POSITIVE-UNLABELED LEARNING FOR AGENT SAFETY FALSE ALARM AUDITING

Anonymous authors
Paper under double-blind review

ABSTRACT

Safety monitors help safeguard language-model agents interacting with external tools and environments, but conservative monitoring can generate many false alarms, consuming extensive review resources and weakening trust in alerts. Because false and genuine alarms often remain interleaved in native monitor scores, obtaining a reliable cutoff still requires substantial manual verification. In practice, a small set of verified-safe non-alarmed trajectories may be available while alarms remain unlabeled, naturally casting false-alarm auditing as a positive–unlabeled (PU) ranking problem. The key challenge is *monitor-induced selection*, since observed safe references are accepted by the monitor, while the hidden safe alarms of interest are precisely those it incorrectly flags, making the observed positives poorly representative of the positives to be recovered. To address this challenge, we propose a two-stage framework in which *Trust-aware PU Supervision* adapts safe references toward the alarm domain and protects plausible false alarms from excessive negative pressure, while *Reliability-gated Rank Distillation* consolidates consistent ordering preferences from multiple PU reference models into a single student. *Consensus-guided Structural Refinement* then improves the student ranking using hierarchical safe-reference support, alarm relations, and predicted reference consensus. The framework requires no alarm safety labels for fitting and leaves the underlying monitor unchanged. Across mainstream safety monitors, our method achieves a macro AUPRC of 0.6444, outperforming eight evaluated PU baselines by 5.27–16.98 absolute percentage points; compared with PULDA, the strongest evaluated PU baseline, it recovers 33.3% more false alarms at a 5% review budget. Anonymous code is available at `https://anonymous.4open.science/r/safe-alarm-ranking-c4d2/`.

1 INTRODUCTION

Language-model agents increasingly perform real-world tasks through multi-round interactions with external tools and environments. During execution, they may be exposed to prompt injection attacks or take actions that violate safety constraints, potentially causing serious harm (Debenedetti et al., 2024). An agent's execution trajectory records its intermediate actions, observations, and tool interactions, providing a natural basis for safety monitoring mechanisms that detect unsafe behavior (Liu et al., 2026; Xiang et al., 2025; Chen et al., 2025). Existing trajectory monitors assess such execution traces using learned evaluators or explicit safety criteria and produce safety scores, risk judgments, or intervention signals when potentially unsafe behavior is detected. These approaches have demonstrated effective detection of unsafe agent behavior across a range of settings, including coding, web interaction, and tool-use tasks (Liu et al., 2026; Xiang et al., 2025; Chen et al., 2025).

However, this seemingly strong performance in unsafe-behavior detection comes with a substantial cost. Agent trajectories vary considerably in structure and format across tasks and datasets, while the safety of an individual action often depends on both the task context and the preceding execution history (Yuan et al., 2024; Chen et al., 2026). To reduce the risk of overlooking unsafe behavior across such heterogeneous settings, safety monitors often favor sensitivity, for example by adopting permissive alarm thresholds or overly inclusive safety criteria that also flag borderline cases. Our empirical study reveals that this design choice results in a substantial number of false alarms: across mainstream safety monitors, 39.4%–57.9% of triggered alarms actually correspond to safe trajectories and should therefore be regarded as false positives (FPs) (Figure 1a). Such a high false-positive rate

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

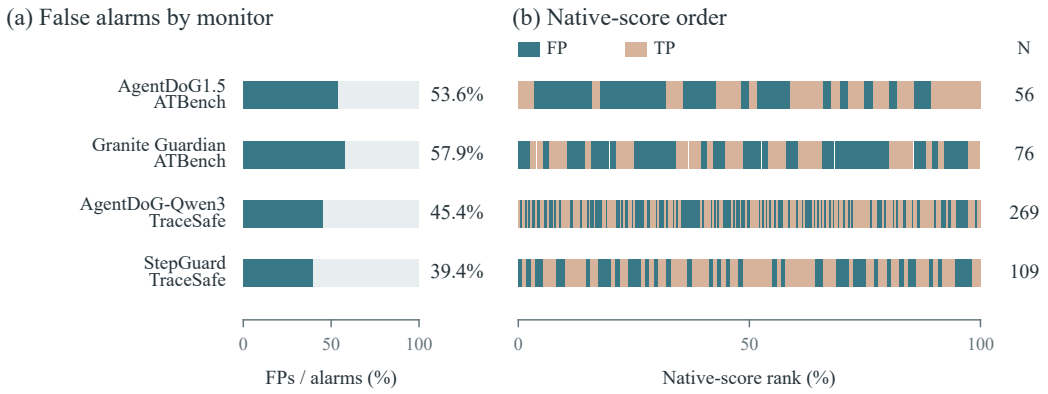


Figure 1: **False alarms and overlap in the native score order.** (a) Share of safe trajectories among alarms in four dataset–monitor settings. (b) The native monitor-score ordering for the same four settings, with lower-risk alarms on the left. Each tile is one alarm (teal for FP, sand for TP); row widths are normalized to cohort size, and N gives the alarm count. All 510 alarms are shown. FP and TP remain interleaved within each order, leaving no clean score cutoff. Ground-truth labels determine tile colors only, not the ordering; ties in native scores are broken by sample ID.

is consequential. Repeated false warnings can erode users' trust in the monitoring system and reduce their willingness to respond to subsequent alerts (Chancey et al., 2017). Moreover, each false alarm must be investigated before it can be dismissed, incurring additional human effort and computational overhead (Alahmadi et al., 2022; Yang et al., 2024; Oliver et al., 2024).

A natural way to prioritize likely false alarms for review is to reuse the monitor's original safety score, possibly with an additional cutoff. Our empirical study shows that false and true alarms often receive similar scores and remain mixed in the native ordering (Figure 1b). With this overlap, moving the cutoff cannot reliably separate the two groups. Even if stronger future monitors produced better-separated scores, choosing a cutoff that preserves genuine alarms would require manually verifying many cases. That verification is impractical under a limited review budget.

Let us reconsider the alarm-isolation problem from a different perspective. In practice, it is often easy to obtain a small set of verified-safe trajectories, either from routine benign traces that do not trigger alarms or from the monitor's training data. Because safety monitors are typically conservative, trajectories that they confidently classify as safe provide reliable positive examples. Treating safety as the class of interest, these verified-safe trajectories form the observed positive set, while false alarms hidden within the alarm pool constitute latent positives. The task therefore becomes one of identifying positive-class samples from an otherwise unlabeled alarm pool, naturally giving rise to a positive-unlabeled (PU) learning formulation (Elkan & Noto, 2008). However, this setting differs fundamentally from standard PU learning. The safety monitor itself has already partitioned the two types of safe trajectories across its alarm boundary: observed positives correspond to safe behavior that the monitor accepts, whereas the latent positives of interest are precisely the safe trajectories that it incorrectly rejects. Consequently, the observed positive set may not be representative of the false alarms in the unlabeled pool. Learning directly from these positives without accounting for this selection mechanism risks reproducing the monitor's existing decision pattern rather than correcting its errors (Bekker & Davis, 2018).

In this work, to prioritize false-alarm agent trajectories identified by safety monitors, we propose a two-stage positive–unlabeled ranking framework that can prioritize most-likely false alarms in an auditing queue. In the first stage, we adapt the verified-safe non-alarms towards the alarm pool and reduce excessive negative pressure on plausible false alarms to mitigate the shift between the safe references and the false alarms, which helps us develop much more effective PU classifiers. But due to the uncertainty of PU learning process, we found that trained classifiers with different initializations provide distinctive auditing queues. Therefore, we propose a reliability-gated distillation strategy to distill the consistent alarm orderings across multiple PU classifiers into a single student. Then, in the second stage, we can obtain an initial auditing queue by ranking the scores predicted by the student

PU classifier. To further exploit structure across task contexts, we refine the auditing queue with consensus-guided structural refinement driven by hierarchical safe-reference support, target-alarm relations, and reference agreement. Extensive experiments on two benchmark datasets and four dataset–monitor settings demonstrated the superiority of our proposed framework in prioritizing false alarms of agent trajectories compared with a variety of state-of-the-art potential baseline approaches. In summary, our main contributions are:

- We identify the issue of frequent false alarms raised by current agent safety monitors, and to our best knowledge, propose the first effective framework to prioritize most-likely false alarms in the auditing queue.
- We formulate agent false-alarm auditing as a special positive–unlabeled ranking problem under monitor-conditioned selection and develop a series of designs to mitigate the shift between positives and hidden positives and search for consistency across PU learning processes and task contexts.
- We conduct extensive experiments against established PU methods, direct auditing controls based on native monitor scores and frozen representations, and local LLM re-judgment. Across mainstream safety monitors, our method achieves a macro AUPRC of 0.6444, outperforming eight evaluated PU baselines by 5.27–16.98 absolute percentage points and recovering 33.3% more false alarms than PULDA at only a 5% auditing budget (Table 1).

