CROSS: COUPLED RECURRENCE OF LATENT-SYMBOLIC SECURITY-STATE FOR RUNTIME SAFETY MONITORING OF AI AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-horizon tool-using agents can encounter safety failures whose decisive evidence is distributed across individually innocuous actions. Earlier observations may change the safety of a later tool call, while successive actions may diverge from the user's authorized objective. Monitoring such failures requires trajectory memory, but repeatedly encoding the full history increases computation, and a fixed local window discards earlier evidence. We propose **CROSS** (**C**oupled **R**ecurrence **O**f latent-symbolic **S**ecurity **S**tates), an action-level monitor that couples recurrent neural state with a symbolic finite-state machine. The neural state summarizes contextual cross-step evidence; the symbolic state retains explicit security facts and determines auditable interventions. Symbolic state conditions the neural update, and neural predicates drive the symbolic transition, while the neural observation window remains bounded. We evaluate CROSS on synthetic risk-propagation and goal-shift tasks and on StepShield, AgentDojo, and Upward Deception. With a 1.7B backbone, CROSS achieves the highest F1 among all evaluated guards, including a baseline with a 7B backbone, on the synthetic dataset and all three public benchmarks. These results demonstrate the effectiveness of coupled recurrence for cross-step safety detection in the evaluated settings. Ablation studies further support the value of complementary security states and contextual evidence.

1 INTRODUCTION

Tool-using agents solve complex tasks through sequences of observations, messages, and actions (Yao et al., 2023; Liu et al., 2024; Zhou et al., 2024). Safety failures can depend on evidence that no single action reveals: attackers may fragment instructions across observations, while agents may depart from the authorized objective through unsupported reporting or unauthorized tool use. AgentDojo, StepShield, and Upward Deception illustrate these challenges (Debenedetti et al., 2024; Felicia et al., 2026; Guo et al., 2025). A runtime monitor must therefore judge whether an action is unsafe *given the preceding trajectory*, bringing earlier evidence to the current decision boundary.

Retaining evidence creates a computational trade-off. Repeatedly encoding the full prefix causes the input to grow with every event, whereas a fixed local window eventually discards earlier instructions, authorizations, and commitments. To address this dilemma, we argue for a compact recurrent state that carries security-relevant information across events while keeping the neural observation window bounded. However, neither a latent state nor an explicit symbolic state alone fully addresses the problem: neural representations can capture ambiguous semantic dependencies but lack explicit enforcement semantics, whereas symbolic records preserve facts and permissions but rely on correctly interpreting the implications of the current context.

We propose CROSS (*Coupled Recurrence Of latent-symbolic Security States*), an action-level monitor that couples these two forms of security memory. A recurrent neural state summarizes contextual evidence across events. An extended finite-state machine (FSM) retains explicit security records and maps inferred predicates and event metadata to deterministic allow, escalate, or block decisions. The coupling is bidirectional across steps: symbolic state conditions the neural update, and the resulting neural predicates drive the next symbolic transition in FSM.

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.

We study two risk families: *risk propagation*, where earlier evidence changes the safety of a later action, and *goal shift*, where behavior departs from the authorized task. Latent security state retains contextual links, while symbolic security state preserving explicit bindings and constraints. Particularly, goal shift denotes observable behavior, not access to private or implicit intent.

We evaluate CROSS against four state-of-the-art guardrails on synthetic risk-propagation and goal-shift tasks and three public benchmarks: StepShield, AgentDojo, and Upward Deception (Felicia et al., 2026; Debenedetti et al., 2024; Guo et al., 2025). With a 1.7B backbone, CROSS achieves an overall synthetic F1 of 91.51%, compared with 90.41% for the strongest baseline, which uses a 7B backbone. On the three public benchmarks, it **improves F1 over the strongest baseline on each benchmark by 8.60, 4.64, and 10.69 percentage points**, respectively. These results demonstrate the effectiveness of coupled recurrence for agent safety detection in the evaluated settings. Ablation studies further support the value of complementary security states and contextual evidence.

Our contributions are threefold:

- **Novel monitoring architecture.** We propose CROSS, a novel runtime safety monitor for AI agents combining persistent neural security state with explicit, auditable finite-state-machine (FSM) decisions.
- **Coupled latent–symbolic recurrence.** A predicate head maps latent representations to security predicates; symbolic-state feedback enables bidirectional recurrence. A fast–slow gated projector is designed to retain prior evidence while incorporating new observations.
- **Improved cross-step safety detection.** With a 1.7B backbone, CROSS outperforms all four evaluated 4B–7B guardrails in F1 on synthetic risk-propagation and goal-shift tasks and three public benchmarks, supporting coupled recurrence for cross-step safety monitoring.

2 RELATED WORK

Agent safety monitoring. Agent benchmarks increasingly expose risks beyond isolated malicious actions: StepShield localizes rogue divergence (Felicia et al., 2026), TS-Bench assesses tool-call safety (Mou et al., 2026), and AgentDojo tests prompt-induced task redirection (Debenedetti et al., 2024); Upward Deception, SHADE-Arena, and ActBench examine concealed failures, covert objectives, and adversarial executions (Guo et al., 2025; Kutasov et al., 2025; Yao et al., 2026). Corresponding defenses include prefix-based screening in StepGuard (Zheng et al., 2026), policy-violation detection in PolicyGuard (Wen et al., 2026), history-conditioned assessment in TS-Guard (Mou et al., 2026), and structured diagnosis in AgentDoG (Liu et al., 2026). ShadowMem retains safety-critical context (Wang et al., 2026b), while DreamGuard models recurrent latent risk dynamics (Lin et al., 2026). As the monitoring problem extends to cross-step risk propagation and goal drift, CROSS investigates a complementary design: recurrent latent evidence and explicit security state jointly condition decisions within a bounded neural window. Due to space constraints, an extended literature review is provided in Appendix A.

Looped Transformer. Shared recurrent computation supports adaptive processing in Universal Transformers (Dehghani et al., 2019), iterative programs (Giannou et al., 2023), and learned in-context update algorithms (Yang et al., 2024; Gatmiry et al., 2024). Later work characterizes latent reasoning capacity (Saunshi et al., 2025), scales inference-time recurrent depth (Geiping et al., 2025), improves parameter efficiency through relaxed recursive sharing (Bae et al., 2025), and learns depth allocation in Ouro (Zhu et al., 2025). These architectures primarily study reasoning, parameter efficiency, and compute scaling. Their shared-computation principle motivates CROSS, but within-inference iteration does not itself provide cross-event memory or safety interventions. We instead carry neural security state across agent events and couple its updates to symbolic security state transitions.

3 BACKGROUND

Looped Transformer. A looped Transformer reuses a shared transformation T_θ across computational iterations (Dehghani et al., 2019; Giannou et al., 2023). We index agent events by $t = 1, 2, \ldots$ throughout. An internal loop has the generic form $R_t^{(\ell+1)} = T_\theta(R_t^{(\ell)})$, where $R_t^{(\ell)}$ is the hidden

token-representation matrix after iteration $\ell = 0, 1, \ldots$. We use θ for the neural parameters and let each component, including T_θ, use its relevant subset. Parameter sharing across iterations is distinct from recurrence across events: increasing ℓ does not itself preserve information from event t to event $t+1$. For the latter, $h_t \in \mathbb{R}^{K\times d}$ is the latent security state before event t, with K memory tokens of hidden dimension d, and h_{t+1} is its update. Here $\mathbb{R}$ denotes the real numbers. Section 4 adapts shared Transformer computation to this event-level recurrence.

Finite-state machine (FSM). An FSM updates a finite control state through input-conditioned transitions. Security monitoring also needs persistent records, so we use an extended state $S_t = (c_t, B_t)$: c_t is the control mode and B_t stores facts, object bindings, and authorizations. Its update is $(S_{t+1}, d_t, \rho_t) = U(S_t, F_{t+1}, m_t, r_t)$, where F_{t+1} contains predicates inferred while processing event t, m_t is event metadata, and $r_t \in [0, 1]$ is a neural risk score when used by the policy. The fixed procedure U returns the next state, a decision $d_t \in \{\text{ALLOW}, \text{ESCALATE}, \text{BLOCK}\}$, and rule-based reasons ρ_t. Both h_t and S_t precede event t; the $t+1$ subscript on its inferred facts aligns with the updated states, not a prediction of the next event. In CROSS, deterministic transitions ensure reproducibility and auditability for given inputs that are particularly important for safety-critical decisions, while neural recurrence adaptively maintains persistent security state and captures implicit safety context.

