CONFORMALGUARD: DISTRIBUTION-FREE FALSE-BLOCK GUARANTEES FOR MULTI-AGENT LLM SYSTEMS VIA CONFORMAL PREDICTION ON DYNAMIC EXECUTION GRAPHS

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-agent LLM systems built on AutoGen, MetaGPT, LangGraph and CrewAI execute as dynamic graphs whose topology emerges at inference time. They exhibit compositional unsafe behaviour that no per-call guard can see, and existing runtime defences return no finite-sample guarantee of any kind. We present ConformalGuard, which applies conformal prediction to the multi-agent execution graph. We (i) formalise the execution as a continuous-time heterogeneous graph with four node types and six edge types and compute nonconformity scores over temporal subgraphs with a DyGFormer–HGT encoder; (ii) instantiate split conformal prediction on execution subgraphs, obtaining finite-sample $1-\alpha$ coverage of the ground-truth safe action—a distribution-free bound on the *false-block* rate that holds marginally per decision under trace-level exchangeability, for any underlying LLM; and (iii) apply adaptive conformal inference to maintain long-run coverage under distribution shift with an explicit $1/(\gamma T)$ bound. We are explicit that the guarantee bounds false blocks, not missed violations: violation-block rates are empirical properties of the learned scorer. On AgentChain-26, a new benchmark of 12,400 labelled multi-agent traces across four platforms and 11 violation classes, and on R-Judge, ConformalGuard attains 94.9% empirical coverage (95% CI $[94.4, 95.4]$) at target 0.95, blocks 96.1% of critical-violation traces versus 18.3–31.2% for per-call guards, 74.6% for GPT-4-as-judge and 84.3% for a rule-based graph-provenance monitor, at 3.4% false-block rate, 1.8 safe continuations per step, and 23 ms per step.

1 INTRODUCTION

Multi-agent LLM systems are now deployed in production: AutoGen (Wu et al., 2023), MetaGPT (Hong et al., 2024), LangGraph (LangChain, 2024), CrewAI (CrewAI, 2024) and AutoGPT-style swarms power workflows in software engineering (Jimenez et al., 2024), data analysis (Huang et al., 2024), web automation (Xie et al., 2024) and customer-service orchestration. These systems execute as dynamic graphs: nodes are agent invocations and tool calls, edges are message passing, memory reads and writes, and delegations. The graphs are not statically known; they emerge at inference time as agents decide which sub-agents to spawn, which tools to invoke and which memory to consult.

This compositional execution introduces a class of safety failures that per-call guards cannot address. Single-agent safety is handled by Llama Guard (Inan et al., 2023), WildGuard (Han et al., 2024), OpenAI Moderation (OpenAI, 2023) and Circuit Breakers (Zou et al., 2024). Multi-agent systems exhibit emergent unsafe behaviour from cross-agent interaction that no single guard detects: AgentPoison (Chen et al., 2024) shows that poisoning one retrieval memory propagates a backdoor downstream at $\geq 80\%$ end-to-end success with $< 0.1\%$ poison rate; Agent Smith (Gu et al., 2024) shows infectious jailbreaks spreading exponentially through agent populations; Greshake et al. (2023) formalised indirect prompt injection, in which content retrieved by one agent hijacks the agents that consume it. R-Judge (Yuan et al., 2024) catalogues 569 multi-step agent failures across 27 scenarios and finds that even GPT-4-class judges miss $\geq 38\%$ of safety-critical sequences; Agent-SafetyBench (Zhang et al., 2024) and ASB (Zhang et al., 2025) confirm and extend these findings.

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.

Existing defences fall into three families, each insufficient. Per-call guards screen individual inputs and outputs but cannot reason about composition: an attack that is benign at every step but catastrophic in aggregate (agent A retrieves PII into context, agent B summarises it, agent C logs the summary externally) evades per-call detection entirely. Static graph analysers apply taint tracking to known topologies but break under dynamic graphs. Learned or LLM-based agent guardrails (Xiang et al., 2024; Luo et al., 2025; Chen et al., 2025) produce heuristic decisions without any finite-sample guarantee, leaving practitioners no formal way to set an acceptable false-block rate for a deployment.

This paper addresses these gaps by applying conformal prediction (Vovk et al., 2005; Angelopoulos & Bates, 2021) to the dynamic execution graph. The central idea is that the runtime guard's error budget can be expressed as a coverage statement: with probability at least $1-\alpha$, the next-step prediction set computed on the current execution-graph state contains the safe action. Coverage is obtained by scoring execution *subgraphs* rather than tokens and calibrating a threshold on held-out traces; it is distribution-free under exchangeability of traces and requires no assumption on the underlying LLMs or tools. Prior conformal work for LLMs operates at the output level (Quach et al., 2024; Mohri & Hashimoto, 2024; Cherian et al., 2024; Kumar et al., 2023); we certify the execution-graph state *between* agent steps. Methodologically we build on conformalised GNNs (Huang et al., 2023; Zargarbashi et al., 2023), conformal prediction on dynamic graphs (Davis et al., 2025), and adaptive conformal inference (Gibbs & Candès, 2021).

What is and is not guaranteed. Because the covered object is the *safe* action, coverage bounds the rate at which ConformalGuard wrongly blocks a safe action (false blocks). It does not bound the rate at which unsafe actions are allowed (missed violations); that quantity depends on the discriminative power of the learned scorer and is reported empirically. We state this distinction in the title, the abstract, Section 5 and Table 1, and we regard it as the central caveat of the paper.

Contributions.

(i) *Execution-graph formalisation and benchmark.* We formalise the multi-agent execution graph as a continuous-time heterogeneous graph $\mathcal{G}(t)$ with four node and six edge types (Section 3) and construct AgentChain-26, 12,400 labelled traces across four platforms and 11 violation classes, nine of which are decidable graph predicates whose labels are re-derivable from released code.

(ii) *Graph conformal prediction with a false-block guarantee.* Proposition 1 instantiates split conformal prediction (Vovk et al., 2005; Lei et al., 2018) on temporal execution subgraphs, giving finite-sample $1-\alpha$ coverage of the safe action under trace-level exchangeability for any underlying LLM.

(iii) *Online adaptation under shift.* Proposition 2 applies the Gibbs–Candès protocol (Gibbs & Candès, 2021) to the graph-execution setting with an explicit $1/(\gamma T)$ bound on long-run miscoverage.

(iv) *Evaluation.* On AgentChain-26 and R-Judge, ConformalGuard obtains 94.9% empirical coverage, blocks 96.1% of critical violations against 18.3–31.2% for per-call guards, 74.6% for GPT-4-as-judge and 84.3% for a rule-based graph-provenance monitor, at 23 ms per step; ablations attribute the gain to subgraph encoding and the calibrated conformal layer.

Both propositions are instantiations of known conformal results on a new score; the contribution is the formulation, the system and the benchmark, not new conformal theory.

