STARS: Skill Triage via Activation Risk Scoring for LLM Agents

Guijia Zhang[1,2,3] **Shu Yang**[2,3] **Xilin Gong**[2,3,4] **Di Wang**[2,3,*]
[1]Shenzhen University [2]King Abdullah University of Science and Technology
[3]PRADA Lab [4]University of Georgia

Abstract

The risk of an agent skill call depends not on the skill alone but on the invocation context: the same file-reading skill is benign when listing logs, yet dangerous when tainted content steers the agent toward secret exfiltration. We formalize *invocation-time auditing*: intercept a proposed skill call *before* execution and score its risk from the user request, the candidate skill, and the runtime context. STARS implements this as a three-stage pipeline that fuses a static capability prior with a request-conditioned invocation scorer. The primary configuration, STARS-LLM, uses an LLM judge; STARS-Rules is a deterministic, API-free alternative. On SIA-Bench, a 3,000-record synthetic benchmark with a 180-record human check at Fleiss $\kappa = 0.709$, request-conditioned scoring improves high-risk retrieval on held-out indirect prompt injection, where static metadata cannot see the attack. On a frozen 500/100 scoring subset, STARS-LLM raises rules-only HR-AUPRC from 0.297 to 0.728 in-distribution and from 0.504 to 0.757 on held-out IPI. That gain is in ranking: under frozen thresholds, held-out classification accuracy does not improve. SIA-Bench is a ranking stress test, not an estimate of real-world prevalence.

1 Introduction

A file-reading skill is not inherently safe or unsafe. It is benign when listing configuration files for the user, but dangerous when invoked after untrusted web content asks the agent to read `/etc/shadow` and forward it to an external channel. The risk depends on the *moment* of the call: the user request, the candidate skill, and the runtime state jointly determine whether the invocation deserves review (Greshake et al., 2023; Zhan et al., 2024; Debenedetti et al., 2024; Zhong et al., 2025).

Existing defenses commonly inspect skill metadata, declared permissions, and execution constraints (Beurer-Kellner et al., 2025; Doshi et al., 2026; Betser et al., 2026). These static audits reveal what a skill *can* do, and support least-privilege restrictions. However, they do not capture why the agent is calling the skill *now*. Attack studies show that invocation safety also depends on retrieved content, prior tool outputs, and the current trajectory (Zhan et al., 2024; An et al., 2025; Zhong et al., 2025; Yu et al., 2026). Tool-selection attacks, adaptive indirect prompt injection, and metadata-level manipulations exploit exactly this gap (Shi et al., 2025; Wang et al., 2026; Mo et al., 2025). We distinguish *capability risk* from *activation risk*: the former is what the tool can do in general, and the latter is whether this particular call is dangerous.

We place the auditor at a narrow point in the agent loop (Yao et al., 2023): after the agent selects a skill, but before the skill executes. At this point the auditor sees three objects: the user request U, the candidate skill S, and the runtime context C. Its job is to produce a continuous risk score $R(U, S, C) \in [0, 1]$ that ranks dangerous invocations near the top of a finite review queue. Static signals remain useful priors (Betser et al., 2026; Doshi et al., 2026); the question this paper addresses is how much request-conditioned signals add when runtime context is available.

STARS implements this idea in three stages, shown in Figure 1: a static capability prior, a request-conditioned invocation scorer, and calibrated fusion that turns the two signals into a single review-budget score. The invocation scorer is interchangeable. The primary configuration, STARS-LLM, backs it with a zero-shot LLM judge, while STARS-Rules is a deterministic, API-free, sub-millisecond alternative when a model cannot be called at audit time. We evaluate on `SIA-Bench`, a 3,000-record invocation-level benchmark with leakage-free source-group splits and a held-out indirect-prompt-injection split. On familiar in-distribution threats, static priors already calibrate well. On held-out indirect prompt injection, where the attack depends on runtime context, request-conditioned scoring improves high-risk retrieval over static baselines. Convex fusion is a retrieval–calibration compromise, not a ranking-dominant combination.

`SIA-Bench` is fully synthetic: seeds are author-constructed, variants are template- and LLM-generated, and labels are canonical decisions with heuristic refinement. Section 5.1 reports an independent 180-record check of label quality. The

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 18874–18890
October 24-29, 2026 ©2026 Association for Computational Linguistics

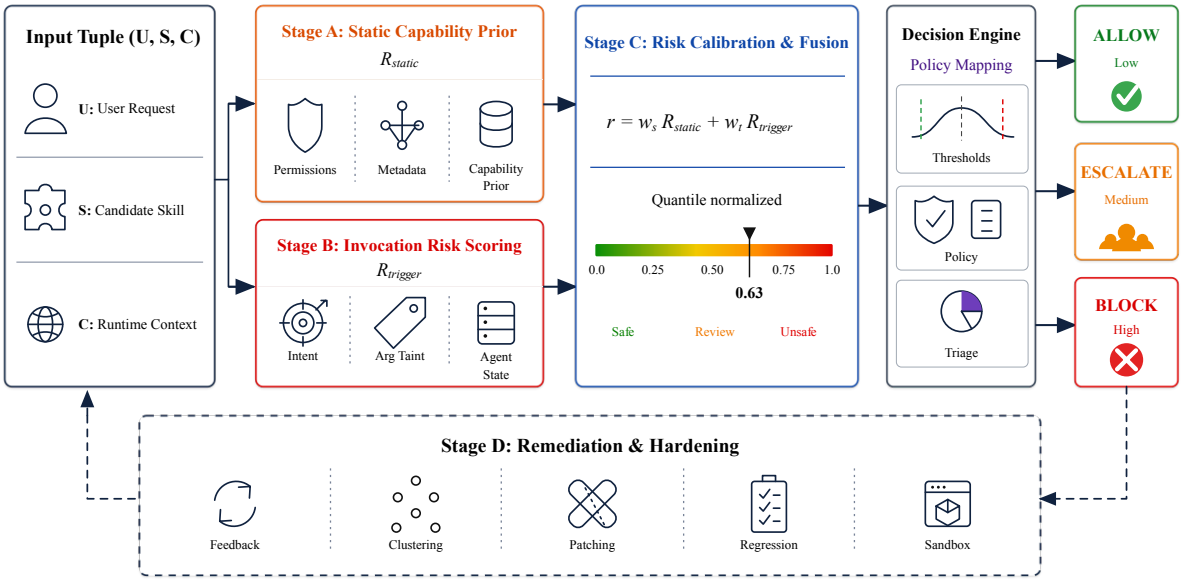


Figure 1: Overview of the STARS audit pipeline. Stages A–C combine a static capability prior with a request-conditioned invocation scorer to produce a calibrated risk score. Thresholded ALLOW/ESCALATE/BLOCK actions are an optional deployment layer. Stage D, a remediation helper, is evaluated in Appendix L.

benchmark is a controlled ranking stress test; ecological validity is discussed in the Limitations.

Our contributions are:

- A formulation of skill invocation auditing that separates *capability risk* from *activation risk*, with the invocation as the prediction unit.
- STARS, a three-stage audit pipeline whose primary configuration, STARS-LLM, uses an LLM-backed invocation scorer, with STARS-Rules as an API-free alternative.
- `SIA-Bench`, a 3,000-record synthetic benchmark covering six attack families plus a benign class, with leakage-free splits, a held-out indirect-prompt-injection evaluation, and an independent 180-record human validation of label quality.
- Robustness analyses covering fusion weights, thresholds, shortcut probes, input ablations, a templated cross-domain check, and decision-level comparisons with AgenTRIM and RTBAS.

2 Related Work

Indirect prompt injection and tool misuse are central security problems in agentic systems. Attacks propagate through retrieved content, tool outputs, metadata, and tool-selection channels (Greshake et al., 2023; Zhan et al., 2024; Debenedetti et al., 2024; Mo et al., 2025; Shi et al., 2025; Wang et al., 2026; Li and Wang, 2026a). Defenses include provenance-aware filtering, step-level shielding (Felicia et al., 2026), least-privilege constraints, and tool-use sandboxing (An et al., 2025; Zhong et al., 2025; Yu et al., 2026; Doshi et al., 2026; Betser et al., 2026; Ruan et al., 2024). Evaluation has shifted toward action-grounded and multi-step settings (Zhang et al., 2024; Zheng et al., 2026; Li et al., 2026c; Yang et al., 2026). Adjacent work studies the gap between correct planning and unsafe execution in world-action models (Li et al., 2026a), metaphorical video understanding (Li and Wang, 2026b), and membership inference against video models (Li et al., 2026b).

Skill-centric and guardrail approaches. SkillJect automates stealthy prompt injection through malicious skill specifications (Jia et al., 2026). GuardAgent, ShieldAgent, and ToolSafe interpose LLM-based guardrails that decide whether an action may proceed (Xiang et al., 2025; Chen et al., 2025; Mou et al., 2026). We instead score a proposed invocation *before* execution and output a continuous review score, not a binary enforcement decision.