2 PRELIMINARIES

Safety monitoring and LLM-based judgment. Existing work assesses agent safety through learned risk prediction, policy verification, and direct LLM judgment. Granite Guardian detects risks in model inputs and outputs (Padhi et al., 2025), while AgentDoG provides contextual risk diagnosis for execution trajectories (Liu et al., 2026). StepGuard checks candidate tool actions before execution and balances safe and unsafe supervision to reduce over-defense and under-defense (Zheng et al., 2026). Policy-based approaches include GuardAgent, which translates safety requirements into executable checks (Xiang et al., 2025), and ShieldAgent, which combines policy modeling with logical verification (Chen et al., 2025). R-Judge evaluates the ability of general-purpose LLMs to judge safety from interaction records (Yuan et al., 2024). These studies focus on improving safety judgments and interventions. Our work addresses a complementary task after alarms are raised: using limited verified-safe supervision to prioritize likely false alarms for human review.

Positive–unlabeled learning. PU methods learn from labeled positives and an unlabeled mixture through risk estimation, distribution alignment, or sample-level supervision. Prior work includes nnPU, which uses non-negative risk estimation to reduce overfitting (Kiryo et al., 2017), and DC-PU, which constrains positive and negative risks to balance classification errors (Li et al., 2025). Dist-PU aligns predicted label distributions with class-prior information (Zhao et al., 2022), and PULDA combines this alignment with functional-margin learning (Jiang et al., 2023). Robust-PU iteratively selects reliable negative examples (Zhu et al., 2023), PUL-CPBF constructs supervision using classifiers with different probability boundaries (Li et al., 2024), and HolisticPU identifies unlabeled examples through their prediction trends (Wang et al., 2023). NcPU addresses unreliable PU supervision through representation alignment (Zhao et al., 2026). Related work also addresses nonrandom positive selection (Bekker & Davis, 2018) and AUC optimization from PU data (Sakai et al., 2018). Our setting connects these directions through monitor-induced selection: verified-safe non-alarms and hidden safe alarms share a safety label but are selected by opposite monitor decisions. Our method adapts safe supervision and ranking refinement to recover false alarms under a fixed review budget.

Setting and observed data. Let τ_i denote an agent execution trajectory and $y_i \in \{+1, -1\}$ its safety label, with $+1$ indicating safe behavior and -1 indicating unsafe behavior. A safety monitor M_ψ with fixed configuration ψ provides a representation $\mathbf{h}_i \in \mathbb{R}^d$ and an alarm decision $a_i \in \{0, 1\}$, where $a_i = 1$ indicates an alarm. We observe a small set $\mathcal{D}_P$ of verified-safe non-alarmed trajectories ($y_i = +1$, $a_i = 0$), a set $\mathcal{D}_U$ of unlabeled training alarms, and a separate target cohort $\mathcal{C} = \{\tau_i\}_{i=1}^N$ with $a_i = 1$ for every member of both alarm sets. Safety is the positive class in our PU formulation; members of $\mathcal{D}_P$ are observed positives and are termed *TN references* relative to the monitor's alarm decision. Frozen representations, task features, and isolation-group identifiers are available for all

three sets. Safety labels are observed only for $\mathcal{D}_P$ during fitting; the labels and false-alarm proportions of $\mathcal{D}_U$ and $\mathcal{C}$ are unknown.

Problem definition. The auditor produces an ordering of the N target alarms for human review. Under a review budget $B \in \{1, \ldots, N\}$, only the first B alarms are inspected, and the objective is to maximize the number of false alarms ($y_i = +1$) among them. The ordering may depend on the unlabeled target features; target safety labels are used only for evaluation after the order is fixed. The auditor determines review priority while preserving all original alarm decisions.

3 METHOD

3.1 OVERVIEW

The difficulty is that the known-safe trajectories passed the monitor, whereas the false alarms we seek were selected by it. Treating every alarm as a negative example can therefore bury the very trajectories that reviewers should see first. Stage I learns a PU auditor in two steps. Trust-aware supervision transfers TN references toward the alarm domain and reduces negative pressure on plausible FPs. Reliability-gated distillation then combines the ordering preferences of several frozen reference auditors in one student. Stage II freezes that student and refines its queue with hierarchical TN support and relations among target alarms. A predicted measure of reference agreement controls the final per-alarm adjustment. Both stages may use unlabeled target features; neither uses target safety labels for fitting. Figure 2 follows these three components from left to right.

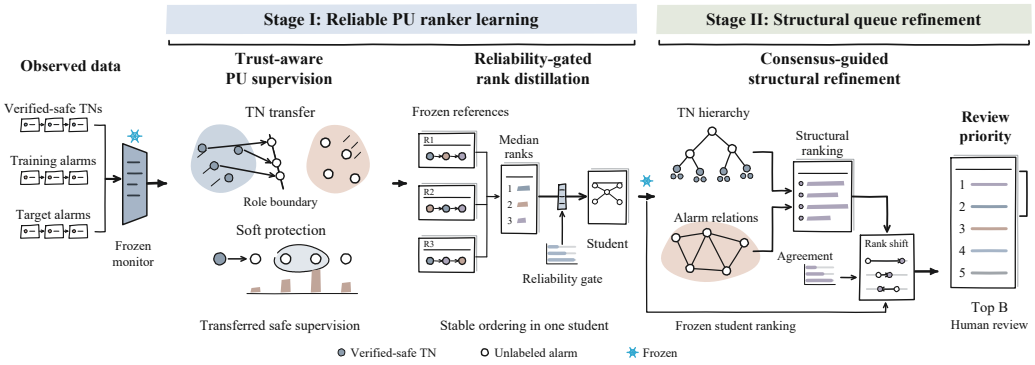


Figure 2: **Overview of our method.** Stage I learns one student through trust-aware TN transfer and soft protection (§3.2.1), followed by median-rank supervision with a shared reliability gate (§3.2.2). Stage II freezes the student, combines hierarchical TN support and target-alarm relations into a structural ranking, and uses an agreement head to scale the final rank shift (§3.3). Components 1–2 form Stage I and component 3 forms Stage II; the final queue uses no test-time reference ensemble.

3.2 STAGE I: RELIABLE PU RANKER LEARNING

Precursor, reference, and student auditors share a PU objective. Its common part regularizes predicted safety scores across verified-safe and alarm batches through a distribution-alignment loss $\mathcal{L}_{\text{align}}$, and adds a group-structured representation mixing loss $\mathcal{L}_{\text{mix}}$: $\mathcal{L}_{\text{base}} = \mathcal{L}_{\text{align}} + \lambda_{\text{mix}}(t)\mathcal{L}_{\text{mix}}$. Here t is the training step and $\lambda_{\text{mix}}(t)$ is its mixing weight. Appendix A.3 gives the terms and optimization. The two components below determine how safe references and other alarms contribute to fitting.

3.2.1 TRUST-AWARE PU SUPERVISION

Transferring safe references. A role classifier, fitted out of fold on training data, distinguishes TN references from training alarms using monitor and task representations. Its decision boundary describes the monitor's *selection* of trajectories into the alarm pool; it is not a boundary between safe and unsafe behavior. For a TN reference $\tau_p \in \mathcal{D}_P$, let $\widetilde{\mathbf{h}}_p$ be its nearest feasible representation

on that role boundary, with the task representation and monitor-vector norm held fixed. A transfer weight $\lambda_p^{\mathrm{tr}} \in [0,1]$ reflects boundary proximity and the displacement needed to reach it. The positive comparison score is

$$s_p^{\mathrm{ref}} = (1-\lambda_p^{\mathrm{tr}}) f_\theta(\mathbf{h}_p) + \lambda_p^{\mathrm{tr}} f_\theta(\widetilde{\mathbf{h}}_p). \tag{1}$$

Thus a well-supported transfer uses more of the boundary-oriented score, while a doubtful transfer stays closer to the observed TN. The role classifier, projections, and weights remain fixed as f_θ learns. Appendix A.2 specifies their construction.

Protecting uncertain alarms. Changing the TN comparison score alone does not prevent a plausible FP from being repeatedly pushed below it. We assign each alarm a *soft protection* weight $q_i^{\mathrm{prot}} \in [0,1]$, computed separately in each pool $\mathcal{V} \in \{\mathcal{D}_U, \mathcal{C}\}$. Training-alarm evidence combines running auditor scores and proximity to the role boundary. Target-alarm evidence also uses frozen precursor predictions, with transfer trust and allocation overlap moderating their influence. Precursors initialize protection; the independent reference auditors below provide the distillation targets.

Let $m_\mathcal{V}$ be the protection mass assigned to pool $\mathcal{V}$. An order-preserving allocator $\mathcal{A}_{m_\mathcal{V}}$ maps alarm evidence e_i to $q_i^{\mathrm{prot}} = \sigma(e_i - \xi)$, where the shared offset ξ is chosen so that $\sum_{i\in\mathcal{V}} q_i^{\mathrm{prot}} = m_\mathcal{V}$. Here $\sigma(z) = (1+\exp(-z))^{-1}$ is the logistic function. The mass is $m_\mathcal{V} = \lceil \widehat{\rho}_{\mathrm{tr}} |\mathcal{V}| \rceil$. The training-only quantity $\widehat{\rho}_{\mathrm{tr}}$ measures TN-supported safe mass among alarms; it is not claimed to identify their global FP prevalence (Appendix A.6). The mass $m_\mathcal{V}$ is distinct from the human review budget B.