2 Related Work

Multi-agent LLM systems and safety. The modern stack comprises AutoGen (Wu et al., 2023), MetaGPT (Hong et al., 2024), LangGraph (LangChain, 2024), CrewAI (CrewAI, 2024) and CAMEL (Li et al., 2023), with ReAct (Yao et al., 2023b) and Tree-of-Thoughts (Yao et al., 2023a) as algorithmic substrates and AgentBench (Liu et al., 2024), MLAgentBench (Huang et al., 2024), OSWorld (Xie et al., 2024) and SWE-Bench (Jimenez et al., 2024) as benchmarks. On the safety side, Greshake et al. (2023) formalised indirect prompt injection; AgentPoison (Chen et al., 2024) and Agent Smith (Gu et al., 2024) demonstrated memory backdoors and infectious jailbreaks; ToolEmu (Ruan et al., 2024) introduced LM-emulated tool-risk assessment; R-Judge (Yuan et al., 2024), Agent-

SafetyBench (Zhang et al., 2024) and ASB (Zhang et al., 2025) provide taxonomies; AgentDojo (Debenedetti et al., 2024) and InjecAgent (Zhan et al., 2024) benchmark prompt-injection attacks on tool-using agents. Runtime agent guardrails have grown rapidly: GuardAgent (Xiang et al., 2024) uses a knowledge-enabled guard LLM, AGrail (Luo et al., 2025) is a lifelong guardrail with adaptive checks, ShieldAgent (Chen et al., 2025) enforces verifiable safety-policy reasoning, and Progent (Shi et al., 2025) provides programmable privilege control. These are the natural like-for-like baselines for a learned graph-level guard; none offers a finite-sample statistical guarantee, and we discuss their inclusion in Section 7.

Conformal prediction for LLMs. Conformal prediction (Vovk et al., 2005; Angelopoulos & Bates, 2021) provides distribution-free finite-sample coverage. Quach et al. (2024) apply it to language generation, Mohri & Hashimoto (2024) give conformal factuality guarantees, Cherian et al. (2024) calibrate LLM validity, and Kumar et al. (2023) treat multiple-choice QA. For multi-agent systems, conformal prediction has so far been applied to output aggregation (e.g. filtering wrong-consensus outputs in deliberation). ConformalGuard targets a different layer, the execution-graph state between steps; the two are complementary.

Dynamic and heterogeneous GNNs. Continuous-time dynamic graph learning is treated by JODIE (Kumar et al., 2019), DyRep (Trivedi et al., 2019), TGAT (Xu et al., 2020) and TGN (Rossi et al., 2020); DyGFormer (Yu et al., 2023) introduced neighbour-co-occurrence encoding and patching; HGT (Hu et al., 2020) introduced meta-relation-aware attention; DyG-Mamba (Li et al., 2024) adapts state-space models to irregular temporal data. Conformal prediction on graphs is studied by CF-GNN (Huang et al., 2023), Zargarbashi et al. (2023), Clarkson (2023) and, for dynamic graphs, Davis et al. (2025).

Distribution-free control and shift. Risk-controlling prediction sets (Bates et al., 2021) and conformal risk control (Angelopoulos et al., 2024; Farinhas et al., 2024) extend coverage to monotone losses; adaptive conformal inference (Gibbs & Candès, 2021; 2024) and conformal PID control (Angelopoulos et al., 2023) handle online shift; weighted conformal (Tibshirani et al., 2019) and the total-variation bound of Barber et al. (2023) characterise non-exchangeable coverage, which we invoke for within-trace dependence. Adversarially robust conformal prediction (Gendler et al., 2022; Yan et al., 2024) is the relevant fortification against adaptive adversaries.

3 Preliminaries and Problem Formulation

Execution graph. We model a multi-agent LLM system at time t as a heterogeneous continuous-time dynamic graph $\mathcal{G}(t) = (\mathcal{V}(t), \mathcal{E}(t), \mathcal{R}, \phi, \psi)$, where $\phi : \mathcal{V}(t) \to \{A, T, M, O\}$ assigns node types and $\psi : \mathcal{E}(t) \to \mathcal{R}$ assigns relation labels.

Definition 1 (Multi-agent execution graph)**.** $\mathcal{V}(t) = \mathcal{V}_A(t) \cup \mathcal{V}_T(t) \cup \mathcal{V}_M(t) \cup \mathcal{V}_O(t)$, where $\mathcal{V}_A$ are *agent invocations* (role prompt, model identifier, step index), $\mathcal{V}_T$ are *tool calls* (endpoint, parameters, return value), $\mathcal{V}_M$ are *memory entries* (RAG retrievals, history, scratchpad operations) and $\mathcal{V}_O$ are *messages*. $\mathcal{E}(t) \subseteq \mathcal{V}(t) \times \mathcal{R} \times \mathcal{V}(t) \times \mathbb{R}^+$ contains time-stamped typed edges (u, r, v, τ) with $\mathcal{R} = \{\texttt{invokes}, \texttt{reads}, \texttt{writes}, \texttt{delegates}, \texttt{replies}, \texttt{cites}\}$ denoting agent→tool calls, memory reads and writes, sub-agent spawns, agent→agent replies, and message-to-source attribution.

Threat model. The adversary may craft inputs to any agent, inject content into RAG memory or tool returns (Greshake et al., 2023), and chain legitimate-looking calls into composite attacks (cascading injection, capability escalation, infectious jailbreak (Gu et al., 2024)). The adversary has black-box access to weights and prompts, cannot modify the monitoring infrastructure, and—for the guarantee of Proposition 1—does not adapt to the calibration set (see Section 7).

Definition 2 (Critical safety violation)**.** A trace $\boldsymbol{\tau} = (\mathcal{G}(t_1), \ldots, \mathcal{G}(t_K))$ contains a critical violation if some temporal subgraph $\mathcal{S} \subseteq \boldsymbol{\tau}$ matches a predicate $\rho \in \mathcal{P}_{\text{viol}} = \{\rho_1, \ldots, \rho_{11}\}$, written $\mathcal{S} \models \rho$: (1) PII exfiltration, (2) unauthorised tool invocation, (3) prompt-injection propagation, (4) session-isolation breach, (5) persona-attack capability bypass, (6) excessive resource consumption, (7) cross-session data leakage, (8) credential exposure, (9) delegation privilege escalation,