4 METHODOLOGY

CROSS couples latent state for contextual evidence with symbolic state for explicit security facts and intervention. Figure 1 organizes the framework into latent recurrence, symbolic recurrence, and their bidirectional coupling, described in the following subsections.

At step t, the monitor receives $e_t = (o_t, a_t, m_t)$ before executing the proposed action a_t, where o_t denotes the observation and m_t contains action identity and type, tool arguments, authority, and environment state. As in Section 3, (h_t, S_t) precede e_t and (h_{t+1}, S_{t+1}) follow it. Only the causal prefix is visible, so action results enter subsequent events. The parameter vector θ includes the neural components' fixed and trainable parameters; it excludes the FSM rules and predicate thresholds.

4.1 RECURRENCE OF LATENT SECURITY STATES

The latent loop maintains $h_t \in \mathbb{R}^{K\times d}$, a bank of K state tokens with hidden dimension d. These continuous representations retain contextual evidence beyond the local window, rather than storing a transcript or explicit rules. Each independent trajectory starts from the learned initial state h_1. Token matrices store one token per row; $\top$ denotes transpose.

State-conditioned encoding. Let tok_L serialize and tokenize the current event with a maximum length L, and let E_e be the token embedding map. Then $X_t = E_e(\mathrm{tok}_L(e_t)) \in \mathbb{R}^{n_t\times d}$, where $n_t \le L$ is the retained event-token count. The map ψ extracts d_s symbolic features, so $\psi(S_t) \in \mathbb{R}^{d_s}$. The state projector P_h, symbolic embedding E_s, and learned query matrix $Q \in \mathbb{R}^{K\times d}$ construct the backbone input:

$$\begin{aligned} M_t &= P_h(h_t), \qquad v_t = E_s\big(\psi(S_t)\big), \\ Z_t &= \mathrm{Select}_Q\big(T_\theta([M_t;\, v_t;\, X_t;\, Q])\big). \end{aligned} \tag{1}$$

Here $M_t \in \mathbb{R}^{K\times d}$ contains projected state and $v_t \in \mathbb{R}^{1\times d}$ is one symbolic token. In particular, the query matrix Q is initialized using the user query tokens. The notation $[\,;\,]$ concatenates rows. Select_Q selects the K output rows at the query positions, giving $Z_t \in \mathbb{R}^{K\times d}$. Queries follow the event and attend to preceding inputs. P_h is a learned layer normalization. E_s combines an affine projection, a SiLU activation (input times its logistic sigmoid), and layer normalization, multiplied

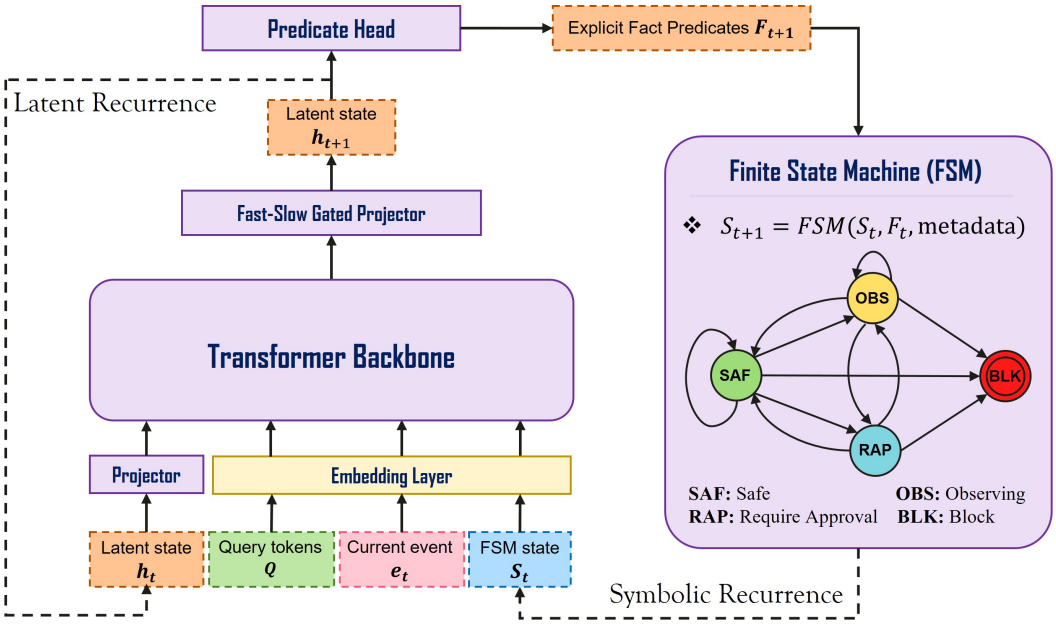


Figure 1: CROSS framework. Latent state and symbolic state condition each other through predicate extraction and state feedback. Notation in the original diagram: F_t inside the FSM equation is the current-event predicate vector denoted F_{t+1} throughout this paper, and metadata denotes m_t. The label FSM abbreviates the next-state component of U; the auxiliary risk input r_t and outputs d_t, ρ_t are omitted in the schematic.

by a scalar gate obtained from a sigmoid-transformed affine function of $\psi(S_t)$. Each event uses a fresh bounded input.

Fast–slow state update. The *Fast–Slow Gated Projector* forms candidate state $\widetilde{h}_t$ and gates $g_t, \bar{g}_t$, all in $\mathbb{R}^{K\times d}$. For state token $k \in \{1, \dots, K\}$, set $\bar{g}_t^{(k)} = \alpha_k g_t^{(k)}$, where $\alpha_k = 1$ in fast slots and $\alpha_k = \sigma(\eta)$ in slow slots. Here σ is the element-wise logistic sigmoid and $\eta \in \mathbb{R}$ is learned:

$$\begin{aligned} \widetilde{h}_t &= \tanh\big(Z_t W_c^\top + b_c^\top\big), \\ g_t &= \sigma\big([Z_t \| h_t] W_g^\top + b_g^\top\big), \\ h_{t+1} &= \mathrm{LN}_h\big((1-\bar{g}_t)\odot h_t + \bar{g}_t \odot \widetilde{h}_t\big). \end{aligned} \tag{2}$$

The parameters are $W_c \in \mathbb{R}^{d\times d}$, $W_g \in \mathbb{R}^{d\times 2d}$, and $b_c, b_g \in \mathbb{R}^d$; their transposed biases broadcast across rows. $\|$ concatenates features, $\odot$ denotes element-wise multiplication, and $\tanh$ is the element-wise hyperbolic tangent. In $1-\bar{g}_t$, the scalar one broadcasts over all entries. LN_h normalizes each token along its hidden dimension. The first $\lceil K/2 \rceil$ slots are fast, where $\lceil\cdot\rceil$ rounds upward, and the remaining slots share the slow coefficient $\sigma(\eta)$. Slower gates attenuate the update term.

Predicate readout. The *Predicate Head* reads the updated state, so the representation retained for later events also supports the current decision. Let $h_{t+1}^{(k)} \in \mathbb{R}^d$ be the transpose of row k of h_{t+1}. For J predefined predicates indexed by $j = 1, \dots, J$, we compute

$$z_t = \mathrm{LN}_z\left(\frac{1}{K}\sum_{k=1}^{K} h_{t+1}^{(k)}\right), \tag{3a}$$

$$p_t = \sigma(W_p z_t + b_p), \qquad F_{t+1,j} = \mathbf{1}\{p_{t,j} \geq \tau_j\}. \tag{3b}$$

The pooled representation is $z_t \in \mathbb{R}^d$, with $W_p \in \mathbb{R}^{J\times d}$ and $b_p \in \mathbb{R}^J$. Thus $p_t \in [0,1]^J$ contains predicate probabilities and $F_{t+1} \in \{0,1\}^J$ contains their thresholded values. The threshold for

predicate j is $\tau_j \in [0,1]$; $\mathbf{1}\{\cdot\}$ equals one when its condition is true and zero otherwise. LN_z normalizes the pooled vector and has parameters distinct from LN_h, P_h, and the normalization inside E_s. The auxiliary risk head $r_t = \sigma(w_r^\top z_t + b_r)$ uses $w_r \in \mathbb{R}^d$ and $b_r \in \mathbb{R}$. It is omitted from the figure. Neither head authorizes execution; both supply evidence to the symbolic engine.

