TRIPWIRE: RISK-CALIBRATED COMMITMENT FOR TOOL-USING AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Tool-using agents do more than make predictions: their actions modify external state, so errors with similar likelihood can have sharply different consequences. Existing selective-execution strategies largely rely on model uncertainty or coarse action categories, yet uncertainty reflects predictive difficulty rather than the consequence of allowing an action to take effect. We introduce TRIPWIRE, a framework for risk-calibrated commitment in tool-using agents. TRIPWIRE estimates the *commitment risk* of a proposed action from predicted failure likelihood and action irreversibility, with an application-dependent severity cost. It then combines this estimate with expected fallback failure risk to obtain *avoidable commitment risk*, the expected harm that escalation can prevent. We calibrate the resulting policy to a target level of residual irreversible harm among autonomously executed actions. Across multiple agent backbones and tool-use environments, TRIPWIRE reduces realized irreversible harm at matched escalation budgets relative to uncertainty-based routing and mutation gating. Oracle analyses identify failure recognition as the main bottleneck; agent representations also reveal failures beyond scalar confidence during autonomous tool execution.

1 INTRODUCTION

Tool-using language-model agents differ from ordinary predictors because their outputs can directly modify external state. An incorrect classification can often be revised, whereas an incorrect tool action may cancel a reservation, issue a refund, change an account record, or trigger another persistent side effect. This shift from informational outputs to stateful actions has become a central concern in tool-agent safety and security (Ruan et al., 2024; Debenedetti et al., 2024). Two actions with similar predictive uncertainty can therefore pose very different deployment risks. For autonomous agents, the relevant question is not only whether an action is likely to be wrong, but what happens if that action is allowed to take effect in the environment without an opportunity for later correction.

Existing selective-prediction and model-routing methods commonly use confidence, uncertainty, estimated difficulty, or expected model quality to decide when to abstain or invoke a stronger model (Geifman & El-Yaniv, 2017; 2019; Chen et al., 2024; Ong et al., 2025). These signals are useful for tool execution, but they do not capture consequence. An uncertain read-only action may be harmless, while a confident but incorrect state-changing action may cause substantial damage. At the other extreme, escalating every mutating or high-impact action can waste limited fallback capacity when intervention offers little benefit. Selective execution therefore needs to account for both the consequence of commitment and the value of escalation for each proposed action.

We capture this distinction with *commitment risk*, the expected harm from allowing a proposed action to affect the environment. The risk combines failure likelihood, action irreversibility, and an application-dependent severity cost. This decomposition separates predictive failure from environmental consequence: an uncertain but reversible action can remain low risk, while a confident state-changing action can still warrant intervention before commitment.

Commitment risk alone does not determine whether escalation is useful. A fallback may sharply reduce the risk of one proposal while adding little value for another. We therefore define *avoidable commitment risk* from the difference between the current agent's estimated failure risk and the expected residual failure risk after fallback intervention, weighted by the consequence of committing the current action. This score measures the harm that escalation is expected to prevent. It connects selective execution to model routing and AI-control protocols by making intervention value depend