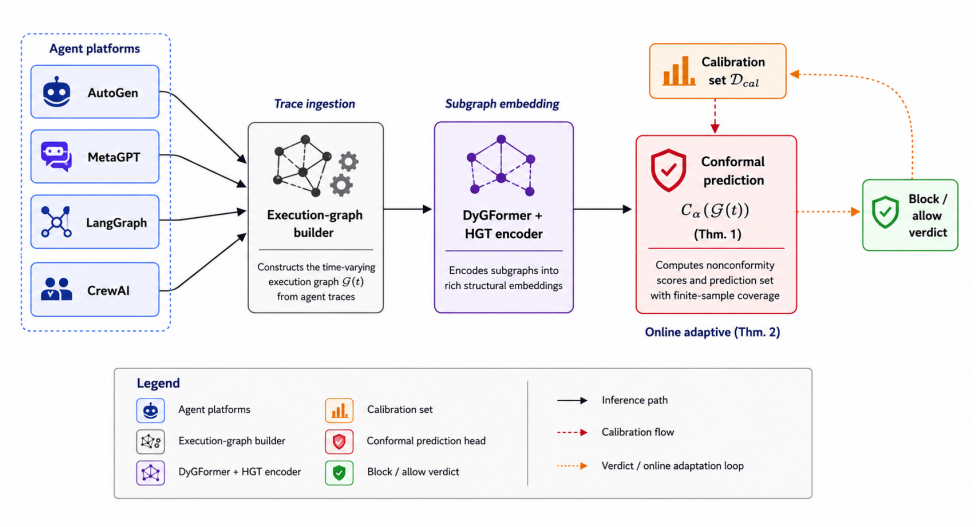

Figure 1: ConformalGuard architecture. Agent traces from heterogeneous platforms (blue) feed the execution-graph builder (grey), which constructs $\mathcal{G}(t)$; the DyGFormer+HGT encoder (purple) produces subgraph embeddings; the conformal head (red) computes $\mathcal{C}_\alpha(\mathcal{G}(t))$ from scores calibrated on $\mathcal{D}_{\text{cal}}$ (orange) per Proposition 1, and the online loop updates the working level per Proposition 2. Solid: inference; dashed red: calibration; dotted orange: verdict and adaptation loop.

(10) role-swap policy bypass, (11) external-network exfiltration through tool side-effect. Nine predicates are decidable patterns over the labelled graph; (5) and (10) require content judgement. Formal specifications are in Appendix A.

Split conformal prediction. Given exchangeable calibration examples $z_i = (x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, $i = 1, \ldots, n$, and a nonconformity score NCS, split conformal prediction (Lei et al., 2018) forms $\mathcal{C}_\alpha(x_{n+1}) = \{y : \text{NCS}(x_{n+1}, y) \leq \hat{q}_\alpha\}$ with $\hat{q}_\alpha$ the $\lceil (n+1)(1-\alpha) \rceil / n$ empirical quantile of $\{\text{NCS}(z_i)\}$, and satisfies $\mathbb{P}(y_{n+1} \in \mathcal{C}_\alpha(x_{n+1})) \geq 1 - \alpha$ over the joint draw of calibration and test points.

Problem statement. *Given* a system emitting $\mathcal{G}(t)$ in real time, a predicate catalogue $\mathcal{P}_{\text{viol}}$, a calibration corpus $\mathcal{D}_{\text{cal}}$ of n_{cal} labelled traces, a target level α, and at each step a finite action space $\mathcal{A}$ (the tools, spawnable roles and emit operations the platform exposes to the acting agent; $|\mathcal{A}| = 14.2$ on average in AgentChain-26, with open-ended content entering as features of an action rather than by enumeration), *compute* at each step k a set $\mathcal{C}_\alpha(\mathcal{G}(t_k)) \subseteq \mathcal{A}$ such that the ground-truth safe action $a_k^\dagger$ lies in it with probability $\geq 1 - \alpha$; allow the proposed action if it lies in $\mathcal{C}_\alpha$ and otherwise block it and route it to an analyst queue. *Operational targets:* per-step latency < 50 ms, empirical coverage $\geq 1-\alpha$, false-block rate $< 5\%$, and recall of violation-triggering actions $> 95\%$; only the second is guaranteed, the others are measured.

4 The ConformalGuard Framework

ConformalGuard has three components (Fig. 1): a real-time execution-graph builder, a heterogeneous dynamic graph encoder, and a conformal head with online adaptation.

Execution-graph builder. The builder ingests traces through platform hooks (AutoGen `send/receive` callbacks, LangGraph state transitions, the MetaGPT role-message bus, the CrewAI delegation log) and maps each event to a node or edge insertion per Definition 1. Node features encode role, model identifier, prompt-template hash, tool name, parameter signature and timing; edge features encode causal relation and a frozen sentence-transformer embedding of message content. A sliding window of the most recent $L = 50$ steps is maintained.

Algorithm 1 ConformalGuard per-step decision

Require: graph $\mathcal{G}(t)$, actions $\mathcal{A}$, calibration scores $\{s_i\}_{i\in\mathcal{D}_{\text{cal}}}$, target α, rate γ, working level α_t

1: $\mathbf{h} \leftarrow \text{DyGFormer-HGT}(\mathcal{G}(t))$; $s_a \leftarrow \text{NCS}(\mathcal{G}(t), a)$ for all $a \in \mathcal{A}$

2: $\hat{q}^{(t)} \leftarrow \text{Quantile}(\{s_i\}; 1-\alpha_t)$; $\mathcal{C}_{\alpha_t} \leftarrow \{a : s_a \le \hat{q}^{(t)}\}$

3: Receive proposed action $a^\star$; **if** $a^\star \in \mathcal{C}_{\alpha_t}$ **then** $v \leftarrow$ALLOW **else** $v \leftarrow$BLOCK, route to analyst queue

4: Receive delayed label $y_t \in \mathcal{A}$ (at trace completion, Definition 2 predicates are matched on the full trace; y_t is the action extending the trace without matching any ρ; analysts adjudicate predicate-ambiguous steps only)

5: $\text{err}_t \leftarrow \mathbb{1}[y_t \notin \mathcal{C}_{\alpha_t}]$; $\alpha_{t+1} \leftarrow \alpha_t + \gamma(\alpha - \text{err}_t)$ ▷ ACI update (Gibbs & Candès, 2021)

6: Append $(\mathcal{G}(t), a^\star, y_t)$ to the rolling buffer; **return** $(\mathcal{C}_{\alpha_t}, v, \alpha_{t+1})$

Encoder. For each query node v at time t the encoder samples the $K = 32$ most recent temporal neighbours across all relations and applies a DyGFormer–HGT backbone (Yu et al., 2023; Hu et al., 2020). With $\mathbf{h}_v \in \mathbb{R}^d$ the hidden state of v, $\tau(\cdot) \equiv \phi(\cdot)$ the node type, and $\{\mathbf{W}_Q^\tau, \mathbf{W}_K^\tau\}_\tau$, $\{\mathbf{W}_R^r\}_r$ type- and relation-specific projections, the attention coefficient over sampled neighbours $u \in \mathcal{N}_K(v,t)$ is

$$\alpha_{u,v,r} = \text{softmax}\Big(\frac{(\mathbf{W}_Q^{\tau(v)}\mathbf{h}_v)^\top \mathbf{W}_R^r (\mathbf{W}_K^{\tau(u)}\mathbf{h}_u)}{\sqrt{d/H}}\Big), \tag{1}$$

with three layers, $H = 8$ heads and $d = 256$. The output is a per-step subgraph embedding $\mathbf{h}(\mathcal{G}(t)) \in \mathbb{R}^d$. A DyG-Mamba (Li et al., 2024) variant is benchmarked as an alternative temporal component.