Let $\ell_{i,\mathcal{V}}$ be the comparison loss for alarm i, averaged over the transferred TN scores s_p^{ref}. With $\eta_\mathcal{C} \geq 0$ denoting the relative use of the unlabeled target cohort, we train against

$$\mathcal{L}_{\mathrm{cmp},\mathcal{V}} = \frac{\sum_{i\in\mathcal{V}} (1-q_i^{\mathrm{prot}}) \ell_{i,\mathcal{V}}}{\sum_{i\in\mathcal{V}} (1-q_i^{\mathrm{prot}})}, \qquad \mathcal{J}_{\mathrm{PU}} = \frac{\mathcal{L}_{\mathrm{cmp},\mathcal{D}_U} + \eta_\mathcal{C} \mathcal{L}_{\mathrm{cmp},\mathcal{C}}}{1+\eta_\mathcal{C}}. \tag{2}$$

Protection reduces comparison pressure; it does not assign an alarm a safe label. The two pools retain separate updating memories. A reference auditor minimizes $\mathcal{L}_{\mathrm{base}} + \mathcal{J}_{\mathrm{PU}}$.

3.2.2 Reliability-Gated Rank Distillation

Consensus positions. Even with adapted supervision, independently fitted PU auditors can place uncertain alarms in different positions. We use a frozen bank $\{f_{\phi_k}\}_{k=1}^K$ and transfer its ordering, rather than its raw score scale, into one student. Let $\mathrm{Rank}(\mathbf{a})$ denote ascending midpoint ranks normalized to $(0,1)$; equal values receive equal average ranks. For reference k, define $r_i^{(k)} = \mathrm{Rank}((f_{\phi_k}(\mathbf{h}_j))_{j=1}^N)_i$. Their median and a fixed soft allocation are defined using $\mathrm{logit}(u) = \log(u/(1-u))$:

$$\bar{r}_i = \mathrm{median}_{1\leq k\leq K}\, r_i^{(k)}, \qquad \mathbf{q}^{\mathrm{ref}} = \mathcal{A}_{m_\mathcal{C}}\big(\mathrm{logit}(\bar{\mathbf{r}})\big). \tag{3}$$

The median limits the effect of an extreme reference position. $\mathbf{q}^{\mathrm{ref}}$ is fixed during student fitting and encodes review-order preference, not a safety label.

Pairwise transfer. For target alarms $i \neq j$, let $w_{ij}^{\mathrm{dist}} = q_i^{\mathrm{ref}}(1-q_j^{\mathrm{ref}})|\bar{r}_i - \bar{r}_j|$. Let $s_i = f_\theta(\mathbf{h}_i)$ be the student score and $\mu \geq 0$ the ranking margin. The distillation loss is

$$\mathcal{L}_{\mathrm{dist}} = \frac{\sum_{i\neq j} w_{ij}^{\mathrm{dist}} \log(1+\exp(\mu - s_i + s_j))}{\sum_{i\neq j} w_{ij}^{\mathrm{dist}}}. \tag{4}$$

Both pair directions remain in the loss. Their unequal weights favor the reference-preferred direction; the rank gap downweights nearly tied positions. The gap is a measure of *separation*, not pair-specific agreement among references.

One reliability gate. Agreement alone cannot reveal a bias shared by all references. We therefore use a cohort-level gate $\lambda_{\mathrm{dist}} = o_{\mathrm{ref}}(\bar{\lambda}^{\mathrm{tr}})^2$, where $o_{\mathrm{ref}} \in [0,1]$ is the soft-allocation overlap of a fixed reference pair and $\bar{\lambda}^{\mathrm{tr}}$ is mean TN transfer weight. A lower overlap or weaker transfer reduces the influence of distillation. The student minimizes

$$\mathcal{L}_{\mathrm{train}} = \mathcal{L}_{\mathrm{base}} + \frac{\mathcal{J}_{\mathrm{PU}} + \lambda_{\mathrm{dist}} \mathcal{L}_{\mathrm{dist}}}{1+\lambda_{\mathrm{dist}}}. \tag{5}$$

The base terms retain their weights; only PU comparison and distillation are blended. Reference ranks and the gate stay fixed while the student's protection memory updates. Stage I returns one student, and keeps reference-rank dispersion for Stage II (Appendix A.4).

3.3 Stage II: Consensus-Guided Structural Refinement

The student captures a common ordering, but that ordering does not explicitly express how TN references support alarms at different granularities or how alarms relate within the target cohort. We freeze the student and use those two kinds of structure to refine its review queue. No student parameter is updated in this stage.

Hierarchical TN support. Training role–group centers form a root–parent–leaf hierarchy. When local TN support is sparse, the hierarchy allows a score to borrow support from a broader ancestor. Holding the student score fixed, we form three *score views* on the cohort: $\mathbf{s}^{\mathrm{local}}$ uses local support, $\mathbf{s}^{\mathrm{flat}}$ shrinks directly from leaf to root, and $\mathbf{s}^{\mathrm{hier}}$ shares support through the parent. Their matched contrast defines

$$\mathbf{s}^{(0)} = \mathbf{s}^{\mathrm{local}} + (\mathbf{s}^{\mathrm{hier}} - \mathbf{s}^{\mathrm{flat}}). \tag{6}$$

The difference adds parent-level information without counting the shared student score twice. Appendix B.1 defines the support and the three views.

Relations among target alarms. The three views also suggest preferences between alarms. We retain a cross-group directed pair (i, j) only when the root view and at least one other support view agree that i should precede j. Let $\mathcal{E}$ be this edge set, $w_{ij}^{\mathrm{str}} \geq 0$ its support weight, and $\delta_{ij} \geq 0$ its desired separation. A sigmoid transform of the ranks of $\mathbf{s}^{(0)}$ yields $\mathbf{p} \in (0,1)^N$. We solve the following constrained problem, where $\mathrm{Ber}(u)$ denotes a Bernoulli distribution with mean u, KL is its Kullback–Leibler divergence, and $[z]_+ = \max\{z, 0\}$:

$$\begin{aligned} \min_{\mathbf{v}\in\Omega(\mathbf{p})} \quad & \frac{1}{N}\sum_i \mathrm{KL}(\mathrm{Ber}(v_i) \,\|\, \mathrm{Ber}(p_i)) + \frac{\sum_{(i,j)\in\mathcal{E}} w_{ij}^{\mathrm{str}}[\delta_{ij} - v_i + v_j]_+^2}{\sum_{(i,j)\in\mathcal{E}} w_{ij}^{\mathrm{str}}}, \\ \Omega(\mathbf{p}) &= \{\mathbf{v} \in [0,1]^N : \mathbf{1}^\top \mathbf{v} = \mathbf{1}^\top \mathbf{p}\}. \end{aligned} \tag{7}$$

The first term keeps the solution near the hierarchy-supported initialization; the second penalizes violated relations. The constraints retain its total soft mass. With no supported relations, the solution is $\mathbf{p}$. Let $\widehat{\mathbf{v}}$ be the finite solver output. The structural ordering is formed with the endpoint-normalized ascending rank Rank_{01}, which maps the smallest and largest positions to 0 and 1 (Appendix A.1):

$$\mathbf{s}^{\mathrm{str}} = (1-\alpha)\,\mathrm{Rank}_{01}(\mathbf{s}^{(0)}) + \alpha\,\mathrm{Rank}_{01}(\widehat{\mathbf{v}}), \tag{8}$$

where $\alpha \in [0,1]$ controls the contribution of cohort relations. Appendices A.1 and B.2 give the operators and solver details.

Agreement-guided readout. We use the reference bank to estimate how stable the ranking of each alarm is. Its rank spread and relative agreement target are

$$d_i = \max_{1\leq k\leq K} r_i^{(k)} - \min_{1\leq k\leq K} r_i^{(k)}, \qquad c_i^\star = 1 - \mathrm{Rank}(\mathbf{d})_i. \tag{9}$$

A ridge head predicts $c_i \in [0,1]$ from standardized frozen monitor features. It is fitted once per cohort to $c_i^\star$, without alarm safety labels, and shared across student-seed repetitions. Unlike the cohort-level distillation gate, c_i controls an *individual* alarm's final adjustment.

Let $\mathbf{r}^{\mathrm{stu}} = \mathrm{Rank}((f_\theta(\mathbf{h}_i))_{i=1}^N)$ and $\mathbf{r}^{\mathrm{str}} = \mathrm{Rank}(\mathbf{s}^{\mathrm{str}})$. For a fixed adjustment scale $\beta \geq 0$, final review priority is

$$s_i^{\mathrm{out}} = r_i^{\mathrm{str}} + \beta c_i (r_i^{\mathrm{str}} - r_i^{\mathrm{stu}}). \tag{10}$$

Predicted agreement determines how far to move along the structural correction; it is not itself evidence of safety. The final scores need not lie in $[0,1]$ and are sorted in descending order. Appendix B.3 specifies feature standardization, ridge fitting, and score handling.

The fitted cohort is scored with one student, the agreement head, and stored structural statistics. A new cohort requires its own reference ranks and pool-dependent statistics; it does not require changing the frozen monitor's decisions.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Protocol. We evaluate four dataset–monitor settings on ATBench (Li et al., 2026) and TraceSafe (Chen et al., 2026), covering AgentDoG (Liu et al., 2026), Granite Guardian (Padhi et al., 2025), and StepGuard (Zheng et al., 2026). Each setting provides 40 verified-safe non-alarms and unlabeled training alarms, with 510 target alarms in total. Target labels remain sealed until all rankings are fixed, and all safety monitors remain frozen.