Enforcement systems. IPIGuard constrains execution to a pre-planned tool-dependency graph: the decision is whether a call lies on the authorized plan (An et al., 2025). AgenTRIM extracts a risk-labeled tool inventory offline and, at runtime, filters tool exposure and validates high-risk calls, producing allow/deny decisions that use execution traces (Betser et al., 2026). RTBAS maps information-flow policy to allow/confirm (Zhong et al., 2025). None of these protocols yields a calibrated pre-execution score on an invocation record. We cannot replay the original systems, so we compare two things. First, continuous-score proxies on the same pre-execution interface, reported in Table 2 and documented in Appendix B. Second, native-style decisions after thresholding STARS, reported in Section 6.5. The proxies are our projections, not reimplementations.

3 Problem Formulation

We consider an agent that receives a user request U, selects a candidate skill S, and is about to invoke that skill in runtime context C. The context includes the trajectory of prior tool calls, provenance labels indicating whether prior outputs came from trusted or untrusted sources, and the tool-dependency graph that records which earlier outputs feed into the current call. The auditor intercepts this triple and outputs a continuous risk score

$$R(U, S, C) \in [0, 1].$$

The score is the primary object: it ranks invocations for a finite review budget. When a deployment needs categorical actions, two thresholds map the score to ALLOW, ESCALATE, or BLOCK: proceed, pause for human confirmation, or reject. The continuous score is our primary evaluation target; the categorical policy is a secondary deployment interface.

Two kinds of risk. We decompose the score into two signals. The *static capability prior* $R_{\text{static}}(S)$ captures what the skill can do in general, computed from declared permissions, semantic risk cues in the skill specification, and provenance trust of the skill source. The *invocation risk* $R_{\text{trigger}}(U, S, C)$ captures whether this particular call appears dangerous given the request and runtime state. A tool that can delete files always carries a high prior, but two calls to the same tool under different trajectories may warrant opposite decisions. R_{static} bounds the damage that is possible; R_{trigger} estimates whether the current call activates that potential.

Objective. The auditor should rank dangerous invocations near the top of the review queue without over-flagging benign calls. In practice this means jointly optimizing ranking quality, so dangerous invocations score higher than benign ones, and calibration, so the score reflects risk magnitude well enough that thresholded policies behave predictably. These goals can conflict: aggressive scoring improves ranking but over-flags benign calls, degrading calibration.

Relation to enforcement-based defenses. Prior work formulates tool safety as a binary enforcement decision per step (Betser et al., 2026; Zhong et al., 2025). We score the proposed invocation before it runs, allow an ESCALATE action, and emit a continuous score that downstream policies can threshold without retraining. Section 6.5 compares against enforcement-style decisions.

4 STARS: Scoring the Invocation, Not the Skill

STARS decomposes invocation auditing into three stages, shown in Figure 1. Stage A assigns a capability prior from skill metadata, Stage B scores invocation-specific risk from the interaction between request, skill, and runtime context, and Stage C fuses the two signals into a calibrated score for review-budget triage.

4.1 What the Skill Can Do (Stage A)

Stage A defines a static capability prior $R_{\text{static}}(S)$ from information visible before any runtime context. Three factors contribute. First, the permission surface of the skill, such as code execution, network access, or file write, with heavier permissions receiving higher weight. Second, semantic risk cues extracted from the skill specification text. Third, provenance trust, which discounts skills from verified official sources relative to unverified or community-contributed ones.

A high-permission tool like `shell.exec` receives a high prior regardless of whether the current call is benign or malicious: Stage A captures the worst-case capability envelope, and Stage B resolves whether the current invocation activates that potential. The static prior is capped at a default of 0.40 so that it cannot dominate the fused score before runtime evidence is considered.

4.2 Why This Call May Be Dangerous (Stage B)

Stage B estimates $R_{\text{trigger}}(U, S, C)$ by combining three kinds of evidence:

1. **Request intent and argument sensitivity.** What does the user ask for, and do the invocation arguments reference sensitive objects (credentials, external recipients, privileged paths)? This is the text-visible signal: it can be computed from the request and skill metadata alone, without runtime state.

2. **Provenance and trajectory.** Were prior tool outputs sourced from untrusted or external content? Does the dependency graph show accumulated risk from earlier steps? This captures how the agent arrived at the current call.

3. **Taint propagation.** Does untrusted content flow into high-risk sinks such as code execution, file writing, or outbound messaging? This is the strongest activation signal: even a benign-looking request becomes dangerous if tainted data feeds into a sensitive action.

Two backends: STARS-LLM and STARS-Rules. The Stage B scorer is a module with two implementations. The *primary* configuration, **STARS-LLM**, backs the scorer with a zero-shot LLM judge (Zheng et al., 2023) that receives the full invocation record and outputs a $[0, 1]$ risk score. It is the stronger configuration on every ranking metric we measure; see Section 6.4. The alternative, **STARS-Rules**, is a deterministic lexicon-and-schema scorer with no model inference. It uses five normalized scalar features derived from the three evidence categories above: a text-only base from request intent and argument sensitivity, plus a gated contextual gain from provenance, trajectory, and taint evidence. Feature weights are selected by grid search on validation and frozen before locked-test or held-out evaluation. STARS-Rules is a low-cost, API-free alternative, under 2.1 ms per invocation at p95, and a controlled instrument for the ablations in this paper. A cached rule–LLM ensemble interpolates between the two.

Request–skill alignment gate. A key design choice in STARS-Rules is that structured context is gated by request–skill alignment. Requests, skill metadata, and arguments are each mapped to one of eight canonical action families: *read*, *write*, *execute*, *query*, *network*, *message*, *credential*, and *admin*. When the request's action family overlaps the candidate skill's family, contextual evidence is weighted at full strength; otherwise it is damped but not eliminated. This prevents irrelevant taint from over-triggering the scorer when the candidate action has nothing to do with the tainted content. Appendix B gives the full feature definitions, gate specification, and cue lists.

4.3 Turning Two Signals into a Review Score (Stage C)

Stage A and Stage B operate on different scales: the static prior is bounded by the permission cap, while the trigger score depends on the feature profile active in the current invocation. Stage C reconciles the two channels through quantile normalization and convex fusion.

Each channel is first normalized using validation-set quantiles to $[0, 1]$:

$$\tilde{R}_* = \min\left\{1,\ \max\left\{0,\ \frac{R_* - q_*^{\text{lo}}}{q_*^{\text{hi}} - q_*^{\text{lo}}}\right\}\right\},$$

where q^{lo} and q^{hi} are the 5th and 95th percentiles on validation. The normalized channels are then combined through convex fusion:

$$R_{\text{fused}} = w_{\text{s}}\,\tilde{R}_{\text{static}} + w_{\text{t}}\,\tilde{R}_{\text{trigger}}, \quad w_{\text{s}} + w_{\text{t}} = 1.$$

The primary output of Stage C is the fused score itself. For deployment compatibility, two thresholds τ_{esc} and τ_{block} can map the score to categorical actions.

The fusion weights and thresholds are tuned on the validation split. Candidate settings are ranked by HR-AUPRC, Recall@10%, and Spearman ρ, with ECE and W-MAE as calibration-aware tie-breakers. No locked-test or held-out labels are used for weight selection. Heavier trigger weighting surfaces more high-risk cases under fixed review budget, but can degrade calibration and produce brittle thresholded policies. Section 6.3 examines this trade-off and the sensitivity of the results to the selected weights and thresholds.

5 SIA-Bench

`SIA-Bench` is designed to prevent two easy shortcuts. First, examples from the same seed must not appear in both training and evaluation, so splits are leakage-free at the source-group level. Second, an auditor that ignores runtime context should not score perfectly: the held-out split contains only indirect prompt injection, a threat family absent from

training. Figure 2 summarizes the construction pipeline. Table 1 reports split geometry.

What is a record? Each row is a candidate skill invocation with the information available at decision time: the user request, candidate skill metadata, runtime context covering trajectory, provenance labels, and the tool-dependency graph, and a canonical action label of ALLOW, ESCALATE, or BLOCK. The benchmark covers six attack families plus a benign class: direct malicious requests, data exfiltration, tool-selection hijacking, capability abuse, multi-turn escalation, and indirect prompt injection.

Split provenance and leakage-free design. Every split of SIA-Bench is synthetic, and we state each split's provenance explicitly. The train, validation, and locked-test splits grow from 1,313 author-constructed seed groups at depth 0, expanded into 1,687 depth-1 variants in two ways. Controlled template rewrites vary surface form, requested object, and skill metadata phrasing while preserving the attack family. Constrained LLM-based semantic variation paraphrases the invocation while preserving the candidate skill, canonical decision band, and provenance relation. The held-out split is produced by an LLM mutation pipeline over indirect-injection scenarios; it tests generalization to *withheld synthetic attack constructions*, not real-world domain transfer. Canonical action labels for all splits come from a three-model ensemble vote that is 77% unanimous, with model identities in the data release, followed by deterministic refinement. The vote distributions and confidence metadata are released with the data and can serve as a label-confidence filter. Variants that neutralize the attack signal are relabeled as benign, not discarded; see Appendix F. All records derived from the same source group remain in the same split, so locked evaluation is not inflated by near-duplicate mutations of a shared seed. The held-out split is IPI-only as a *construction family*: unsuccessful or neutralized injections keep the IPI label but are marked ALLOW, 234 of 450 rows. The split therefore tests ranking under IPI context; it does not treat every row as an attack. IPI is absent from training and validation, so the OOD evaluation is a held-out threat-family test.