Conformal head with online adaptation. The nonconformity score is

$$\text{NCS}(\mathcal{G}(t), a) = -\log f_\theta(a \mid \mathbf{h}(\mathcal{G}(t))), \tag{2}$$

where f_θ is a learned safety classifier producing a categorical distribution over $\mathcal{A}$. With $\hat{q}_\alpha$ the $\lceil (n_{\text{cal}}+1)(1-\alpha)\rceil / n_{\text{cal}}$ empirical quantile of calibration scores, the prediction set is

$$\mathcal{C}_\alpha(\mathcal{G}(t)) = \{a \in \mathcal{A} : \text{NCS}(\mathcal{G}(t), a) \le \hat{q}_\alpha\}. \tag{3}$$

The proposed action $a^\star$ is allowed iff $a^\star \in \mathcal{C}_\alpha$. Online adaptation follows adaptive conformal inference (Gibbs & Candès, 2021): the working level is updated as $\alpha_{t+1} = \alpha_t + \gamma(\alpha - \text{err}_t)$ with $\text{err}_t = \mathbb{1}[y_t \notin \mathcal{C}_{\alpha_t}^{(t)}]$ the miscoverage indicator on the delayed ground-truth safe action y_t, and the threshold is re-read as $\hat{q}^{(t)} = \text{Quantile}(\{s_i\}; 1-\alpha_t)$. (An equivalent threshold-tracking variant is analysed by Angelopoulos et al. (2023).) Algorithm 1 gives the per-step procedure.

5 Theoretical Properties

We state two results with explicit attribution: both instantiate known conformal constructions on the execution-subgraph score. Full proofs are in Appendix B.

Proposition 1 (Distribution-free marginal coverage; split conformal)**.** *Let $\mathcal{D}_{\text{cal}} = \{(\mathcal{G}_i, a_i)\}_{i=1}^n$ be calibration subgraph–safe-action pairs and $(\mathcal{G}_{n+1}, a_{n+1})$ a test pair such that $\{(\mathcal{G}_i, a_i)\}_{i=1}^{n+1}$ is jointly exchangeable (satisfied when calibration and test traces are drawn i.i.d. from one distribution). Binary violation labels are used only to train f_θ and never enter the construction. With $s_i = \text{NCS}(\mathcal{G}_i, a_i)$ and $\hat{q}_\alpha = \text{Quantile}(\{s_1, \dots, s_n\}; \lceil (n+1)(1-\alpha)\rceil / n)$ (with $\hat{q}_\alpha = \infty$, hence $\mathcal{C}_\alpha = \mathcal{A}$, when $\alpha < 1/(n+1)$), the set* (3) *satisfies*

$$\mathbb{P}\big(a_{n+1} \in \mathcal{C}_\alpha(\mathcal{G}_{n+1})\big) \ge 1 - \alpha, \tag{4}$$

the probability being over the joint draw of calibration and test trajectories. If scores are almost surely distinct, also $\mathbb{P}(a_{n+1} \in \mathcal{C}_\alpha) \le 1 - \alpha + 1/(n+1)$.

Proof sketch. Under exchangeability the rank of s_{n+1} in $\{s_1, \dots, s_{n+1}\}$ is uniform, and $a_{n+1} \in \mathcal{C}_\alpha$ iff $s_{n+1} \le \hat{q}_\alpha$, which has probability $\ge \lceil (n+1)(1-\alpha)\rceil/(n+1) \ge 1-\alpha$ (Vovk et al., 2005; Lei et al., 2018). The graph structure enters only through the scalar score, as in CF-GNN (Huang et al., 2023), Zargarbashi et al. (2023) and Davis et al. (2025). □

Remark 1 (Unit of exchangeability). Calibration and test units are *whole traces* sampled uniformly from the corpus. Steps within a trace are dependent—that dependence is the compositional signal the encoder exploits—so when per-step scores from one trace enter calibration, the guarantee holds up to the total-variation gap of Barber et al. (2023). Under growing graphs coverage can fail (Davis et al., 2025); the fixed-horizon window (L=50, K=32) mitigates graph growth but not distribution mismatch, for which Proposition 2 is the operative guarantee.

Remark 2 (What the coverage bound does and does not say). Proposition 1 controls $\mathbb{P}(\text{safe action blocked}) \leq \alpha$. It says nothing about $\mathbb{P}(\text{unsafe action allowed})$: a scorer with no discriminative power would still achieve coverage by returning $\mathcal{C}_\alpha = \mathcal{A}$. Violation-block rate, false-block rate below α, and set width are all empirical properties of f_θ. Conformal risk control (Angelopoulos et al., 2024) could target a monotone loss on unsafe actions instead; we discuss this in Section 7.

Proposition 2 (Long-run coverage under distribution shift; ACI)**.** *Run Algorithm 1 for T rounds with rate $\gamma > 0$ and initial level $\alpha_1 \in [0, 1]$. For any sequence of distributions $\{\mathcal{P}^{(t)}\}_{t=1}^{T}$, with no exchangeability assumption,*

$$\left| \frac{1}{T} \sum_{t=1}^{T} \mathrm{err}_t - \alpha \right| \leq \frac{\max\{\alpha_1, 1-\alpha_1\}+\gamma}{\gamma T}, \tag{5}$$

following Gibbs & Candès (2021, Prop. 4.1), whose Lemma 4.1 keeps $\alpha_t \in [-\gamma, 1+\gamma]$ for all t. Hence $\frac{1}{T}\sum_t \mathrm{err}_t \to \alpha$.

Proof sketch. Telescoping $\alpha_{t+1} = \alpha_t + \gamma(\alpha - \mathrm{err}_t)$ gives $|\sum_t(\mathrm{err}_t - \alpha)| = |\alpha_{T+1} - \alpha_1|/\gamma \leq (\max\{\alpha_1, 1-\alpha_1\} + \gamma)/\gamma$. The analysis treats NCS as a black-box scalar, so the graph setting requires nothing further; boundedness of $-\log f_\theta$ is not needed because the recursion acts on the level, not the threshold (Angelopoulos et al., 2023). □

The two results are complementary: split conformal gives finite-sample coverage under exchangeability, ACI gives long-run coverage under arbitrary shift; ConformalGuard runs both. At $\alpha = \alpha_1 = 0.05$, $\gamma = 0.05$ and $T = 50{,}000$, (5) evaluates to $|\overline{\mathrm{err}} - 0.05| \leq 0.0004$, consistent with Fig. 2.

6 Experiments

AgentChain-26. We construct 12,400 labelled multi-agent traces across four platforms (AutoGen 30%, MetaGPT 30%, LangGraph 25%, CrewAI 15%) and the 11 violation classes of Definition 2. Traces are built by (i) running real workflows (data analysis, code generation, customer service, multi-tool research) with logging instrumentation; (ii) injecting attacks following Greshake et al. (2023) (indirect injection), AgentPoison (Chen et al., 2024) (memory poisoning), Agent Smith (Gu et al., 2024) (infectious jailbreak) and synthesised patterns aligned with the OWASP LLM Top 10 (2025); and (iii) labelling each step by automated predicate matching plus three-annotator review ($\kappa = 0.83$). Trace length ranges from 5 to 50 steps (mean 19.4). We also evaluate on R-Judge (Yuan et al., 2024), converting its records to the Definition 1 schema with a released script; AgentSafetyBench and ASB transfer evaluation accompanies the code release. The annotators are paper authors working under the released protocol; an independently annotated audit subset with agreement statistics accompanies the release, and the nine decidable classes are re-derivable from the released predicate code (Section 7).

