SKILLLINEAGE: BLACK-BOX PROVENANCE AUDITING OF AGENT SKILLS ACROSS LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Agent skills package procedures, scripts, and reference materials for reuse across language models, but this portability makes their origin difficult to audit once they are copied, rewritten, or reconstructed behind another agent's API. We study *black-box skill provenance auditing*: determining whether an opaque deployment derives from a protected skill or its outputs, rather than from an independently developed solution to the same task. The challenge is that provenance is easily confounded by two effects: independent same-task skills can exhibit similar functionality, while the same skill can appear different when deployed on another model. We introduce SKILLLINEAGE, to our knowledge the first passive fingerprinting framework for black-box agent skill provenance auditing, based on local behavioral correspondence. Using an owner-controlled model, it selects skill-sensitive contexts and queries the suspect with matching response prefixes. The resulting profile records continuation agreement at selected response positions and is compared with references generated with and without the protected skill. The audit uses only returned text, without prior modification of the protected skill or knowing its host model. Across five released skills, 240 deployments on three open-source models, and four black-box APIs, SKILLLINEAGE achieves a 0.9977 AUROC on the three primary skills using only Llama-based references.

1 INTRODUCTION

Agent skills turn specialized expertise into reusable procedures, allowing developers to share capabilities across models without repeating the underlying engineering effort. Building on agents that combine reasoning, tools, and accumulated experience (Yao et al., 2023; Schick et al., 2023; Shinn et al., 2023; Wang et al., 2024a), skills package task-specific instructions, scripts, and reference materials for reuse across agents and language models (Zhang et al., 2025a; Xu & Yan, 2026; Jiang et al., 2026; Ling et al., 2026). A useful skill captures decisions that make a workflow reliable: which tools to use, how to order operations, how to handle exceptions, and how to validate results. Developing and testing these procedures requires expertise and engineering effort. Their value is measurable: SkillsBench reports an average improvement of 16.2 percentage points from curated skills, while self-generated skills provide no average benefit in its evaluation (Li et al., 2026).

Yet this portability creates an accountability gap: a skill can be adapted and deployed behind another agent's API while its origin becomes invisible to its developer. With only black-box access, the developer cannot inspect the underlying skill files or internal implementation, making attribution and licensing terms difficult to audit. Establishing skill provenance therefore provides an evidentiary foundation for attribution and the review of downstream reuse. Developers can permit reuse while retaining a means to investigate whether downstream services derive from their work. Prompt-extraction and cloning attacks show that hidden instructions can be recovered or approximated from outputs (Zhang et al., 2024b;a; Qian et al., 2025). Recent skill-stealing attacks extend this threat to reusable procedures, reconstructing hidden skills through black-box interactions and returned artifacts (Wang et al., 2026c; Tsai et al., 2026).

This motivates our **black-box skill provenance auditing**: assessing whether an opaque agent service carries evidence of derivation from a particular skill or its outputs, without access to its internal instructions or execution traces. The distinction from prompt auditing is the unit of provenance: a portable procedure whose implementation can span instructions, scripts, and reference materials. A

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.

query may expose only some of its workflow steps, conditional rules, or validation choices, while rewriting and reconstruction can replace their textual realization. The **central challenge** is to recover evidence of these procedural choices and distinguish derivation from independent implementations of the same functionality. Independent skills can share correct answers, standard procedures, and public tools, producing *task-induced similarity*; successful task completion alone is therefore insufficient evidence of shared provenance. Meanwhile, moving the same skill to another model introduces *host-induced variation* in how its procedures are followed and expressed. The audit must isolate source-specific behavior from both effects without knowing or reproducing the suspect host.

Existing protection mechanisms leave two gaps for this setting. **(1) Auditing without modifying the skill.** Watermarking deliberately introduces identifiable signals before release (Kirchenbauer et al., 2023; Kuditipudi et al., 2024; Dathathri et al., 2024; Huang et al., 2026; Meng et al., 2026; Gao et al., 2026). Such signals cannot establish the provenance of copies made before they were embedded. Auditing an already released, unmodified skill instead requires a fingerprint extracted from its existing behavior. **(2) Separating skill provenance from host and task effects.** Model fingerprints identify model identity or derivation (Gao et al., 2025; Pasquini et al., 2025; Shao et al., 2026; Xu et al., 2024; Zhang et al., 2025b; Yamabe et al., 2025), whereas the skill can move independently of its host. Prompt-provenance methods test instruction reuse through model behavior (Levin et al., 2025; Lee et al., 2026). Concurrent work also studies cross-model prompt fingerprints (Chen et al., 2026b) and visible-package skill reuse (Chen et al., 2026a). For hidden skills, however, whole-response similarity mixes local procedural evidence with shared task content, host style, and earlier generation choices. The unresolved requirement is to retain source-specific correspondence across hosts while separating independently developed skills for the same task.

We introduce **SKILLLINEAGE**, the *first* black-box behavioral fingerprinting method for skill provenance auditing. Given only an agent's returned text, SKILLLINEAGE extracts a behavioral fingerprint and tests whether the suspect exhibits behavior induced by the protected skill. It requires no access to skill files, model weights, logits, host identity, or execution traces, and **does not modify the protected skill with watermarks or triggers**. Instead of comparing complete answers, SKILLLINEAGE probes matched, skill-relevant response positions under shared prefixes and aggregates the resulting local continuations into a correspondence fingerprint. This fingerprint is then compared with owner-side references collected with and without the protected skill to assess provenance.

We evaluate five released SkillsBench packages (Li et al., 2026), covering planning, network analysis, spreadsheets, 3D mesh analysis, and radar vital-sign extraction, with 240 deployment cases across three open-source language models and four API deployments, including the closed-source GPT-4o-mini and Gemini-2.5-Flash models. The evaluation includes original and rewritten skills, reconstructions following recent skill-stealing methods (Wang et al., 2026c; Tsai et al., 2026), independently developed same-task skills, and other non-source alternatives. Using references constructed only on a separate Llama model, SKILLLINEAGE achieves a macro pooled-host AUROC of 0.9977, outperforming all baselines. These results demonstrate that skill provenance can remain detectable from black-box behavior after host changes.

Our contributions are threefold: (1) formulate black-box provenance auditing for portable agent skills with independent same-task development as a key alternative; (2) develop SKILLLINEAGE, which fingerprints the protected skill without modifying it, using local behavioral correspondence, owner-side references, and text-only access to the suspect; (3) establish its effectiveness across rewritten and reconstructed skills on different host models.

2 RELATED WORK

Agent skills and reconstruction. Agent skills build on language-model agents that store experience, refine written procedures, and reuse executable programs (Shinn et al., 2023; Zhao et al., 2024; Wang et al., 2024a). Recent work further studies how skills are learned, transferred, selected, and reused across models and environments (Wang et al., 2026a; Yu et al., 2026; Ling et al., 2026; Zhang et al., 2026; Sha et al., 2026). This portability also makes skills easier to copy or reconstruct. Prompt-extraction benchmarks and adaptive attacks show that protecting literal instructions does not prevent useful behavior from being recovered through interaction (Wang et al., 2024b; Toyer et al., 2024; Debenedetti et al., 2024; Yang et al., 2025b). Recent skill-stealing methods extend this setting

from recovering hidden prompts to reconstructing structured skill artifacts (Wang et al., 2026c; Tsai et al., 2026). These works highlight the need to trace the origin of copied or reconstructed skills.

Watermark-based attribution. Watermarking provides explicit provenance evidence when protection can be introduced before release. Text watermarks encode deliberately assigned signals in generated tokens; prior work studies their robustness to transformations and attacks such as paraphrasing, removal, and stealing (Kirchenbauer et al., 2023; 2024; Kuditipudi et al., 2024; Christ et al., 2024; Rastogi & Pruthi, 2024; Jovanović et al., 2024). Other schemes study whether identifying signals persist through distillation (Pan et al., 2025; Lu et al., 2025; Cui et al., 2025), while agent watermarks place such signals in plans or actions (Huang et al., 2026; Meng et al., 2026; Gao et al., 2026). PromptCOS similarly supports content-only verification by modifying the protected prompt and jointly optimizing a verification query and output signal (Yang et al., 2026). These approaches use evidence deliberately introduced before copying occurs. SKILLLINEAGE extracts a skill behavioral fingerprint from agent's output text, leaving skill materials unchanged.

Model identity and ancestry. Model fingerprints use active queries, response distributions, and tool-choice patterns to identify the model behind an endpoint (Gubri et al., 2024; Pasquini et al., 2025; Shao et al., 2026; Gao et al., 2025; Wang et al., 2026b), or trace its ancestry under adaptation and composition (Xu et al., 2024; Zhang et al., 2025b; Yamabe et al., 2025; Yan et al., 2026; Xiong et al., 2026). Natural stylistic and behavioral signatures (Suzuki et al., 2025; Bitton et al., 2025) and text-only membership tests (He et al., 2025) provide related evidence about model identity and training history. The attribution target in these studies is the model; our target is a mounted skill that can move independently of its host. Host-dependent style is therefore a confounding factor in skill auditing, and a useful fingerprint must retain evidence of the procedure across model changes.

Prompt and skill provenance. Prompt provenance is closely related to our setting and, in principle, its behavioral auditing methods can also be applied to skills. Prompt Detective statistically compares sampled responses (Levin et al., 2025), BBF compares output signatures across models (Chen et al., 2026b), and PROMPRINT optimizes discriminative queries using a white-box reference model (Lee et al., 2026). However, skill provenance is a more complex auditing problem: the useful evidence lies less in general linguistic patterns and more in the skill-specific functional behavior induced during task execution. As a result, directly transferring prompt-level fingerprints may provide weak discrimination for skills. Recent SkillTrace instead studies skill provenance from expression, implementation, and operational traces extracted from *visible* suspect skill packages (Chen et al., 2026a). Our setting is stricter: the suspect skill itself is inaccessible, and auditing must rely only on the returned text of the deployed agent.

3 PROBLEM SETTING

Skill provenance. A deployed agent service is $E = (M, S, R, \theta)$, where M is the host model, S the mounted skill, R the runtime and resources, and θ the generation configuration. We study the provenance of S, rather than the identity of M. Given an owner's source skill $S^\star$ and a public task description C, we call a suspect skill source-related if it was constructed from $S^\star$ or from outputs produced by $S^\star$ before the audit began.

The hypotheses concern source history rather than textual similarity:

$$H_1 : \ S \text{ derives from } S^\star \text{ or its outputs}, \qquad H_0 : \ S \text{ does not derive from } S^\star \text{ or its outputs}. \quad (1)$$

H_1 includes direct copying, rewriting, and behavioral reconstruction. H_0 includes no mounted skill ($S = \varnothing$), unrelated skills, and skills developed independently for the same public task C. The last case is the critical alternative: two skills may share task capability, public tools, and reasonable implementation choices without sharing a source. Source-relatedness describes provenance only and does not imply that the reuse was unauthorized.