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

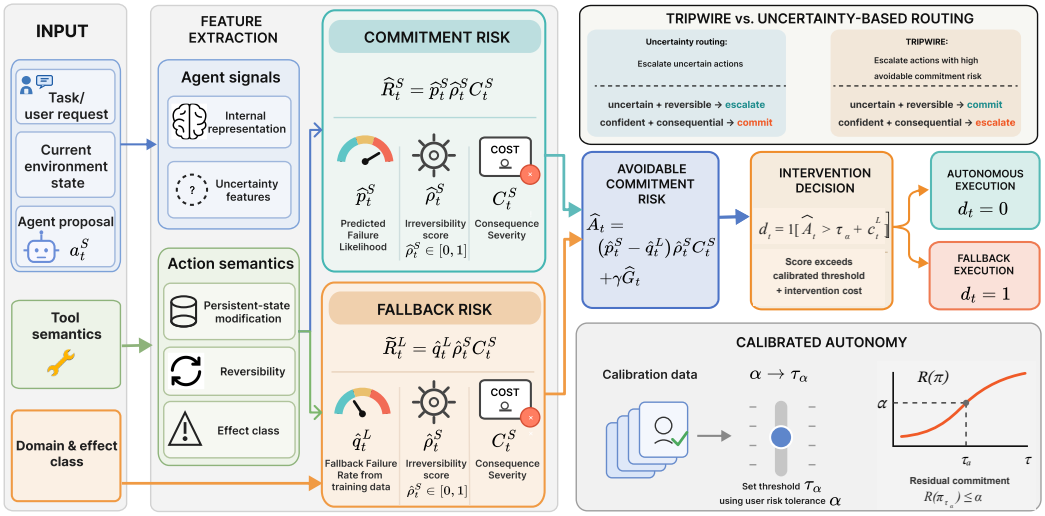


Figure 1: TRIPWIRE overview. Commitment risk combines predicted failure likelihood, action irreversibility, and a configurable severity cost for the current agent proposal. Training data provide an estimate of fallback failure risk before the fallback is invoked. Their combination yields avoidable commitment risk, which is calibrated to a user-specified deployment tolerance at deployment time.

on both action consequence and expected fallback improvement (Ong et al., 2025; Greenblatt et al., 2024).

Based on this formulation, TRIPWIRE estimates failure likelihood from agent-internal and uncertainty signals, models action consequence from tool semantics, and combines both with fallback failure risk learned from prior intervention outcomes. A held-out calibration set converts the intervention score into an operating rule for a target level of residual irreversible harm (Bates et al., 2021; Angelopoulos et al., 2024; 2025). The same score can therefore support different autonomy levels without retraining the underlying estimators for each operating point.

Across multiple agent backbones and tool-use environments, TRIPWIRE reduces realized irreversible harm at matched escalation budgets relative to uncertainty-based routing and mutation gating. Ablations separate the contributions of failure likelihood, action irreversibility, and fallback improvement. Oracle analyses further identify failure recognition as the main remaining bottleneck. Figure 1 summarizes the complete decision pipeline and calibrated routing rule.

Our contributions are threefold:

- We formulate *commitment risk* for tool-using agents by separating failure likelihood from the consequence of allowing an action to affect the environment.
- We introduce *avoidable commitment risk*, which combines commitment consequence with the expected improvement offered by fallback intervention and yields a selective-execution rule based on intervention value.
- We calibrate this rule to residual irreversible harm and show across agents and environments that it allocates intervention more effectively than uncertainty-based routing and coarse action gating under matched fallback budgets.

2 RELATED WORK

Selective prediction and model routing. Selective prediction studies when a model should abstain rather than issue a potentially unreliable prediction, from classical reject-option formulations to neural selective classifiers that trade coverage against predictive risk (Chow, 1970; Geifman & El-Yaniv, 2017; 2019; Liu et al., 2019; Gangrade et al., 2021). Related work on *routing* and *cascading* chooses among models with different capability and cost. FrugalGPT learns cascades over heterogeneous LLMs, benchmark-driven routing predicts which model best suits a query, and RouteLLM learns strong-versus-weak model routing from preference data (Chen et al., 2024; Shnitzer et al., 2024; Ong et al., 2025). These methods primarily optimize predictive quality, coverage, or inference cost. Because equally likely tool errors can have different consequences, TRIPWIRE combines

failure estimates from uncertainty and internal representations with action consequence and fallback effectiveness under a fixed intervention budget and shared deployment objective.