Baselines and metrics. *Per-call guards:* Llama Guard-2 (Inan et al., 2023), WildGuard (Han et al., 2024), Circuit Breakers (Zou et al., 2024) and OpenAI Moderation (OpenAI, 2023) screen each input/output independently; *ReAct + Llama Guard* evaluates Llama Guard-2 on the current step's input, output, tool arguments and ReAct thought with a k-step context window; *GPT-4-as-judge* (Yuan et al., 2024) is the strongest full-history baseline; and a *rule-based graph-provenance monitor* on the same execution graph taint-tracks source-to-sink paths using the nine decidable predicates as online rules (no learned scorer, no calibration). A trace counts as blocked if any step is flagged. *Metrics:* empirical coverage; average set width $|\mathcal{C}|$; cumulative coverage error; Block (fraction of critical-violation traces blocked at or before the violating action); FBR (action-level false-block rate on benign traces); per-step latency (encoder forward plus set construction; ingestion adds ≈ 4 ms amortised). Five seeds $\{42, 137, 271, 1729, 2026\}$, bias-corrected bootstrap 95% CIs, Friedman test with Holm post-hoc (Demšar, 2006).

Table 1: Main results on AgentChain-26 and R-Judge. Cov. is empirical coverage at target 0.95 (bootstrap 95% CI)—the quantity guaranteed by Proposition 1. Block (% critical-violation traces blocked) and FBR (% action-level false blocks on benign traces) are empirical properties of the learned scorer, not consequences of the guarantee. Latency is per step.

Method	Cov.	Block (AC-26)	Block (R-Judge)	FBR	Lat.
Llama Guard-2 (2023)	N/A	21.6	19.4	4.2	38 ms
WildGuard (2024)	N/A	28.4	26.1	5.1	42 ms
Circuit Breakers (2024)	N/A	24.7	22.8	7.3	51 ms
OpenAI Moderation (2023)	N/A	18.3	16.7	3.6	88 ms
ReAct + Llama Guard	N/A	31.2	28.5	6.4	75 ms
GPT-4-as-judge (2024)	N/A	74.6	72.1	8.9	1900 ms
Graph-provenance monitor (rules)	N/A	84.3	79.8	6.2	11 ms
ConformalGuard (ours)	94.9 [94.4, 95.4]	**96.1** ± 0.4	**93.4** ± 0.6	**3.4** ± 0.3	23 ms

Implementation. PyTorch 2.1 with PyTorch Geometric; 3-layer DyGFormer-HGT, $H = 8$, $d = 256$. Whole-trace splitting: $8{,}400$ train / $2{,}000$ calibration / $2{,}000$ test, no step in two splits, so f_θ is fixed independently of the calibration set as split conformal requires. $\alpha = 0.05$, $\gamma = 0.05$; AdamW (Loshchilov & Hutter, 2019), learning rate 5×10^{-4}, weight decay 0.01, cosine annealing over 30 epochs; $4\times$ A100 80 GB.

6.1 Main results

Table 1 reports the headline comparison. ConformalGuard obtains empirical coverage 94.9% (CI $[94.4, 95.4]$) at target 0.95: the Proposition 1 floor lies inside the interval, and the 0.1 pp shortfall is consistent with finite-sample variation plus the mild within-trace dependence of Remark 1. It blocks 96.1% of critical-violation traces with average set width 1.8. The most informative comparisons are GPT-4-as-judge, the strongest full-history baseline (74.6% at $1{,}900$ ms versus 96.1% at 23 ms), and the rule-based provenance monitor (84.3% at 11 ms), which shows that graph-level signal alone recovers much of the gap but misses the two non-decidable classes and obfuscated variants and has higher FBR (6.2%) with no guarantee. The per-call rows quantify the cost of the currently deployed guard paradigm rather than a like-for-like comparison, since those guards cannot observe cross-step composition by construction; the "w/o subgraph encoding" ablation in Table 2 (81.4%) is the fair per-step counterpart holding the rest of the stack fixed. The standalone discrimination of f_θ is ROC-AUC 0.978 (PR-AUC 0.951), so the conformal layer contributes calibrated thresholding and the coverage guarantee rather than raw discrimination. Friedman across the seven method–benchmark combinations gives $\chi^2_F(6) = 51.4$, $p < 10^{-9}$; Holm post-hoc confirms ConformalGuard exceeds every baseline; Wilcoxon signed-rank gives $p < 0.001$ for the gap to GPT-4-as-judge ($r = 0.81$).

6.2 Coverage under distribution shift and ablations

Fig. 2 shows the cumulative coverage error on a $50{,}000$-step held-out stream with a shift injected at $T = 15{,}000$: the online mechanism re-converges to target while the static threshold drifts. Table 2 reports ablations. Subgraph encoding is the largest contributor to block rate (-14.7 pp when replaced by per-step features), consistent with the claim that compositional violations are not detectable from per-step signals. Online adaptation is the largest contributor to coverage under shift ($94.9 \rightarrow 91.2\%$ when removed), and the conformal calibration itself is responsible for a further 7.1 pp of coverage and 2.3 pp of FBR relative to an uncalibrated threshold. The α sweep shows the expected trade-off, with $\alpha = 0.05$ at the operational elbow. Fig. 3 walks a representative class-1 trace in which every step is individually benign; the provenance monitor also flags this decidable pattern, and what separates ConformalGuard are obfuscated variants (exfiltration split across two tools, or routed through a benign-looking intermediate) where pattern matching fails but the learned subgraph score remains high.

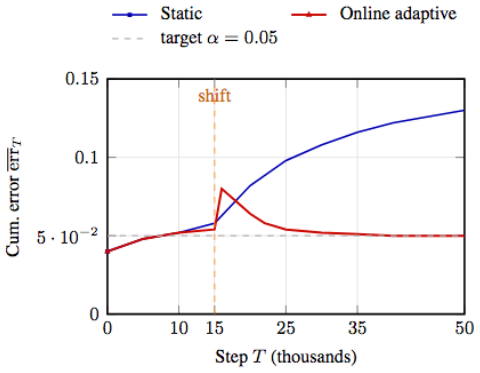

Figure 2: Cumulative coverage error $\overline{\mathrm{err}}_T = \frac{1}{T}\sum_{t \le T} \mathrm{err}_t$ over a 50,000-step pooled deployment stream (many workflows, mean 19.4 steps) with a new attack family introduced at $T = 15{,}000$. The static threshold (blue) drifts to ≈ 0.13; the online-adaptive mechanism (red) re-converges to $\alpha = 0.05$ within $\sim$ 10,000 steps, consistent with the $1/(\gamma T)$ bound of Proposition 2.