Adversary's knowledge and capability. The adversary knows the public task C and may obtain the source skill $S^\star$ or access a service running it. It can copy, rewrite, reorganize, or reconstruct the skill, and freely choose the suspect host and deployment configuration. We assume the suspect skill is constructed before auditing and remains fixed during measurement.

Auditor's goal. The auditor owns $S^\star$ and seeks to distinguish source-related skills from independent alternatives for the same task. It controls a trusted reference model that can run with and without

the source skill, and performs all probe and reference construction locally. Auditing requires only black-box text access to the suspect: the defender submits a task and response prefix and observes the returned continuation. No access to the suspect host identity, skill files, model internals, or tool traces is required, and no watermark or trigger is inserted into the source skill.

4 SKILLLINEAGE

In this section, we explain why an output fingerprint that distinguishes skills on one host may fail after a host change. We then develop SKILLLINEAGE by choosing queries and response contexts that help compare skills across hosts.

4.1 DESIGN RATIONALE

Output distribution shifts caused by host changes. A model uses its own knowledge and the mounted skill to answer a probe. The answer also depends on the runtime and generation settings:

$$Y \sim P(\cdot \mid x, S, M, R, \theta). \tag{2}$$

Here x is the probe, S the skill, M the model, and R, θ the deployment settings. A fingerprint extracted from Y therefore contains the effects of these factors together.

If the verifier has prior access to the exact host on which the suspect skill will later be deployed, it can first mount the protected skill $S^\star$ on that host and construct a host-specific reference fingerprint $f(M, S^\star)$. At audit time, the suspect fingerprint $f(M, S)$ is then compared against this pre-profiled reference. Because both fingerprints are collected on the same host, host-induced variation is largely removed, and existing fingerprinting methods can perform well under this setting. For example, on our test, Prompt Detective (Levin et al., 2025) and ZeroPrint (Shao et al., 2026) achieve same-host AUROCs of 0.9529 and 0.9764, respectively, pooled over Qwen, InternLM, and GLM.

This setting, however, assumes that the verifier knows in advance where the protected skill will be redeployed and can profile $S^\star$ on that exact host before the audit. Such an assumption is generally unavailable in post-hoc provenance auditing: a copied or reconstructed skill may be deployed on an arbitrary, previously unseen model or API.

When the suspect model is unknown, the verifier may have references only on a different, local model. The comparison then includes differences due to both the skill and the model. A low similarity may indicate a different skill, or simply a different model running the same skill. Formally, the fingerprint difference can be written as

$$f(M_v, S) - f(M_r, S^\star) = \underbrace{f(M_r, S) - f(M_r, S^\star)}_{\text{changing the skill on the reference host}} + \underbrace{f(M_v, S) - f(M_r, S)}_{\text{changing the host for the same skill}}. \tag{3}$$

Here M_r and M_v are the reference and suspect hosts. The second term is zero in a same-host comparison. Across hosts, it may dominate the comparison and reverse the distance ordering of candidates. In our testing, replacing same-host references with Llama references reduces the Prompt Detective (Levin et al., 2025) and ZeroPrint (Shao et al., 2026) AUROCs on the same PDDL diagnostic to 0.4848 and 0.8788.

4.2 THE SKILLLINEAGE AUDIT

Potential solutions. One possible solution is to separate the model and skill contributions, for example by assuming $f(M, S) = f_{\text{model}}(M) + f_{\text{skill}}(S)$, and extracting only the skill component. This separation is difficult because the model and skill jointly determine the answer: the effect of a skill can change with the model. We instead seek probes that make the skill's influence on the response more pronounced. Such probes ask about procedures specified by the skill, so that answering them depends more on the skill than on the model's general knowledge.

This gives our probe-design objective. If host changes cause only bounded fingerprint shifts, a larger skill-induced difference is harder for those shifts to erase. We therefore seek probes that make the effect of the skill large, as formalized below.

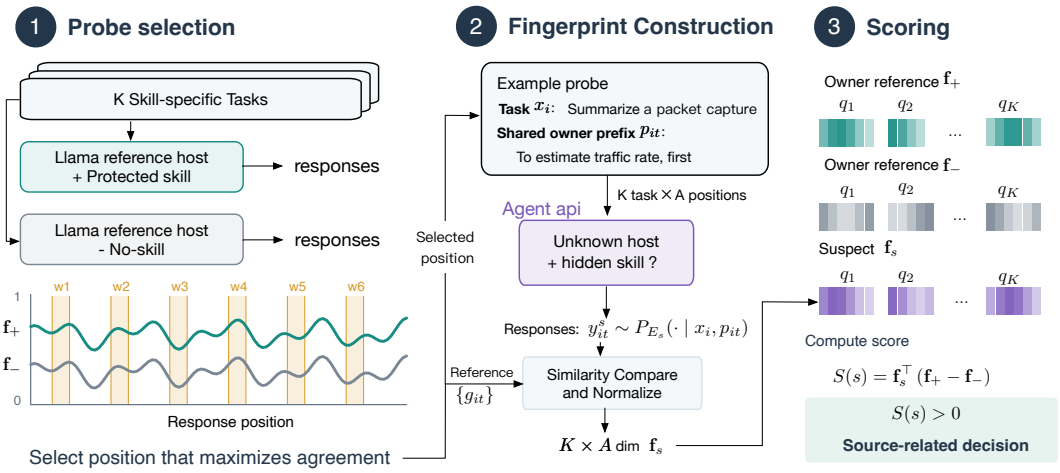


Figure 1: SKILLLINEAGE overview. We select response windows where the protected skill increases agreement with owner targets, then compare suspect continuations under the same owner prefixes. The resulting local agreement profile is scored against protected-skill and no-skill references to audit provenance across hosts.

Proposition 1 (Probe selection under bounded host shifts)**.** *Let f_Q be the fingerprint obtained with probes Q, and fix a skill state S to distinguish from $S^\star$. Suppose $\|f_Q(M_v, S) - f_Q(M_r, S)\|_2 \le B$ for all allowed probe sets and hosts, with a common bound B. Then*

$$\|f_Q(M_v, S) - f_Q(M_r, S^\star)\|_2 \ge \|f_Q(M_r, S) - f_Q(M_r, S^\star)\|_2 - B. \tag{4}$$

Under a fixed budget K, this lower bound is maximized by

$$Q^\star \in \arg\max_{Q:\,|Q|=K} \underbrace{\|f_Q(M_r, S) - f_Q(M_r, S^\star)\|_2}_{\textit{effect of changing the skill}}. \tag{5}$$

The proposition motivates increasing the skill effect without requiring an explicit model–skill decomposition (proof: Appendix A.1). We next specify how to select probes, represent the measured behavior as a fingerprint, and compare that fingerprint with references to obtain a provenance score.

Building on Proposition 1, SKILLLINEAGE implements the probe-design principle in three stages. Probe selection finds response regions where the protected skill has a strong observable effect. Fingerprint construction preserves that effect as an aligned local profile. Provenance scoring compares the profile with two owner-controlled references. Figure 1 summarizes the SKILLLINEAGE audit.

Probe selection. The ideal objective in Equation (5) depends on the unknown suspect skill state S. We therefore instantiate the contrast on the owner-controlled reference host by using No-Skill as a fixed alternative to the protected skill. The search has two steps. The owner constructs K skill-specific task questions covering distinct procedures in the protected skill. A question states the task but does not reveal the protected procedure. The owner then uses the reference host M_r to locate the informative regions of each response.

For question x_i, the response generated with $S^\star$ supplies candidate prefixes and owner targets. Let $u_i(w; M, S)$ be the local agreement with these targets in window w, as defined in Equation (8). The reference-side gain and the jth selected window are

$$G_i(w) = u_i(w; M_r, S^\star) - u_i(w; M_r, \varnothing), \qquad W_{ij} \in \arg\max_{w \in \mathcal{W}_i^{(j)}} G_i(w). \tag{6}$$

Both terms in $G_i(w)$ use the same reference host, question, prefix, and target; the only difference is whether the protected skill is mounted. A large positive $G_i(w)$ therefore means that the skill makes the model follow the owner target more closely in window w. We use this difference to approximate the skill effect in Proposition 1.

After selecting one window, we remove nearby windows from the next candidate set $\mathcal{W}_i^{(j)}$. This prevents several selected windows from measuring the same part of a response. We repeat the selection

until each of the K questions has A windows. The resulting questions, owner texts, and relative window positions are held fixed for every suspect deployment in an evaluation. See Appendix C.3 and Appendix B for probe examples and selection settings.

Fingerprint construction. Skill-specific questions alone do not ensure that a fingerprint preserves the elicited skill information. Different hosts can organize and phrase a full answer differently even when they follow the same procedure. SKILLLINEAGE instead compares local continuations under the same owner prefix and retains the agreement pattern across response positions.

Given a task and visible prefix, supplied via native prefill or an ordinary continuation prompt, suspect deployment E_s returns

$$y_{it}^s \sim P_{E_s}(\cdot \mid x_i, p_{it}), \tag{7}$$

where x_i denotes the ith task, p_{it} denotes the owner prefix immediately before observation t, and y_{it}^s denotes the text returned by the suspect at that position. Each y_{it}^s is obtained from an independently restarted next-token request; predictions are not chained into a free-running continuation. The saved relative position identifies the corresponding region of the owner response on each host, while every retained record still pairs one returned token with its exact owner target. One fingerprint coordinate is the mean semantic agreement within the selected window:

$$a_{ij}^s = \frac{1}{|W_{ij}|} \sum_{t \in W_{ij}} \rho(g_{it}, y_{it}^s), \qquad \rho(g, y) = \frac{e(g)^\top e(y)}{\|e(g)\|_2 \|e(y)\|_2}, \tag{8}$$

where g_{it} denotes the next-token text in the fixed owner response at observation t of task x_i, and $\rho(g, y)$ denotes the cosine similarity between the embeddings of owner target text g and returned text y. The encoder $e(\cdot)$ is the fixed `all-MiniLM-L6-v2` (Reimers & Gurevych, 2019); W_{ij} denotes the jth selected window for task x_i, containing $|W_{ij}| = 20$ retained local observations. The same statistic defines $u_i(w; M, S)$ during reference-side selection. Each question contributes A coordinates. Concatenating them in fixed task–position order and normalizing the uncentered profile gives

$$\begin{aligned} \mathbf{a}_i^s &= [a_{i1}^s, \ldots, a_{iA}^s]^\top, \qquad \mathbf{a}_s = [(\mathbf{a}_1^s)^\top, \ldots, (\mathbf{a}_K^s)^\top]^\top \in \mathbb{R}^{KA}, \\ \mathbf{f}_s &= \mathcal{N}(\mathbf{a}_s), \qquad\qquad \mathcal{N}(\mathbf{v}) = \frac{\mathbf{v}}{\|\mathbf{v}\|_2}. \end{aligned} \tag{9}$$

We use $K = 8$ questions and $A = 6$ windows per question, producing a 48-dimensional fingerprint for each suspect. Each coordinate records agreement at one selected response position. We keep the six scores for each question separate because different positions can reflect different steps of the procedure. Averaging them would reduce the fingerprint to eight values and remove this information. Section 5.4 compares this position-wise representation with coarser and modified alternatives.