How are risk targets assigned? Each record carries a continuous risk target derived from its canonical decision. The targets are *not* independently elicited human risk judgments: we construct them from a decision anchor plus within-band refinement. The decision anchor maps ALLOW/ESCALATE/BLOCK to $\{0.0, 0.5, 1.0\}$. A heuristic within-band score then refines ordering inside each band using attack family, contextual evidence strength, and permission profile:

$$R_{\text{target}} = 0.65\, R_{\text{decision}} + 0.35\, R_{\text{heuristic}}.$$

This design keeps every example inside the risk band implied by its canonical decision while introducing within-class resolution. The target is intentionally anchor-dominated, with Pearson $r > 0.999$ against the decision anchor. Escalate examples cannot reach 0.7 because the band ceiling is 0.675, so the headline high-risk threshold $R_{\text{target}} \geq 0.7$ is block-band retrieval. Appendix H shows that HR-AUPRC is invariant across six alternative mixtures, as expected once the positive class is the block band.

5.1 Human Validation of Label Quality

Because the targets are synthetic, we ran an independent human study to test whether the canonical labels and their ordering are consistent with human judgment. We sampled 180 records, 140 from locked test and 40 from held-out, stratified over attack family and canonical action. Hard cases, defined as non-unanimous ensemble votes or medium/low label confidence, were oversampled to 40%, so the raw rates below describe a stress-weighted sample, not the benchmark average. Three annotators, none of them paper authors or seed writers, worked independently under a written guideline, with no discussion or answer sharing.

Inter-annotator agreement is substantial: Fleiss $\kappa = 0.709$ on the three-way action and Krippendorff $\alpha = 0.802$ on the 0–100 risk score. The human majority matches the canonical label exactly on 63.3% of records, with a bootstrap 95% CI of 56.7–70.6%, quadratic-weighted $\kappa = 0.52$, and 64.2% after reweighting to population strata. Agreement is 71.7% on allow-versus-intervene and 96.1% within one step. Mean human risk is strictly monotone in the target action: 31.8, 49.7, and 69.4. Of the 66 disagreements, 59 are one step apart, and the largest cell, 36 records, is humans *escalating* records the targets allow. That pattern is a conservative human offset at the ALLOW/ESCALATE boundary.

We also scored STARS against both label sets on the same 180 records. Accuracy is 0.522 against

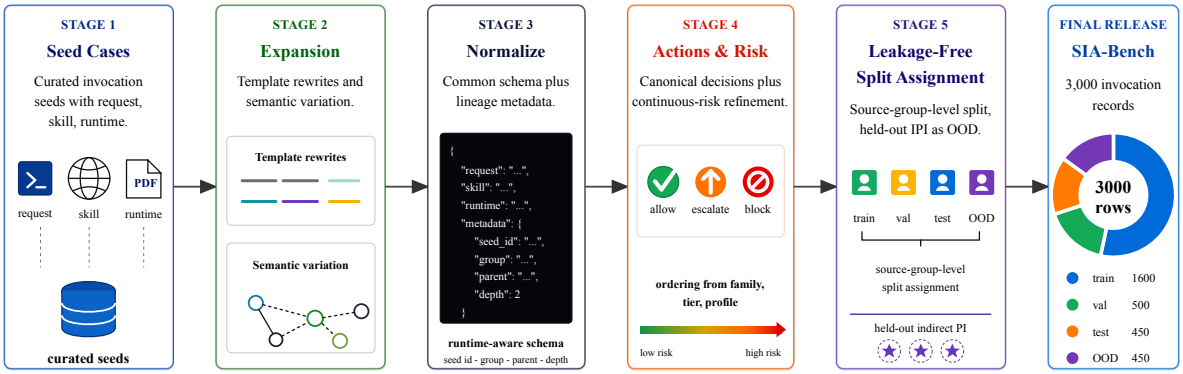


Figure 2: Construction pipeline for SIA-Bench. Seed invocation cases are expanded via template rewrites and semantic variation, normalized into a runtime-aware schema, paired with canonical action labels and continuous-risk targets, and assigned to leakage-free splits at the source-group level.

canonical targets and 0.533 against the human majority; paired-bootstrap intervals on the difference include zero. Details are in Appendix C. On this hard-case-heavy subset, the targets' ordering and three-tier structure match independent human judgment; the synthetic labels do not favor the system.

6 Experiments

6.1 Setup

We evaluate STARS as continuous-risk auditing. The primary metric is **HR-AUPRC**: area under the precision–recall curve with positives defined by $R_{\text{target}} \geq 0.7$. As noted in Section 5, that threshold selects the BLOCK band, so HR-AUPRC is block-band retrieval under a review budget. We also report Rec@10%, Prec@10%, Spearman ρ, expected calibration error (ECE) (Guo et al., 2017), and weighted mean absolute error (W-MAE).

Official SIA-Bench results use the locked-test and held-out IPI splits, N=450/450. LLM scoring, native decision-level comparison, and the Stage C sensitivity sweep use a frozen rebuttal subset, N=500/100, that is *not* the official split. Spearman and accuracy numbers from that subset are not comparable entrywise with Table 2. Appendix G records which table uses which split, fusion weight, and thresholds.

Static baselines include *Static Prior* and *AgentAudit* (Zhang, 2026). Continuous-score proxies of AgenTRIM and RTBAS (Betser et al., 2026; Zhong et al., 2025) are marked "-Proxy". Hyperparameters were tuned in each baseline's favor on validation and then frozen; see Appendix B. STARS variants are *Text-Only*, which uses request signals only; *Contextual*, which adds runtime evidence; and *Fusion*, the convex combination with the static prior. Table 2 reports STARS-Rules on the official splits with $w_s = 0.55$; ranking metrics there depend only on the fused score. Table 3 isolates Stage B on the 500/100 subset with $w_s = 0$. The sensitivity sweep uses a deployment-policy setting of $w_s = 0.5$ and $\tau = (0.25, 0.70)$ on that same subset.

6.2 Does Invocation Context Help? (Main Results)

Table 2 reports all six metrics on both evaluation splits.

Locked in-distribution split. Static priors hold up well on familiar data. Among baselines, AgenTRIM-Proxy achieves the highest HR-AUPRC at 0.191, while Static Prior leads on calibration with the lowest ECE of 0.149. Fusion obtains the best overall HR-AUPRC of 0.204, improving over AgenTRIM-Proxy by 7%, but this comes at a calibration cost: its ECE rises to 0.277. Contextual scoring achieves the strongest rank correlation (Spearman 0.112) without sacrificing calibration. The in-distribution picture is thus mixed: request-conditioned scoring improves continuous-risk ranking, but static priors still calibrate better when the threat distribution is familiar.

Held-out indirect prompt injection. The gap widens when the threat family is unseen. Static Prior drops sharply to 0.339 HR-AUPRC with near-zero recall at 10% budget, because indirect prompt

Split	N	Skills	Groups	Allow	Esc.	Block	D0	D1	Ben.	Dir.	Exf.	Hij.	Abus.	MTE	IPI
Train	1600	421	897	888	443	269	897	703	663	166	156	130	155	330	0
Val	500	138	163	276	139	85	163	337	207	52	49	41	48	103	0
Test	450	132	151	248	126	76	151	299	187	46	44	37	43	93	0
OOD	450	95	102	234	52	164	102	348	0	0	0	0	0	0	450
Total	3000	476	1313	1646	760	594	1313	1687	1057	264	249	208	246	526	450

Table 1: Dataset statistics for SIA-Bench. D0 and D1 denote lineage depths 0 and 1. Ben., Dir., Exf., Hij., Abus., MTE, and IPI abbreviate benign, direct malicious, data exfiltration, tool-selection hijack, capability abuse, multi-turn escalation, and indirect prompt injection. The OOD split is IPI-only as a construction family; 234 of 450 rows are labeled ALLOW because the injection was neutralized or never reached a high-risk sink.

injection exploits runtime context that static metadata cannot see. Fusion reaches 0.439 HR-AUPRC, a 12% improvement over the best proxy baseline. It also leads on recall at 10% review budget, achieving 0.152 versus 0.121 for AgenTRIM-Proxy, and on rank correlation. Contextual scoring without the static prior still calibrates best, with ECE of 0.289.

A retrieval–calibration trade-off runs through these results. Static priors remain the stronger calibrator on familiar data. Request-conditioned scoring adds most value when the threat depends on runtime context that static metadata cannot capture. Fusion is the compromise: it is the best retriever in this table, not the best-calibrated scorer.

Error analysis. False positives arise in two recurring patterns: tainted context that is irrelevant to the candidate action, and benign arguments that happen to look credential-like. False negatives occur when the malicious dependency is oblique: a prior tool output implicitly instructs exfiltration without naming the target field, or a multi-turn setup distributes the attack across several low-risk-looking steps. Appendix N probes these failure modes with a targeted stress set.

6.3 Retrieval–Calibration Trade-off and Sensitivity to Weights and Thresholds

Figure 3 visualizes the trade-off across the full grid of fusion weights on validation. Each point is one (w_s, w_t) setting; points along the frontier offer the best available HR-AUPRC for a given ECE budget. Heavier trigger weighting consistently improves high-risk retrieval but tends to worsen calibration.