Baselines and metrics. We compare against eight PU methods: nnPU (Kiryo et al., 2017), DC-PU (Li et al., 2025), Dist-PU (Zhao et al., 2022), Robust-PU (Zhu et al., 2023), PULDA (Jiang et al., 2023), PUL-CPBF (Li et al., 2024), HolisticPU (Wang et al., 2023), and feature-adapted NcPU (Zhao et al., 2026). All methods use the same frozen features and data splits. We report macro AUPRC/AUROC (↑), AUGRC (↓), and false alarms recovered under 5%, 10%, and 20% review budgets. Full implementation and evaluation details are in Appendix A.

4.2 RANKING PERFORMANCE AND REVIEW YIELD

Our method achieves macro AUPRC/AUROC/AUGRC of $64.44/61.70/22.85$***, improving over the strongest PU baseline by*** $5.27/2.32/0.56$ ***favorable points and obtaining the strongest result across all*** 96 ***evaluated setting–seed–baseline comparisons. The advantage is particularly pronounced under limited review budgets: at*** 5%***, we recover*** 22.67 ***false alarms versus PULDA's*** 17.00 ***(***$+33.3\%$***), while the gain remains substantial at*** 20% ***(***65.00 ***versus*** 54.33***,*** $+19.6\%$***).***

Table 1: **Ranking performance and review yield.** Upper: two ATBench settings. Lower: four-setting macro mean±std over seeds $\{13, 37, 73\}$ $(100\times)$ and summed FP yield. **Bold** and underlining mark the best and second-best values.

	ATBench AgentDoG1.5-Qwen3.5-2B			ATBench Granite-Guardian-3.1-2B		
Method (year)	AUPRC ↑	AUROC ↑	AUGRC ↓	AUPRC ↑	AUROC ↑	AUGRC ↓
nnPU (2017)	61.75±0.09	65.94±0.15	19.66±0.04	60.73±0.24	50.97±0.35	21.09±0.09
Dist-PU (2022)	70.04±2.76	70.00±3.57	18.65±0.89	68.16±1.12	57.81±1.57	19.43±0.38
Robust-PU (2023)	62.02±0.55	66.75±0.71	19.46±0.18	60.24±0.19	50.78±0.14	21.14±0.03
PULDA (2023)	72.89±0.92	73.59±0.51	17.76±0.13	69.82±0.65	58.43±1.17	19.28±0.29
HolisticPU (2023)	61.85±1.18	65.77±1.14	19.71±0.28	57.97±0.54	48.06±0.53	21.80±0.13
PUL-CPBF (2024)	61.95±2.06	64.70±2.09	19.97±0.52	61.56±2.24	51.28±4.26	21.02±1.04
DC-PU (2025)	51.23±8.92	47.35±11.13	24.30±2.75	51.72±2.30	42.99±4.03	23.04±0.98
NcPU (2026)	60.46±1.50	64.57±1.69	20.00±0.42	61.19±0.39	51.52±0.69	20.96±0.17
Ours	**79.48**±0.65	**76.67**±0.59	**17.00**±0.15	**73.91**±0.81	**62.55**±0.86	**18.27**±0.21

	Average over all four combinations			FPs found over all four combinations		
Method (year)	AUPRC ↑	AUROC ↑	AUGRC ↓	Top 5%	Top 10%	Top 20%
nnPU (2017)	53.41±0.17	55.15±0.14	24.45±0.03	12.67	24.67	52.33
Dist-PU (2022)	57.78±0.45	58.08±0.59	23.73±0.15	16.00	29.33	54.33
Robust-PU (2023)	53.14±0.17	54.99±0.16	24.49±0.04	12.00	24.00	50.00
PULDA (2023)	59.17±0.25	59.38±0.40	23.41±0.10	17.00	31.00	53.33
HolisticPU (2023)	53.55±0.43	55.28±0.38	24.42±0.09	13.33	25.00	52.33
PUL-CPBF (2024)	52.96±1.19	54.03±1.98	24.74±0.49	11.00	25.00	48.33
DC-PU (2025)	47.46±2.78	47.50±4.21	26.26±1.08	11.33	19.33	43.67
NcPU (2026)	53.71±0.28	55.59±0.50	24.35±0.12	14.67	24.67	51.67
Ours	**64.44**±0.21	**61.70**±0.20	**22.85**±0.05	**22.67**	**37.00**	**65.00**

The 5%, 10%, and 20% budgets review 27, 52, and 104 alarms, respectively, across 510 target alarms containing 239 hidden FPs.

4.3 Large-scale External Validation

We further evaluate on Agent-SafetyBench (Zhang et al., 2024), yielding 4,054 held-out alarms from 1,400 evaluation cases and two frozen monitors. ***Our method improves PULDA from*** 24.39 ***to*** 28.72 ***AUPRC and increases Top-***5% ***recovery from*** 66 ***to*** 94 ***false alarms*** (+42.4%)***.*** The AUPRC improvement is supported by a case-level 95% bootstrap interval of $[3.12, 5.58]$. Even the $K = 1$ variant outperforms a three-model PULDA ensemble (26.15 versus 25.42), while the full method maintains its advantage as the target cohort grows from 1,034 to 4,054 alarms. This consistent scaling indicates that the improvement transfers beyond the smaller primary evaluation to substantially larger audit pools.

4.4 Direct LLM Re-judgment

We compare against a zero-shot Qwen2.5-7B-Instruct judge that independently re-evaluates each policy, task, and trajectory without access to labels, safe references, or monitor scores. Although direct semantic re-judgment improves over the native monitor order, our method achieves substantially higher AUPRC (64.44 versus 57.78) and recovers 4.67, 9.00, and 15.00 more false alarms at the 5%, 10%, and 20% budgets, respectively (Figure 4(a,d)). ***This gap shows that cohort-level PU supervision provides information beyond independent semantic re-evaluation of individual alarms.***

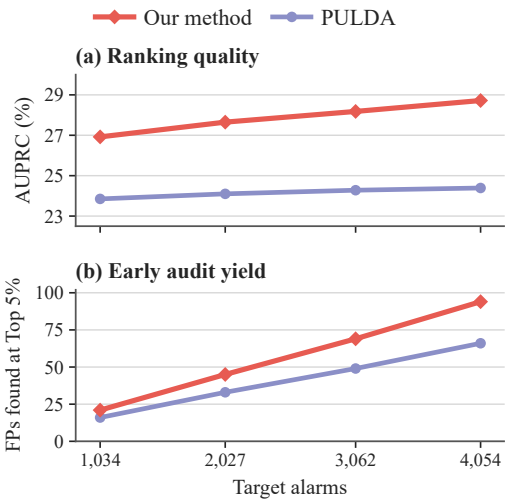


Figure 3: **External cohort scaling.** Macro AUPRC and Top-5% false-alarm recovery.

4.5 Ablation and Robustness

Each component contributes consistently: trust-aware supervision, rank distillation, and structural refinement increase AUPRC from 58.42 ***to*** 60.35***,*** 61.92***, and*** 64.44***, respectively, while improving all three ranking metrics at every stage.***

Table 4: **Cumulative component ablation.** Rows add one module; metrics use a 100× scale. **Bold**/underlining: best/second-best.

ID	Configuration	AUPRC ↑	AUROC ↑	AUGRC ↓
(1)	Base PU student	58.42	57.80	23.650
(2)	+ Trust-aware PU supervision	60.35	59.92	23.210
(3)	+ Reliability-gated rank distillation	<u>61.92</u>	<u>61.34</u>	<u>22.955</u>
(4)	**+ Structural queue refinement (Ours)**	**64.44**	**61.70**	**22.846**

Each component improves all three metrics; together they add 6.02 AUPRC and 3.90 AUROC points and reduce AUGRC by 0.804.

The improvement is not explained by simple ensembling: our single-reference variant reaches 62.35 AUPRC, exceeding the three-model PULDA ensemble at 60.24. The method is also robust to its training-only prior-mass estimate: across a 0.5–8× range, AUPRC remains 61.52–64.44 and Top-5% recovery remains 21.67–22.67. Performance further improves with additional verified-safe supervision, rising from 59.28 AUPRC with 10 references to 66.45 with 80. At the default 40-reference setting, $K = 3$ captures nearly all of the observed benefit from increasing the reference bank (Figure 4(b–f)). Together, these results show that the gains persist across both supervision levels and reference-bank configurations.

Table 2: **Large-scale external validation on Agent-SafetyBench.** Ranking metrics are two-monitor macro scores (100×); FP columns report recovered false alarms at each review budget.

Method	AUPRC ↑	AUROC ↑	AUGRC ↓	FP@5% ↑	FP@10% ↑	FP@20% ↑
Dist-PU	24.16	57.20	40.97	66	110	165
PULDA	24.39	58.57	40.74	66	106	186
PULDA ensemble	25.42	59.45	40.21	71	114	195
Our method ($K = 1$)	26.15	60.32	39.68	76	125	208
Our method ($K = 3$)	**28.72**	**63.45**	**37.82**	**94**	**152**	**241**

Table 3: **Local LLM re-judgment control.** Four-setting macro ranking and summed FP yield; the Judge uses its frozen bidirectional score. **Bold**/underlining: best/second-best.