Provenance scoring. The owner computes two fingerprints on the reference host: $\mathbf{f}_+$ with the protected skill and $\mathbf{f}_-$ without a mounted skill. These are the only reference deployments used by the primary method. The signed provenance score is

$$S(s) = \mathbf{f}_s^\top (\mathbf{f}_+ - \mathbf{f}_-), \qquad \widehat{H}(s) = H_1 \text{ iff } S(s) > 0. \tag{10}$$

Because all three profiles are unit normalized, $S(s)$ is the suspect's cosine similarity to original minus its cosine similarity to no-skill. The zero boundary follows directly from this comparison and is not fitted on an evaluation host. At audit time, the readout requires neither the suspect host identity nor labeled suspect deployments. Its output is a relative provenance score, not a posterior probability.

5 EXPERIMENTS

5.1 EVALUATION PROTOCOL

Evaluation skills and deployment. We study five released SkillsBench packages (Li et al., 2026): `pddl-skills` for formal planning (Ghallab et al., 1998), `pcap-analysis` for network captures, `xlsx` for spreadsheets, `mesh-analysis` for 3D meshes, and `radar-vital-signs` for radar vital-sign extraction. For each target, the source-related group contains original, paraphrased, and two skill stealing attacks Daydreaming (Tsai et al., 2026) and Wang et al. skill stealing (Wang et al., 2026c); The non-source group contains five or six independently developed same-task skills, six or five skills for other tasks, respectively, and one no-skill deployment (see Appendix C.1).

Table 1: **Skill provenance performance.** Llama-3.1-8B is the only reference model; Qwen3-8B, InternLM2.5-7B, and GLM-4-9B-0414 are evaluation hosts. Each target–host pair contains four positive and twelve negative deployments. Each entry is AUROC for one target–host pair. Macro averages the three skills' pooled-host AUROCs (five-skill results: Table 6 in Appendix D.1).

	PDDL			PCAP			XLSX			Macro
Method	**Qwen**	**Intern**	**GLM**	**Qwen**	**Intern**	**GLM**	**Qwen**	**Intern**	**GLM**	**AUC ↑**
Prompt Detective (Levin et al., 2025)	0.5000	0.5000	0.4583	0.5000	0.5000	0.5000	0.5000	0.5000	0.6250	0.5093
MET (Gao et al., 2025)	0.3750	0.8125	0.9167	0.7917	**1.0000**	0.8750	0.3125	0.5000	0.6458	0.5841
LLMmap (Pasquini et al., 2025)	0.7292	0.7500	0.6875	0.0208	0.3750	0.3333	0.5625	0.5833	0.7917	0.5285
ZeroPrint (Shao et al., 2026)	0.8125	0.8958	**1.0000**	0.6667	0.5833	0.7500	0.7708	0.6667	0.3125	0.6960
BBF (Chen et al., 2026b)	0.8333	0.7396	0.8333	0.5104	0.8958	0.6354	0.4583	0.5208	0.9167	0.6705
SKILLLINEAGE	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**0.9792**	**0.9977**

We build the reference library on Llama-3.1-8B-Instruct and evaluate candidate deployments on three open-source models: Qwen3-8B, InternLM2.5-7B, and GLM-4-9B-0414 (Grattafiori et al., 2024; Yang et al., 2025a; Cai et al., 2024; Z.ai, 2025). For each target, SKILLLINEAGE evaluates every candidate deployment on the same eight tasks, with six observation windows per task. A fixed all-MiniLM-L6-v2 encoder measures local agreement between owner targets and returned text (Reimers & Gurevych, 2019); the 48 window scores form the fingerprint described in Section 4.2.

Baselines. Since no prior work directly addresses skill provenance auditing, we adapt five representative methods from prompt provenance, model equality testing, and model fingerprinting to the skill provenance setting. Prompt Detective uses the permutation-test p-value between candidate and reference response banks as its score (Levin et al., 2025). Model Equality Testing uses the negative MMD–Hamming discrepancy between the same two response banks (Gao et al., 2025). LLMmap maps eight generic-query replies to a trace vector and scores its negative Euclidean distance from the Llama original trace (Pasquini et al., 2025). ZeroPrint estimates a response-Jacobian fingerprint from perturbed inputs and scores its Pearson correlation with the reference skill fingerprint (Shao et al., 2026). BBF compares full-response embedding centroids with an owner-side unrelated reference; its score is the fraction of queries with a positive similarity gap (Chen et al., 2026b). Higher scores indicate greater similarity to the protected reference (implementation appears in Appendix C.4).

Metrics. Following the hypothesis test in Eq. (1), we evaluate how well the provenance score separates source-related deployments (H_1) from non-source deployments (H_0) without using the suspect host identity. Each target–host pair contains four positive and twelve negative deployments, giving 240 deployments across five skills and three open-source hosts. We use **AUROC** to measure the ranking of H_1 over H_0 across decision thresholds. For each target, deployment scores are pooled across the three evaluation hosts to obtain a pooled-host AUROC, and Macro AUROC is the mean over the targets reported in each table.

5.2 MAIN RESULTS

Table 1 reports the cross-host provenance results and comparisons for the three primary target skills. Across the three primary targets, **SKILLLINEAGE achieves strong cross-host provenance discrimination.** Using references constructed only on Llama, SKILLLINEAGE achieves a 0.9977 macro pooled-host AUROC. All 36 source-related deployments are correctly identified. PDDL and PCAP achieve perfect discrimination across the three hosts, while XLSX obtains a pooled AUROC of 0.9931. The four false positives on XLSX include three independently developed same-task skills and one other-task skill (Figure 2(b)). Extending evaluation to `Mesh` and `Radar` skills yield pooled-host AUROCs of 0.9954 and 0.9167, respectively (see Appendix D.1).

SKILLLINEAGE substantially outperforms existing baselines. The strongest baseline, ZeroPrint (Shao et al., 2026), reaches a macro pooled-host AUROC of 0.6960. Prompt Detective (Levin et al., 2025), Model Equality Testing (MET) (Gao et al., 2025), and LLMmap (Pasquini et al., 2025) achieve 0.5093, 0.5841, and 0.5285, respectively. BBF (Chen et al., 2026b) obtains 0.6705. These results show that methods designed for prompt reuse, model equality, or model fingerprinting do not directly provide reliable skill-provenance discrimination when the reference and suspect deployments use different host models. Further host-transfer and candidate-family analyses show that Prompt

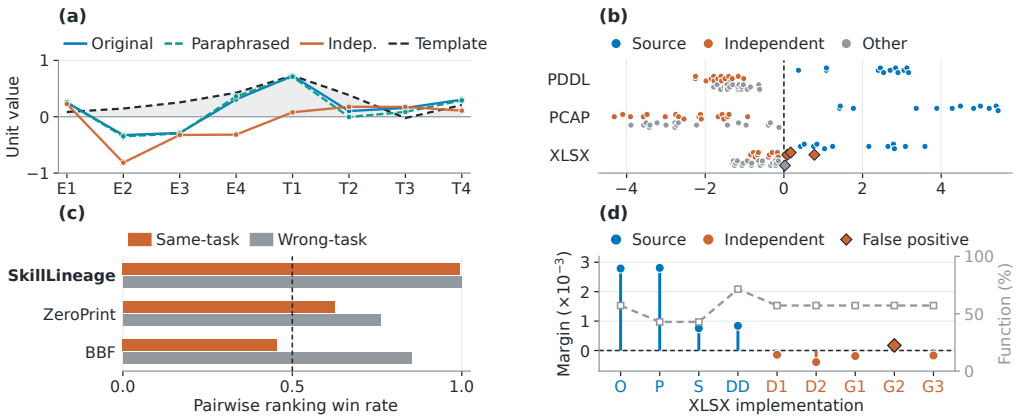


Figure 2: **Skill fingerprints and provenance of SKILLLINEAGE.** (a) Skill fingerprint visualization across eight probes. (b) Scatter plots of provenance scores for positive and negative candidate deployments. (c) Provenance discrimination of independent same-task skills and wrong-task skills. (d) XLSX provenance margins (left) and task pass rates (right) on Qwen. O: Original; P: Paraphrased; S: Steal; DD: Daydreaming; "D" and "G": DeepSeek and GLM-generated independent skills.

Detective loses cross-host separation and BBF struggles with independent same-task skills (see Appendix D.1 and Appendix D.3 for detailed analysis).

5.3 DISTINGUISHING PROVENANCE FROM TASK FUNCTIONALITY

We next ask whether the strong discrimination above reflects provenance itself, rather than differences in task functionality. **Independent implementations of the same task are substantially harder negatives than skills for unrelated tasks.** As shown in Figure 2(c), ZeroPrint (Shao et al., 2026) and BBF (Chen et al., 2026b) obtain much higher pairwise ranking accuracy against wrong-task skills than against independently developed same-task skills. In contrast, SKILLLINEAGE maintains a 0.9948 ranking accuracy on the same-task comparisons. This result shows that distinguishing a protected skill from an unrelated capability is an easier problem and does not by itself demonstrate provenance discrimination.

The provenance signal persists even when observed task performance is matched. On Qwen, original and five independent XLSX implementations pass the same four of seven tasks, with identical pass and fail patterns (Figure 2(d)). Paraphrased, steal, and daydreaming pass three, three, and five tasks, respectively. All four source-related candidates nevertheless rank above every independent implementation. These results indicate that our skill-auditing fingerprints capture provenance-related behavior rather than task performance alone.

5.4 CONTRIBUTIONS OF THE DESIGN CHOICES

We next examine which components of SKILLLINEAGE are responsible for its cross-host provenance signal. Table 2 performs controlled substitutions at three stages of the audit: probe design, fingerprint representation, and reference-based scoring. Each diagnostic changes one component while holding the remaining choices fixed.

Skill-specific probes expose provenance-relevant behavior. We compare our probes with neutral questions matched to the same task domain. Although the neutral probes strongly reveal host identity, they provide little separation between source-related and independently developed skills. Skill-specific probes recover this separation, including the harder same-task case. Thus, behavior useful for identifying the host is not necessarily useful for identifying the skill.