We select the operating point by ranking metrics first and use ECE/W-MAE as tie-breakers. Appendix I compares that rule with a threshold-first selector on the official splits: the latter raises malicious recall but incurs 15–34% false-block rates, and task completion falls from 0.92 to 0.73 on locked test. Those thresholds, $\tau = (0.60, 0.75)$ versus a threshold-first pair, are not the unused ranking default $\tau = (0.10, 0.70)$ listed for Table 2. Ranking metrics are invariant to τ.

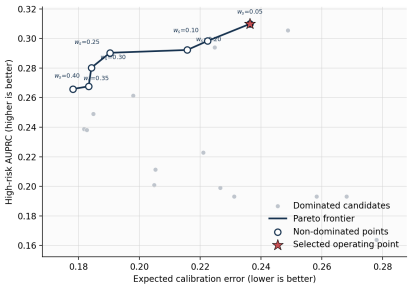


Figure 3: Validation sweep over fusion weights. Each point is one operating point; the frontier shows the retrieval–calibration trade-off.

Weight and threshold sensitivity. We swept the Stage C grid, 21 fusion weights times 140 threshold pairs, on the frozen 500/100 rebuttal subset, not the official 450/450 splits. This sweep uses a deployment-policy setting of $w_s = 0.5$ and $\tau = (0.25, 0.70)$. On that subset, test Spearman stays within 0.30–0.41 over the full $[0, 1]$ weight range, with an optimum of 0.411 at $w_s = 0.45$ and 0.407 at $w_s = 0.5$. Those Spearman values are not comparable with Table 2, where Fusion is 0.098 on official locked test. An unconstrained validation macro-F1 search would pick $(0.2, 0.5)$, whose test macro-F1 of 0.4501 is indistinguishable from this sweep's 0.4493. As a diagnostic only, an oracle-on-test threshold pick would add at most 0.065 macro-F1.

Runtime overhead. The STARS-Rules audit adds less than 2.1 ms per invocation at p95, and the quantile normalization and fusion step is negligible. The STARS-LLM scorer trades speed for stronger ranking at approximately 4.5 s per invocation. This latency gap motivates a practical two-tier architecture: rule-based pre-filtering handles the majority of calls, and selective LLM re-scoring is reserved for cases near the escalation threshold.

Method	HR-AUPRC ↑	Rec@10% ↑	Prec@10% ↑	Spearman ↑	ECE ↓	W-MAE ↓
Locked test						
Static and prior-work proxies						
Static Prior†	0.188	**0.122**	**0.206**	0.031	**0.149**	0.291
Agent-Audit	0.169	0.090	0.152	0.015	0.318	0.319
AgenTRIM-Proxy	0.191	0.111	0.187	0.044	0.175	0.295
RTBAS-Proxy	0.185	0.100	0.169	0.028	0.213	0.302
Request-conditioned STARS variants (STARS-Rules backend)						
Text-Only	0.189	0.066	0.111	0.073	0.217	**0.288**
Contextual	0.195	0.105	0.178	**0.112**	0.216	0.288
Fusion	**0.204**	0.118	0.200	0.098	0.277	0.321
Held-out indirect prompt injection						
Static and prior-work proxies						
Static Prior†	0.339	0.023	0.083	0.011	0.298	0.391
Agent-Audit	0.380	0.114	0.416	0.164	0.435	0.435
AgenTRIM-Proxy	0.392	0.121	0.441	0.149	0.312	0.398
RTBAS-Proxy	0.378	0.107	0.391	0.136	0.329	0.410
Request-conditioned STARS variants (STARS-Rules backend)						
Text-Only	0.399	0.110	0.400	0.125	0.304	**0.380**
Contextual	0.405	0.146	0.533	0.127	**0.289**	0.380
Fusion	**0.439**	**0.152**	**0.556**	**0.172**	0.298	0.395

† Static Denylist produces identical scores and is omitted. Bold = best; underline = second-best within each split.

Table 2: Official SIA-Bench splits, N=450/450, STARS-Rules, $w_s = 0.55$. Ranking metrics ignore τ. STARS-LLM is Table 3 on the 500/100 subset. Fusion is the strongest retriever here; Contextual and Static Prior often calibrate better.

Backend	HR-AUPRC	Rec@10	Spear.	ECE	p50 ms
In-distribution slice (n=500)					
STARS-Rules	0.297	0.360	0.344	0.140	0.8
STARS-LLM	**0.728**	**0.717**	**0.552**	0.108	4254
Ensemble	0.700	0.740	0.550	**0.114**	10
Held-out IPI slice (n=100)					
STARS-Rules	0.504	0.171	0.205	0.361	2.0
STARS-LLM	**0.757**	**0.217**	**0.667**	**0.283**	4627
Ensemble	0.685	0.183	0.589	0.295	17

Table 3: Stage B backends on the frozen 500/100 rebuttal subset with $w_s = 0$, not the official 450/450 splits. STARS-LLM is the primary configuration; STARS-Rules is the API-free alternative. Ensemble latency assumes cached LLM scores.

6.4 STARS-LLM: The Primary Configuration

Table 2 used STARS-Rules on the official splits so that every method is deterministic. The primary configuration instead backs Stage B with a zero-shot GPT-5.4-mini risk scorer (Zheng et al., 2023). Table 3 isolates Stage B on the frozen 500/100 subset with $w_s = 0$. A 40/60 rule–LLM ensemble caches the LLM score and blends it with the rule output. On this subset, Trigger-Only Rules already outranks Fusion-55/45, HR-AUPRC 0.297 versus 0.282, which is why we report the LLM comparison without the static channel: adding Stage A can dilute a strong trigger signal.

On this subset STARS-LLM raises rules-only HR-AUPRC from 0.297 to 0.728 in-distribution and from 0.504 to 0.757 on held-out IPI. In-distribution Spearman rises from 0.344 to 0.552. The cached ensemble recovers most of the ranking at about 10 ms median latency, versus about 4.3 s for a live judge call.

The gain is a ranking gain. Under thresholds frozen on validation, held-out accuracy does not improve, falling from 0.380 to 0.330. A stronger scorer reorders the queue; thresholds that are not retuned under shift can ignore that ordering.

Input ablations on the same subset confirm that the signal is invocation-specific. With the rule backend frozen, capability-only input collapses ranking to Spearman 0.00 on both slices. Emptying the runtime context drops held-out Spearman from 0.205 to 0.119, shuffling the context pairing to 0.099, and shuffling requests to 0.071. See Appendix D.

6.5 Are the Results Robust?

Four analyses test whether the conclusions depend on artifacts of the seeded benchmark construction.

Cross-domain transfer. We generate 300 invocation records with GPT-5.4 across five unseen domains, finance, healthcare, DevOps, legal, and IoT, with no overlap with SIA-Bench seeds, and score them with frozen validation weights. The generator emits templated batches, 32 unique IDs among 300 rows, and Stage A scores in this dump are not clipped to the paper cap of 0.40.

Fusion reaches 0.710 HR-AUPRC versus 0.557 for Static Prior, with ECE 0.145 versus 0.270. We

Method	HR-AUPRC	Spearman	ECE
Static Prior	0.557	0.279	0.270
Trigger Only	0.569	0.508	0.203
Fusion	**0.710**	**0.592**	**0.145**

Table 4: LLM-generated transfer check on 300 templated records. Stage A cap not applied. Directional only.

read this as a template-level domain-shift check, not real-deployment transfer.

Surface-shortcut probe. A TF-IDF logistic-regression probe trained on 500 labeled train records reaches 0.66 accuracy on the official locked test, above both STARS backends under fixed thresholds, because a model fit to in-distribution labels can exploit surface regularities that STARS never trains on. The probe does not reach the safety-relevant metric: on the 500/100 subset, the LLM judge's BLOCK AUPRC is 0.75 versus 0.35 for the probe. Appendix D reports both numbers with their splits.

Target-feature leakage. Three raw fields, permissions, provenance, and trajectory, appear in both the target heuristic and Stage B, with different formulas. Removing those fields from the target leaves a nearly identical label, Pearson $r > 0.999$ and MAE $= 0.013$, so headline block-band ranking is not an artifact of copying those fields into the target. Feature-drop orderings are in Appendix E. We do not report HR-AUPRC from that dump against Table 2, because it used a different scoring path.

Decision-level comparison. Appendix Table 5 compares STARS-Rules Fusion, thresholded at the deployment-policy pair $\tau = (0.25, 0.70)$ on the 500/100 subset, against native-style AgenTRIM and RTBAS decisions. AgenTRIM catches 82% of unsafe invocations but blocks 85% of benign calls, with task completion 0.344. RTBAS preserves task completion at 0.676 but catches only 25% of unsafe invocations. STARS balances the two, with task completion 0.678 and unsafe recall 0.662, because ESCALATE routes ambiguous cases to review. These numbers use $\tau = (0.25, 0.70)$, not the unused ranking default $\tau = (0.10, 0.70)$ associated with Table 2.

7 Conclusion

On familiar data, static priors and invocation-time scoring are hard to separate. On held-out indirect prompt injection, STARS-Rules Fusion improves high-risk retrieval by 12% over the best proxy baseline on the official splits, because static metadata cannot represent *why* the agent is calling a skill. On the 500/100 subset, STARS-LLM raises HR-AUPRC from 0.297 to 0.728 over rules-only Stage B; under frozen thresholds that ranking gain does not become a classification gain. A practical design blends rule-based pre-filtering with selective LLM re-scoring. We release SIA-Bench, the annotations, and STARS.