	Macro ranking (100×)			FPs recovered		
Method	AUPRC ↑	AUROC ↑	AUGRC ↓	Top 5%	Top 10%	Top 20%
Native monitor score	53.39	55.21	24.71	12.00	24.00	50.00
PULDA	<u>59.17</u>	<u>59.38</u>	<u>23.41</u>	17.00	<u>31.00</u>	<u>53.33</u>
Qwen2.5-7B judge	57.78	57.17	23.97	<u>18.00</u>	28.00	50.00
Our method	**64.44**	**61.70**	**22.85**	**22.67**	**37.00**	**65.00**

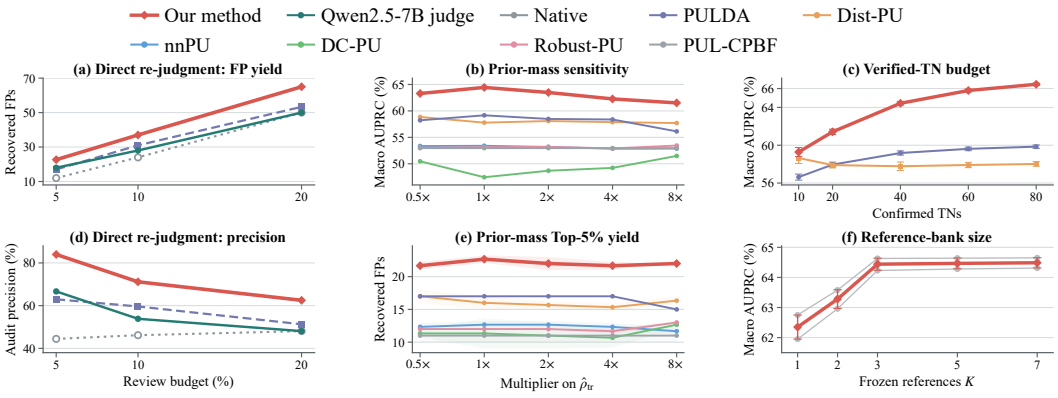


Figure 4: **Controls and sensitivity analyses.** (a,d) Direct LLM re-judgment; (b,e) prior-mass sensitivity; (c) verified-safe supervision; (f) reference-bank size.

4.6 Deployment Cost

With cached features, the default $K = 3$ configuration requires 162.86 seconds of offline fitting on one NVIDIA H200. Deployment uses a single student scorer without reference-model ensembling, keeping the final auditing pipeline compact despite the richer training procedure. Detailed cost analysis is provided in Appendix C.

5 Conclusion

In this work, we study false-alarm auditing for agent safety monitors under monitor-induced selection, where verified-safe non-alarms may poorly represent the hidden safe alarms of interest. We develop a two-stage PU ranking framework that first adapts verified-safe supervision toward the alarm domain and reduces excessive negative pressure on plausible false alarms, then stabilizes ranking preferences across independently trained PU models and further exploits hierarchical safe-reference support and relations among alarms. Across mainstream safety monitors, our method improves macro AUPRC over eight PU baselines by 5.27–16.98 absolute percentage points and recovers 33.3% more false alarms than PULDA at a 5% review budget, with consistent gains on a larger external benchmark of 4,054 held-out alarms. The framework requires no alarm safety labels for fitting and leaves the monitor decisions unchanged.

AI USE STATEMENT

Generative AI tools were used to assist with language editing, manuscript organization, and feedback on presentation. All research decisions, experimental designs, implementations, analyses, and reported results were reviewed and verified by the authors, who take full responsibility for the final content.

ETHICS STATEMENT

This work prioritizes alarms for human review rather than automatically suppressing them. A high-ranked alarm is not certified safe, and the original monitor decisions remain in force. Deployment should keep reviewers in the loop and respect the privacy and access conditions of the underlying agent traces.

REPRODUCIBILITY STATEMENT

The main text and appendix specify the method, evaluation protocol, and fixed training configuration. The anonymous repository contains the core implementation, fixed configurations, sample manifests, frozen predictions, implementation tests, and scripts for regenerating the reported results. Benchmark data and checkpoints follow their original distribution licenses.

REFERENCES

Bushra A. Alahmadi, Louise Axon, and Ivan Martinovic. 99% false positives: A qualitative study of SOC analysts' perspectives on security alarms. In *31st USENIX Security Symposium (USENIX Security 22)*, pp. 2783–2800. USENIX Association, 2022. URL https://www.usenix.org/conference/usenixsecurity22/presentation/alahmadi.

Jessa Bekker and Jesse Davis. Learning from positive and unlabeled data under the selected at random assumption. In *Proceedings of the Second International Workshop on Learning with Imbalanced Domains: Theory and Applications*, volume 94 of *Proceedings of Machine Learning Research*, pp. 8–22. PMLR, 2018. URL https://proceedings.mlr.press/v94/bekker18a.html.

Eric T. Chancey, James P. Bliss, Yusuke Yamani, and Holly A. H. Handley. Trust and the compliance–reliance paradigm: The effects of risk, error bias, and reliability on trust and dependence. *Human Factors*, 59(3):333–345, 2017. doi: 10.1177/0018720816682648. URL https://journals.sagepub.com/doi/10.1177/0018720816682648.

Yen-Shan Chen, Sian-Yao Huang, Cheng-Lin Yang, and Yun-Nung Chen. TraceSafe: A systematic assessment of LLM guardrails on multi-step tool-calling trajectories. *arXiv preprint arXiv:2604.07223*, 2026. URL https://arxiv.org/abs/2604.07223.

Zhaorun Chen, Mintong Kang, and Bo Li. ShieldAgent: Shielding agents via verifiable safety policy reasoning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 8313–8344. PMLR, 2025. URL https://proceedings.mlr.press/v267/chen25ae.html.

Edoardo Debenedetti, Jie Zhang, Mislav Balunovic, Luca Beurer-Kellner, Marc Fischer, and Florian Tramèr. AgentDojo: A dynamic environment to evaluate prompt injection attacks and defenses for LLM agents. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-2636. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/97091a5177d8dc64b1da8bf3e1f6fb54-Abstract-Datasets_and_Benchmarks_Track.html.

Charles Elkan and Keith Noto. Learning classifiers from only positive and unlabeled data. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 213–220. ACM, 2008. doi: 10.1145/1401890.1401920. URL https://dl.acm.org/doi/10.1145/1401890.1401920.

Yangbangyan Jiang, Qianqian Xu, Yunrui Zhao, Zhiyong Yang, Peisong Wen, Xiaochun Cao, and Qingming Huang. Positive-unlabeled learning with label distribution alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(12):15345–15363, 2023. doi: 10.1109/TPAMI.2023.3319431.

Ryuichi Kiryo, Gang Niu, Marthinus C. du Plessis, and Masashi Sugiyama. Positive-unlabeled learning with non-negative risk estimator. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Changchun Li, Yuanchao Dai, Lei Feng, Ximing Li, Bing Wang, and Jihong Ouyang. Positive and unlabeled learning with controlled probability boundary fence. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 27641–27652. PMLR, 2024.

Ximing Li, Yuanchao Dai, Bing Wang, Changchun Li, Jianfeng Qu, and Renchu Guan. Balancing positive and negative classification error rates in positive-unlabeled learning. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-2456.

Yu Li, Haoyu Luo, Yuejin Xie, Yuqian Fu, Zhonghao Yang, Shuai Shao, Qihan Ren, Wanying Qu, Yanwei Fu, Yujiu Yang, Jing Shao, Xia Hu, and Dongrui Liu. ATBench: A diverse and realistic agent trajectory benchmark for safety evaluation and diagnosis. *arXiv preprint arXiv:2604.02022*, 2026. URL https://arxiv.org/abs/2604.02022.

Dongrui Liu, Qihan Ren, Chen Qian, Shuai Shao, Yuejin Xie, Yu Li, Zhonghao Yang, Haoyu Luo, Peng Wang, Qingyu Liu, Binxin Hu, Ling Tang, Jilin Mei, Dadi Guo, Leitao Yuan, Junyao Yang, Guanxu Chen, Qihao Lin, Yi Yu, Bo Zhang, Jiaxuan Guo, Jie Zhang, Wenqi Shao, Huiqi Deng, Zhiheng Xi, Wenjie Wang, Wenxuan Wang, Wen Shen, Zhikai Chen, Haoyu Xie, Jialing Tao, Juntao Dai, Jiaming Ji, Zhongjie Ba, Linfeng Zhang, Yong Liu, Quanshi Zhang, Lei Zhu, Zhihua Wei, Hui Xue, Chaochao Lu, Jing Shao, and Xia Hu. AgentDoG: A diagnostic guardrail framework for AI agent safety and security. *arXiv preprint arXiv:2601.18491*, 2026. doi: 10.48550/arXiv.2601.18491. URL https://arxiv.org/abs/2601.18491.

Jonathan Oliver, Raghav Batta, Adam Bates, Muhammad Adil Inam, Shelly Mehta, and Shugao Xia. Carbon filter: Real-time alert triage using large scale clustering and fast search. *arXiv preprint arXiv:2405.04691*, 2024. doi: 10.48550/arXiv.2405.04691. URL https://arxiv.org/abs/2405.04691.

Inkit Padhi, Manish Nagireddy, Giandomenico Cornacchia, Subhajit Chaudhury, Tejaswini Pedapati, Pierre Dognin, Keerthiram Murugesan, Erik Miehling, Martin Santillan Cooper, Kieran Fraser, et al. Granite Guardian: Comprehensive LLM safeguarding. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2025. URL https://research.ibm.com/publications/granite-guardian-comprehensive-llm-safeguarding.