Local correspondence preserves where the protected skill influences the response. Using the same saved predictions, we compare generated-text traces, probe-level summaries, and the local correspondence profile. Preserving local comparisons gives the clearest separation, particularly for same-task independent skills. Moreover, breaking the alignment between suspect continuations and their corresponding owner targets sharply weakens discrimination, while simpler pooling changes

Table 2: **Counterfactual tests of the three audit stages.** (a) Skill-specific probe design; (b) local correspondence in the fingerprint; (c) negative-reference scoring.

Controlled change (control → design)	All negatives	Independent
(a) Do the probes elicit skill-specific behavior? (PDDL)		
Task-matched neutral probes → skill-specific probes	0.5101 → **0.9798** (+0.4697)	0.5056 → **0.9611** (+0.4556)
Task-irrelevant probes → skill-specific probes	0.6641 → **0.9798** (+0.3157)	0.5111 → **0.9611** (+0.4500)
(b) Does the fingerprint preserve local correspondence? (three-skill macro)		
Probe-wide generated trace → selected local correspondence	0.9402 → **0.9832** (+0.0429)	0.8833 → **0.9889** (+0.1056)
Pool six positions per probe → retain all 48 positions	0.9436 → **0.9832** (+0.0396)	0.9407 → **0.9889** (+0.0481)
Shift owner targets within each probe → correct alignment	0.5320 → **0.9621** (+0.4301)	0.5444 → **0.9500** (+0.4056)
Replace targets across probes → correct alignment	0.5480 → **0.9621** (+0.4141)	0.5201 → **0.9500** (+0.4299)
(c) Does a negative reference term help with the positive centroid fixed? (three-skill macro)		
LLMmap: positive centroid → positive-minus-negative	0.4958 → **0.5833** (+0.0875)	0.4444 → **0.5664** (+0.1219)
ZeroPrint: positive centroid → positive-minus-negative	0.7365 → **0.8173** (+0.0808)	0.6827 → **0.7985** (+0.1157)
SKILLLINEAGE: positive centroid → positive-minus-negative	0.7727 → **0.9832** (+0.2104)	0.7907 → **0.9889** (+0.1981)

have little effect. The gain therefore comes from preserving *which* local behavior corresponds to *which* owner target, rather than merely using a larger representation.

The negative reference contributes at the scoring stage. We hold both the fingerprint and the all-source positive centroid fixed, and only subtract the score to the available Llama non-source centroid. This changes LLMmap from 0.4958 to 0.5833 AUROC, ZeroPrint from 0.7365 to 0.8173, and our SKILLLINEAGE from 0.7727 to 0.9832. The negative reference therefore removes shared similarity that is insufficient for establishing provenance, but does not replace skill-specific probing or local correspondence.

Observations per window have the largest effect in the budget comparisons. We vary probe counts, anchors per probe, and observations per window. Increasing probes from 1 to 6 raises macro AUROC by 7.33 percentage points; increasing anchors per probe from 1 to 6 raises it by 14.43 points; increasing observations per window from 5 to 20 raises it by 16.98 points, the largest gain among these budget changes (see Appendix D.2).

5.5 EXTERNAL BLACK-BOX DEPLOYMENT

We also evaluate SKILLLINEAGE on four black-box APIs: DeepSeek-V4.1-Flash, GLM-4.7, GPT-4o-mini, and Gemini-2.5-Flash; GPT-4o-mini and Gemini-2.5-Flash are closed-source models. Each deployment uses the fixed Llama original and no-skill references, eight fixed probes, six frozen owner prefixes per probe, and at most 20 generated tokens per query. For APIs without native prefill, we prompt the model to continue the owner prefix through ordinary chat messages. Table 3 reports strong cross-API provenance discrimination across skills. The cost per audit remains low across all settings.

Table 3: **SKILLLINEAGE skill auditing AUC on black-box API deployments.** The auditor collect suspect output fingerprint from apis and compute similarity with Llama skill and no-skill references. Macro AUC averages all skills.

API model	PDDL ↑	PCAP ↑	XLSX ↑	Mesh ↑	Macro ↑	Cost per audit (USD)
DeepSeek-V4.1-Flash (552B)	1.0000	1.0000	0.9792	0.9896	0.9922	$0.00310
GLM-4.7 (358B)	1.0000	1.0000	0.9375	0.9792	0.9792	$0.01413
GPT-4o-mini	1.0000	1.0000	0.9167	0.9896	0.9766	$0.01378
Gemini-2.5-flash	0.9583	1.0000	0.9167	0.9375	0.9531	$0.04495

6 CONCLUSION

In this work, we presented SKILLLINEAGE, a black-box framework for agent-skill provenance auditing without inspecting suspect implementations or inserting watermarks into protected skills. It uses local behavioral correspondence under shared prefixes to distinguish source-related skills from independent implementations. Experiments on five skills across three open-source hosts, with additional evaluations on four APIs, demonstrate cross-host provenance discrimination. Functional controls indicate that this signal is not explained by task performance alone. Together, our method and evaluation protocol offer the community a way to study skill reuse beyond functional similarity.

AI USE STATEMENT

Generative AI assisted language editing, literature discovery, LaTeX formatting, and analysis-code review. The authors verified sources, code, and numerical claims, reviewed the manuscript, and take responsibility for all content and artifacts.

ETHICS STATEMENT

This study uses publicly released SkillsBench packages and controlled source-related and independently developed candidates. We evaluate skill-stealing attacks only on local deployments of these public skills to assess audit robustness. The intended use is to help skill developers examine possible reuse through behavioral evidence. Such evidence does not establish whether reuse was unauthorized: false positives remain possible, and shared provenance does not imply a lack of permission. Audit findings should therefore be considered alongside source histories and authorization records before attributing misuse.

REPRODUCIBILITY STATEMENT

Section 4 specifies the audit procedure, fingerprint construction, and scoring rule. Section 5.1 describes the skills, evaluation models, deployment split, and metrics. The supplementary material provides selection and numerical settings, example audit questions, baseline implementations with repository links, and additional results and analyses. Owner responses, observation positions, and Llama references are held fixed across the corresponding suspect deployments.

REFERENCES

Yehonatan Bitton, Elad Bitton, and Shai Nisan. Detecting stylistic fingerprints of large language models. *arXiv preprint arXiv:2503.01659*, 2025.

Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, et al. Internlm2 technical report. *arXiv preprint arXiv:2403.17297*, 2024.

Jialuo Chen, Minghe Wang, Lingqi Jiang, Jianan Ma, Xinhao Deng, Xiaohu Du, Ruixiao Lin, et al. SkillTrace: Multi-trace provenance auditing for LLM-agent skill reuse. arXiv preprint arXiv:2608.05204, 2026a.

Linghan Chen, Yudong Gao, Jiyao Wang, Kaiyan Ji, and Honglong Chen. Do system prompts leave behavioral fingerprints? a large-scale empirical study of clone detection via output similarity. arXiv preprint arXiv:2608.24461, 2026b.

Miranda Christ, Sam Gunn, and Or Zamir. Undetectable watermarks for language models. In *Proceedings of Thirty Seventh Conference on Learning Theory*, 2024.

Xinyue Cui, Johnny Wei, Swabha Swayamdipta, and Robin Jia. Robust data watermarking in language models by injecting fictitious knowledge. In *Findings of the Association for Computational Linguistics: ACL 2025*, 2025.

Sumanth Dathathri, Abigail See, Sumedh Ghaisas, Po-Sen Huang, Rob McAdam, Johannes Welbl, Vandana Bachani, et al. Scalable watermarking for identifying large language model outputs. *Nature*, 2024. doi: 10.1038/s41586-024-08025-4.

Edoardo Debenedetti, Javier Rando, Daniel Paleka, Fineas Silaghi, Dragos Albastroiu, Niv Cohen, Yuval Lemberg, et al. Dataset and lessons learned from the 2024 SaTML LLM capture-the-flag competition. In *Advances in Neural Information Processing Systems: Datasets and Benchmarks Track*, 2024.

Irena Gao, Percy Liang, and Carlos Guestrin. Model equality testing: Which model is this api serving? In *International Conference on Learning Representations*, 2025.

Zheng Gao, Xiaoyu Li, Xiaoyan Feng, Jiaojiao Jiang, Yang Song, Yulei Sui, Zhenchang Xing, et al. Trace: A two-channel robust attribution watermark via complementary embeddings for llm-agent trajectories. *arXiv preprint arXiv:2607.08400*, 2026.

Malik Ghallab, Adele Howe, Craig Knoblock, Drew McDermott, Ashwin Ram, Manuela Veloso, Daniel Weld, et al. Pddl—the planning domain definition language. Technical Report CVC TR-98-003/DCS TR-1165, Yale Center for Computational Vision and Control, October 1998. Version 1.2.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Martin Gubri, Dennis Ulmer, Hwaran Lee, Sangdoo Yun, and Seong Joon Oh. Trap: Targeted random adversarial prompt honeypot for black-box identification. In *Findings of the Association for Computational Linguistics: ACL 2024*, 2024. doi: 10.18653/v1/2024.findings-acl.683.

Yu He, Boheng Li, Liu Liu, Zhongjie Ba, Wei Dong, Yiming Li, Zhan Qin, et al. Towards label-only membership inference attack against pre-trained large language models. In *34th USENIX Security Symposium*, pp. 1609–1628, 2025.

Kaibo Huang, Jin Tan, Yukun Wei, Wanling Li, Zipei Zhang, Hui Tian, Zhongliang Yang, et al. Agentmark: Utility-preserving behavioral watermarking for agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*, 2026. doi: 10.18653/v1/2026.acl-long.573.

Yanna Jiang, Delong Li, Haiyu Deng, Baihe Ma, Xu Wang, Qin Wang, and Guangsheng Yu. Sok: Agentic skills—beyond tool use in llm agents. *arXiv preprint arXiv:2602.20867*, 2026.

Nikola Jovanović, Robin Staab, and Martin Vechev. Watermark stealing in large language models. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Manli Shu, Khalid Saifullah, Kezhi Kong, Kasun Fernando, et al. On the reliability of watermarks for large language models. In *International Conference on Learning Representations*, 2024.

Rohith Kuditipudi, John Thickstun, Tatsunori Hashimoto, and Percy Liang. Robust distortion-free watermarks for language models. *Transactions on Machine Learning Research*, 2024.

Jungmin Lee, Peizhuo Lv, and Yeonjoon Lee. PROMPRINT: Prompt fingerprinting via first-token response for LLM app cloning detection. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 22960–22972. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.1052.

Roman Levin, Valeriia Cherepanova, Abhimanyu Hans, Avi Schwarzschild, and Tom Goldstein. Has my system prompt been used? large language model prompt membership inference. *arXiv preprint arXiv:2502.09974*, 2025.