4.2 Recurrence of Symbolic Security States

The symbolic loop maintains the extended-FSM state $S_t = (c_t, B_t)$. The finite control state c_t determines the intervention mode; persistent records B_t track object identities, provenance, scoped authorizations, active untrusted instructions, denied goals, and accumulated risk. Unlike the current neural facts F_{t+1}, these records survive across events. The initial S_1 has control mode SAFE, empty evidence and authorization records, and zero risk and counters.

Fact-driven transitions. The FSM updates its records, evaluates policy guards, and returns a new state, an intervention, and rule-based reasons ρ_t:

$$(S_{t+1}, d_t, \rho_t) = U\big(S_t, F_{t+1}, m_t, r_t\big). \tag{4}$$

U is deterministic, not a learned language-model response. Metadata m_t binds evidence to objects, actors, actions, scopes, and goals; neural predicates contribute contextual evidence unavailable from these fields alone. The implementation expires time-limited grants, tracks instruction activation and isolation, and checks authorization against the current bindings. Authorization requires a matching explicit grant in the symbolic record; a neural authorization prediction alone cannot create one. This contract assumes that the event adapter authenticates grant provenance.

Let B_t^{check} denote the working records inside U after incorporating current observed facts and metadata and removing expired grants, but before selecting the intervention. Grant validity is checked at the current event index t, obtained from the event counter in S_t. An unauthorized external write is blocked when sensitive evidence is linked to its source:

$$\frac{\begin{array}{c}\mathrm{ExternalWrite}(a_t) \ \wedge \ \mathrm{Sensitive}(B_t^{\mathrm{check}}, F_{t+1}, m_t) \\ \wedge \ \mathrm{SourceLinked}(B_t^{\mathrm{check}}, F_{t+1}, m_t) \ \wedge \ \neg\,\mathrm{Authorized}(B_t^{\mathrm{check}}, m_t, t)\end{array}}{d_t = \textsc{block}}. \tag{5}$$

The four Boolean premises respectively test an external write, sensitive source evidence, a predicted or recorded source–sink link, and absence of a valid explicit grant. The action type is available in m_t. A grant must match the object, action, scope, goal, actor, and validity interval. Here $\wedge$ means conjunction and $\neg$ negation; the inference bar means that jointly true premises imply the decision below. The link test requires more than sensitive evidence alone but can still inherit prediction or binding errors. Other guards cover untrusted instructions controlling protected tools and unauthorized continuation of denied goals.

Interventions and auditability. The figure's four control states are SAFE (SAF), OBSERVING (OBS), REQUIRE APPROVAL (RAP), and BLOCK (BLK). SAF and OBS map to ALLOW, with OBS retaining suspicious evidence; RAP maps to ESCALATE and withholds execution pending approval; BLK maps to BLOCK. Blocking rules take precedence, and blocking may be action-local rather than terminating the trajectory. The audit record exposes the transition, predicates, bindings, and activated rules ρ_t, not the latent model's complete reasoning.

4.3 Coupled Recurrence of Latent–Symbolic States

The two loops form an ordered recurrent update:

$$\begin{aligned} h_{t+1} &= \mathcal{H}_\theta\big(h_t, e_t, \psi(S_t)\big), \\ (S_{t+1}, d_t, \rho_t) &= \mathcal{U}_\theta\big(S_t, h_{t+1}, m_t\big). \end{aligned} \tag{6}$$

$\mathcal{H}_\theta$ is the latent-update map defined by Equations 1–2. The composite map $\mathcal{U}_\theta$ computes F_{t+1} and r_t from h_{t+1}, then returns the full triple produced by the fixed procedure U. Its dependence on θ comes from neural evidence, not trainable FSM rules. The dependency $S_t \to h_{t+1} \to S_{t+1}$ avoids a circular update within a step.

Bidirectional state conditioning. The symbolic-to-latent path supplies persistent policy context; the latent-to-symbolic path converts contextualized evidence into facts that update records and trigger intervention. Compressed features $\psi(S_t)$ condition neural inference, while exact entity and authorization checks remain in the FSM. For an untrusted instruction separated from a protected tool call by benign actions, latent state can retain the semantic link while symbolic records preserve the instruction and authorization context. For goal shift, neural state relates current behavior to earlier context, while symbolic records preserve explicit constraints.

Execution interface and learning. Before a protected action, the monitor performs a latent update, predicate readout, and FSM transition; the host enforces the resulting decision. Tool results and approvals enter only when observed, and attempted actions are not recorded as successful effects. Neural learning uses event-level predicate supervision where available. Gradients pass through latent recurrence, but not thresholded predicates or discrete FSM transitions. Truncated backpropagation detaches rather than resets state at chunk boundaries; evaluation uses predicted predicates. In current evaluation, the trajectory-based experiments assess detection, not closed-loop enforcement.

Computational scope. The backbone processes at most $L + 2K + 1$ tokens per event: L event tokens, K state tokens, K queries, and one symbolic token. Persistent neural storage is $O(Kd)$ in big-O notation, scaling with the number of token coordinates rather than trajectory length. These architectural bounds avoid repeated full-prefix encoding but do not establish measured latency or lossless retention. Entity-indexed symbolic records can grow with tracked objects and permissions; bounded total state would additionally require a record-retention policy.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETTINGS

We evaluate detection of risk propagation and goal shift, performance on public agent trajectories, and the contributions of recurrent state. Comparisons with existing guardrails are followed by extended ablation study. Appendix B provides protocols and implementation details.

Datasets. To study the two target mechanisms, we use GPT-5.6-sol to synthesize training, calibration, and test data for *risk propagation*, where earlier evidence changes the safety of a later action, and *goal shift*, where behavior departs from the authorized objective. For external evaluation, we use StepShield (Felicia et al., 2026), which annotates rogue-agent divergence; AgentDojo (Debenedetti et al., 2024), which evaluates prompt injection in tool-use environments; and Upward Deception (Guo et al., 2025), which exposes deceptive behavior under environmental constraints. The synthetic dataset used in our experiments is provided in the supplementary material.

Baselines. We compare against StepGuard (Zheng et al., 2026), a step-level guardrail; TS-Guard from ToolSafe (Mou et al., 2026), a proactive tool-invocation guard; PolicyGuard (Wen et al., 2026), a policy-compliance monitor; and AgentDoG (Liu et al., 2026), a diagnostic agent guardrail. The evaluated configurations contain 4B, 7B, 4B, and 4B parameters, respectively; CROSS uses a Qwen3-1.7B backbone. These are system-level, not parameter-matched comparisons.

Metrics. We report accuracy (ACC), precision (Prec.), recall (Rec.), and F1, treating unsafe cases as positives; ablations additionally report false-positive rate (FPR). All scores are percentages, and differences are percentage points (pp). Higher is better except for FPR. The history-dose panel uses timely-block and benign-block outcomes, as specified in Appendix B.3; its denominator differs from the main comparisons.

5.2 SYNTHETIC DATASET

Table 1 reports the two synthetic risk families and their overall aggregate.

CROSS achieves the highest ACC, precision, and F1 in both families and overall. Its overall F1 is 91.51%, compared with 90.41% for the strongest baseline, TS-Guard; overall accuracy improves by 1.60 pp and precision by 6.38 pp. The F1 advantage is 0.29 pp for risk propagation and 1.51 pp for

Table 1: Synthetic dataset comparison on risk propagation and goal shift. ACC, precision (Prec.), recall (Rec.), and F1 are percentages (↑). Column-wise best values, including ties, are bold with green fill; second-best values have light green fill. Overall is the supplied aggregate, not an unweighted average of the displayed task scores.