Tomoya Sakai, Gang Niu, and Masashi Sugiyama. Semi-supervised AUC optimization based on positive–unlabeled learning. *Machine Learning*, 107(4):767–794, 2018. doi: 10.1007/s10994-017-5678-9.

Bernhard Schölkopf, John C. Platt, John Shawe-Taylor, Alex J. Smola, and Robert C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001. doi: 10.1162/089976601750264965.

Xinrui Wang, Wenhai Wan, Chuanxing Geng, Shaoyuan Li, and Songcan Chen. Beyond myopia: Learning from positive and unlabeled data through holistic predictive trends. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

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. GuardAgent: Safeguard LLM agents via knowledge-enabled reasoning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 68316–68342. PMLR, 2025. URL https://proceedings.mlr.press/v267/xiang25a.html.

Limin Yang, Zhi Chen, Chenkai Wang, Zhenning Zhang, Sushruth Booma, Phuong Cao, Constantin Adam, Alexander Withers, Zbigniew Kalbarczyk, Ravishankar K. Iyer, and Gang Wang. True attacks, attack attempts, or benign triggers? an empirical measurement of network alerts in a security operations center. In *33rd USENIX Security Symposium (USENIX Security 24)*, pp. 1525–1542. USENIX Association, 2024. URL https://www.usenix.org/conference/usenixsecurity24/presentation/yang-limin.

Tongxin Yuan, Zhiwei He, Lingzhong Dong, Yiming Wang, Ruijie Zhao, Tian Xia, Lizhen Xu, Binglin Zhou, Fangqi Li, Zhuosheng Zhang, Rui Wang, and Gongshen Liu. R-Judge: Benchmarking safety risk awareness for LLM agents. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 1467–1490. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-emnlp.79.

Zhexin Zhang, Shiyao Cui, Yida Lu, Jingzhuo Zhou, Junxiao Yang, Hongning Wang, and Minlie Huang. Agent-safetybench: Evaluating the safety of LLM agents. *arXiv preprint arXiv:2412.14470*, 2024. URL https://arxiv.org/abs/2412.14470.

Hengwei Zhao, Zhengzhong Tu, Zhuo Zheng, Wei Wang, Junjue Wang, Rusty Feagin, and Wenzhe Jiao. Noisy-pair robust representation alignment for positive-unlabeled learning. In *International Conference on Learning Representations*, 2026.

Yunrui Zhao, Qianqian Xu, Yangbangyan Jiang, Peisong Wen, and Qingming Huang. Dist-PU: Positive-unlabeled learning from a label distribution perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14461–14470, 2022.

Zhijie Zheng, Yu Li, Chen Qian, Yuqian Fu, Yanwei Fu, Lu Sheng, Jing Shao, and Dongrui Liu. StepGuard: Learning step-level guardrails with scalable supervision and safety-utility balancing. *arXiv preprint arXiv:2608.24777*, 2026. URL https://arxiv.org/abs/2608.24777.

Zhangchi Zhu, Lu Wang, Pu Zhao, Chao Du, Wei Zhang, Hang Dong, Bo Qiao, Qingwei Lin, Saravan Rajmohan, and Dongmei Zhang. Robust positive-unlabeled learning via noise negative sample self-correction. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 3663–3673, 2023. doi: 10.1145/3580305.3599491.

A Training Definitions and Implementation

We provide the implementation details required to reproduce the reported system. All training diagnostics use only verified-safe TNs and unlabeled training alarms, while target safety labels remain sealed until all ranking scores are fixed. The same configuration is used across all reported dataset–monitor settings.

A.1 Rank Operators

For $\mathbf{a} \in \mathbb{R}^n$, let $\mathrm{arank}(\mathbf{a})$ denote ascending ranks with midpoint tie handling. Reference models use

$$\mathrm{Rank}(\mathbf{a})_i = \frac{\mathrm{arank}(\mathbf{a})_i - \frac{1}{2}}{n}. \tag{11}$$

Structural refinement uses $\mathrm{Rank}_{01}(\mathbf{a})_i = (\mathrm{arank}(\mathbf{a})_i - 1)/(n-1)$ for $n > 1$. Sample ID resolves remaining ties. We use $\sigma(z) = (1 + \exp(-z))^{-1}$ and $\mathrm{logit}(u) = \log(u/(1-u))$.

A.2 TN Transfer and Soft Protection

Transfer diagnostics. A five-fold, isolation-group-disjoint logistic role classifier is fitted on verified TNs and training alarms using normalized monitor and task representations. For TN τ_p, let ℓ_p be its held-out role logit, A_{role} the task-block AUC, and $\mathbf{t}_p$ its normalized task representation. With role-classifier parameters $\varphi = (\mathbf{w}_h, \mathbf{w}_t, b)$, define

$$\mathbf{a} = \frac{\mathbf{w}_h}{\|\mathbf{w}_h\|_2}, \qquad c_p = -\frac{\mathbf{w}_t^\top \mathbf{t}_p + b}{\|\mathbf{w}_h\|_2}, \qquad \mathbf{x}_p = \frac{\mathbf{h}_p}{\|\mathbf{h}_p\|_2}.$$

The nearest role-boundary representation preserving task features and hidden norm is

$$\widetilde{\mathbf{h}}_p = \|\mathbf{h}_p\|_2 \left[c_p \mathbf{a} + \sqrt{1 - c_p^2} \frac{\mathbf{x}_p - (\mathbf{a}^\top \mathbf{x}_p)\mathbf{a}}{\|\mathbf{x}_p - (\mathbf{a}^\top \mathbf{x}_p)\mathbf{a}\|_2} \right]. \tag{12}$$

Let $d_p = \|\widetilde{\mathbf{h}}_p / \|\mathbf{h}_p\|_2 - \mathbf{x}_p\|_2$. Transfer support is

$$u_p^{\mathrm{role}} = \exp\left[- \left(\frac{1 - A_{\mathrm{role}}}{0.02} \right)^8 - |\ell_p|^8 \right], \qquad u_p^{\mathrm{geom}} = \exp\left[- \left(\frac{1 - A_{\mathrm{role}}}{0.02} \right)^8 - \left(\frac{d_p}{0.12} \right)^8 \right]. \tag{13}$$

With $c_{75} = \exp[-Q_{0.75}(|\ell|)^8]$ computed from training TNs,

$$\lambda_p^{\mathrm{tr}} = c_{75} u_p^{\mathrm{role}} + (1 - c_{75}) \min(u_p^{\mathrm{role}}, u_p^{\mathrm{geom}}) \in [0, 1]. \tag{14}$$

All transfer quantities are fixed before PU fitting.

Soft protection. Each alarm receives evidence e_i from running ranking information and role-boundary proximity. The order-preserving allocator produces

$$q_i^{\mathrm{prot}} = \sigma(e_i - \xi), \qquad \sum_{i \in \mathcal{V}} q_i^{\mathrm{prot}} = m_{\mathcal{V}}, \tag{15}$$

where ξ is obtained by bisection. Training and target pools maintain separate memories. Protection adjusts comparison strength while preserving the original unlabeled status of every alarm.

Task-conditioned comparisons. For normalized task representations $\mathbf{t}_p, \mathbf{t}_i$, define

$$D_{pi} = \max\{0, 2 - 2\mathbf{t}_p^\top \mathbf{t}_i\}, \qquad \kappa_{pi} = \exp[-D_{pi}/(2h_{\mathrm{task}})],$$

where h_{task} is the median positive training TN–alarm distance. For TN minibatch $\mathcal{B}_P$, the comparison term is

$$\ell_{i,\mathcal{V}} = \sum_{p \in \mathcal{B}_P} \nu_{pi,\mathcal{V}} \log(1 + \exp(\mu - s_p^{\mathrm{ref}} + s_i)). \tag{16}$$

These weights redistribute comparison pressure without introducing alarm labels.

A.3 PU Objective and Reference Fits

Every fitted scorer uses the common objective

$$\mathcal{L}_{\mathrm{base}} = \mathcal{L}_{\mathrm{align}} + \lambda_{\mathrm{mix}}(t) \mathcal{L}_{\mathrm{mix}}. \tag{17}$$

For TN and alarm minibatches, let $\bar{p}_P$ and $\bar{p}_U$ be their mean predicted safety probabilities. With $\zeta_T(z) = T^{-1} \log(1 + \exp(Tz))$ and $D_T(p, \rho) = \zeta_T(p - \rho) + \zeta_T(\rho - p)$, the alignment term is

$$\mathcal{L}_{\mathrm{align}} = 2\rho(1 - \bar{p}_P) + \begin{cases} D_T(\bar{p}_U, \rho), & \text{first alarm minibatch,} \\ D_T(\widetilde{p}_U, \rho)/(1 - a_U), & \text{otherwise,} \end{cases} \tag{18}$$

where $\widetilde{p}_U = a_U \widetilde{p}_U^{\mathrm{prev}} + (1 - a_U)\bar{p}_U$ and $\rho = \widehat{\rho}_{\mathrm{tr}}$.