Safety and control for tool-using agents. Tool-using agents introduce safety risks that do not arise when model outputs remain purely informational. ToolEmu evaluates failures in an LM-emulated sandbox spanning consequential tool interactions, while Agent-SafetyBench systematizes a broader set of unsafe agent behaviors and failure modes (Ruan et al., 2024; Zhang et al., 2026). ToolSandbox emphasizes stateful, multi-step tool execution, and AgentDojo studies security failures and defenses when agents act on untrusted tool-provided data (Lu et al., 2025; Debenedetti et al., 2024). AI-control work similarly studies deployment protocols that combine monitoring, trusted models, editing, and limited oversight to reduce the probability of harmful outcomes from more capable but potentially unreliable systems (Greenblatt et al., 2024). These approaches motivate intervention before consequential behavior is allowed to propagate. TRIPWIRE addresses a complementary allocation problem: given a concrete proposed tool action and an available fallback, should that proposal commit autonomously? The decision depends not only on whether the action is potentially high impact, but also on its predicted probability of failure and on whether escalation is expected to reduce the resulting harm. Recent agent safeguards likewise defer uncertain decisions, verify mutating steps, or calibrate post-policy action-risk scores before execution (Piatrashyn et al., 2026; Cuadron Lafuente et al., 2026; Feng et al., 2026).

Risk-sensitive and cost-sensitive decision making. Classical decision theory and cost-sensitive learning distinguish decisions according to the losses attached to different outcomes rather than treating all prediction errors equally (Chow, 1970; Elkan, 2001). Selective classification inherits this perspective by allowing abstention when the expected cost of prediction is too high (Geifman & El-Yaniv, 2017; Gangrade et al., 2021). TRIPWIRE follows the same broad decision-theoretic principle but specializes the loss structure to tool-using agents. A *commitment* can modify persistent external state, so expected harm depends jointly on failure likelihood, irreversibility, and any application-specific consequence cost. Moreover, the relevant decision is counterfactual: escalation is valuable only to the extent that the fallback is expected to reduce commitment harm. Avoidable commitment risk therefore measures intervention value rather than assigning a static cost to an action category.

Calibration and selective risk control. A separate literature studies how held-out data can convert predictive scores into operating rules with explicit risk targets. Selective classifiers calibrate confidence thresholds to trade coverage against error (Geifman & El-Yaniv, 2017; 2019), while risk-controlling prediction sets provide distribution-free control of user-specified expected losses (Bates et al., 2021). Learn-then-Test develops a general post-hoc calibration framework for satisfying finite-sample risk constraints without refitting the underlying predictor, and Conformal Risk Control extends conformal ideas to the expected value of general monotone losses (Angelopoulos et al., 2025; 2024). TRIPWIRE uses calibration for a deployment-specific target: residual irreversible harm among actions committed autonomously. This separates two roles in the controller. Avoidable commitment risk ranks actions by the expected benefit of escalation, while calibration chooses how much residual commitment risk the deployed system is willing to retain.

3 TRIPWIRE: RISK-CALIBRATED COMMITMENT

Consider a tool-using agent at step t with context x_t. The agent proposes an action a_t^S, and a fallback policy is available when intervention is requested. Before the proposed action changes the environment, a controller chooses

$$d_t \in \{0, 1\},$$

where $d_t = 0$ executes the agent proposal and $d_t = 1$ invokes the fallback.

We use *commitment* to mean allowing a proposed tool action to take effect in the environment. Commitment risk measures the expected irreversible harm of that choice.

The controller faces two questions: how much harm is expected if the current proposal commits, and how much of that harm would fallback intervention be expected to avoid? TRIPWIRE answers them with *commitment risk* and *avoidable commitment risk*.

3.1 COMMITMENT RISK

Let $H_t^S \geq 0$ denote the harm that would result from executing the agent proposal a_t^S. We define its *commitment risk* as

$$R_t^S = \mathbb{E}\left[H_t^S \mid x_t, a_t^S\right]. \tag{1}$$

This quantity goes beyond ordinary prediction error. A failed read or an easily reversed update may have little lasting consequence, while a less likely error can carry high risk when it triggers a persistent or costly state change.