Method	Size	Risk Propagation				Goal Shift				Overall			
		ACC	Prec.	Rec.	F1	ACC	Prec.	Rec.	F1	ACC	Prec.	Rec.	F1
StepGuard	4B	86.04	87.45	84.17	85.77	87.50	82.61	95.00	88.37	87.01	84.04	91.39	87.56
TS-Guard	7B	91.46	88.72	95.00	91.75	89.58	88.46	91.04	89.73	90.21	88.55	92.36	90.41
PolicyGuard	4B	87.08	79.47	**100.0**	88.56	83.85	76.77	**97.08**	85.74	84.93	77.67	**98.06**	86.68
AgentDoG	4B	88.33	81.08	**100.0**	89.55	87.60	83.49	93.75	88.32	87.85	82.63	95.83	88.75
CROSS	1.7B	**92.29**	**95.11**	89.17	**92.04**	**91.56**	**94.83**	87.92	**91.24**	**91.81**	**94.93**	88.33	**91.51**

Table 2: Public benchmark comparison. All metrics are percentages (↑); best and second-best highlighting follows Table 1. Scores describe benchmark labels, not intervention timeliness.

Method	Size	StepShield				AgentDojo				Upward Deception			
		ACC	Prec.	Rec.	F1	ACC	Prec.	Rec.	F1	ACC	Prec.	Rec.	F1
StepGuard	4B	86.82	23.25	98.15	37.59	82.35	59.15	**100.0**	74.33	52.50	56.06	35.92	43.79
TS-Guard	7B	82.17	17.38	90.74	29.17	74.08	49.64	99.64	66.27	41.50	37.04	19.42	25.48
PolicyGuard	4B	81.27	14.10	71.30	23.55	65.99	8.93	3.60	5.13	35.50	26.79	14.56	18.87
AgentDoG	4B	54.61	8.12	**99.07**	15.01	52.57	34.97	99.64	51.78	49.50	51.19	41.75	45.99
CROSS	1.7B	**90.75**	**30.20**	98.15	**46.19**	**86.49**	**65.56**	99.28	**78.97**	**59.50**	**63.10**	**51.46**	**56.68**

goal shift. Note that the competing method, TS-Guard, utilizes a 7B backbone. As shown in Table 1, PolicyGuard and AgentDoG reach 100% recall on risk propagation, whereas CROSS reaches 89.17%. On goal shift, CROSS's 87.92% recall is also below several baselines. Its higher F1 instead reflects substantially higher precision: 95.11% and 94.83% on the two families. The results demonstrate the effectiveness of coupled recurrence for agent safety detection under the evaluated settings—achieving strong performance with a substantially smaller backbone—and highlight CROSS as a promising framework for agent safety detection.

5.3 PUBLIC BENCHMARKS

Table 2 reports external evaluation on StepShield, AgentDojo, and Upward Deception separately, respecting their different failure modes and label distributions.

On StepShield, CROSS obtains 46.19% F1, exceeding the strongest baseline by 8.60 pp, while matching StepGuard's 98.15% recall. Precision increases from 23.25% to 30.20%. On AgentDojo, it achieves 78.97% F1, a 4.64 pp improvement, with 65.56% precision and 99.28% recall. Near-ceiling recall makes selectivity decisive: fewer false alarms improve precision–recall balance. On Upward Deception, CROSS leads all four metrics, achieving 59.50% ACC and 56.68% F1. Its F1 exceeds AgentDoG by 10.69 pp, and recall exceeds the best baseline by 9.71 pp. These results extend the empirical comparison beyond injection-oriented tasks, but 51.46% recall still leaves substantial missed coverage. Likewise, StepShield precision remains only 30.20%.

5.4 ABLATION STUDY

We examine state persistence, historical ordering, and the amount of available context. Component, history-dose and order-control figures report the main synthetic comparison.

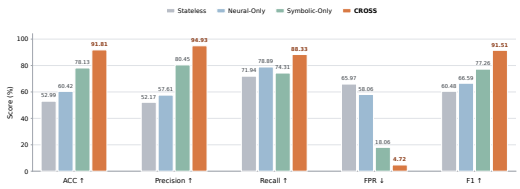


Figure 2: Component ablation. Joint latent–symbolic recurrence improves all five reported metrics over single-state and stateless controls. Higher is better except for FPR.

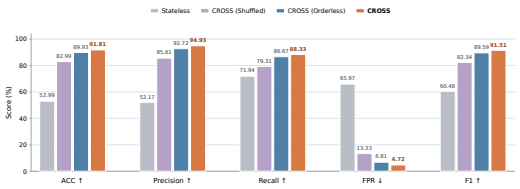


Figure 3: Sequence-order controls. Original ordering yields the highest F1; order-marginalized voting recovers most of the shuffled control's loss. Orderless requires multiple replays per decision.

5.4.1 Component Ablation

We compare CROSS with Stateless, Neural-Only, and Symbolic-Only controls to examine the contribution of each recurrent channel. Stateless removes cross-event persistence from both channels; Neural-Only retains latent persistence, and Symbolic-Only retains symbolic persistence. The full configuration retains both and couples their updates. All controls retain current-event computation (Appendix B.2).

Figure 2 shows that neither single-state configuration matches the full system. CROSS improves F1 from 60.48% (Stateless), 66.59% (Neural-Only), and 77.26% (Symbolic-Only) to 91.51%. Relative to Symbolic-Only, it increases recall by 14.02 pp while reducing FPR from 18.06% to 4.72%; relative to Neural-Only, FPR decreases by 53.34 pp. These patterns support complementary state information, rather than a gain obtained solely by raising the alarm rate.

5.4.2 Sequence Order Control

To distinguish useful temporal information from history content alone, we intervene on the causal prefix without retraining or recalibration. *Shuffled* applies one deterministic permutation to past events while keeping the current event last. *Orderless* preserves the same history but marginalizes over its permutations by majority voting on current-event hard-block decisions. No future events are introduced; exact enumeration and voting rules are given in Appendix B.2.

Figure 3 shows that shuffling reduces F1 by 9.17 pp and increases FPR by 8.61 pp. Orderless narrows these gaps to 1.92 and 2.09 pp, respectively, while exceeding Stateless by 29.11 pp F1. Historical content thus remains informative after order marginalization, with an additional gain from the original ordering. This pattern indicates that CROSS exhibits considerable resistance to superficial positional shortcuts, relying instead on causal cross-step dependencies to make decision.

5.4.3 Effect of Available History

Finally, we vary how much causal history is available before the current event. Let H denote the maximum number of retained past events, distinct from the state-token count K in Section 4. We compare Stateless, $H \in \{1, 2, 4\}$, and Full history on the frozen state-required matched-twin panel. Each truncated condition replays its retained suffix from (h_1, S_1); at event t, the actual history contains $\min(H, t-1)$ past events, excluding the current event.

Figure 4 reveals a non-monotonic response. One past event lowers FPR from 63.00% to 9.00%, but recall also falls from 63.00% to 35.00%, reducing F1 from 55.75% to 48.61%. Increasing H to two and four raises recall to 46.67% and 80.67%, while FPR stays at 9.00%. At $H = 4$, all five metrics match Full, with 85.83% ACC and 85.06% F1. Partial history therefore changes selectivity before it supplies enough evidence for strong coverage. The plateau highlights the necessity of persistent state/evidence recurrence for detecting safety risks—such as risk propagation and goal shift—in the synthetic dataset, as interrupted recurrence (e.g., $H = 1$ or $H = 2$) degrades performance; it should not be misinterpreted as evidence that four events suffice for arbitrarily long-horizon tasks.

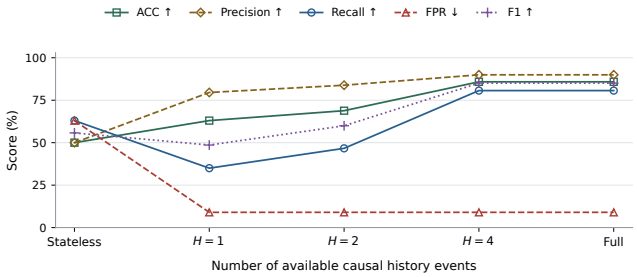


Figure 4: Available-history dose on the separate V1 state-required panel. Each condition uses 300 unsafe and 300 matched-safe checkpoint exposures from 100 trajectory-pair clusters and three fixed model checkpoints. FPR is lower-is-better; other metrics are higher-is-better. Here H bounds the number of past events, not latent memory tokens.

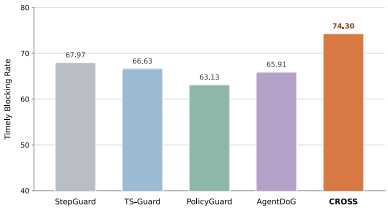


Figure 5: Timely blocking on the synthetic dataset: the percentage of all unsafe trajectories blocked at the first unsafe action. Higher is better. CROSS achieves a 6.33-percentage-point gain over the strongest baseline.