For group-structured mixing, let $\mathbf{c}_i$ be the role–isolation-group centroid, $\boldsymbol{\mu}_{\chi_i}$ the corresponding role mean, $\mathbf{b}_i = \mathbf{c}_i - \boldsymbol{\mu}_{\chi_i}$, and $\mathbf{u}_i = \mathbf{h}_i - \mathbf{c}_i$. For a cross-role pair (i, j),

$$\begin{aligned} \mathbf{h}_i^{\mathrm{mix}} = \xi_i \boldsymbol{\mu}_{\chi_i} + (1 - \xi_i)\boldsymbol{\mu}_{\chi_j} + \sqrt{\xi_i}\mathbf{b}_i + \sqrt{1 - \xi_i}\mathbf{b}_j \\ + \xi_i \mathbf{u}_i + (1 - \xi_i)\mathbf{u}_j. \end{aligned} \tag{19}$$

Same-role pairs use standard interpolation.

With detached soft targets $\widehat{y}_i$ and $p_i^{\mathrm{mix}} = \sigma(\mathrm{clip}(f_\theta(\mathbf{h}_i^{\mathrm{mix}}), -10, 10))$,

$$\mathcal{L}_{\mathrm{mix}} = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \left[\xi_i \, \mathrm{BCE}(p_i^{\mathrm{mix}}, \widehat{y}_i) + (1 - \xi_i) \, \mathrm{BCE}(p_i^{\mathrm{mix}}, \widehat{y}_{\varpi(i)}) \right]. \tag{20}$$

A.4 Reference Ranks and Student Distillation

Reference models use identical data, architecture, and PU objectives with independent initialization. Their frozen outputs are converted to normalized ranks, whose coordinatewise median yields $\bar{\mathbf{r}}$.

The fixed reference allocation is

$$\mathbf{q}^{\mathrm{ref}} = \mathcal{A}_{m_{\mathcal{C}}}(\mathrm{logit}(\bar{\mathbf{r}})).$$

For equal-mass allocations,

$$o(\mathbf{q}, \mathbf{q}') = \frac{\sum_i \min(q_i, q'_i)}{m_{\mathcal{C}}},$$

and the cohort-level distillation weight is

$$\lambda_{\mathrm{dist}} = o_{\mathrm{ref}}(\bar{\lambda}^{\mathrm{tr}})^2.$$

Reference ranks, pair weights, and the gate remain fixed while the student is optimized, so the reference bank contributes relative ordering information throughout distillation.

A.5 Optimization and Fixed Settings

All scorers use a two-hidden-layer MLP with width 1000. Adam uses learning rate 10^{-3}, weight decay 10^{-4}, and gradient clipping at 5. Each fit runs for 400 epochs with TN/alarm minibatches of size 64, with the first 200 epochs forming the common warm-up. All reported dataset–monitor settings use the same optimization schedule and hyperparameters; seeds 13, 37, and 73 differ only in initialization and minibatch order.

Table 5: **Fixed implementation settings.** Values are shared across all four dataset–monitor cells and all seeds.

Component	Setting
PU scorer	two hidden layers; width 1000 each; Adam, learning rate 10^{-3}, weight decay 10^{-4}; gradient clip 5
Training	400 epochs; TN/alarm batches $64/64$; $\max\{1, \lceil \lvert\mathcal{D}_U\rvert/64 \rceil\}$ updates per epoch; 200-epoch warm-up
Ranking and mixing	margin 0.6; post-warm-up $\lambda_{\mathrm{mix}} = 4.2$; $\mathrm{Beta}(11, 11)$ mixing; protection-memory EMA 0.5
Distribution alignment	smoothing temperature $T = 3.5$; alarm-mean EMA coefficient $a_U = 0.85$
Role and transfer	five group-disjoint role folds; logistic $C = 1$; L-BFGS tolerance 10^{-10}; role scale 1; displacement scale 0.12
Reference bank	$K = 3$; midpoint ranks; coordinatewise median; protection mass $\lceil \widehat{\rho}_{\mathrm{tr}} \lvert\mathcal{V}\rvert \rceil$
Structural readout	16 PCA components; root/parent/leaf branching $1/2/2$; 50 projected iterations; solver-rank blend 0.25
Agreement head	target standardization; constant-feature threshold 10^{-10}; Ridge strength 100; output clip $[0, 1]$; final coefficient 0.20
Evaluation	seeds $13, 37, 73$; checkpoint epoch 400; ascending sample ID resolves score ties

A.6 TN-Supported Mass and Sensitivity

The training-only quantity $\widehat{\rho}_{\mathrm{tr}}$ controls the amount of alarm mass eligible for soft protection rather than estimating global false-alarm prevalence.

For a TN-like upper tail $\mathcal{H}_t = \{\tau : g_\varphi(\tau) \geq t\}$,

$$R_t = \frac{\Pr_{\mathcal{D}_U}(\mathcal{H}_t)}{\Pr_{\mathcal{D}_P}(\mathcal{H}_t)} = \pi_U \kappa_t + (1 - \pi_U)\delta_t, \tag{21}$$

which motivates the observed tail ratio as a training-only support statistic.

Isolation groups are split into three folds and the statistic is computed from held-out role-classifier scores. A single fixed construction is used across all reported settings before target labels are accessed.

We evaluate $\{0.5, 1, 2, 4, 8\}\widehat{\rho}_{\mathrm{tr}}$ for all applicable methods. Our method remains strongest throughout the sweep, with AUPRC of 61.52–64.44 and Top-5% recovery of 21.67–22.67, compared with the strongest baseline operating points of 59.17 and 17.00, respectively.

Table 6: **Training-only TN-supported mass.** Values are computed without alarm safety labels.

Dataset	Monitor	$\widehat{\rho}_{\mathrm{tr}}$
ATBench	AgentDoG1.5-Qwen3.5-2B	0.02381
ATBench	Granite-Guardian-3.1-2B	0.02368
TraceSafe	AgentDoG-Qwen3-4B	0.01000
TraceSafe	StepGuard-4B	0.01266

Table 7: **Prior-mass sensitivity.** Four-setting macro AUPRC under multipliers of the training-only TN-supported mass.

Method	$0.5\times$	$1\times$	$2\times$	$4\times$	$8\times$
nnPU	53.36	53.41	53.18	52.86	52.87
DC-PU	50.48	47.46	48.69	49.22	51.49
Dist-PU	58.86	57.78	58.09	57.88	57.71
Robust-PU	53.04	53.14	53.21	52.91	53.45
PUL-CPBF	52.96	52.96	52.96	52.96	52.96
PULDA	58.23	59.17	58.48	58.38	56.13
Ours	**63.30**	**64.44**	**63.49**	**62.27**	**61.52**

Table 8: **Evaluation cells and fixed protocol.** Training alarms are unlabeled during fitting; target FP counts are used only for evaluation. Each cell contains 40 verified TNs.

Dataset	Monitor	Training alarms	Target alarms	Target FPs
ATBench	AgentDoG1.5-Qwen3.5-2B	249	56	30
ATBench	Granite-Guardian-3.1-2B	300	76	44
TraceSafe	AgentDoG-Qwen3-4B	949	269	122
TraceSafe	StepGuard-4B	504	109	43
Total target pool		—	510	239

Macro AUPRC, AUROC, and AUGRC give equal weight to the four settings. At review budget b, we inspect $\lceil bN_c \rceil$ alarms per setting, sum recovered FPs, and average over seeds. Checkpoints are fixed at epoch 400.

B Structural Queue Refinement

Structural refinement is a frozen-student readout using training-derived TN support, unlabeled target representations, and reference-rank consensus.

B.1 Hierarchical TN Support

Training role–group centers define a root–parent–leaf hierarchy. When local TN support is weak, support shrinks toward the nearest reliable ancestor:

$$u_g = \rho_g u_g^{\mathrm{local}} + (1-\rho_g) u_{\mathrm{par}(g)}, \qquad \rho_g \in [0,1]. \tag{22}$$

The reliability ρ_g depends only on training support and geometry. The local, flat, and hierarchical views share the same student contribution, so their contrast isolates additional hierarchical support.

B.2 Consensus-Supported Relations

Cross-group alarm relations are retained only when the root view and at least one more specific support view agree on direction. Starting from the hierarchy-supported initialization, we solve

$$\begin{aligned}\min_{\mathbf{v}\in\Omega(\mathbf{p})} \quad & \frac{1}{N}\sum_i \mathrm{KL}(\mathrm{Ber}(v_i)\|\,\mathrm{Ber}(p_i)) \\ & + \frac{\sum_{(i,j)\in\mathcal{E}} w_{ij}^{\mathrm{str}}[\delta_{ij}-v_i+v_j]_+^2}{\sum_{(i,j)\in\mathcal{E}} w_{ij}^{\mathrm{str}}}, \\ \Omega(\mathbf{p}) = \left\{\mathbf{v}\in[0,1]^N : \mathbf{1}^\top\mathbf{v} = \mathbf{1}^\top\mathbf{p}\right\}&.\end{aligned} \tag{23}$$

We use 50 projected iterations with monotone backtracking. The student remains frozen throughout refinement.

B.3 Reference Consensus and Final Readout

Reference dispersion for alarm i is $d_i = \max_k r_i^{(k)} - \min_k r_i^{(k)}$, with relative consensus target

$$c_i^\star = 1 - \mathrm{Rank}(\mathbf{d})_i. \tag{24}$$

A ridge head predicts this target from standardized frozen monitor representations. It is fitted once per target cohort using reference ranks only, without alarm safety labels.