Limitations

Synthetic construction is the primary limitation. Every record in SIA-Bench is synthetic: seeds are author-constructed, variants are template- and LLM-generated, the held-out split is an LLM mutation pipeline, and risk targets come from ensemble decisions plus heuristic refinement. Absolute metrics should not be read as deployment performance. The 180-record human study checks label ordering, not the full benchmark. A TF-IDF probe, reported in Appendix D, shows that the synthetic text contains learnable regularities.

Frozen thresholds under shift. On the 500/100 subset the shortcut probe transfers its class boundary to held-out data better than our frozen thresholds, 0.53 versus 0.38 accuracy, and STARS-LLM's ranking gain does not survive fixed-threshold classification. Deployments should retune thresholds under shift even when ranking transfers.

Scope of the audit. Held-out evaluation covers IPI only. Skills overlap across splits, so a purely static prior can exploit skill identity. We audit one invocation at a time; multi-step deception across several low-risk-looking calls can evade that view. AgenTRIM and RTBAS rows are our pre-execution proxies, not the original systems; see Appendix B. LLM scoring and the decision-level table use a 500/100 dump, not the official 450/450 splits; see Appendix G. Open directions include seeding from real agent traces and learned replacements for the lexicon scorer.

Ethical Considerations

This work studies defensive agent-safety auditing. Benchmark records are invocation-level artifacts that do not require running attacks against live services. We release SIA-Bench records, group-safe splits, the full STARS implementation for both

backends, proxy baselines, the shortcut probe, evaluation scripts, prompt templates, and generation logs, with secrets and PII removed.

The human validation study (Section 5.1) involved three annotators who were not paper authors or seed writers. Annotators were informed of the study's purpose, worked independently under a written guideline with no discussion or answer sharing, and were compensated for their time. The annotation interface showed all annotators the same lexical keyword highlighting, which is disclosed in the released protocol. The study assesses label quality of synthetic records only and involves no personal, sensitive, or user-derived data. The per-annotator data, guideline, and analysis scripts are released with the paper.

Deployments should treat scores as decision support; high-impact actions should retain human confirmation.

Acknowledgments

Di Wang is the corresponding author. He is supported in part by funding BAS/1/1689-01-01, RGC/3/7125-01-01, FCC/1/5940-20-05, FCC/1/5940-06-02, and the King Abdullah University of Science and Technology (KAUST) Center of Excellence for Generative AI, under award number 5940, as well as a gift from Google.

References

Hengyu An, Jinghuai Zhang, Tianyu Du, Chunyi Zhou, Qingming Li, Tao Lin, and Shouling Ji. 2025. IPIGuard: A novel tool dependency graph-based defense against indirect prompt injection in LLM agents. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 1023–1039, Suzhou, China. Association for Computational Linguistics.

Roy Betser, Shamik Bose, Amit Giloni, Chiara Picardi, Sindhu Padakandla, and Roman Vainshtein. 2026. AgenTRIM: Tool risk mitigation for agentic AI. *arXiv preprint arXiv:2601.12449*.

Luca Beurer-Kellner, Beat Buesser, Ana-Maria Creţu, Edoardo Debenedetti, Daniel Dobos, Daniel Fabian, Marc Fischer, David Froelicher, Kathrin Grosse, Daniel Naeff, Ezinwanne Ozoani, Andrew Paverd, Florian Tramèr, and Václav Volhejn. 2025. Design patterns for securing LLM agents against prompt injections. *arXiv preprint arXiv:2506.08837*.

Zhaorun Chen, Mintong Kang, and Bo Li. 2025. ShieldAgent: Shielding agents via verifiable safety policy reasoning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *PMLR*, pages 8313–8344.

Edoardo Debenedetti, Jie Zhang, Mislav Balunović, Luca Beurer-Kellner, Marc Fischer, and Florian Tramèr. 2024. AgentDojo: A dynamic environment to evaluate prompt injection attacks and defenses for LLM agents. In *Advances in Neural Information Processing Systems 37*. Datasets and Benchmarks Track.

Aarya Doshi, Yining Hong, Congying Xu, Eunsuk Kang, Alexandros Kapravelos, and Christian Kästner. 2026. Towards verifiably safe tool use for LLM agents. *arXiv preprint arXiv:2601.08012*.

Gloria Felicia, Michael Eniolade, Jinfeng He, Zitha Sasindran, Hemant Kumar, Milan Hussain Angati, and Sandeep Bandarupalli. 2026. StepShield: When, not whether to intervene on rogue agents. *arXiv preprint arXiv:2601.22136*.

Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. 2023. Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection. In *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security*.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *PMLR*, pages 1321–1330.

Xiaojun Jia, Jie Liao, Simeng Qin, Jindong Gu, Wenqi Ren, Xiaochun Cao, Yang Liu, and Philip H. S. Torr. 2026. SkillJect: Automating stealthy skill-based prompt injection for coding agents with trace-driven closed-loop refinement. *arXiv preprint arXiv:2602.14211*.

Qi Li and Xinchao Wang. 2026a. Sponge tool attack: Stealthy denial-of-efficiency against tool-augmented agentic reasoning. *arXiv preprint arXiv:2601.17566*.

Qi Li and Xinchao Wang. 2026b. Vimu: Benchmarking video metaphorical understanding. *arXiv preprint arXiv:2605.14607*.

Qi Li, Xingyi Yang, and Xinchao Wang. 2026a. BadWAM: When world-action models dream right but act wrong. *arXiv preprint arXiv:2607.15207*.

Qi Li, Runpeng Yu, and Xinchao Wang. 2026b. VidSME: Membership inference attacks against large video understanding models. *Advances in Neural Information Processing Systems*, 38:111572–111596.

Xu Li, Simon Yu, Minzhou Pan, Yiyou Sun, Bo Li, Dawn Song, Xue Lin, and Weiyan Shi. 2026c. Unsafer in many turns: Benchmarking and defending multi-turn safety risks in tool-using agents. *arXiv preprint arXiv:2602.13379*.

Kanghua Mo, Li Hu, Yucheng Long, and Zhihao Li. 2025. Attractive metadata attack: Inducing LLM agents to invoke malicious tools. In *Advances in Neural Information Processing Systems*.

Yutao Mou, Zhangchi Xue, Lijun Li, Peiyang Liu, Shikun Zhang, Wei Ye, and Jing Shao. 2026. ToolSafe: Enhancing tool invocation safety of LLM-based agents via proactive step-level guardrail and feedback. *arXiv preprint arXiv:2601.10156*.

Yangjun Ruan, Honghua Dong, Andrew Wang, Silviu Pitis, Yongchao Zhou, Jimmy Ba, Yann Dubois, Chris J. Maddison, and Tatsunori Hashimoto. 2024. ToolEmu: Identifying the risks of LM agents with an LM-emulated sandbox. In *International Conference on Learning Representations*.

Jiawen Shi, Zenghui Yuan, Guiyao Tie, Pan Zhou, Neil Zhenqiang Gong, and Lichao Sun. 2025. Prompt injection attack to tool selection in LLM agents. *arXiv preprint arXiv:2504.19793*.

Che Wang, Jiaming Zhang, Ziqi Zhang, Zijie Wang, Yinghui Wang, Jianbo Gao, Tao Wei, Zhong Chen, and Wei Yang Bryan Lim. 2026. AdapTools: Adaptive tool-based indirect prompt injection attacks on agentic LLMs. *arXiv preprint arXiv:2602.20720*.

Zhen Xiang, Linzhi Zheng, Yanjie Li, Junyuan Hong, Qinbin Li, Han Xie, Jiawei Zhang, Zidi Xiong, Chulin Xie, Carl Yang, Dawn Song, and Bo Li. 2025. GuardAgent: Safeguard LLM agents via knowledge-enabled reasoning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *PMLR*, pages 68316–68342.

Zhi Yang, Runguo Li, Qiqi Qiang, Jiashun Wang, Fangqi Lou, Mengping Li, Dongpo Cheng, Rui Xu, Heng Lian, Shuo Zhang, Xiaolong Liang, Xiaoming Huang, Zheng Wei, Zhaowei Liu, Xin Guo, Huacan Wang, Ronghao Chen, and Liwen Zhang. 2026. FinVault: Benchmarking financial agent safety in execution-grounded environments. *arXiv preprint arXiv:2601.07853*.

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

Qiang Yu, Xinran Cheng, and Chuanyi Liu. 2026. Defense against indirect prompt injection via tool result parsing. *arXiv preprint arXiv:2601.04795*.

Qiusi Zhan, Zhixiang Liang, Zifan Ying, and Daniel Kang. 2024. InjecAgent: Benchmarking indirect prompt injections in tool-integrated large language model agents. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 10471–10506, Bangkok, Thailand. Association for Computational Linguistics.

Haiyue Zhang. 2026. Agent audit: Static security analysis for ai agent applications. Based on OWASP Agentic Top 10 (2026) threat model.

Zhexin Zhang, Shiyao Cui, Yida Lu, Jingzhuo Zhou, Junxiao Yang, Hongning Wang, and Minlie Huang. 2024. Agent-SafetyBench: Evaluating the safety of LLM agents. *arXiv preprint arXiv:2412.14470*.