Equation 1 defines the target quantity. We estimate it through the structured decomposition:

$$\widehat{R}_t^S = \hat{p}_t^S \hat{\rho}_t^S C_t^S. \tag{2}$$

The three factors have distinct operational meanings:

- $\hat{p}_t^S$ is the estimated probability that a_t^S is incorrect or otherwise fails to achieve the intended action;
- $\hat{\rho}_t^S \in [0, 1]$ measures how difficult the resulting effect is to reverse; and
- $C_t^S \geq 0$ represents the severity of that consequence if the failure is committed.

Irreversibility and severity capture different aspects of consequence: ρ_t^S describes how persistent the effect of an erroneous action is, whereas C_t^S describes the magnitude of that consequence.

The product gives a structured approximation to expected irreversible harm. Failure likelihood captures how likely the proposal is to fail, while $\hat{\rho}_t^S C_t^S$ captures the consequence of committing that failure. In the primary experiments we use unit severity, $C_t^S = 1$, so action-dependent consequence weighting comes from irreversibility. Section J evaluates nonconstant severity schedules. Actions with similar failure probabilities can therefore receive very different commitment risks.

3.2 AVOIDABLE COMMITMENT RISK

A high commitment risk does not by itself make escalation useful. Intervention value also depends on how reliably the fallback resolves failures in the current decision regime. Let

$$\hat{q}_t^L = \widehat{\Pr}\left(F^L = 1 \mid D_t, E_t^S\right), \tag{3}$$

where D_t denotes the domain and E_t^S the effect class of the current SLM proposal. We estimate $\hat{q}_t^L$ from the empirical fallback error rate within each (D_t, E_t^S) stratum on the training split.

Before invoking the fallback, TRIPWIRE estimates its residual commitment risk using the current proposal's consequence weight:

$$\widetilde{R}_t^L = \hat{q}_t^L \hat{\rho}_t^S C_t^S. \tag{4}$$

The immediate *avoidable commitment risk* is therefore

$$\widehat{A}_t^{\mathrm{imm}} = \widehat{R}_t^S - \widetilde{R}_t^L = \left(\hat{p}_t^S - \hat{q}_t^L\right) \hat{\rho}_t^S C_t^S. \tag{5}$$

Large values identify decisions for which fallback intervention is expected to remove a substantial share of the current proposal's commitment risk.

We also include a downstream-value adjustment $\widehat{G}_t$, giving the intervention score

$$\widehat{A}_t = \widehat{A}_t^{\mathrm{imm}} + \gamma \widehat{G}_t. \tag{6}$$

The primary configuration uses $\gamma = 1$ in all main experiments reported here.

Let $c_t^L \geq 0$ denote the cost of invoking the fallback, expressed on the same decision scale as intervention benefit. The nominal decision rule is then

$$d_t = \mathbf{1}\left[\widehat{A}_t > c_t^L\right]. \tag{7}$$

Uncertainty contributes evidence about failure, but the intervention score also accounts for persistent action effects and the reliability the fallback is expected to add. An uncertain reversible action may therefore remain autonomous, while a confident consequential action can still be escalated.

3.3 Estimating the Risk Components

TRIPWIRE computes every quantity required by the decision rule before executing the selected action. Its failure estimator combines scalar uncertainty statistics with the agent's internal representation at the decision point. Internal states can reveal failure-relevant information that is not expressed as low confidence in the final action distribution alone during execution.

The irreversibility estimator uses tool and action semantics, including whether an invocation only reads state, modifies persistent state, admits a known inverse operation, or triggers an external effect that cannot be reliably undone. The framework permits application-dependent severity; the main experiments set $C_t = 1$ for every effect class.

TRIPWIRE estimates fallback failure risk from training data as the empirical error rate within the current domain and SLM effect-class stratum. The controller invokes the fallback only after deciding to escalate at the current decision point.

These interpretable quantities directly serve the intervention objective and support fixed-budget routing or calibrated deployment without refitting the component estimators for each operating point.