Predicted agreement scales the final structural displacement. The same readout coefficient is used across all reported settings. Deployment requires only the frozen student and stored cohort statistics, without reference-model ensembling.

C Data Provenance and Experimental Controls

C.1 Benchmarks and Monitor Identifiers

ATBench follows Li et al. (2026); AgentDoG follows Liu et al. (2026); TraceSafe refers to TraceSafe-Bench (Chen et al., 2026); Granite Guardian follows Padhi et al. (2025); and StepGuard follows Zheng et al. (2026). Table 8 lists the exact benchmark–monitor pairings and sample counts.

C.2 Computational Cost

With cached monitor and task features, the default $K = 3$ configuration requires 162.86 seconds of offline cohort-level fitting on one NVIDIA H200. After fitting, deployment uses one student scorer together with lightweight stored cohort statistics, without reference-model ensembling.

Table 9: **Resource usage for the default $K = 3$ configuration.** Measurements start from cached monitor and task features on one NVIDIA H200.

Resource	$K = 3$
Total elapsed time (s)	162.86
Peak GPU allocation (MiB)	138.93
Peak GPU reservation (MiB)	164.00
Peak process memory (MiB)	2223.91

C.3 Local LLM-as-a-Judge Control

We evaluate Qwen2.5-7B-Instruct as a zero-shot semantic re-judgment baseline. For each target alarm, the model receives the safety policy, task instruction, and serialized trajectory, but no target label, verified-safe reference, target-set statistic, or monitor confidence. The prompt and bidirectional scoring rule are fixed before labels are opened.

Direct re-judgment reaches 57.78 macro AUPRC, while our method reaches 64.44 and recovers an additional 4.67, 9.00, and 15.00 false alarms at the 5%, 10%, and 20% review budgets, respectively.

C.4 Deployment Protocol

The method is designed for periodic batch auditing. Deployment uses the frozen student and stored structural statistics, with cohort-level statistics constructed using the same fixed procedure for each audit batch.

Reproducibility. The anonymous supplement contains fixed configurations, sample manifests, frozen predictions, implementation tests, and table-regeneration scripts. Target labels are accessed only after prediction outputs are fixed.

D Additional Experimental Controls

D.1 Selection and Representation Controls

Role propensity and SAR-EM test whether modeling monitor selection alone is sufficient, while deterministic representation baselines test whether the ranking can be reduced to proximity to verified TNs. The complete method outperforms both classes of controls.

D.2 Prior-Mass Sensitivity

The full sensitivity sweep is reported in Table 7 and visualized in Figure 4(b,e). Our method remains above the strongest baseline throughout the complete evaluated range.

D.3 Ensemble Control

Three-model logit averaging improves PULDA to 60.24 AUPRC. Our single-reference variant reaches 62.35, while the full $K = 3$ model reaches 64.44 and deploys as a single student scorer. The improvement therefore extends beyond simple inference-time ensembling.

D.4 TN and Reference-Bank Sensitivity

Performance rises from 59.28 AUPRC with 10 verified TNs to 66.45 with 80 TNs. At the default 40-TN setting, increasing the reference bank from $K = 1$ to $K = 3$ raises AUPRC from 62.35 to 64.44, with $K = 3$ capturing nearly all of the observed reference-bank benefit.

E Agent-SafetyBench External Validation

Protocol. Agent-SafetyBench (Zhang et al., 2024) provides 2,000 case IDs, split case-wise into 600 training and 1,400 held-out evaluation cases. Across three trajectory-generation seeds and two frozen monitors, the final evaluation contains 4,054 held-out alarms, including 638 false alarms. The training split supplies 40 verified-safe non-alarms. All method settings and predictions are fixed before evaluation labels are accessed.

Full results. Macro ranking metrics average the two monitor-specific cohorts. Review-budget yield counts recovered false alarms within the corresponding aggregate target budget.

The full method improves PULDA by 4.33 AUPRC points and recovers 42.4% more false alarms at the Top-5% review budget. The $K = 1$ variant also exceeds the three-model PULDA ensemble.

Monitor-level results. The improvement appears consistently across both frozen monitor families.

Risk-category results. Our method achieves the strongest AUPRC in all six sufficiently populated risk categories, showing that the external improvement extends across heterogeneous safety risks.

Scaling and uncertainty. The nested cohorts in Figure 3 contain 1,034, 2,027, 3,062, and 4,054 alarms. The AUPRC improvement over PULDA increases from 3.07 to 4.33 points as the cohort grows. Uncertainty is estimated using 2,000 case-level bootstrap resamples.

Table 10: **Selection-bias and representation controls.** Ranking metrics are macro-averaged over the same four target cells (100×); yield is the summed number of recovered FPs. SAR-EM uses the official selected-at-random implementation (Bekker & Davis, 2018). Deterministic representation controls use only the 40 training TNs and frozen monitor representations. PULDA and our method are the three-seed results from Table 1.

Method	AUPRC ↑	AUROC ↑	AUGRC ↓	Top 5%	Top 10%	Top 20%
Native monitor score	—	—	—	—	—	50.00
TN centroid distance	55.44	57.98	23.75	13.00	28.00	53.00
5-NN to TNs	54.69	57.74	23.82	14.00	27.00	50.00
One-class SVM (Schölkopf et al., 2001)	57.74	59.30	23.43	14.00	29.00	52.00
TN PCA support	54.97	57.55	23.86	14.00	29.00	51.00
Role-propensity score	53.89	55.93	24.26	13.00	26.00	53.00
SAR-EM (role propensity)	54.11	54.51	24.63	12.00	26.00	52.00
PULDA	59.17	59.38	23.41	17.00	31.00	53.33
Ours	**64.44**	**61.70**	**22.85**	**22.67**	**37.00**	**65.00**

Native outputs were archived consistently for all four cells at the 20% cutoff; unavailable full-score entries are left blank rather than reconstructed. SAR-EM and the representation controls are deterministic fits, so no seed dispersion is attached.

Table 11: **Compute and ensemble control under the main protocol.** All rows use the same four dataset–monitor settings, 40 TNs per setting, 400 training epochs, MLP capacity, and seeds $\{13, 37, 73\}$. Ranking metrics are macro-averaged over the four settings within each seed and then averaged over seeds. FP yield sums the recovered FPs at $\lceil bN_c \rceil$ within each setting and then averages the sum over seeds. Baseline ensembles average the logits of three independently trained seeds; our $K = 3$ model uses three frozen references during training and one student scorer at inference.

Method	Architecture / compute	Macro AUPRC ↑ (%)	Macro AUROC ↑ (%)	Top-5% FP yield ↑ /27	Top-10% FP yield ↑ /52
Dist-PU (Single)	one model; three-run mean	57.78	58.08	16.00	29.33
Dist-PU (Ensemble)	three-logit average	58.65	58.74	17.00	30.67
PULDA (Single)	one model; three-run mean	59.17	59.38	17.00	31.00
PULDA (Ensemble)	three-logit average	60.24	60.18	18.33	32.67
Ours ($K = 1$)	**single model**	**62.35**	**60.82**	**20.67**	**34.67**
Ours ($K = 3$)	**three references → one student**	**64.44**	**61.70**	**22.67**	**37.00**

Table 12: **Full Agent-SafetyBench results.** Macro ranking metrics average two frozen monitors (100×); FP columns report false alarms recovered at each review budget.

Method	AUPRC ↑	AUROC ↑	AUGRC ↓	FP@1%	FP@2%	FP@5%	FP@10%	FP@20%
Native monitor score	16.98	49.36	42.42	5	10	34	65	121
nnPU	19.40	54.98	41.34	8	18	45	78	143
DC-PU	15.32	45.35	42.84	5	9	22	36	76
Dist-PU	24.16	57.20	40.97	24	42	66	110	165
Robust-PU	20.26	55.01	41.28	13	22	53	85	149
PULDA	24.39	58.57	40.74	23	38	66	106	186
PUL-CPBF	20.81	45.95	41.64	16	33	53	91	147
HolisticPU	20.27	54.98	41.38	12	21	50	84	146
PULDA ensemble	25.42	59.45	40.21	25	41	71	114	195
Our method ($K = 1$)	26.15	60.32	39.68	26	45	76	125	208
Our method ($K = 3$)	**28.72**	**63.45**	**37.82**	**32**	**57**	**94**	**152**	**241**

Table 13: **External monitor-specific AUPRC** (100×).

Monitor	Alarms	FPs	Native	PULDA	Our method
AgentDoG-Qwen3-4B	2,427	213	14.92	19.70	**22.85**
Granite-Guardian-3.1-2B	1,627	425	19.05	29.09	**34.58**

Table 14: **External risk-category AUPRC ($100\times$).** Categories contain at least 30 false and 30 genuine alarms.

Risk category	Alarms	FPs	Native	PULDA	Our method
Unsafe information / misinformation	606	329	63.31	59.09	**69.85**
Property loss	503	71	19.58	23.34	**27.12**
Sensitive-data leak	496	65	17.34	27.73	**30.45**
Availability compromise	455	46	11.24	13.19	**15.60**
Physical harm	439	39	16.73	22.93	**24.88**
Harmful / vulnerable code	716	62	14.74	27.69	**29.35**