Jingnan Zheng, Yanzhen Luo, Jingjun Xu, Bingnan Liu, Yuxin Chen, Chenhang Cui, Gelei Deng, Chaochao Lu, Xiang Wang, An Zhang, and Tat-Seng Chua. 2026. Risky-bench: Probing agentic safety risks under real-world deployment. *arXiv preprint arXiv:2602.03100*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-Bench and chatbot arena. In *Advances in Neural Information Processing Systems 36*.

Peter Yong Zhong, Siyuan Chen, Ruiqi Wang, McKenna McCall, Ben L. Titzer, Heather Miller, and Phillip B. Gibbons. 2025. RTBAS: Defending LLM agents against prompt injection and privacy leakage. *arXiv preprint arXiv:2502.08966*.

A Decision-Level Comparison

Method	Unsafe Rec	False Blk	Task Cpl	Esc Rate
In-distribution slice (n=500)				
AgenTRIM-Proxy	**0.821**	0.851	0.344	0.000
RTBAS-Proxy	0.248	**0.149**	0.676	0.170
STARS (Fusion)	0.662	0.316	**0.678**	0.372
Held-out IPI slice (n=100)				
AgenTRIM-Proxy	**0.765**	0.898	0.440	0.000
RTBAS-Proxy	0.333	**0.204**	0.560	0.270
STARS (Fusion)	0.549	0.449	**0.550**	0.450

Table 5: Native-style decisions on the 500/100 subset with $\tau = (0.25, 0.70)$, the deployment-policy pair, not the unused ranking default associated with Table 2. AgenTRIM-Proxy has high unsafe recall but blocks most benign calls. STARS uses ESCALATE to trade recall against task completion.

B Scorer, Proxy-Baseline, and Dataset Construction Details

Stage B feature computation. The contextual scorer is deterministic. f_{intent} is a normalized maximum over request-text cue families for destructive action, credential or secret seeking, exfiltration, policy bypass, and externally directed messaging. f_{arg} scores invocation arguments that contain sensitive file paths, credential-like field names, account identifiers, external recipients, database tables, or outbound payloads. f_{prov} maps provenance tiers for prior tool outputs into risk weights, with unverified or external content receiving higher weight than official or trusted content. f_{traj} aggregates prior

Family	Lexical/schema cues
read	read, list, open, fetch, download, export, path
write	write, save, create, update, delete, upload, overwrite
execute	run, execute, shell, code, command, script, subprocess
query	query, search, sql, database, retrieve, lookup
network	url, http, webhook, api, post, request, socket
message	email, slack, send, forward, notify, recipient
credential	token, key, password, secret, credential, oauth
admin	permission, role, admin, sudo, install, policy

Table 6: Action-family cue groups used by the deterministic request–skill alignment gate.

risk state over the dependency graph available before the candidate invocation, using only ancestor tool outputs and policy state already present in the record. f_{taint} records whether untrusted or externally sourced content flows into high-risk sinks such as code execution, file writing, database mutation, or outbound communication. The dependency graph is not allowed to inspect future tool outputs.

Request–skill alignment gate. The gate $g(U, S, C)$ is described in Section 4.2. Each of the eight canonical action families (*read, write, execute, query, network, message, credential, admin*) is defined by a keyword list matched against the normalized request text, skill metadata, and invocation arguments. Table 6 reports the cue groups used in the paper configuration. Matching is case-folded and applied to request text, skill names/descriptions, permission strings, and argument keys. If multiple families match, the family set is their union. If request and skill families overlap, $g = 1$; if either side is empty or disjoint, $g = 0.08$. The mapping is deterministic and requires no learned parameters.

Proxy-baseline documentation. The proxy baselines are documented per the following schema: what the original method does, what our projection implements, which deviations the invocation-level interface forces, and which choices are ours. All proxies operate on invocation records; none can observe execution traces at decision time. Hyperparameters were tuned in each baseline's favor on validation, then frozen. Proxy rows are reference points on this interface, not reimplementations.

AgenTRIM-Proxy. The original AgenTRIM combines static offline constraint extraction with runtime per-step validation over execution traces. Our projection extracts three factors from each invocation record: (1) permission-surface score (same as Stage A static prior, capped at 0.40); (2) per-step constraint-violation count (binary flag for each of AgenTRIM's eight static constraint categories that would be triggered by the invocation arguments); (3) runtime per-step divergence flag (whether the candidate skill diverges from the expected trajectory inferred from prior tool outputs). The three factors are min-max normalized on the validation split and averaged with equal weights into a $[0, 1]$ risk score. Forced deviation: AgenTRIM's post-execution validation signals are unavailable pre-execution. Our choice: equal factor weighting after validation-favoring normalization.

RTBAS-Proxy. The original RTBAS makes binary allow/confirm decisions from information-flow constraints and trajectory context. Our projection maps that binary decision into a two-tier continuous score: invocations that would be "confirmed" (high-risk) receive a score in $[0.5, 1.0]$ linearly interpolated by taint evidence strength; invocations that would be "allowed" (low-risk) receive a score in $[0, 0.5]$ linearly interpolated by provenance trust (unverified $\rightarrow$ 0.5; official $\rightarrow$ 0.0). Forced deviation: RTBAS's interactive confirmation loop is collapsed into a score band. Our choice: the linear interpolation rules within each band. Both proxies use the same frozen validation-split normalization ranges as the main STARS methods; no parameters are re-tuned on the locked test or held-out splits.

IPIGuard-style replay and SkillJect-derived slices. Appendix M aligns subsets of SIA-Bench with the threat models of IPIGuard and SkillJect; those replays follow the same documentation schema and are not protocol-matched replacements for the original evaluations.

Weight selection. The Stage B feature weights and Stage C fusion weights are selected on the validation split using the same ranking-and-calibration criteria described in Section 4.3. They are not inherited from prior work and are not optimized on the locked test or held-out IPI split. After validation selection, the weights, quantile-normalization ranges, and thresholds are frozen for all reported locked evaluations.

C Human Annotation Study Details

Protocol. We sampled 180 records (140 locked-test / 40 held-out), stratified over attack family and canonical action, with hard cases (non-unanimous ensemble votes or medium/low label confidence) oversampled to 40%. Three annotators, none of them paper authors or seed writers, worked in-

Reference labels	Accuracy	Macro-F1
Canonical targets	0.522 (0.450–0.600)	0.399
Human majority	0.533 (0.461–0.606)	0.454
Δ (canonical − human)	-0.011 (-0.094, $+0.067$)	-0.056 (-0.139, $+0.029$)

Table 7: STARS scored against canonical targets versus independent human-majority labels on the same 180 records (95% bootstrap CIs in parentheses). Performance against human labels is not lower, so the synthetic targets do not favor the system.

Signal	Target	Scorer	Overlap
attack_family, evidence_tier	✓	–	Label metadata
decision anchor, votes	✓	–	Supervision
skill.permissions	✓	✓	Shared field
context.provenance	✓	✓	Shared field
context.trajectory	✓	✓	Shared field
request.text/arguments	–	✓	Scorer-only
context.dep_graph	–	✓	Scorer-only

Table 8: Signal overlap between target construction and STARS scoring.

dependently under a written guideline; no discussion or answer sharing occurred, and the interface showed every annotator the same lexical keyword highlighting, which is disclosed in the protocol. Each annotator provided a three-way action decision (ALLOW/ESCALATE/BLOCK) and a 0–100 risk score per record.

Agreement. Inter-annotator agreement: Fleiss $\kappa = 0.709$ on the three-way action; Krippendorff $\alpha = 0.802$ on the risk score. Of the 66 disagreements with canonical labels, 59 are one step apart. The largest cell (36) is humans escalating records the targets allow. The 7/180 ALLOW–BLOCK flips remain in the release as label-uncertainty cases.

Does the labeling favor the system? Table 7 scores STARS at the frozen paper operating point against both label sets. Performance against human labels is not lower; paired bootstrap intervals on the difference include zero.

D Shortcut Probe and Input Ablations

Surface-shortcut probe. A TF-IDF logistic-regression probe trained on 500 labeled train records reaches 0.66 accuracy and 0.50 macro-F1 on the official locked test. STARS never trains on benchmark text. On high-risk ranking the LLM judge is far beyond the probe: BLOCK AUPRC is 0.75 versus 0.35 on the 500/100 subset. The probe transfers its class boundary to held-out data better than frozen STARS thresholds, 0.53 versus 0.38 accuracy.

Input ablations. On the 500/100 subset, with the rule backend frozen: capability-only input collapses ranking to Spearman 0.00 on both slices; emptying runtime context drops held-out Spearman from 0.205 to 0.119; shuffling the context pairing drops it to 0.099; shuffling requests drops it to 0.071.

Backend end-to-end policy metrics. Same 500/100 subset, $w_s = 0$ for ranking, thresholds frozen on validation: STARS-Rules reaches Spearman 0.344 / AUPRC 0.315 / accuracy 0.596 / macro-F1 0.449 on the in-distribution slice and 0.205 / 0.545 / 0.380 / 0.345 on held-out; STARS-LLM reaches 0.552 / 0.753 / 0.642 / 0.567 and 0.667 / 0.769 / 0.330 / 0.346. The judge's gain is in ranking; held-out accuracy does not improve.