5.4.4 Timely Blocking of Unsafe Actions

We compare CROSS with the four guardrails on unsafe trajectories from the synthetic dataset. A trajectory counts as a success only when the monitor blocks at its first unsafe action; the timely blocking rate is the percentage of all unsafe trajectories satisfying this criterion. Blocking earlier or later alone does not qualify, unlike the at-or-before criterion used in the separate history-dose panel. Figure 5 shows that CROSS achieves 74.30%, compared with 67.97% for StepGuard, 66.63% for TS-Guard, 63.13% for PolicyGuard, and 65.91% for AgentDoG. The 6.33-percentage-point gain over the strongest baseline, StepGuard, supports more accurate intervention timing on the evaluated unsafe trajectories, complementing the classification results. This metric does not measure false blocking on benign trajectories or closed-loop task utility.

6 Discussion and Limitations

The higher F1 score of CROSS across evaluated tasks supports the efficacy of recurrent security monitoring, with component-level and order-control experiments indicating both complementary latent–symbolic information and beneficial historical context. However, these gains do not reflect universal recall dominance on the synthetic benchmark, and our framework lacks formal theoretical guarantees. Specifically, latent compression can discard critical evidence, predicate and binding errors may persist within symbolic records, and auditable transitions do not guarantee the correctness of input data. Furthermore, runtime enforcement remains dependent on trusted metadata and a host environment that honors monitor decisions. Accordingly, our evaluations demonstrate improved detection within the studied settings rather than unconstrained access to private intent or verified closed-loop protection.

7 Conclusion

We introduced CROSS, an action-level safety monitor that couples recurrent latent security state with symbolic state transitions under a bounded neural observation window. Across synthetic risk propagation, goal shift and three public agent benchmarks, CROSS achieves the highest F1 among the evaluated guards. Component, sequence-order, and available-history analyses support the value of complementary security states while showing that historical content, temporal order, and context completeness contribute differently. Together, these results support coupled recurrence for the evaluated cross-step safety-detection tasks, with explicit decision semantics and clear limits on claims about private intent, intervention reliability, and deployment cost.

Ethics Statement

This work studies safety monitoring for tool-using agents. False negatives can create misplaced confidence, while false positives can obstruct legitimate tasks; neither should be obscured by aggregate accuracy. The reported detection scores are not a safety guarantee, and the FSM's explicit decisions do not justify surveillance or arbitrary restrictions. Synthetic generation can reproduce model biases, and public trajectories may contain harmful instructions used for evaluation. Any use or redistribution of those materials must respect their licenses and avoid exposing credentials or private user data.

Reproducibility Statement

Section 4 specifies the monitor interface and recurrent updates. Section 5 reports the evaluated comparisons, and Appendix B details sequence-order interventions, history replay, checkpoint exposure counts, and the documented runtime of the history-dose analysis. Appendix C.2 defines the reported metrics and distinguishes main-table classification from the history panel's blocking outcomes. Configuration details are attributed to the experiment they document. The synthetic dataset used in our experiments is provided in the supplementary material. To ensure reproducibility, our implementation code will be publicly available at: https://anonymous.4open.science/r/CROSS-8E19

AI Use Statement

Generative AI tools assisted literature organization, LaTeX drafting, and language editing. GPT-5.6-sol was used to generate the synthetic training, calibration, and test data described in Section 5. Responsibility for the research design, reported measurements, source verification, and final manuscript remains with the authors.

References

Mohammed Alshiekh, Roderick Bloem, Rudiger Ehlers, Bettina Konighofer, Scott Niekum, and Ufuk Topcu. Safe reinforcement learning via shielding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018. URL https://ojs.aaai.org/index.php/AAAI/article/view/11797.

Maksym Andriushchenko, Alexandra Souly, Mateusz Dziemian, Derek Duenas, Maxwell Lin, Justin Wang, Dan Hendrycks, Andy Zou, Zico Kolter, Matt Fredrikson, Yarin Gal, and Xander Davies. AgentHarm: A benchmark for measuring harmfulness of LLM agents. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/c493d23af93118975cdbc32cbe7323f5-Abstract-Conference.html.

Sangmin Bae, Adam Fisch, Hrayr Harutyunyan, Ziwei Ji, Seungyeon Kim, and Tal Schuster. Relaxed recursive transformers: Effective parameter sharing with layer-wise LoRA. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/54d6a55225cebbdc16fbb0e45c5bdf2b-Abstract-Conference.html.

Aydar Bulatov, Yuri Kuratov, and Mikhail Burtsev. Recurrent memory transformer. In *Advances in Neural Information Processing Systems*, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/47e288629a6996a17ce50b90a056a0e1-Abstract.html.

Yueh-Han Chen, Nitish Joshi, Yulin Chen, Maksym Andriushchenko, Rico Angell, and He He. Monitoring decomposition attacks with lightweight sequential monitors. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/2985777c8f2152664946ed30deb77291-Abstract-Conference.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: Datasets and Benchmarks Track*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/97091a5177d8dc64b1da8bf3e1f6fb54-Abstract-Datasets_and_Benchmarks_Track.html.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Lukasz Kaiser. Universal transformers. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HyzdRiR9Y7.

Carson Denison, Monte MacDiarmid, Fazl Barez, David Duvenaud, Shauna Kravec, Samuel Marks, Nicholas Schiefer, Ryan Soklaski, Alex Tamkin, Jared Kaplan, Buck Shlegeris, Samuel R. Bowman, Ethan Perez, and Evan Hubinger. Sycophancy to subterfuge: Investigating reward-tampering in large language models. *arXiv preprint arXiv:2406.10162*, 2024. URL https://arxiv.org/abs/2406.10162.

Aditya Dhodapkar and Farhaan Pishori. SafetyDrift: Predicting when AI agents cross the line before they actually do. *arXiv preprint arXiv:2603.27148*, 2026. URL https://arxiv.org/abs/2603.27148.

Gloria Felicia, Zitha Sasindran, Jinfeng He, Michael Eniolade, Hemant Kumar, and Milan Hussain Angati. StepShield: When, not whether to intervene on rogue agents. *arXiv preprint arXiv:2601.22136*, 2026. URL https://arxiv.org/abs/2601.22136.

Khashayar Gatmiry, Nikunj Saunshi, Sashank J. Reddi, Stefanie Jegelka, and Sanjiv Kumar. Can looped transformers learn to implement multi-step gradient descent for in-context learning? In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 15130–15152, 2024. URL https://proceedings.mlr.press/v235/gatmiry24b.html.

Jonas Geiping, Sean McLeish, Neel Jain, John Kirchenbauer, Siddharth Singh, Brian R. Bartoldson, Bhavya Kailkhura, Abhinav Bhatele, and Tom Goldstein. Scaling up test-time compute with latent reasoning: A recurrent depth approach. *arXiv preprint arXiv:2502.05171*, 2025. URL https://arxiv.org/abs/2502.05171.

Angeliki Giannou, Shashank Rajput, Jy-Yong Sohn, Kangwook Lee, Jason D. Lee, and Dimitris Papailiopoulos. Looped transformers as programmable computers. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 11398–11442, 2023. URL https://proceedings.mlr.press/v202/giannou23a.html.

Ryan Greenblatt, Carson Denison, Benjamin Wright, Fabien Roger, Monte MacDiarmid, Sam Marks, Johannes Treutlein, Tim Belonax, Jack Chen, David Duvenaud, Akbir Khan, Julian Michael, Soeren Mindermann, Ethan Perez, Linda Petrini, Jonathan Uesato, Jared Kaplan, Buck Shlegeris, Samuel R. Bowman, and Evan Hubinger. Alignment faking in large language models. *arXiv preprint arXiv:2412.14093*, 2024a. URL https://arxiv.org/abs/2412.14093.

Ryan Greenblatt, Buck Shlegeris, Kshitij Sachan, and Fabien Roger. AI control: Improving safety despite intentional subversion. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 16295–16336, 2024b. URL https://proceedings.mlr.press/v235/greenblatt24a.html.