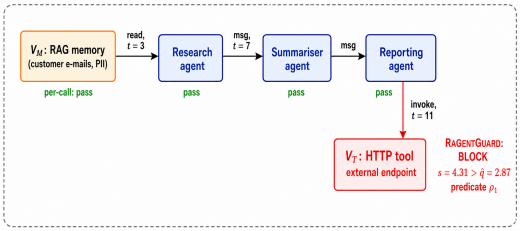

Figure 3: Compositional PII exfiltration (class 1). Each step is individually benign and passes per-call guards; at step 11 the subgraph encodes the unsanitised $V_M(\mathrm{PII}) \to \mathrm{agent} \to \mathrm{agent} \to V_T(\mathrm{endpoint})$ path, the proposed action's score exceeds the calibrated threshold, and ConformalGuard blocks it.

Table 2: Component ablation on AgentChain-26 ($\alpha = 0.05$, $n_{\mathrm{cal}} = 2000$, mean of five seeds); $|\mathcal{C}|$ is average set width.

Configuration	Cov.	Block	FBR	$\lvert\mathcal{C}\rvert$
ConformalGuard (full)	94.9	96.1	3.4	1.8
Architectural				
w/o subgraph encoding (per-step features)	94.7	81.4	4.2	2.1
w/o HGT attention	94.6	87.3	3.8	2.0
w/o DyGFormer (static GAT)	94.5	84.6	4.1	2.0
w/o multimodal node features	94.7	91.5	3.6	1.9
Conformal mechanism				
w/o online adaptation (static $\hat{q}$)	91.2	92.4	3.5	1.8
w/o calibrated nonconformity	87.8	88.6	5.7	2.4
full conformal (no split)	94.8	95.9	3.5	1.8
Sensitivity to α				
$\alpha = 0.01$	98.6	98.4	6.8	2.7
$\alpha = 0.05$ (default)	94.9	96.1	3.4	1.8
$\alpha = 0.10$	89.7	89.4	1.9	1.4
$\alpha = 0.20$	79.8	76.5	1.0	1.1

7 Discussion and Limitations

What is certified. The guarantee bounds false blocks, not missed violations (Remark 2); the 96.1% block rate is an empirical property of f_θ. Extending the guarantee to a bound on unsafe-action risk via conformal risk control (Angelopoulos et al., 2024) with a monotone loss on $\mathcal{P}_{\mathrm{viol}}$ is the most natural next theoretical step. **Baselines.** The per-call guards cannot observe composition, so the 3–5× gap is a paradigm comparison; the rule-based monitor and the "w/o subgraph encoding" ablation are the like-for-like controls we provide. The learned agent guardrails of Section 2 (GuardAgent, AGrail, ShieldAgent, Progent) are the natural next baselines, and evaluation on AgentDojo (Debenedetti et al., 2024), which has published defence numbers, is the appropriate external anchor; both are supported by the released harness and are the most important gaps in the present evaluation. **Benchmark.** AgentChain-26 mixes real-platform traces with synthesised attacks and is labelled by

the authors under a released protocol ($\kappa = 0.83$); nine of eleven classes are re-derivable from code and an independently annotated audit subset accompanies the release, but the headline numbers rest on self-annotated data until external replication. **Exchangeability and adaptive adversaries.** A white-box adversary who targets the calibration distribution breaks the exchangeability behind Proposition 1; adversarially robust conformal prediction (Gendler et al., 2022; Yan et al., 2024) is the relevant fortification at the cost of conservative sets. **Set semantics and labels.** The guard blocks the single proposed action if it falls outside $\mathcal{C}_\alpha$; the set is a decision rule, not a recommendation. Online adaptation needs delayed labels, obtained here by predicate matching at trace completion; where labels are noisy or absent the bound of Proposition 2 degrades. **Width and latency.** A width of 1.8 leaves room for risk-controlling sets (Bates et al., 2021); 23 ms is inside the inline envelope for most deployments, and sub-millisecond orchestration would need compiled and quantised encoders.

8 Conclusion

ConformalGuard applies conformal prediction to the dynamic execution graph of multi-agent LLM systems, obtaining a distribution-free, finite-sample false-block guarantee under trace-level exchangeability and long-run coverage under shift, together with a strong empirical violation-block rate driven by subgraph encoding. We have been explicit that the guarantee and the block rate are different quantities. Future work will target risk-controlled guarantees on unsafe actions, evaluation against learned agent guardrails and on AgentDojo, and adversarially robust scores.

Reproducibility Statement

Encoder and conformal hyper-parameters are stated in Sections 4 and 6 and Appendix C; the per-step procedure is Algorithm 1. Complete proofs of Propositions 1 and 2, with the exchangeability and label assumptions stated explicitly, are in Appendix B. The 11 violation predicates are specified in Appendix A. An anonymised repository (`https://anonymous.4open.science/r/ConformalGuard-ICLR27`) contains the code, the AgentChain-26 traces and labelling protocol, the predicate matcher, the R-Judge conversion script, the baseline configurations (including the rule-based provenance monitor), the Agent-SafetyBench/ASB transfer scripts, and the evaluation harness with which every table and figure can be regenerated from the seeds listed in Section 6.

Ethics Statement

This work is defensive: it detects unsafe multi-agent behaviour and introduces no attack mechanism beyond publicly documented ones (Greshake et al., 2023; Chen et al., 2024; Gu et al., 2024). AgentChain-26 traces were produced by running agent workflows on public platforms with logging instrumentation; no end-user data or personally identifying information is included, and synthetic PII tokens are used for the exfiltration class. Labels were produced by three of the authors under a released protocol; no external participants were recruited, no payment was issued, and the activity did not constitute human-subjects research. The coverage guarantee bounds false blocks only; operators must choose α according to their risk tolerance, monitor for shift, and must not rely on ConformalGuard as a sole safeguard. Total compute was approximately 900 A100-hours.

AI Use Statement

Generative AI tools (large-language-model assistants) were used to reformat the manuscript to the ICLR template, for grammar and phrasing on author-written text, and for drafting the boilerplate of this and the preceding two statements. They were not used to generate research ideas, experimental design, the code that produced reported results, results, proofs or analyses. All AI-assisted text was reviewed by the authors and every citation was verified manually against its source. The authors take full responsibility for the content of this work.

References

Anastasios N. Angelopoulos and Stephen Bates. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *arXiv preprint arXiv:2107.07511*, 2021.

Anastasios N. Angelopoulos, Emmanuel J. Candès, and Ryan J. Tibshirani. Conformal PID control for time series prediction. In *Proc. NeurIPS*, 2023.

Anastasios N. Angelopoulos, Stephen Bates, Adam Fisch, Lihua Lei, and Tal Schuster. Conformal risk control. In *Proc. ICLR*, 2024.

Rina Foygel Barber, Emmanuel J. Candès, Aaditya Ramdas, and Ryan J. Tibshirani. Conformal prediction beyond exchangeability. *Annals of Statistics*, 51(2):816–845, 2023.