Xiangyi Li, Yimin Liu, Wenbo Chen, Bingran You, Zonglin Di, Yifeng He, Shenghan Zheng, et al. Skillsbench: Benchmarking how well agent skills work across diverse tasks. *arXiv preprint arXiv:2602.12670*, 2026.

George Ling, Shanshan Zhong, and Richard Huang. Agent skills: A data-driven analysis of claude skills for extending large language model functionality. *arXiv preprint arXiv:2602.08004*, 2026.

Xinyang Lu, Jingtan Wang, Zitong Zhao, Zhongxiang Dai, Chuan-Sheng Foo, See Kiong Ng, and Bryan Kian Hsiang Low. Wasa: Watermark-based source attribution for large language model-generated data. In *Findings of the Association for Computational Linguistics: ACL 2025*, 2025.

Wenlong Meng, Chen Gong, Terry Yue Zhuo, Fan Zhang, Kecen Li, Zheng Liu, Zhou Yang, et al. Watermarking LLM agent trajectories. In *International Conference on Machine Learning (ICML)*, 2026.

Leyi Pan, Aiwei Liu, Shiyu Huang, Yijian Lu, Xuming Hu, Lijie Wen, Irwin King, et al. Can llm watermarks robustly prevent unauthorized knowledge distillation? In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, 2025.

Dario Pasquini, Evgenios M. Kornaropoulos, and Giuseppe Ateniese. Llmmap: Fingerprinting for large language models. In *34th USENIX Security Symposium*, 2025.

Yi Qian, Fei Peng, Hao Wu, Ligeng Chen, and Bing Mao. Uncovering prompt elements: Cloning system prompts from behavioral traces. In *40th IEEE/ACM International Conference on Automated Software Engineering*, pp. 457–468, 2025. doi: 10.1109/ASE63991.2025.00045.

Saksham Rastogi and Danish Pruthi. Revisiting the robustness of watermarking to paraphrasing attacks. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024. doi: 10.18653/v1/2024.emnlp-main.1005.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, 2019. doi: 10.18653/v1/D19-1410.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, et al. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems*, 2023.

Tianming Sha, Yue Zhao, Lichao Sun, and Yushun Dong. SkillCenter: A large-scale source-grounded skill library for autonomous AI agents. *arXiv preprint arXiv:2607.07676*, 2026.

Shuo Shao, Yiming Li, Hongwei Yao, Yifei Chen, Yuchen Yang, and Zhan Qin. Reading between the lines: Towards reliable black-box llm fingerprinting via zeroth-order gradient estimation. In *Proceedings of the ACM Web Conference 2026*, pp. 2637–2648, 2026. doi: 10.1145/3774904.3792196.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, 2023.

Teppei Suzuki, Ryokan Ri, and Sho Takase. Natural fingerprints of large language models. *arXiv preprint arXiv:2504.14871*, 2025.

Sam Toyer, Olivia Watkins, Ethan Mendes, Justin Svegliato, Luke Bailey, Tiffany Wang, Isaac Ong, et al. Tensor trust: Interpretable prompt injection attacks from an online game. In *International Conference on Learning Representations*, 2024.

Yu-Lin Tsai, Yu-An Lu, Ci-Yang Tsai, Muxi Lyu, Raluca Ada Popa, and Chia-Mu Yu. Daydreaming: Stealing hidden agent skills through black-box task interaction. *arXiv preprint arXiv:2608.26733*, 2026.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, et al. Voyager: An open-ended embodied agent with large language models. *Transactions on Machine Learning Research*, 2024a.

Jiongxiao Wang, Qiaojing Yan, Yawei Wang, Yijun Tian, Soumya Smruti Mishra, Zhichao Xu, Megha Gandhi, et al. Reinforcement learning for self-improving agent with skill library. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*, pp. 1529–1550, 2026a. doi: 10.18653/v1/2026.acl-long.69.

Junlin Wang, Tianyi Yang, Roy Xie, and Bhuwan Dhingra. Raccoon: Prompt extraction benchmark of llm-integrated applications. In *Findings of the Association for Computational Linguistics: ACL 2024*, 2024b. doi: 10.18653/v1/2024.findings-acl.791.

Xun Wang, Bihe Zhao, Michael Backes, Franziska Boenisch, and Adam Dziedzic. AgentProv: Auditing agentic LLM API providers via tool-use policy probes. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2026b.

Zihan Wang, Rui Zhang, Yu Liu, Chi Liu, Qingchuan Zhao, Hongwei Li, and Guowen Xu. Black-box skill stealing attack from proprietary llm agents: An empirical study. *arXiv preprint arXiv:2604.21829*, 2026c.

Zixun Xiong, Gaoyi Wu, Qingyang Yu, Mingyu Derek Ma, Lingfeng Yao, Miao Pan, Xiaojiang Du, et al. iSeal: Encrypted fingerprinting for reliable LLM ownership verification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2026. doi: 10.1609/aaai.v40i42.40909.

Jiashu Xu, Fei Wang, Mingyu Derek 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. doi: 10.18653/v1/2024.naacl-long.180.

Renjun Xu and Yang Yan. Agent skills for large language models: Architecture, acquisition, security, and the path forward. *arXiv preprint arXiv:2602.12430*, 2026. Accepted at the Agent Skills '26 Workshop at the ACM Conference on AI and Agentic Systems 2026.

Shojiro Yamabe, Futa Kai Waseda, Tsubasa Takahashi, and Koki Wataoka. MergePrint: Merge-resistant fingerprints for robust black-box ownership verification of large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, 2025. doi: 10.18653/v1/2025.acl-long.342.

Yuliang Yan, Haochun Tang, Shuo Yan, and Enyan Dai. DuFFin: A dual-level fingerprinting framework for LLMs IP protection. In *Findings of the Association for Computational Linguistics: EACL 2026*, 2026. doi: 10.18653/v1/2026.findings-eacl.273.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a.

Yong Yang, Changjiang Li, Qingming Li, Oubo Ma, Haoyu Wang, Zonghui Wang, Yandong Gao, et al. Prsa: Prompt stealing attacks against real-world prompt services. In *34th USENIX Security Symposium*, 2025b.

Yuchen Yang, Yiming Li, Hongwei Yao, Enhao Huang, Shuo Shao, Yuyi Wang, Zhibo Wang, et al. PromptCOS: Towards content-only system prompt copyright auditing for LLMs. In *2026 IEEE Symposium on Security and Privacy (SP)*, 2026.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023.

Simon Yu, Gang Li, Weiyan Shi, and Peng Qi. PolySkill: Learning generalizable skills through polymorphic abstraction for continual learning. In *International Conference on Learning Representations*, 2026.

Z.ai. GLM-4-9B-0414. Hugging Face model card, 2025.

Barry Zhang, Keith Lazuka, and Mahesh Murag. Equipping agents for the real world with agent skills. Anthropic Engineering, October 2025a.

Chi Zhang, Yimin Liu, Xinze Chen, and Ping Ji. What keeps agent skills from being reusable? evidence from 138k SKILL.md files. *arXiv preprint arXiv:2608.08453*, 2026.

Collin Zhang, John Xavier Morris, and Vitaly Shmatikov. Extracting prompts by inverting llm outputs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024a. doi: 10.18653/v1/2024.emnlp-main.819.

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*, 2025b.

Yiming Zhang, Nicholas Carlini, and Daphne Ippolito. Effective prompt extraction from language models. In *Conference on Language Modeling*, 2024b.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. ExpeL: LLM agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19632–19642, 2024. doi: 10.1609/aaai.v38i17.29936.

APPENDIX ROADMAP

A Method proofs: proof and stability bounds for Section 4.
B Method details: selection and scoring algorithm for Section 4.2.
C Experimental settings: candidate counts, SKILLLINEAGE settings, audit questions, and baseline implementations for Sections 4.2 and 5.1.
D Supplementary results: complete provenance results and host/task diagnostics for Section 5.2 (D.1); ablations and budget analyses for Section 5.4 (D.2); candidate-family and functional controls for Section 5.3 (D.3).
E Limitations and future work: configuration drift and skills internalized in model parameters.

A THEORETICAL PROOFS FOR THE METHOD

The analysis in Section 4 is supported here by the probe-design proof and bounds on fingerprint and score changes under host shifts.

A.1 PROBE DESIGN AND SCORE STABILITY

We prove Proposition 1 from Section 4.2, then bound host-induced changes in fingerprint separation and the provenance margin (Equation 10).

Proof of Proposition 1. Apply the reverse triangle inequality to Equation (3) and bound the host term by B. Since B is independent of Q, maximizing the resulting lower bound is equivalent to maximizing its skill term. □

Same-host separation and host shift. Fix the probes, runtime, generation settings, and fingerprint construction. For protected skill $S^\star$ and reference host M_r, define

$$\begin{aligned} d_r(S) &= \|f(M_r, S) - f(M_r, S^\star)\|_2, \\ d_v(S) &= \|f(M_v, S) - f(M_r, S^\star)\|_2, \\ \delta(S) &= \|f(M_v, S) - f(M_r, S)\|_2. \end{aligned} \tag{11}$$

For fixed fingerprints, the reverse triangle inequality gives

$$|d_v(S) - d_r(S)| \le \delta(S).$$

Repeated-sampling errors, when present, contribute additional perturbations.

Proposition 2 (Preserving separation across hosts)**.** *Suppose source-related S^+ and same-task independent S^- satisfy $d_r(S^+) \le a < b \le d_r(S^-)$, with host shifts bounded by ϵ_+ and ϵ_-, respectively. Then*

$$d_v(S^-) - d_v(S^+) \ge (b - a) - (\epsilon_+ + \epsilon_-). \tag{12}$$

Their ordering is preserved if $b - a > \epsilon_+ + \epsilon_-$.

Proof. The distance bound gives $d_v(S^+) \le a + \epsilon_+$ and $d_v(S^-) \ge b - \epsilon_-$. Subtract the first from the second. □

The condition preserves pairwise ordering when the reference-host separation exceeds the combined host shifts. Here $f(M_r, S)$ describes the hypothetical deployment of the suspect skill on the reference host.

Normalized-vector perturbation. For nonzero $\mathbf{x}$ and $\mathbf{x}+\mathbf{r}$,

$$\left\| \frac{\mathbf{x}+\mathbf{r}}{\|\mathbf{x}+\mathbf{r}\|_2} - \frac{\mathbf{x}}{\|\mathbf{x}\|_2} \right\|_2 \le \left\| \frac{\mathbf{x}+\mathbf{r}}{\|\mathbf{x}+\mathbf{r}\|_2} - \frac{\mathbf{x}+\mathbf{r}}{\|\mathbf{x}\|_2} \right\|_2 + \frac{\|\mathbf{r}\|_2}{\|\mathbf{x}\|_2}$$
$$= \frac{\big|\|\mathbf{x}+\mathbf{r}\|_2 - \|\mathbf{x}\|_2\big| + \|\mathbf{r}\|_2}{\|\mathbf{x}\|_2} \le \frac{2\|\mathbf{r}\|_2}{\|\mathbf{x}\|_2}. \tag{13}$$

The final inequality uses the reverse triangle inequality.