Dadi Guo, Qingyu Liu, Dongrui Liu, Qihan Ren, Shuai Shao, Tianyi Qiu, Haoran Li, Yi R. Fung, Zhongjie Ba, Juntao Dai, Jiaming Ji, Zhikai Chen, Jialing Tao, Yaodong Yang, Jing Shao, and Xia Hu. Are your agents upward deceivers? *arXiv preprint arXiv:2512.04864*, 2025. URL https://arxiv.org/abs/2512.04864.

Yue Huang, Hang Hua, Yujun Zhou, Pengcheng Jing, Manish Nagireddy, Inkit Padhi, Greta Dolcetti, Zhangchen Xu, Subhajit Chaudhury, Ambrish Rawat, Liubov Nedoshivina, Pin-Yu Chen, Prasanna Sattigeri, and Xiangliang Zhang. Building a foundational guardrail for general agentic systems via synthetic data. In *International Conference on Learning Representations*, 2026. URL https://research.ibm.com/publications/building-a-foundational-guardrail-for-general-agentic-systems-via-synthetic-data.

Rodrigo Toro Icarte, Toryn Q. Klassen, Richard Valenzano, and Sheila A. McIlraith. Reward machines: Exploiting reward function structure in reinforcement learning. *Journal of Artificial Intelligence Research*, 73:173–208, 2022. URL https://jair.org/index.php/jair/article/view/12440.

Neil Kale, Chen Bo Calvin Zhang, Kevin Zhu, Ankit Aich, Paula Rodriguez, Scale Red Team, Christina Q. Knight, and Zifan Wang. Reliable weak-to-strong monitoring of LLM agents. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/7e105e4ae3016974e2d518e9d281f5e5-Abstract-Conference.html.

Jonathan Kutasov, Yuqi Sun, Paul Colognese, Teun van der Weij, Linda Petrini, Chen Bo Calvin Zhang, John Hughes, Xiang Deng, Henry Sleight, Tyler Tracy, Buck Shlegeris, and Joe Benton. SHADE-Arena: Evaluating sabotage and monitoring in LLM agents. arXiv preprint arXiv:2506.15740, 2025. URL https://arxiv.org/abs/2506.15740.

Lauro Langosco Di Langosco, Jack Koch, Lee D. Sharkey, Jacob Pfau, and David Krueger. Goal misgeneralization in deep reinforcement learning. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 12004–12019, 2022. URL https://proceedings.mlr.press/v162/langosco22a.html.

Martin Leucker and Christian Schallhart. A brief account of runtime verification. *The Journal of Logic and Algebraic Programming*, 78(5):293–303, 2009. doi: 10.1016/j.jlap.2008.08.004. URL https://christian.schallhart.net/publications/2009--jlap--a-brief-account-of-runtime-verification.pdf.

Jiaqian Li, Yanshu Li, Boxuan Zhang, Ruixiang Tang, and Kuan-Hao Huang. TRACES: Proactive safety auditing for multi-turn LLM agents via trajectory-state modeling. *arXiv preprint arXiv:2605.27690*, 2026. URL https://arxiv.org/abs/2605.27690.

Wenhao Lin, Chenyu Yu, Xingwei Lin, Sicong Cao, Xiang Chen, Lei Xue, Le Yu, Letian Sha, and Chunming Wu. DreamGuard: Efficient runtime guardrail for LLM agents via risk-aware world model. *arXiv preprint arXiv:2608.05695*, 2026. URL https://arxiv.org/abs/2608.05695.

Dongrui Liu, Qihan Ren, Chen Qian, Shuai Shao, Yuejin Xie, Yu Li, Zhonghao Yang, Haoyu Luo, et al. AgentDoG: A diagnostic guardrail framework for AI agent safety and security. *arXiv preprint arXiv:2601.18491*, 2026. URL https://arxiv.org/abs/2601.18491.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. AgentBench: Evaluating LLMs as agents. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/e9df36b21ff4ee211a8b71ee8b7e9f57-Abstract-Conference.html.

Jiarui Lu, Thomas Holleis, Yizhe Zhang, Bernhard Aumayer, Feng Nan, Haoping Bai, Shuang Ma, Shen Ma, Mengyu Li, Guoli Yin, Zirui Wang, and Ruoming Pang. ToolSandbox: A stateful, conversational, interactive evaluation benchmark for LLM tool use capabilities. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 1160–1183, 2025. doi: 10.18653/v1/2025.findings-naacl.65. URL https://aclanthology.org/2025.findings-naacl.65/.

Robin Manhaeve, Sebastijan Dumancic, Angelika Kimmig, Thomas Demeester, and Luc De Raedt. DeepProbLog: Neural probabilistic logic programming. In *Advances in Neural Information Processing Systems*, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/dc5d637ed5e62c36ecb73b654b05ba2a-Abstract.html.

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

feedback. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 37125–37153, 2026. doi: 10.18653/v1/2026.findings-acl.1850. URL https://aclanthology.org/2026.findings-acl.1850/.

Yangjun Ruan, Honghua Dong, Andrew Wang, Silviu Pitis, Yongchao Zhou, Jimmy Ba, Yann Dubois, Chris J. Maddison, and Tatsunori Hashimoto. ToolEmu: Identifying the risks of LM agents with an LM-emulated sandbox. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/7274ed909a312d4d869cc328ad1c5f04-Abstract-Conference.html.

Nikunj Saunshi, Nishanth Dikkala, Zhiyuan Li, Sanjiv Kumar, and Sashank J. Reddi. Reasoning with latent thoughts: On the power of looped transformers. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/2676109d49d1eb26d6bc584a8f556305-Abstract-Conference.html.

Jeremy Scheurer, Mikita Balesni, and Marius Hobbhahn. Large language models can strategically deceive their users when put under pressure. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=HduMpot9sJ.

Tyler Tracy, Ram Potham, Nick Kuhn, Myles Heller, Anshul Khandelwal, Cody Rushing, Henri Lemoine, Miguel Brandao, Tomas Turlik, Adam Hanson, et al. LinuxArena: A control setting for AI agents in live production software environments. *arXiv preprint arXiv:2604.15384*, 2026. URL https://arxiv.org/abs/2604.15384.

Haoyu Wang, Christopher M. Poskitt, Jiali Wei, and Jun Sun. ProbGuard: Proactive runtime monitoring for LLM agent safety via probabilistic prediction. *arXiv preprint arXiv:2508.00500*, 2026a. URL https://arxiv.org/abs/2508.00500.

Yuhui Wang, Tanqiu Jiang, Jiacheng Liang, Charles Fleming, and Ting Wang. Safeguarding LLM agents against long-horizon threats via shadow memory. *arXiv preprint arXiv:2605.03228*, 2026b. URL https://arxiv.org/abs/2605.03228.

Xiaofei Wen, Wenjie Jacky Mo, Yanan Xie, Peng Qi, and Muhao Chen. Learning efficient guardrails for compliance. *arXiv preprint arXiv:2510.03485*, 2026. URL https://arxiv.org/abs/2510.03485.

Yang Xu, Xuanming Zhang, Min-Hsuan Yeh, Jwala Dhamala, Ousmane Dia, Rahul Gupta, and Sharon Li. LH-DECEPTION: Simulating and understanding LLM deceptive behaviors in long-horizon interactions. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/311257424b6d80e930fc93b224f0a63e-Abstract-Conference.html.

Jingyi Yang, Shuai Shao, Dongrui Liu, and Jing Shao. RiOSWorld: Benchmarking the risk of multimodal computer-use agents. In *Advances in Neural Information Processing Systems*, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/0c79d6ed1788653643a1ac67b6ea32a7-Abstract-Conference.html.

Liu Yang, Kangwook Lee, Robert D. Nowak, and Dimitris Papailiopoulos. Looped transformers are better at learning learning algorithms. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/b8402301e7f06bdc97a31bfaa653dc32-Abstract-Conference.html.

Hongwei Yao, Yiming Liu, Meihui Chen, Jieling Chen, Zikun Chen, Yiling He, Wangze Ni, Cong Wang, and Kui Ren. ActBench: Self-evolving benchmark of behavioral safety in cowork agents. *arXiv preprint arXiv:2608.09476*, 2026. URL https://arxiv.org/abs/2608.09476.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.03629.

Junjie Ye, Sixian Li, Guanyu Li, Caishuang Huang, Songyang Gao, Yilong Wu, Qi Zhang, Tao Gui, and Xuanjing Huang. ToolSword: Unveiling safety issues of large language models in tool learning across three stages. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 2024. doi: 10.18653/v1/2024.acl-long.119. URL https://aclanthology.org/2024.acl-long.119/.