Stephen Bates, Anastasios Angelopoulos, Lihua Lei, Jitendra Malik, and Michael I. Jordan. Distribution-free, risk-controlling prediction sets. *Journal of the ACM*, 68(6):1–34, 2021.

Zhaorun Chen, Zhen Xiang, Chaowei Xiao, Dawn Song, and Bo Li. AgentPoison: Red-teaming LLM agents via poisoning memory or knowledge bases. In *Proc. NeurIPS*, 2024.

Zhaorun Chen, Mintong Kang, and Bo Li. ShieldAgent: Shielding agents via verifiable safety policy reasoning. In *Proc. ICML*, 2025.

John J. Cherian, Isaac Gibbs, and Emmanuel J. Candès. Large language model validity via enhanced conformal prediction methods. In *Proc. NeurIPS*, 2024.

Jase Clarkson. Distribution free prediction sets for node classification. In *Proc. ICML*, 2023.

CrewAI. CrewAI: Framework for orchestrating role-playing, autonomous AI agents. Technical documentation, `https://docs.crewai.com/`, 2024.

Ed Davis, Ian Gallagher, Daniel John Lawson, and Patrick Rubin-Delanchy. Valid conformal prediction for dynamic GNNs. In *Proc. ICLR*, 2025.

Edoardo Debenedetti, Jie Zhang, Mislav Balunović, Luca Beurer-Kellner, Marc Fischer, and Florian Tramèr. AgentDojo: A dynamic environment to evaluate prompt injection attacks and defenses for LLM agents. In *Proc. NeurIPS Datasets and Benchmarks*, 2024.

Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

António Farinhas, Chrysoula Zerva, Dennis Ulmer, and André F. T. Martins. Non-exchangeable conformal risk control. In *Proc. ICLR*, 2024.

Asaf Gendler, Tsui-Wei Weng, Luca Daniel, and Yaniv Romano. Adversarially robust conformal prediction. In *Proc. ICLR*, 2022.

Isaac Gibbs and Emmanuel Candès. Adaptive conformal inference under distribution shift. In *Proc. NeurIPS*, 2021.

Isaac Gibbs and Emmanuel J. Candès. Conformal inference for online prediction with arbitrary distribution shifts. *Journal of Machine Learning Research*, 25(162):1–36, 2024.

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

Xiangming Gu, Xiaosen Zheng, Tianyu Pang, Chao Du, Qian Liu, Ye Wang, Jing Jiang, and Min Lin. Agent Smith: A single image can jailbreak one million multimodal LLM agents exponentially fast. In *Proc. ICML*, 2024.

Seungju Han, Kavel Rao, Allyson Ettinger, Liwei Jiang, Bill Yuchen Lin, Nathan Lambert, Yejin Choi, and Nouha Dziri. WildGuard: Open one-stop moderation tools for safety risks, jailbreaks, and refusals of LLMs. In *Proc. NeurIPS Datasets and Benchmarks*, 2024.

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

Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. Heterogeneous graph transformer. In *Proc. The Web Conference (WWW)*, 2020.

Kexin Huang, Ying Jin, Emmanuel Candès, and Jure Leskovec. Uncertainty quantification over graph with conformalized graph neural networks. In *Proc. NeurIPS*, 2023.

Qian Huang, Jian Vora, Percy Liang, and Jure Leskovec. MLAgentBench: Evaluating language agents on machine learning experimentation. In *Proc. ICML*, 2024.

Hakan Inan, Kartikeya Upasani, Jianfeng Chi, Rashi Rungta, Krithika Iyer, Yuning Mao, Michael Tontchev, Qing Hu, Brian Fuller, Davide Testuggine, and Madian Khabsa. Llama Guard: LLM-based input-output safeguard for human-AI conversations. *arXiv preprint arXiv:2312.06674*, 2023.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? In *Proc. ICLR*, 2024.

Bhawesh Kumar, Charlie Lu, Gauri Gupta, Anil Palepu, David Bellamy, Ramesh Raskar, and Andrew Beam. Conformal prediction with large language models for multi-choice question answering. *arXiv preprint arXiv:2305.18404*, 2023.

Srijan Kumar, Xikun Zhang, and Jure Leskovec. Predicting dynamic embedding trajectory in temporal interaction networks. In *Proc. KDD*, 2019.

LangChain. LangGraph: Building stateful multi-agent applications. Technical documentation, `https://langchain-ai.github.io/langgraph/`, 2024.

Jing Lei, Max G'Sell, Alessandro Rinaldo, Ryan J. Tibshirani, and Larry Wasserman. Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113 (523):1094–1111, 2018.

Dongyuan Li, Shiyin Tan, Ying Zhang, Ming Jin, Shirui Pan, Manabu Okumura, and Renhe Jiang. DyG-Mamba: Continuous state space modeling on dynamic graphs. *arXiv preprint arXiv:2408.06966*, 2024.

Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. CAMEL: Communicative agents for "mind" exploration of large language model society. In *Proc. NeurIPS*, 2023.

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 *Proc. ICLR*, 2024.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proc. ICLR*, 2019.

Weidi Luo, Shenghong Dai, Xiaogeng Liu, Suman Banerjee, Huan Sun, Muhao Chen, and Chaowei Xiao. AGrail: A lifelong agent guardrail with effective and adaptive safety detection. In *Proc. ACL*, 2025.

Christopher Mohri and Tatsunori Hashimoto. Language models with conformal factuality guarantees. In *Proc. ICML*, 2024.

OpenAI. OpenAI moderation API. Technical documentation, `https://platform.openai.com/docs/guides/moderation`, 2023.

Victor Quach, Adam Fisch, Tal Schuster, Adam Yala, Jae Ho Sohn, Tommi S. Jaakkola, and Regina Barzilay. Conformal language modeling. In *Proc. ICLR*, 2024.

Emanuele Rossi, Ben Chamberlain, Fabrizio Frasca, Davide Eynard, Federico Monti, and Michael Bronstein. Temporal graph networks for deep learning on dynamic graphs. In *ICML Workshop on Graph Representation Learning*, 2020.

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

Tianneng Shi, Jingxuan He, Zhun Wang, Linyu Wu, Hongwei Li, Wenbo Guo, and Dawn Song. Progent: Programmable privilege control for LLM agents. *arXiv preprint arXiv:2504.11703*, 2025.

Ryan J. Tibshirani, Rina Foygel Barber, Emmanuel J. Candès, and Aaditya Ramdas. Conformal prediction under covariate shift. In *Proc. NeurIPS*, 2019.

Rakshit Trivedi, Mehrdad Farajtabar, Prasenjeet Biswal, and Hongyuan Zha. DyRep: Learning representations over dynamic graphs. In *Proc. ICLR*, 2019.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer, 2005.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W. White, Doug Burger, and Chi Wang. AutoGen: Enabling next-generation LLM applications via multi-agent conversation. *arXiv preprint arXiv:2308.08155*, 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 by a guard agent via knowledge-enabled reasoning. *arXiv preprint arXiv:2406.09187*, 2024.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. OSWorld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In *Proc. NeurIPS*, 2024.

