench	75.60	75.80	2.66	2.74	0.78	2.10	-0.64

Gemini 2.0 Flash; success rates are percentages and Steps count successful tasks. Capacity and per-step time use the original Table 6 definitions.

Table 15: AgentMark observation-rewriting results from Table 2 (Huang et al., 2026).

Observation	Behavior match (%)	KL (nats)	Bitstring recovery (%)
No rewriting	100.00	0.000	100.00
Semantic rewriting	49.45	3.227	16.84

ALFWorld-OOD, 134 tasks and 2,326 steps. Behavior match, KL, and per-step bitstring recovery retain their source definitions. Source spreads for the rewriting row are 16.90, 1.802, and 19.56, respectively.

AgentMark's Steps metric counts successful tasks, while its capacity and bit-recovery measurements concern behavioral signature recovery. The matched RouterMAS experiment instead records all-episode cost and fixed-FPR decision-budget detection; the source panels retain their native units.

G.3 MoE EXPERT-ROUTING WATERMARKING

Table 16: MoE watermark verification reported in PathMark Table 1 (Gao et al., 2026).

MoE backbone	Method	Wiki-103 WSR	ptb-text WSR	MMW WSR
Qwen1.5-MoE-2.7B	KGW	81.67	83.33	91.67
Qwen1.5-MoE-2.7B	LearnMark	78.50	64.30	94.29
Qwen1.5-MoE-2.7B	IFMark	90.00	90.00	100.00
Qwen1.5-MoE-2.7B	EaaW	96.88	100.00	93.75
Qwen1.5-MoE-2.7B	PathMark	100.00	100.00	100.00
Mixtral-8x7B	KGW	90.00	91.67	93.33
Mixtral-8x7B	LearnMark	87.10	90.00	95.70
Mixtral-8x7B	IFMark	100.00	100.00	90.00
Mixtral-8x7B	EaaW	100.00	96.88	96.88
Mixtral-8x7B	PathMark	100.00	99.00	100.00
Phi-3.5-MoE	KGW	83.33	88.33	96.67
Phi-3.5-MoE	LearnMark	87.14	90.00	97.14
Phi-3.5-MoE	IFMark	100.00	90.00	90.00
Phi-3.5-MoE	EaaW	100.00	100.00	90.63
Phi-3.5-MoE	PathMark	100.00	100.00	100.00
Qwen3-30B-A3B	KGW	86.67	95.00	98.33
Qwen3-30B-A3B	LearnMark	94.28	92.86	97.14
Qwen3-30B-A3B	IFMark	100.00	100.00	100.00
Qwen3-30B-A3B	EaaW	100.00	100.00	100.00
Qwen3-30B-A3B	PathMark	100.00	100.00	100.00

WSR is the source watermark-success metric; model rows and verification access are those of the original MoE experiment.

Table 17: MoE task utility reported in PathMark Table 2 (Gao et al., 2026).

Task / subject	Baseline (%)	PathMark (%)
MMLU global facts 5-shot	38.0	38.0
MMLU machine learning 5-shot	42.9	42.0
MMLU professional law 5-shot	45.4	43.5
MMLU high-school physics 5-shot	41.7	38.4
MMLU four-subject average	41.6	40.6
GSM8K 8-shot	55.0	53.0

The original clean and watermarked model conditions and task-score units are retained.

Table 18: PathMark quantization results from Table 4 (Gao et al., 2026).

Method	8-bit WSR (%)	8-bit PPL	4-bit WSR (%)	4-bit PPL
LearnMark	72.85	17.3	68.57	17.9
IFMark	90.00	13.6	90.00	14.8
EaaW	96.88	13.3	93.75	14.1
PathMark	100.00	12.8	100.00	13.9

Quantization, verification success, and perplexity follow the source MoE protocol.

The PathMark panels retain the original expert-routing model and verification modes. WSR, task utility, and perplexity are separate axes. Its forced inter-agent counterpart in the main experiment is the explicitly named delegation adaptation.

G.4 Trajectory-data watermarking

ActHook's detection and utility panels retain their downstream-training protocols, including the source's distinction between detection tasks and agent-utility benchmarks. These observations complement the routing-artifact tests in Section 5.6.

Table 19: Agent-trajectory utility reported in ActHook Table 2 (Meng et al., 2026).

Evaluation task	Method	Success (%)	Turns	Tokens
MATH	Original	60.0	3.7	1204
MATH	Fine-tuned no watermark	75.3	3.5	1319
MATH	ActHook standalone	75.3	3.0	1212
MATH	ActHook contextual	75.3	3.2	975
SimpleQA	Original	59.1	3.4	297
SimpleQA	Fine-tuned no watermark	75.8	2.7	257
SimpleQA	ActHook standalone	75.3	2.9	276
SimpleQA	ActHook contextual	77.1	2.6	228
SWE-Bench Lite	Original	0.0	3.0	2255
SWE-Bench Lite	Fine-tuned no watermark	13.0	49.3	9291
SWE-Bench Lite	ActHook standalone	12.3	50.6	9313
SWE-Bench Lite	ActHook contextual	12.7	50.4	9318

ActHook uses downstream trajectory training. Source task-performance and execution-overhead metrics are retained in this panel.

Table 20: ActHook detection AUC reported in Figure 3 (Meng et al., 2026).

Detection dataset	Standalone AUC	Contextual AUC	CodeMark AUC
MATH	99.5	96.3	56.7
SimpleQA	99.6	87.9	56.6
SWE-Smith	88.3	94.2	53.1

AUC follows the source trajectory-data provenance experiment and is reported independently of fixed-FPR routing detection.

H AI-Use Statement

AI tools assisted literature organization, mathematical exposition, experiment planning, and LaTeX preparation. The numerical finite-state checks were executed by the supplied program.