With fixed $\mathbf{w} = \mathbf{f}_+ - \mathbf{f}_-$ from Equation 10, define

$$S^{\text{raw}}_{\mathbf{w}}(\mathbf{a}) = \mathbf{w}^\top \mathcal{N}(\mathbf{a}). \tag{14}$$

Proposition 3 (Uncentered margin stability)**.** *For the implemented uncentered profile, let* $\mathbf{a}' = \lambda\mathbf{a}+\mathbf{r}$ *with* $\lambda > 0$ *and* $\|\mathbf{r}\|_2 < \lambda\|\mathbf{a}\|_2$*. Then*

$$|S^{\text{raw}}_{\mathbf{w}}(\mathbf{a}') - S^{\text{raw}}_{\mathbf{w}}(\mathbf{a})| \le \frac{2\|\mathbf{w}\|_2\|\mathbf{r}\|_2}{\lambda\|\mathbf{a}\|_2}. \tag{15}$$

Proof. Apply Equation (13) with $\mathbf{x} = \lambda\mathbf{a}$ and then Cauchy–Schwarz against $\mathbf{w}$. □

For the raw profile, a common offset contributes $b\mathbf{1}$ to the perturbation $\mathbf{r}$ and enters the score-change bound above.

For the numerical zero-norm threshold $\tau = 10^{-12}$, the bound applies when both profile norms exceed τ. A sufficient condition is $\lambda\|\mathbf{a}\|_2 - \|\mathbf{r}\|_2 > \tau$ and $\|\mathbf{a}\|_2 > \tau$.

B Additional Method Details

The audit in Section 4.2 is detailed below through its window-selection settings and scoring algorithm. Collection and numerical settings appear in Appendix C.2.

B.1 Selection and scoring algorithm

Algorithm 1 implements window selection and scoring for Section 4.2. For Equation 6, the candidate grid has 89 equally spaced positions from 0.08 to 0.95. We retain six positions with minimum spacing 0.075, breaking score ties by position, and store them in response order. Here p_{it} is the owner prefix for task x_i at observation t, g_{it} is the next-token text in the fixed owner response, and y^s_{it} is the text returned by the suspect. The function $\rho(g, y)$ is the cosine similarity of their text embeddings, defined in Equation 8.

Algorithm 1 Text-only provenance audit.

Require: Tasks $\{x_i\}_{i=1}^8$, owner prefixes/targets, reference traces T_+, T_-, suspect endpoint E_s
Ensure: Margin m and decision $\hat{y}$

for $i = 1, \dots, 8$ **do**
  $\{W_{ij}\}_{j=1}^6 \leftarrow \textsc{SelectWindows}(T_+^{(i)}, T_-^{(i)})$ ▷ Eq. 6
end for
$\mathbf{f}_\pm \leftarrow \mathcal{N}(\textsc{Profile}(T_\pm, W))$
for $i = 1, \dots, 8$ **do**
  $y^s_{it} \leftarrow E_s(x_i, p_{it})$ for each $t \in \bigcup_j W_{ij}$
  $a^s_{ij} \leftarrow |W_{ij}|^{-1} \sum_{t \in W_{ij}} \rho(g_{it}, y^s_{it})$ for $j = 1, \dots, 6$
end for
$\mathbf{f}_s \leftarrow \mathcal{N}(\text{vec}([a^s_{ij}]))$
$m \leftarrow \mathbf{f}_s^\top(\mathbf{f}_+ - \mathbf{f}_-)$
return $(m, \mathbb{I}[m > 0])$

C EXPERIMENTAL SETTINGS

Reproducing the audit and evaluation in Sections 4.2 and 5.1 requires the deployment definitions, collection settings, audit questions, and baseline configurations given below.

C.1 SKILLS AND DEPLOYMENTS

Table 4 breaks down the evaluation split in Section 5.1 by candidate family and target skill. Each candidate runs on all three open-source evaluation hosts. Source-related candidates include Original, Paraphrased, Daydreaming, and the target-specific skill-stealing reconstruction. Independent candidates include implementations generated with GLM and DeepSeek.

Table 4: **Skill deployment counts for Section 5.1.** Two skill stealing attacks represents Daydreaming (Tsai et al., 2026) and Wang et al. skill stealing (Wang et al., 2026c)

	Source-related				Non-source				
Target	**Original**	**Paraphrased**	**Daydreaming**	**Skill stealing**	**Independent**	**Wrong task**	**No-Skill**	**Per host**	**Total**
PDDL	1	1	1	1	5	6	1	16	48
PCAP	1	1	1	1	6	5	1	16	48
XLSX	1	1	1	1	5	6	1	16	48
Mesh	1	1	1	1	5	6	1	16	48
Radar	1	1	1	1	5	6	1	16	48
All	5	5	5	5	26	29	5	80	240

C.2 SKILLLINEAGE SETTINGS

The collection procedure and numerical conventions below implement the audit in Section 4.2 under the protocol in Section 5.1. For each task x_i, we use a fixed owner response r_i. The deployment loads the skill as a system message and the task as a user message. The collector uses the host tokenizer and chat template, places the owner prefix after the assistant marker, and requests one token with greedy decoding. Every query starts from its assigned owner prefix. The auditor supplies the task and prefix and observes the returned text. Each record contains the owner target (`gold`), returned token (`generated`), and token offsets (`start`/`end`).

We trim whitespace and retain records with nonempty owner and returned strings. For a retained sequence of length n, a window at relative position u starts at $k = \mathrm{round}(u(n-20))$ and averages 20 records using Equation 8. We use nearest-integer rounding. Each trace contains at least 20 retained records; missing probes and empty traces raise an error.

MiniLM encodes the owner and returned strings. We normalize the embeddings and clip their dot product to $[-1, 1]$ to obtain local agreement. Vectors with norm at most 10^{-12} are mapped to zero. A zero profile receives margin zero under the decision rule in Equation 10.

C.3 AUDIT QUESTIONS

The eight PDDL questions in Table 5 illustrate the task-specific probe design described in Section 4.2.

C.4 BASELINE IMPLEMENTATIONS

For the baselines in Section 5.1 and Table 1, we specify the queries, sampling settings, representations, and decision rules.

Prompt Detective[1] and **MET**[2]: Prompt Detective and MET use a shared bank of ten independently sampled responses per task and condition: eight tasks, 80 generated responses, temperature 1.0, top-$p = 1$, no top-k truncation, and a 50-token cap. We collect a separate sample for the Original reference. Prompt Detective uses its released `bert-base-uncased` representation and 10,000-permutation test; its provenance score is the p-value against the Llama Original sample. MET uses the

[1] https://github.com/amazon-science/prompt-membership-inference
[2] https://github.com/i-gao/model-equality-testing

Table 5: **Example audit tasks for PDDL.**

Probe	Task prompt
pddl-ev-01	A plan reaches the goal only if actions are reordered. Explain how to verify ordering constraints rather than checking the final state alone.
pddl-ev-02	The domain uses negative preconditions and conditional effects. State what must be checked before selecting a planner and how validation should be performed.
pddl-ev-03	Define a deterministic output convention when several valid PDDL plans have equal length, without changing plan correctness.
pddl-ev-04	A problem file references an object type not declared in the domain. Describe the expected failure handling and what evidence should be returned.
pddl-tr-01	Given a PDDL domain and problem, describe the steps for loading them, generating a plan, saving it in standard action format, and validating it.
pddl-tr-02	A planner returns no plan. Give a diagnostic procedure that distinguishes syntax errors, unsupported features, unsatisfied preconditions, and a genuinely unreachable goal.
pddl-tr-03	Explain how objects, typed parameters, predicates, preconditions, and effects interact when checking whether a grounded PDDL action is applicable.
pddl-tr-04	A generated plan looks plausible but fails validation. Describe how to identify the first invalid action and report the violated condition.

released MMD–Hamming statistic on the first 512 Unicode code points, padded for shorter responses. Its score is negative MMD against the same reference.

LLMmap[3]: We retain its eight released generic queries, greedy decoding, and 256-token response cap. The released `multilingual-e5-large-instruct` encoder and inference network produce a trace vector. The native provenance adaptation scores negative Euclidean distance to the Llama Original vector.

ZeroPrint[4]: We use the two targeted task probes and four GloVe-based lexical perturbations per probe. Each perturbed input substitutes three words using ten-nearest-neighbor candidates with seed 1000. Ten inputs with 20 responses each give 200 responses per condition, at temperature 0.7, top-$p = 0.9$, top-$k = 50$, and a 512-token cap. Its `all-mpnet-base-v2` embeddings and ridge coefficient $\alpha = 10^{-3}$ estimate a local response Jacobian; the two base-query estimates are averaged. Pearson correlation to Llama Original supplies the provenance score.

BBF: Following Chen et al. (2026b), we construct 64 task queries and a same-domain, different-task Llama reference per target. DQO ranks queries by increasing Original–unrelated centroid cosine similarity. Table 1 uses the first 16, with three samples each, temperature 0.7, and a 400-token output cap. MiniLM-L6-v2 encodes answers with a 256-token input cap; normalized embeddings are averaged per query. The score is the fraction of positive query-level similarity gaps, evaluated by deployment-level AUROC; the binary rule is a strict majority.

Binary-decision scope: SKILLLINEAGE uses a zero threshold with Llama references. BBF uses its native strict-majority rule without requiring the suspect host identity. The other four baselines provide ranking scores but no native binary skill-provenance rule under our unknown-host protocol, so their BAcc entries are N/A. Prompt Detective's 5% permutation test assumes a shared host; MET's 5% test concerns equality of complete output distributions. LLMmap returns the closest registered host class. ZeroPrint calibrates a Pearson-similarity threshold on labeled model-family comparisons by maximizing Youden's $J = \mathrm{TPR} - \mathrm{FPR}$.

D SUPPLEMENTARY RESULTS

The results below extend the main comparison (Section 5.2), design-choice and budget analyses (Section 5.4), and provenance–functionality controls (Section 5.3).

D.1 SUPPLEMENTARY MAIN RESULTS

We expand Section 5.2 with five-skill results and deployment margins, and examine host and task effects underlying Section 5.3.