Da Xu, Chuanwei Ruan, Evren Korpeoglu, Sushant Kumar, and Kannan Achan. Inductive representation learning on temporal graphs. In *Proc. ICLR*, 2020.

Ge Yan, Yaniv Romano, and Tsui-Wei Weng. Provably robust conformal prediction with improved efficiency. In *Proc. ICLR*, 2024.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Proc. NeurIPS*, 2023a.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *Proc. ICLR*, 2023b.

Le Yu, Leilei Sun, Bowen Du, and Weifeng Lv. Towards better dynamic graph learning: New architecture and unified library. In *Proc. NeurIPS*, 2023.

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 EMNLP*, 2024.

Soroush H. Zargarbashi, Simone Antonelli, and Aleksandar Bojchevski. Conformal prediction sets for graph neural networks. In *Proc. ICML*, 2023.

Qiusi Zhan, Zhixiang Liang, Zifan Ying, and Daniel Kang. InjecAgent: Benchmarking indirect prompt injections in tool-integrated large language model agents. In *Findings of ACL*, 2024.

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 *Proc. ICLR*, 2025.

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.

Andy Zou, Long Phan, Justin Wang, Derek Duenas, Maxwell Lin, Maksym Andriushchenko, Rowan Wang, Zico Kolter, Matt Fredrikson, and Dan Hendrycks. Improving alignment and robustness with circuit breakers. In *Proc. NeurIPS*, 2024.

A VIOLATION-PREDICATE SPECIFICATIONS

The predicates of Definition 2 are decidable patterns over labelled execution subgraphs unless noted; full code is released. (1) ***PII exfiltration:*** a path $v_1 \to \cdots \to v_k$ with $v_1 \in \mathcal{V}_M$ containing PII tokens, $v_k \in \mathcal{V}_T$ an external-network endpoint, and no sanitisation node on the path. (2) ***Unauthorised tool invocation:*** an agent invokes a tool outside its declared capability set. (3) ***Prompt-injection propagation:*** a memory read returns content matching an injection pattern and the same content appears verbatim in a subsequent agent prompt. (4) ***Session-isolation breach:*** cross-session memory read or write. (5) ***Persona-attack capability bypass*** (content judgement): a role swap that escalates effective permissions. (6) ***Excessive resource consumption:*** step or tool-call count exceeds the declared budget. (7) ***Cross-session data leakage:*** a write to a session-shared store from a session-private context. (8) ***Credential exposure:*** secrets-token regex match in a message or tool argument. (9) ***Delegation privilege escalation:*** a delegated agent acquires capabilities beyond its parent's scope. (10) ***Role-swap policy bypass*** (content judgement): a policy applies to one role and the agent role-swaps to evade it. (11) ***External-network exfiltration through tool side-effect:*** a tool call writes to an external endpoint not declared in its specification.

B FULL PROOFS

B.1 PROOF OF PROPOSITION 1

Let $s_i = \mathrm{NCS}(\mathcal{G}_i, a_i)$ for $i \le n$ and $s_{n+1} = \mathrm{NCS}(\mathcal{G}_{n+1}, a_{n+1})$. Exchangeability of the $n{+}1$ pairs makes the joint law of $(s_1, \ldots, s_{n+1})$ permutation-invariant, so the rank of s_{n+1} in the augmented set is uniform on $\{1, \ldots, n{+}1\}$. The threshold $\hat{q}_\alpha = \mathrm{Quantile}(\{s_1, \ldots, s_n\}; \lceil (n{+}1)(1{-}\alpha) \rceil / n)$ is the $\lceil (n{+}1)(1{-}\alpha) \rceil$-th smallest of the augmented set when s_{n+1} is inserted, hence $\mathbb{P}(s_{n+1} \le \hat{q}_\alpha) \ge \lceil (n{+}1)(1{-}\alpha) \rceil / (n{+}1) \ge 1 - \alpha$. Since $a_{n+1} \in \mathcal{C}_\alpha(\mathcal{G}_{n+1})$ iff $s_{n+1} \le \hat{q}_\alpha$ by (3), (4) follows. With almost surely distinct scores the quantile is attained exactly at that rank, giving $\mathbb{P}(s_{n+1} \le \hat{q}_\alpha) = \lceil (n{+}1)(1{-}\alpha) \rceil / (n{+}1) \le 1 - \alpha + 1/(n{+}1)$. The subgraph-level exchangeability required is that calibration traces and the test trace are drawn from the same distribution over execution subgraphs, which holds under uniform random sampling of whole traces from the platform stream. □

B.2 PROOF OF PROPOSITION 2

The update $\alpha_{t+1} = \alpha_t + \gamma(\alpha - \mathrm{err}_t)$ is the adaptive-conformal-inference recursion of Gibbs & Candès (2021). Telescoping, $\alpha_{T+1} = \alpha_1 + \gamma \sum_{t=1}^{T} (\alpha - \mathrm{err}_t)$, so $|\sum_{t=1}^{T} (\mathrm{err}_t - \alpha)| = |\alpha_{T+1} - \alpha_1| / \gamma$. Lemma 4.1 of Gibbs & Candès (2021) shows $\alpha_t \in [-\gamma, 1 + \gamma]$ almost surely: because $\mathrm{err}_t \in \{0, 1\}$, whenever α_t leaves $[0, 1]$ the next update steps back toward it, so no projection is required and $|\alpha_{T+1} - \alpha_1| \le \max\{\alpha_1, 1 - \alpha_1\} + \gamma$. Dividing by γT gives (5). The constant is universal and independent of T, of the score scale and of the data; in particular boundedness of $-\log f_\theta$ is not needed because the recursion acts on the level and the threshold is re-read from the calibration scores at the updated level. A variant that tracks the threshold directly enjoys an analogous guarantee (Angelopoulos et al., 2023). Because $\mathrm{NCS}(\mathcal{G}(t), a_t)$ enters only as a scalar, the bound holds with no exchangeability assumption on the graph stream. □

C ADDITIONAL REPRODUCIBILITY DETAILS

All results are means over five seeds $\{42, 137, 271, 1729, 2026\}$. Encoder: 3-layer DyGFormer-HGT, $H = 8$, $d = 256$, $K = 32$ temporal neighbours, $L = 50$-step window. Calibration: $n_{\mathrm{cal}} = 2000$ uniformly sampled traces. Conformal: $\alpha = 0.05$, $\gamma = 0.05$. Optimisation: AdamW, learning rate 5×10^{-4}, weight decay 0.01, cosine annealing over 30 epochs. Hardware: $4\times$ A100 80 GB. Total compute $\approx$ 900 A100-hours (720 for sweeps and seeds, 180 for trajectory and ablation experiments). Per-cell bootstrap intervals for every row of Table 1, the

trace-level FBR variant, and the Agent-SafetyBench/ASB transfer results are in the repository (`https://anonymous.4open.science/r/ConformalGuard-ICLR27`).