E Full Target-Feature Leakage Analysis

Table 8 categorizes all signals used in target construction and STARS scoring. Three raw fields are shared but use different formulas; the request text, arguments, and dependency graph are scorer-only; the decision anchor and attack-family metadata are target-only.

Dropping overlapping Stage B features under a de-overlapped target, on a scoring dump that is *not* Table 2, preserves the same qualitative order: trajectory is the most informative channel, while provenance and taint are nearly interchangeable. Absolute HR-AUPRC from that dump is omitted because it is not on the official 450/450 protocol.

F Dataset Generation Details

The 1,313 source groups are depth-0 invocation cases covering the families in Table 1. All seeds were authored by the co-authors; IPI seeds are held out as a group. No real agent logs were involved. Canonical labels come from a three-model ensemble vote that is 77% unanimous, with model identities in the data release, followed by deterministic refinement. Vote metadata is released as a label-confidence filter. Controlled template rewrites vary surface form, requested object, skill metadata phrasing, and runtime evidence while preserving the source group's action family. Constrained LLM-based semantic variation is used only for paraphrase-style diversification: the prompt instructs the generator to preserve the candidate skill, canonical action band, provenance relation, and attack-family rationale. If a variant removes the

Result	Split	N	w_s	τ	Notes
Table 2	official	450/450	0.55	unused	Rules; ranking only
Table 3	rebuttal dump	500/100	0	unused	Stage B backends
Table 5	rebuttal dump	500/100	0.55	$(0.25, 0.70)$	deployment-policy τ
Sec. 6.3 sweep	rebuttal dump	500/100	0.5	$(0.25, 0.70)$	Spearman 0.30–0.41
Table 14	official	450/450	0.55	$(0.60, 0.75)$ vs. thresh.-first	not Table 2's τ

Table 9: Which result uses which split, fusion weight, and thresholds.

motivating attack condition, it is relabeled, not retained as a positive instance.

This construction improves split integrity but does not eliminate closed-distribution risk. The artifact release includes the source-group identifiers, parent links, generation prompts, generator configuration logs, and exact split files so that later work can audit the taxonomy and add non-seeded attack families.

G Implementation Defaults Used in the Paper

Headline `SIA-Bench` numbers use the official locked-test and held-out IPI splits, N=450/450. LLM scoring, the native decision-level table, and the Stage C sensitivity sweep use a frozen rebuttal dump, N=500/100, that is *not* a subsample of those official files. Spearman and accuracy on that dump are not comparable entrywise with Table 2.

Table 10 records the official-split paper-default weights. The $\tau = (0.10, 0.70)$ pair stored there is unused for Table 2 ranking metrics. Stage A cap $R_{\text{static}} \leq 0.40$ applies to that default; the 500/100 dump and the external transfer dump did not clip Stage A scores.

H Additional Continuous-Risk Target Sensitivity

Tables 11 and 12 replay two archived score files (Contextual and Fusion) under six target mixtures. This is a stability check, not a rerun of Table 2; do not compare the 65/35 rows entrywise with that table. HR-AUPRC is invariant across mixtures because $R_{\text{target}} \geq 0.7$ is the block band.

I Full Selection-Rule Comparison

The main text selects fusion weights by ranking metrics. Table 13 shows two policies from the same Stage C grid on validation; Table 14 evaluates them on the official splits. Ranking metrics are identical; the threshold-first selector raises malicious recall at the cost of false blocks and lower task completion. The continuous-risk-first pair $\tau = (0.60, 0.75)$ is not the unused ranking default $\tau = (0.10, 0.70)$ associated with Table 2. The HR-AUPRC and Spearman columns in Table 14 replay an archived Fusion dump, 0.1935 and 0.2146 on locked test, and should not be compared entrywise with Table 2, where Fusion is 0.204 and 0.098.

J Robustness Across Families, Skills, and Mutation Depth

On the official locked-test split, Contextual has lower grouped weighted MAE than Fusion at family (0.276 vs. 0.342), skill (0.248 vs. 0.286), and mutation (0.288 vs. 0.321) levels. On held-out IPI, Fusion has higher family-level Spearman (0.172 vs. 0.127), while Contextual still has lower grouped W-MAE. Retrieval and calibration trade off the same way after aggregation.

K Additional Ablation on Context Signals

Table 15 is a validation leave-one-out of Stage B context channels. Adding the full context bundle raises HR-AUPRC from 0.216 to 0.231 and Spearman from 0.085 to 0.130 over Text-Only. Dropping any one channel changes little; dropping trajectory hurts Spearman most. Context helps as a bundle, not as a single indispensable feature.

L Secondary Remediation Results

Stage D is a downstream helper, not a primary contribution. Narrower triggering with Contextual yields cleaner patches; Fusion flags more cases and patch quality falls.

M Supplementary Boundary Analyses

`IPIGuard`-style replay recovers signal on held-out IPI but does not beat Contextual under continuous-risk evaluation. On a SkillJect-aligned in-distribution slice, Static Denylist is already stronger than Contextual, HR-AUPRC 0.327 versus 0.196. These slices bound the claim; they are not protocol-matched replacements for the original papers.

N LLM-Assisted Stress-Set Sanity Check

GPT-5.4-mini generated and classified 240 synthetic records targeting ambiguous intent, irrelevant taint, sensitive benign arguments, oblique dependencies, multi-turn setup, and high-capability static priors; GPT-5.4 reviewed all 240. This set is not

Component	Setting	Value
Stage A	Permission-surface weights	code execution $= 1.00$; database $= 1.00$; file read $= 0.65$; file write $= 0.65$; network $= 0.6318$; email $= 0.2366$; file system $= 0.0612$
Stage A	Provenance-trust weights	unverified $= 0.3396$; official $= 0.0896$; community $= 0.0630$
Stage A	Static-score cap	$R_{\text{static}} \leq 0.40$
Stage B	Scorer family	rule-based contextual scorer with feature profile `text_prov_graph_traj`
Stage B	Feature weights	intent cues $= 0.1046$; argument sensitivity $= 0.0493$; provenance trust $= 0.2990$; trajectory state $= 0.3944$; taint propagation $= 0.1526$
Stage B	Context interaction defaults	context-gain scale $= 0.25$; gate floor $= 0.08$; cross-check boost disabled
Stage C	Candidate-space grid	$w_{\text{static}} \in \{0.00, 0.05, \ldots, 1.00\}$; $\tau_{\text{esc}} \in \{0.05, 0.10, \ldots, 0.60\}$; $\tau_{\text{block}} \in \{0.30, 0.35, \ldots, 0.95\}$
Stage C	Default fusion weights	$w_{\text{static}} = 0.55$; $w_{\text{trigger}} = 0.45$
Stage C	Default thresholds	$\tau_{\text{esc}} = 0.10$; $\tau_{\text{block}} = 0.70$
Stage C	Normalization defaults	quantiles $(0.05, 0.95)$; static range $[0.1019,\ 0.4019]$; trigger range $[0.0000,\ 0.3202]$

Table 10: Official-split paper-default weights and normalization ranges for Table 2. See Table 9 for other settings.

Target construction	Method	HR-AUPRC	Rec@10%	Prec@10%	Spearman ρ	ECE ↓	W-MAE ↓
Vote only	Ctx.	0.1876	0.1184	0.2000	0.0494	0.2350	0.3183
	Fus.	0.1935	0.1184	0.2000	0.0529	0.0769	0.3481
Blend 80/20	Ctx.	0.1876	0.1184	0.2000	0.2095	0.2396	0.2995
	Fus.	0.1935	0.1184	0.2000	0.2147	0.0791	0.3292
Blend 65/35 (paper default)	Ctx.	0.1876	0.1184	0.2000	0.2094	0.2439	0.2864
	Fus.	0.1935	0.1184	0.2000	0.2146	0.0807	0.3151
Blend 50/50	Ctx.	0.1876	0.1184	0.2000	0.2094	0.2489	0.2794
	Fus.	0.1935	0.1184	0.2000	0.2146	0.0823	0.3009
Blend 20/80	Ctx.	0.1876	0.1184	0.2000	0.2094	0.2588	0.2747
	Fus.	0.1935	0.1184	0.2000	0.2146	0.0855	0.2728
Heuristic only	Ctx.	0.1876	0.1184	0.2000	0.2094	0.2655	0.2758
	Fus.	0.1935	0.1184	0.2000	0.2146	0.0877	0.2543

Table 11: Locked-test sensitivity under alternative target mixtures. Archived-score replay; Ctx. = Contextual, Fus. = Fusion. HR-AUPRC is invariant because positives are the block band.

part of SIA-Bench and is not used for headline metrics.

Three of 240 canonical decisions changed under GPT-5.4 review, with target-score MAE 0.008. Residual errors cluster on irrelevant taint and sensitive benign arguments as false positives, and on oblique dependencies, multi-turn setup, and alignment-gate misses as false negatives.

Where static priors win. Static controls are strongest when the dangerousness of an invocation is mostly explained by a stable capability surface, not by the current trajectory. This appears in the locked test split, where Static Prior has the best Recall@10%, Precision@10%, and ECE, and in the SkillJect-derived in-distribution slice of Table 17, where Static Denylist reaches HR-AUPRC 0.3273 compared with 0.1955 for Contextual. The AgenTRIM-Proxy baseline similarly performs well on familiar data, with locked-test HR-AUPRC 0.191, because its static-plus-runtime combination captures capability surface effectively. These cases are dominated by familiar high-privilege capabilities, so contextual evidence adds less than a conservative static prior.