[3] https://github.com/pasquini-dario/LLMmap

[4] https://github.com/shaoshuo-ss/ZeroPrint

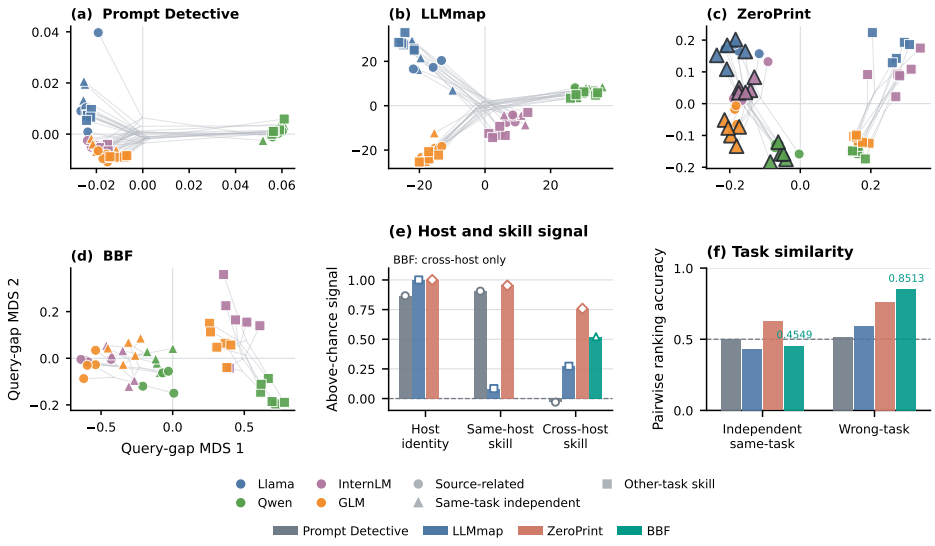


Figure 3: **Host and task similarity can obscure skill provenance.** (a–d) PDDL projections: colors denote hosts, shapes denote candidate families, and lines link the same artifact. (e) Host and provenance discrimination above chance; BBF has cross-host results only. (f) Same-task versus wrong-task ranking accuracy.

Table 6: **Cross-host skill provenance with one reference host.** This table expands Table 1. Within each skill and column, bold marks the best value and underlining the second-best distinct value; ties share formatting.

Method	Qwen ↑	InternLM ↑	GLM ↑	Pooled AUC ↑	BAcc ↑
(a) Target skill: PDDL					
Prompt Detective (Levin et al., 2025)	0.5000	0.5000	0.4583	0.4861	N/A
MET (Gao et al., 2025)	0.3750	0.8125	0.9167	0.5671	N/A
LLMmap (Pasquini et al., 2025)	0.7292	0.7500	0.6875	0.6597	N/A
ZeroPrint (Shao et al., 2026)	0.8125	0.8958	**1.0000**	0.8843	N/A
BBF (Chen et al., 2026b)	0.8333	0.7396	0.8333	0.7975	0.6111
SKILLLINEAGE	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
(b) Target skill: PCAP					
Prompt Detective (Levin et al., 2025)	0.5000	0.5000	0.5000	0.5000	N/A
MET (Gao et al., 2025)	0.7917	**1.0000**	0.8750	0.6782	N/A
LLMmap (Pasquini et al., 2025)	0.0208	0.3750	0.3333	0.3194	N/A
ZeroPrint (Shao et al., 2026)	0.6667	0.5833	0.7500	0.6667	N/A
BBF (Chen et al., 2026b)	0.5104	0.8958	0.6354	0.6181	0.4583
SKILLLINEAGE	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
(c) Target skill: XLSX					
Prompt Detective (Levin et al., 2025)	0.5000	0.5000	0.6250	0.5417	N/A
MET (Gao et al., 2025)	0.3125	0.5000	0.6458	0.5069	N/A
LLMmap (Pasquini et al., 2025)	0.5625	0.5833	0.7917	0.6065	N/A
ZeroPrint (Shao et al., 2026)	0.7708	0.6667	0.3125	0.5370	N/A
BBF (Chen et al., 2026b)	0.4583	0.5208	0.9167	0.5961	0.5000
SKILLLINEAGE	**1.0000**	**1.0000**	**0.9792**	**0.9931**	**0.9444**
(d) Target skill: Mesh					
Prompt Detective (Levin et al., 2025)	0.5000	0.5000	0.5000	0.5000	N/A
MET (Gao et al., 2025)	0.7917	0.5208	0.3958	0.5231	N/A
LLMmap (Pasquini et al., 2025)	0.4167	0.4167	0.7708	0.5023	N/A
ZeroPrint (Shao et al., 2026)	0.6250	0.7917	0.7292	0.6343	N/A
BBF (Chen et al., 2026b)	**1.0000**	0.9896	0.9271	0.9699	0.7917
SKILLLINEAGE	0.9792	**1.0000**	**1.0000**	**0.9954**	**0.9306**
(e) Target skill: Radar					
Prompt Detective (Levin et al., 2025)	0.5000	0.5000	0.5000	0.5000	N/A
MET (Gao et al., 2025)	0.9375	0.8958	0.6667	0.6157	N/A
LLMmap (Pasquini et al., 2025)	0.9792	0.8125	0.6667	0.7708	N/A
ZeroPrint (Shao et al., 2026)	0.3958	0.8750	0.6042	0.6412	N/A
BBF (Chen et al., 2026b)	0.7292	**0.9479**	0.9375	0.7627	0.5833
SKILLLINEAGE	**1.0000**	0.9375	**0.9583**	**0.9167**	**0.8333**

Full Five-Skill Comparison. Table 6 gives the full comparison for PDDL, PCAP, XLSX, Mesh, and Radar, extending Section 5.2. Each completed target–host pair contains four positive and twelve negative deployments.

Host and task similarity. Figure 3 shows why recognizing a host or task need not identify skill provenance. Panels (a–c) project output fingerprint of PDDL candidites on four hosts using Prompt Detective's four-reference similarity profile, LLMmap's trace, and ZeroPrint's Jacobian; (d) projects BBF's 16 query gaps on three evaluation hosts. In (e), provenance uses the same 42 deployments, including 9 source-related, 33 non-source, excluding original and no-skill. In particular, Prompt Detective falls from 0.9529 to 0.4848 AUROC when the reference host changes; LLMmap identifies hosts perfectly but reaches only 0.6364 cross-host provenance AUROC. In the separate three-skill comparison (f), BBF ranks wrong-task negatives at 0.8513 but independent same-task negatives at 0.4549. Together, these results highlight two challenges in provenance auditing: tracing source-related skills across host changes and distinguishing them from independent same-task implementations, as examined in Sections 5.2 and 5.3.

Deployment Margins. Figure 4 shows the individual margins underlying Section 5.2. For PDDL and PCAP, the zero boundary separates source-related and non-source deployments on all three hosts. XLSX non-source margins lie closer to zero, with three independent-skill deployments and one other-task deployment crossing the boundary on Qwen and GLM. All source-related deployments remain positive, while InternLM correctly separates every XLSX candidate.

D.2 Ablation and Mechanism Analyses

To examine the design choices in Section 5.4, we break down representation performance by negative type and vary the probe, anchor, and observation budgets.

In Table 2, arrows compare controlled changes within each diagnostic collection; values are pooled-host AUROC. Independent uses same-task negatives. Block (a) uses PDDL; (b)–(c) report three-skill macro averages. In (c), only the negative-reference term changes.

Local correspondence by negative type. This analysis tests whether the representation gains in Section 5.4 also hold against independently developed skills for the same task. Table 7 breaks down the comparison in Table 2(b) by target and negative type, using the same saved responses and Llama reference libraries. Compared with generated traces, raw 48-D correspondence raises macro AUROC from 0.883 to 0.989 for independent negatives, while it changes from 0.988 to 0.978 for other-task negatives. The largest gain occurs on XLSX independent skills, from 0.656 to 0.967. These results support local correspondence for distinguishing provenance.

Table 7: **Representation results by negative type for Table 2(b).** Entries are pooled-host AUROC; Macro averages the three targets.

	Independent negatives				Other-task negatives			
Representation	**PDDL**	**PCAP**	**XLSX**	**Macro**	**PDDL**	**PCAP**	**XLSX**	**Macro**
Generated trace (per probe)	0.994	**1.000**	0.656	0.883	**1.000**	**1.000**	0.963	0.988
Local correspondence, 8-D	0.983	**1.000**	0.839	0.941	0.981	**1.000**	0.856	0.946
Local 48-D, raw	**1.000**	**1.000**	0.967	0.989	**1.000**	**1.000**	0.935	0.978
Local 48-D, centered	**1.000**	**1.000**	**0.978**	**0.993**	**1.000**	**1.000**	**0.991**	**0.997**

Observation budget. Figure 5 and Table 8 extend the budget analysis in Section 5.4, varying probes K, anchors per probe A, or observations per window L on the same 144-deployment subset covering PDDL, PCAP, and XLSX. The table also reports observation counts and errors on independent skills. Four anchors per probe give AUROC 0.9668 and BAcc 0.8981; ten observations per window give 0.9514 and 0.7454. Shorter windows increase false positives.

D.3 Provenance and Functionality Analyses

Rankings by candidate family and controls with matched task outcomes provide a closer look at the provenance–functionality distinction in Section 5.3 and Figure 2.

Candidate-family ranking analysis. Table 9 expands Figure 2(c) and Table 1 in Sections 5.2 and 5.3 with separate results for each source-related transformation and pairwise error counts. Among the baselines, BBF ranks reconstructed skills most accurately (0.6420), while ZeroPrint leads on paraphrased skills (0.7160) and Daydreaming reconstructions (0.7870).