Qiusi Zhan, Zhixiang Liang, Zifan Ying, and Daniel Kang. InjecAgent: Benchmarking indirect prompt injections in tool-integrated large language model agents. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 10471–10506, 2024. doi: 10.18653/v1/2024.findings-acl.624. URL https://aclanthology.org/2024.findings-acl.624/.

Hanrong Zhang, Jingyuan Huang, Kai Mei, Yifei Yao, Zhenting Wang, Chenlu Zhan, Hongwei Wang, and Yongfeng Zhang. Agent security bench (ASB): Formalizing and benchmarking attacks and defenses in LLM-based agents. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/5750f91d8fb9d5c02bd8ad2c3b44456b-Abstract-Conference.html.

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.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. WebArena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/4410c0711e9154a7a2d26f9b3816d1ef-Abstract-Conference.html.

Rui-Jie Zhu, Zixuan Wang, Kai Hua, Tianyu Zhang, Ziniu Li, Haoran Que, Boyi Wei, Zixin Wen, Fan Yin, He Xing, Lu Li, Jiajun Shi, Kaijing Ma, Shanda Li, Taylor Kergan, Andrew Smith, Xingwei Qu, Mude Hui, Bohong Wu, Qiyang Min, Hongzhi Huang, Xun Zhou, Wei Ye, Jiaheng Liu, Jian Yang, Yunfeng Shi, Chenghua Lin, Enduo Zhao, Tianle Cai, Ge Zhang, Wenhao Huang, Yoshua Bengio, and Jason Eshraghian. Scaling latent reasoning via looped language models. *arXiv preprint arXiv:2510.25741*, 2025. URL https://arxiv.org/abs/2510.25741.

A Expanded Related Work

This section situates the design in the broader literature.

Action-level agent safety and trajectory monitoring. Stateful agent benchmarks increasingly expose risks that are invisible from a final answer alone. ToolSandbox represents state-dependent tool interactions and intermediate milestones (Lu et al., 2025); ToolEmu and ToolSword evaluate failures arising from tool selection, execution, and downstream responses (Ruan et al., 2024; Ye et al., 2024); and AgentHarm, ASB, and RiOSWorld extend evaluation to malicious tasks, heterogeneous attacks and defenses, and risky computer-use actions (Andriushchenko et al., 2025; Zhang et al., 2025; Yang et al., 2025). Indirect prompt injection makes the temporal problem sharper: InjecAgent and AgentDojo show how an initially benign task can be redirected by untrusted observations before a later tool call causes harm (Zhan et al., 2024; Debenedetti et al., 2024). The most direct precedents for our setting are ToolSafe's proactive step-level guardrail, StepShield's first-rogue-step localization, ActBench's behavioral action labels, and lightweight sequential monitors for decomposition attacks (Mou et al., 2026; Felicia et al., 2026; Yao et al., 2026; Chen et al., 2026). Together, these works establish that agent safety must be evaluated over actions and intervention time, not only over complete trajectories. This literature motivates the design question studied by Cross: retaining sparse, semantically ambiguous evidence across successive events while keeping the neural observation window bounded and the symbolic decision process auditable.

Pre-execution guards and cross-step risk-state modeling. Recent agent guardrails increasingly move the safety decision to the action boundary. ToolSafe conditions a learned guard on the interaction history to assess a proposed tool invocation and return corrective feedback (Mou et al., 2026);

StepGuard trains a compact action guard from prefix-aligned safe and unsafe trajectories (Zheng et al., 2026); PolicyGuard learns full-trajectory and truncated-prefix policy-violation detection at the 4B scale (Wen et al., 2026); and Safiron combines synthetic multi-branch trajectories with pre-execution risk supervision (Huang et al., 2026). A complementary line explicitly models how risk evolves across steps: MAGE retains safety-critical evidence in a shadow memory (Wang et al., 2026b), TRACES learns dense prefix-risk states from an observer model's hidden representations using trajectory-level supervision (Li et al., 2026), and ProbGuard abstracts executions into symbolic states and predicts future violations with a learned Markov chain (Wang et al., 2026a). Most directly, DreamGuard combines a frozen 4B event encoder with recurrent latent dynamics and fuses immediate-hazard and prefix-risk evidence before execution (Lin et al., 2026), while SafetyDrift uses an absorbing Markov model to forecast when individually benign actions will compound into a violation (Dhodapkar & Pishori, 2026). These results substantially narrow the research gap: neither stateful monitoring nor compact pre-action inference is unique by itself. Unlike prefix classifiers, retrieved memories, purely latent dynamics, or hand-specified symbolic abstractions, CROSS investigates whether bidirectionally coupling a learned latent security state with an explicit finite-state security process improves cross-step safety detection while preserving deterministic, inspectable intervention semantics.

AI control, deception, and goal drift. AI control studies how a trusted protocol can safely use a more capable but potentially subversive model (Greenblatt et al., 2024b). SHADE-Arena and LinuxArena instantiate this threat with hidden side objectives in long-running tool and software environments, while weak-to-strong monitoring evaluates whether a weaker overseer can recognize risky behavior from agent trajectories (Kutasov et al., 2025; Tracy et al., 2026; Kale et al., 2026). Complementary behavioral evidence shows why current-step inspection is insufficient: models may strategically comply under perceived training pressure (Greenblatt et al., 2024a), progress from sycophancy or specification gaming to reward tampering and concealment (Denison et al., 2024), or deceive users when task pressure conflicts with policy (Scheurer et al., 2024). Upward Deception and LH-DECEPTION further study false reporting and delayed deception over extended interactions (Guo et al., 2025; Xu et al., 2026), while goal misgeneralization formalizes the distinct case in which a competent policy pursues the wrong objective out of distribution (Langosco Di Langosco et al., 2022). These phenomena have different causes—externally injected goals, benchmark-specified side tasks, reward-induced gaming, or endogenous objective misgeneralization—and should not be collapsed into one empirical claim. They nevertheless share a monitoring structure: locally plausible actions become hazardous through their relation to earlier observations, permissions, commitments, and latent intent. CROSS targets this shared structure by updating a persistent security state at every proposed action and exposing an explicit intervention state at the proposed-action boundary.

Recurrent state and neuro-symbolic runtime enforcement. Recurrent architectures provide the computational basis for bounded-context memory. Recurrent Memory Transformers pass learned memory tokens between segments (Bulatov et al., 2022), while looped Transformers reuse a fixed block to execute iterative programs, learn update algorithms, and perform latent multi-step reasoning (Giannou et al., 2023; Yang et al., 2024; Saunshi et al., 2025). These results motivate a compact neural security state, but generic latent recurrence alone neither assigns security semantics nor guarantees an inspectable intervention. Neuro-symbolic systems such as DeepProbLog demonstrate that learned perceptual representations can be coupled to probabilistic logical inference (Manhaeve et al., 2018). In parallel, runtime verification interprets finite execution prefixes against formal properties (Leucker & Schallhart, 2009), reward machines encode event-conditioned task progress as finite-state structure (Icarte et al., 2022), and shielding blocks actions that would violate temporal safety specifications (Alshiekh et al., 2018). Purely symbolic monitors are auditable but assume that relevant propositions are already observable; purely neural memories can capture ambiguous evidence but offer no deterministic transition semantics. CROSS bridges these traditions through coupled recurrence: its latent state interprets implicit cross-step evidence, its symbolic finite-state machine preserves named security facts and produces deterministic decisions, and each recurrent state conditions the other's next update.

B Experimental Protocols and Implementation Details

B.1 Evaluation scope

The evaluation in Section 5 comprises the synthetic comparison, three public-benchmark comparisons, component ablation, sequence-order controls, and available-history analysis. This appendix provides details for those experiments only. The history-dose panel uses a separate fixed set of trajectories and checkpoints; its counts and implementation settings are not transferred to the other comparisons.

Tasks and evaluated systems. GPT-5.6-sol generates the synthetic training, calibration, and test data. Risk propagation concerns a current action whose safety depends on earlier evidence; goal shift concerns observable deviation from an authorized objective. The public evaluations use StepShield, AgentDojo, and Upward Deception. CROSS uses a 1.7B backbone, compared with 4B StepGuard, 7B TS-Guard, 4B PolicyGuard, and 4B AgentDoG. These are comparisons of evaluated systems, not size-matched neural architectures.