Held-out error examples. Manual inspection of held-out IPI errors shows two recurring false-positive patterns. First, externally sourced content can be tainted but irrelevant to the candidate action, such as a user asking for a summary while the context contains unrelated instructions to forward data; the contextual scorer escalates because provenance and taint are high even though the action is not an outbound sink. Second, benign administrative requests that mention sensitive object names can

Target construction	Method	HR-AUPRC	Rec@10%	Prec@10%	Spearman ρ	ECE ↓	W-MAE ↓
Vote only	Ctx.	0.4429	0.1524	0.5556	0.0837	0.3163	0.4209
	Fus.	0.4493	0.1524	0.5556	0.0866	0.0866	0.4333
Blend 80/20	Ctx.	0.4429	0.1524	0.5556	0.1945	0.3246	0.3921
	Fus.	0.4493	0.1524	0.5556	0.1959	0.0932	0.4040
Blend 65/35 (paper default)	Ctx.	0.4429	0.1524	0.5556	0.1945	0.3307	0.3753
	Fus.	0.4493	0.1524	0.5556	0.1959	0.0981	0.3821
Blend 50/50	Ctx.	0.4429	0.1524	0.5556	0.1945	0.3369	0.3666
	Fus.	0.4493	0.1524	0.5556	0.1959	0.1031	0.3601
Blend 20/80	Ctx.	0.4429	0.1524	0.5556	0.1945	0.3492	0.3641
	Fus.	0.4493	0.1524	0.5556	0.1959	0.1130	0.3162
Heuristic only	Ctx.	0.4429	0.1524	0.5556	0.1945	0.3575	0.3665
	Fus.	0.4493	0.1524	0.5556	0.1959	0.1196	0.2869

Table 12: Held-out IPI sensitivity under the same alternative target mixtures. Same replay protocol caveat as Table 11.

Selector	w_{static}	w_{trigger}	τ_{esc}	τ_{block}	HR-AUPRC	Rec@10%	Prec@10%	Spearman ρ	ECE ↓	W-MAE ↓	Macro-F1
Continuous-risk-first	0.55	0.45	0.60	0.75	0.3054	0.2000	0.3400	0.3585	0.0324	0.3062	0.2793
Threshold-first	0.35	0.65	0.30	0.35	0.3054	0.2000	0.3400	0.3585	0.1182	0.2970	0.4468

Table 13: Validation-selected operating points from the same 2,940-candidate Stage C search space. Both selectors share the same ranking metrics; they differ in the thresholded policy. The continuous-risk-first pair $\tau = (0.60, 0.75)$ is *not* the Table 2 default $\tau = (0.10, 0.70)$. Validation task-completion and false-block rates are $0.9384/0.0000$ for continuous-risk-first and $0.7174/0.2029$ for threshold-first.

be over-scored when the arguments look credential-like but the requested operation is read-only or diagnostic. False negatives usually occur when the malicious dependency is oblique, not lexical: a prior tool output may imply that a later message should include a transformed secret, or a multi-turn setup may spread the plan across several low-risk-looking steps.

Held-out indirect prompt injection rescue. In a held-out case where externally sourced email content influences an outbound messaging action, static baselines and uncalibrated contextual scoring all allow the invocation. The calibrated policy escalates it because provenance and taint-flow evidence indicate that the requested outbound action depends on externally sourced instructions. This is the scenario in which Stage C provides the clearest practical value.

O LLM Usage Statement

Language models were used to polish wording and restructure existing text. Appendix N is a marked supplementary check in which GPT-5.4-mini generated synthetic invocation records and GPT-5.4 reviewed them; that set is not part of SIA-Bench and is not used for headline claims. Authors verified all technical content, experiments, results, and the final manuscript.

P Responsible NLP Research Checklist

This appendix mirrors the ARR Responsible NLP Research checklist.

1. **Limitations.** The limitations section covers synthetic construction, frozen thresholds under shift, OOD scope, and deployment boundaries.
2. **Risks and ethics.** The ethics section covers defensive use, artifact release, the human study, and human oversight for high-impact actions.
3. **Dataset documentation.** Section 5 describes `SIA-Bench` (provenance, splits, labels, targets). Further details are in Appendices B, F, G, and H.
4. **Evaluation protocol.** Section 6 defines metrics. Table 2 is the official 450/450 result; Table 3 and Section 6.3 use the 500/100 dump.
5. **Statistical reporting.** We report ranking, recall, precision, Spearman, ECE, and W-MAE. The human study reports bootstrap CIs. LLM, decision-level, and sensitivity numbers use the 500/100 dump; see Appendix G.
6. **Compute and reproducibility.** STARS-Rules is under 2.1 ms per invocation at p95; STARS-LLM is about 4.5 s. Frozen weights and grids are in Appendix G. Code and data are released.

Split	Selector	HR-AUPRC	Spearman ρ	ECE ↓	W-MAE ↓	Macro-F1	Malicious Recall	False Block	Task Completion
Locked test	Continuous-risk-first	0.1935	0.2146	0.0807	0.3151	0.2303	0.0248	0.0000	0.9194
Locked test	Threshold-first	0.1935	0.2146	0.1518	0.3098	0.2884	0.2376	0.1532	0.7258
Held-out OOD	Continuous-risk-first	0.4493	0.1959	0.0981	0.3821	0.2699	0.1574	0.0000	0.8889
Held-out OOD	Threshold-first	0.4493	0.1959	0.1735	0.3781	0.3749	0.5185	0.3376	0.5128

Table 14: Official splits: ranking is identical across selectors; intervention cost is not. Continuous-risk-first $\tau = (0.60, 0.75)$ is not the unused ranking default associated with Table 2. Archived Fusion scores, not Table 2 decimals.

Method	HR-AUPRC ↑	Rec@10% ↑	Prec@10% ↑	Spearman ↑	ECE ↓	W-MAE ↓
Text-Only	0.2160	0.1647	0.2800	0.0847	0.1982	0.2812
Contextual	0.2305	**0.1882**	**0.3200**	0.1299	0.1969	0.2802
No provenance signal	0.2304	**0.1882**	**0.3200**	0.1302	**0.1967**	0.2804
No trajectory signal	**0.2342**	**0.1882**	**0.3200**	0.0978	0.1970	**0.2800**
No taint propagation signal	0.2309	**0.1882**	**0.3200**	**0.1321**	0.1969	0.2803

Table 15: Validation ablation of structured context signals in Stage B. Removing any single context channel changes validation performance only slightly, which suggests cumulative, not single-feature, gains.

Method / split	Req.-patch Acc.	Patch Acc.	Zero-reg. Pass	Triggered
Contextual / test	0.6844	0.3487	0.5000	31
Fusion / test	0.5356	0.1544	0.2547	230
Contextual / OOD	0.5556	0.0388	0.5185	50
Fusion / OOD	0.5111	0.0579	0.2177	326

Table 16: Stage D remediation quality. Req.-patch Acc. denotes whether the generated patch family matches the flagged failure type, and Zero-reg. Pass measures whether generated regression tests pass without obvious regressions.

Comparison	Split	Method	HR-AUPRC	Recall@10%
`IPIGuard` replay	test	`IPIGuard`	0.1522	0.0000
		Static Denylist	**0.3273**	**0.2857**
	ood	`IPIGuard`	0.3637	0.0671
		Contextual	**0.4052**	**0.1463**
`SkillJect`-derived threat slice	test	Static Denylist	**0.3273**	**0.2857**
		Contextual	0.1955	0.0000
	ood	Agent-Audit	0.3799	0.1141
		Contextual	**0.4052**	**0.1463**

Table 17: Supplementary continuous-risk comparisons. These analyses align subsets of `SIA-Bench` with prior threat models, but are not protocol-matched replacements for the main benchmark.

7. **Human subjects and annotators.** Section 5.1 and Appendix C report a three-annotator study on synthetic records only. The ethics section covers consent, independence, compensation, and interface disclosure.

8. **Use of generative AI.** Appendix O discloses writing assistance and the supplementary stress-set check in Appendix N.

Quantity	Value
Generated/classified records	240
GPT-5.4-reviewed records	240
Reviewed valid records	237
Decision agreement	0.988
Target-score MAE	0.008
Verified ALLOW/ESCALATE/BLOCK	96/97/47

Table 18: LLM-assisted stress-set sanity check. GPT-5.4-mini generates and classifies synthetic ambiguous invocation records; GPT-5.4 reviews every record. The check supports the qualitative taxonomy only and is not a replacement for `SIA-Bench`.

GPT-5.4-reviewed error type	Count
False positive: sensitive arguments	40
False negative: alignment gate	36
False negative: multi-turn setup	34
False negative: oblique dependency	32
False positive: irrelevant taint	29
Static-prior win	23
No salient error mode	46

Table 19: Error-type taxonomy from the LLM-assisted stress set. Sensitive-looking benign arguments over-trigger the scorer; oblique dependencies, multi-turn setup, and weak request–skill alignment hide risky calls.