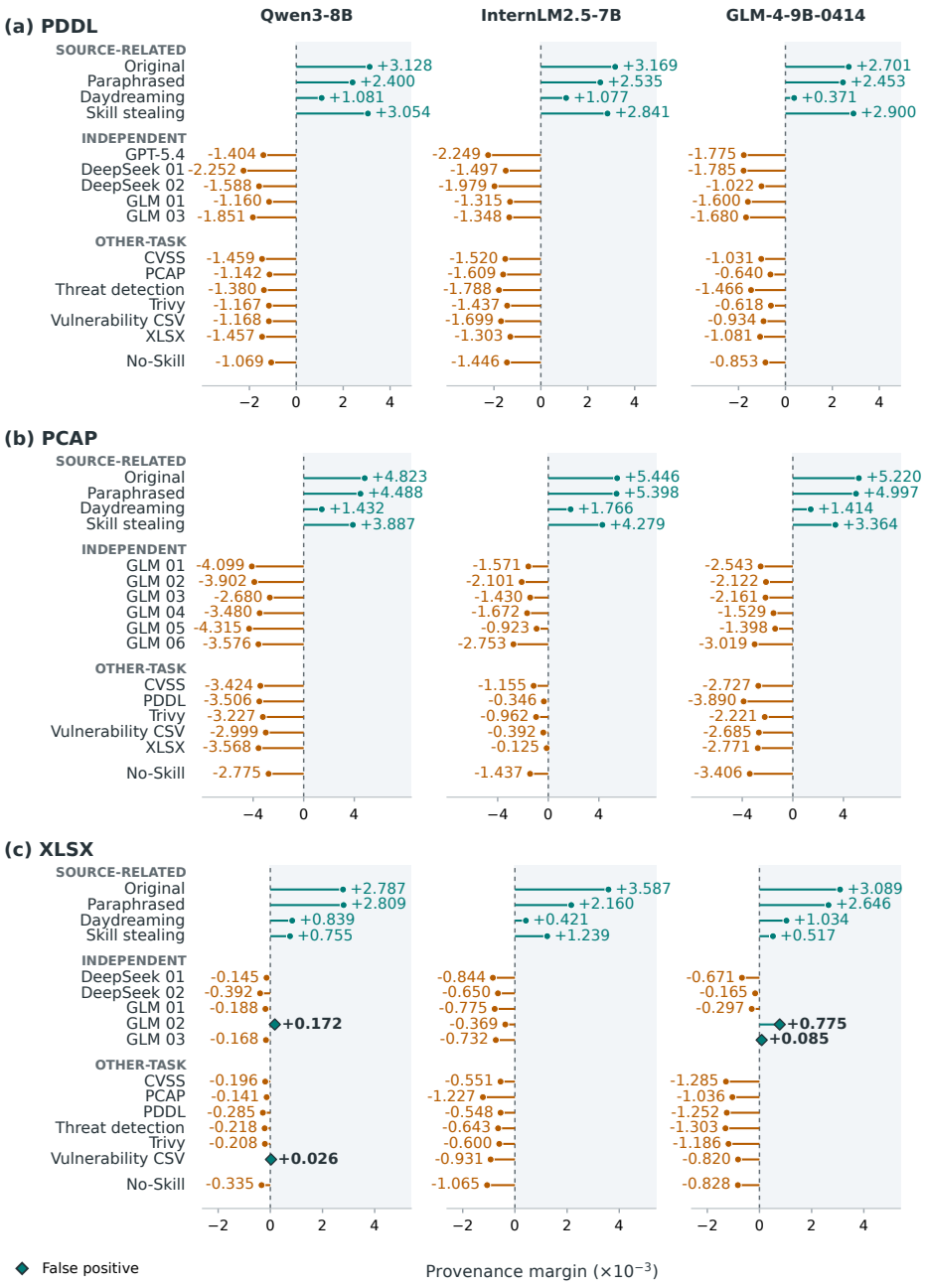


Figure 4: **Deployment margins for the 144-case, three-skill subset in Table 1.** Each point is one deployment, with margins shown beside the points. Columns separate the three hosts; axes share a scale within each task. The dashed line is the zero boundary. Teal denotes positive margins and orange negative margins; diamonds mark the four false positives. Independent rows identify the generating model.

Functional Controls. Tables 10 and 11 supplement the matched-success comparison in Section 5.3 and Figure 2(d). They add check accuracy, the public PracticalSwan control[5], and the individual

[5]We use PracticalSwan's public `excel-sheet` skill for creating and editing Excel workbooks, identified in a follow-up web search and included as an external independent control: https://github.com/PracticalSwan/agent-skills/blob/afbb609dec5b7bf08c8384a013b6b3b535517da2/excel-sheet/SKILL.md.

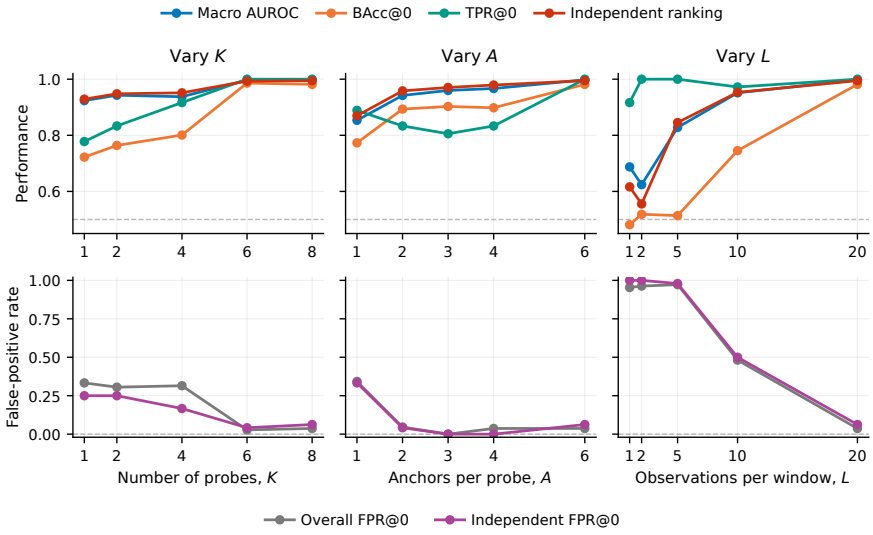


Figure 5: Budget sensitivity for probes K, anchors A, and observations per window L. Performance metrics appear above and false-positive rates below.

Table 8: **Complete budget results.** Coordinates equal KA; slots equal KAL per profile. Unique is the mean and range after deduplicating overlapping windows. Total unique sums those counts over 144 suspects and six Llama reference profiles. Independent ranking uses 576 target-stratified positive–independent comparisons. Best metrics within each budget group are bold.

Budget	Coord.	Slots	Unique/profile	Total unique	AUROC	BAcc	TPR	FPR	Indep. FPR	Indep. rank
Vary K; fix $A = 6, L = 20$										
$K = 1$	6	120	111.5 [93–120]	16728	0.9236	0.7222	0.7778	0.3333	0.2500	0.9288
$K = 2$	12	240	227.8 [201–240]	34166	0.9429	0.7639	0.8333	0.3056	0.2500	0.9479
$K = 4$	24	480	460.3 [416–480]	69046	0.9375	0.8009	0.9167	0.3148	0.1667	0.9514
$K = 6$	36	720	699.7 [654–720]	104948	0.9969	**0.9861**	**1.0000**	**0.0278**	**0.0417**	0.9931
$K = 8$	48	960	933.1 [873–960]	139969	**0.9977**	0.9815	**1.0000**	0.0370	0.0625	**0.9948**
Vary A; fix $K = 8, L = 20$										
$A = 1$	8	160	160.0 [160–160]	24000	0.8534	0.7731	0.8889	0.3426	0.3333	0.8698
$A = 2$	16	320	317.9 [313–320]	47684	0.9421	0.8935	0.8333	0.0463	0.0417	0.9583
$A = 3$	24	480	473.6 [459–480]	71038	0.9599	0.9028	0.8056	**0.0000**	**0.0000**	0.9705
$A = 4$	32	640	630.7 [608–640]	94602	0.9668	0.8981	0.8333	0.0370	**0.0000**	0.9792
$A = 6$	48	960	933.1 [873–960]	139969	**0.9977**	**0.9815**	**1.0000**	0.0370	0.0625	**0.9948**
Vary L; fix $K = 8, A = 6$										
$L = 1$	48	48	47.9 [46–48]	7182	0.6875	0.4815	0.9167	0.9537	1.0000	0.6163
$L = 2$	48	96	95.4 [93–96]	14303	0.6242	0.5185	**1.0000**	0.9630	1.0000	0.5556
$L = 5$	48	240	237.6 [231–240]	35633	0.8279	0.5139	**1.0000**	0.9722	0.9792	0.8455
$L = 10$	48	480	472.1 [453–480]	70810	0.9514	0.7454	0.9722	0.4815	0.5000	0.9531
$L = 20$	48	960	933.1 [873–960]	139969	**0.9977**	**0.9815**	**1.0000**	**0.0370**	**0.0625**	**0.9948**

Table 9: **Candidate-family pairwise ranking accuracy.** Each column pools the three target skills and three evaluation hosts.

Method	Independent	Wrong-task	Paraphrased	Daydreaming	Reconstructed
Prompt Detective	0.5026 [286.5/576]	0.5147 [297/612]	0.5509 [145.5/324]	0.4954 [163.5/324]	0.4954 [163.5/324]
MET	0.6024 [229/576]	0.5735 [261/612]	0.6173 [124/324]	0.5926 [132/324]	0.5556 [144/324]
LLMmap	0.4340 [326/576]	0.5899 [251/612]	0.5000 [162/324]	0.5247 [154/324]	0.5000 [162/324]
ZeroPrint	0.6250 [216/576]	0.7598 [147/612]	0.7160 [92/324]	0.7870 [69/324]	0.5710 [139/324]
BBF	0.4549 [314/576]	0.8513 [91/612]	0.6759 [105/324]	0.6173 [124/324]	0.6420 [116/324]
SKILLLINEAGE	**0.9948** [3/576]	**1.0000** [0/612]	**1.0000** [0/324]	**0.9969** [1/324]	**0.9938** [2/324]

task outcomes. PracticalSwan passes five tasks and has a margin of $+0.000003$, below Original's $+0.002787$.

Table 10: **Executable XLSX success and provenance margins on Qwen3-8B.** D/G denote DeepSeek/GLM-generated independent skills.

Candidate	Passes	Check acc.	Current raw margin
Original	4/7	0.839	+0.002787
Independent D1	4/7	0.835	-0.000145
Independent D2	4/7	**0.892**	-0.000392
Independent G1	4/7	0.827	-0.000188
Independent G2	4/7	0.851	+0.000172
Independent G3	4/7	0.834	-0.000168
Public PracticalSwan	**5/7**	–	+0.000003

Table 11: **Task-level outcomes for Figure 2(d).** One denotes pass and zero denotes fail.

Task	Original	D1	D2	G1	G2	G3
bulk_filter_bold	1	1	1	1	1	1
conditional_color	1	1	1	1	1	1
edit_formula_totals	1	1	1	1	1	1
edit_sparse	0	0	0	0	0	0
moving_avg	0	0	0	0	0	0
reorder_cols	0	0	0	0	0	0
style_header	1	1	1	1	1	1

E Limitations and Future Work

Our current study focuses on *deployment-time skill provenance*: determining, from black-box outputs, whether a suspect agent is operating with a skill derived from the protected target skill. This setting assumes that the relevant skill remains an explicit component of the deployed agent and influences its behavior at inference time. We do not yet consider a stronger setting in which the host model is further trained, fine-tuned, or otherwise adapted using the target skill such that the corresponding capability is internalized into the model parameters and the original skill is no longer attached at inference time. Whether provenance evidence remains detectable after such skill-to-parameter internalization is an open question. Extending black-box skill provenance auditing to this setting is an important direction for future work.