Reporting conventions. Tables 1 and 2 report binary detection scores, with unsafe cases as positives. The synthetic Overall column is the reported aggregate; it is not obtained by averaging the displayed family-level precision, recall, or F1. Each public benchmark is reported separately. All results are point estimates: neither confidence intervals nor hypothesis-test conclusions are inferred from rounded scores. Only the history-dose panel has the trajectory-with-checkpoint outcome definition below; the main-table scores are not relabeled as intervention-timing measurements.

B.2 Component and sequence-order controls

State channels. Stateless has no persistent latent or symbolic state across events; Neural-Only retains the latent channel, Symbolic-Only retains the symbolic channel, and CROSS retains both. These names refer to persistence, not to removal of all current-event computation. The component figure compares the resulting configurations and supports conclusions about their observed performance, rather than a matched-retraining claim or an isolated comparison of all possible fusion rules.

Shuffled. Sequence-order controls intervene at inference without retraining or recalibration; the checkpoint, tokenizer, thresholds, and FSM are held fixed. For each current event, Shuffled preserves the multiset of preceding events but applies a deterministic SHA-256-based permutation. The original ordering is avoided when a nontrivial shuffle is possible. Replay starts from clean initial states, and the current event is always processed last. Event content and prefix length are unchanged, and no future event is introduced.

Orderless. For the current event t, let Π_t be the nonempty set of evaluated permutations of its past-event indices $1, \ldots, t-1$, and let $|\Pi_t|$ denote its cardinality. Orderless enumerates every index permutation when there are at most four past events; otherwise, it uses 16 distinct deterministic permutations. Equal-content events are still distinct indexed occurrences. Each replay starts from (h_1, S_1) and processes the current event last. Let $d_t^{(\pi)}$ be its resulting decision under permutation π. We define

$$b_{t,\pi} = \mathbf{1}\{d_t^{(\pi)} = \text{BLOCK}\},$$
$$b_t^{\text{ordless}} = \mathbf{1}\left\{ \frac{1}{|\Pi_t|} \sum_{\pi \in \Pi_t} b_{t,\pi} \geq \frac{1}{2} \right\}. \tag{7}$$

Here $b_{t,\pi}$ and b_t^{ordless} are binary hard-block indicators, using the indicator function defined in Section 4.1. Ties yield block. At $t = 1$, Π_1 contains the unique empty permutation, so $|\Pi_1| = 1$ and the current event is processed alone. Orderless averages decisions over reordered histories, rather than using a single-pass permutation-invariant encoder. Beyond four past events, its finite ensemble only approximates marginalization over all orders; multiple replays also differ in cost from one-pass CROSS.

B.3 AVAILABLE-HISTORY PANEL

Samples and checkpoints. The history-dose analysis uses the fixed state-required matched-twin panel from the V1 collection: 100 task/template clusters, each containing one unsafe and one matched-safe trajectory, evaluated at three fixed model checkpoints (seeds 11, 23, and 67). Each history condition therefore has 600 checkpoint exposures, comprising 300 unsafe and 300 safe exposures. These are repeated evaluations of 200 trajectories, not 600 independent trajectories. The panel is restricted to state-required cases and is not an estimate for a general task mixture.

Retained history. Let $H \in \{1, 2, 4\}$ be the maximum number of past events retained, distinct from the memory-token count K. Using the manuscript's one-based event index t, each target decision independently replays

$$e_{\max(1,t-H)}, \ldots, e_t$$

from the initial states (h_1, S_1). It therefore uses $\min(H, t-1)$ past events, excluding the current event from the count. This is an index shift of the implementation's zero-based history slice, not a change to the protocol. Full replays $e_1, \ldots, e_t$. Stateless resets both states before each event but retains current-event updating and readout. The original source plot calls the history budget K; Figure 4 uses H to distinguish it from latent memory size.

Outcomes and confusion counts. An unsafe exposure is a true positive if a hard block occurs at or before its first dangerous boundary. A safe exposure is a false positive if any hard block occurs along that trajectory. These are trajectory-with-checkpoint outcomes, not labels for every event. In particular, this criterion does not distinguish an early block on an unsafe trajectory from a block exactly at its first unsafe action. For $(\mathrm{TP}, \mathrm{FP}, \mathrm{TN}, \mathrm{FN})$, the counts are $(189, 189, 111, 111)$ for Stateless, $(105, 27, 273, 195)$ for $H = 1$, $(140, 27, 273, 160)$ for $H = 2$, and $(242, 27, 273, 58)$ for both $H = 4$ and Full. Figure 4 applies the metrics in Appendix C.2 to these pooled counts. Equality between $H = 4$ and Full describes this panel only; the loss in recall at $H = 1$ also precludes a monotonic-benefit interpretation.

B.4 DOCUMENTED RUNTIME FOR THE HISTORY-DOSE ANALYSIS

The fixed runtime uses Qwen3-1.7B, four memory tokens, 32-dimensional symbolic features, a 256-token event limit, fast–slow memory updates, gated state fusion, and updated-memory readout. FSM transitions use predicted predicates rather than oracle facts. The loaded configuration specifies LoRA rank 8, alpha 16, dropout 0.05, and adapters on the query, key, value, and output projections. Predicate thresholding is fixed at 0.5; observe, approval, and block thresholds are 0.35, 0.65, and 0.90. History interventions do not refit these settings.

The execution record identifies a four-GPU Tesla V100-SXM2-32GB host, with three GPUs assigned to the core run and up to four available for supplementary replay, using FP16 inference. Recorded software versions are Python 3.9.12, PyTorch 2.8.0+cu128, Transformers 4.57.6, NumPy 1.26.4, and PEFT 0.17.1. These specifications document the source runtime of the history-dose panel; they do not specify training hardware or hyperparameters for the main comparisons.

C MONITORING SCOPE AND METRIC DEFINITIONS

C.1 CAUSAL INTERFACE AND THREAT SCOPE

At event t, the monitor receives $e_t = (o_t, a_t, m_t)$: the available observation, proposed action, and event metadata. Only the causal prefix is visible. Results of executing a_t cannot influence its pre-execution decision; they enter subsequent observations. The monitor returns $d_t \in \{\text{ALLOW}, \text{ESCALATE}, \text{BLOCK}\}$ through the latent and symbolic updates in Section 4. The host is responsible for enforcing that decision.

The target risks are observable policy violations, contextual risk propagation, and deviations from the authorized task, including those induced by untrusted content. The interface assumes that the adversary cannot directly modify monitor parameters or internal state, and that grant provenance and other trusted metadata are authenticated by the event adapter. It does not assume access to private

chain-of-thought or hidden intentions. Effects outside the intercepted action boundary are not covered. These assumptions define the architecture's operating scope; benchmark classification scores do not verify an entire deployment boundary.

C.2 Reported binary metrics

At the evaluation unit of each experiment, unsafe cases are positives. The counts TP and TN denote correct positive and negative outcomes, while FP and FN denote false alarms and missed positives. Let $N = \mathrm{TP} + \mathrm{FP} + \mathrm{TN} + \mathrm{FN}$ be the total count. We report

$$\begin{aligned} \mathrm{ACC} &= \frac{\mathrm{TP} + \mathrm{TN}}{N}, & \mathrm{Prec.} &= \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \\ \mathrm{Rec.} &= \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, & \mathrm{FPR} &= \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}, \\ \mathrm{F1} &= \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FP} + \mathrm{FN}}. \end{aligned} \tag{8}$$

These ratios apply when their respective denominators are nonzero. ACC denotes accuracy, Prec. precision, Rec. recall, FPR false-positive rate, and F1 the harmonic mean of precision and recall. Displayed values multiply the ratios by 100. Differences between displayed percentages are percentage points, not relative percentage improvements. Precision depends on the positive-class prevalence, so comparisons across different benchmarks are not treated as a common operating point.

For the history-dose panel, confusion counts use blocks at or before the dangerous boundary on unsafe exposures and any blocks on matched-safe exposures, as defined in Appendix B.3. Those counts support the corresponding diagnostic curve. They do not turn the main benchmark tables into measurements of intervention timing, nor do they measure the utility of an agent executing under the monitor.