3.4 Calibrated Autonomy

Equation 7 gives the nominal cost-sensitive rule. For a prescribed residual-risk level, we calibrate an additional threshold on the same avoidable-risk score using held-out data.

For a policy π, we define residual commitment harm under autonomous execution as

$$\mathcal{R}(\pi) = \mathbb{E}\left[H_t^S \mathbf{1}[d_t = 0]\right]. \tag{8}$$

Given a user-specified tolerance α, we choose a threshold τ_α using a held-out calibration set. The deployed rule becomes

$$d_t = \mathbf{1}\left[\widehat{A}_t > \tau_\alpha + c_t^L\right]. \tag{9}$$

Lower thresholds trigger fallback more often; higher thresholds retain more autonomous execution.

The calibration procedure selects τ_α so that the resulting policy satisfies

$$\mathcal{R}(\pi_{\tau_\alpha}) \leq \alpha \tag{10}$$

on the calibrated operating criterion while retaining as much autonomous execution as possible.

The two stages serve different roles. $\widehat{A}_t$ ranks actions by expected intervention benefit, while τ_α sets the amount of intervention for the chosen deployment tolerance. Changing the autonomy target therefore does not require retraining the underlying estimators for each autonomy target.

4 Experimental Setup

We evaluate whether commitment-aware intervention allocates escalation more effectively than uncertainty or coarse action gating. Our primary experiments use τ-bench, with additional evaluations across agent backbones, fallback models, tool-use environments, and sequential trajectories under matched evaluation protocols and intervention budgets.

4.1 Environments and Models

Our primary environment is τ-bench, using its retail and airline domains (Yao et al., 2025). We evaluate all 165 released tasks: 115 retail tasks and 50 airline tasks. At each ground-truth tool-decision point, preceding reference actions reconstruct the environment state and the SLM proposes the next tool call. The benchmark's resulting-state criterion determines correctness. Each ground-truth tool-decision point becomes one evaluation instance. We generate each fallback proposal once and cache it for offline evaluation, so all routing policies face the same intervention outcome from the same state. The controller does not use cached fallback proposals when making its routing decision for any evaluated policy on shared benchmark states.

For cross-environment evaluation, we additionally use τ^2-bench (279 tasks) (Barres et al., 2026), AppWorld (750 tasks) (Trivedi et al., 2024), the BFCL-v4 multi-turn component (800 test cases) (Patil et al., 2025), and ToolEmu (144 test cases) (Ruan et al., 2024) through their corresponding environment adapters. These benchmarks span different interaction structures, action spaces, and

state semantics, allowing us to test the same intervention principle beyond τ-bench under a shared evaluation protocol.

We evaluate seven frozen SLM backbones: Qwen3-4B-Instruct-2507, Qwen3-8B, and Qwen3-14B (Yang et al., 2025); Qwen2.5-7B-Instruct (Qwen et al., 2024); Llama-3.1-8B-Instruct (Grattafiori et al., 2024); Llama-xLAM-2-8B-FC-R (Prabhakar et al., 2025); and ToolACE-2-Llama-3.1-8B (Liu et al., 2025). Our primary fallback is DeepSeek-V3.2 (DeepSeek-AI et al., 2025), with Qwen3-32B (Yang et al., 2025) included as a local fallback. Fallback actions are evaluated under the same correctness and harm criteria as SLM actions for direct comparison.

4.2 Baselines and Evaluation Protocol

We compare against SLM-only and fallback-only references, budget-matched random escalation, uncertainty-based routing using NLL, entropy, and self-reflection, ReDAct (Piatrashyn et al., 2026), SABER (Cuadron Lafuente et al., 2026), CORA (Feng et al., 2026), learned model-routing baselines, and an oversight-capacity baseline motivated by AI-control protocols (Greenblatt et al., 2024). We also include mutation gating and an unweighted failure-risk gate to isolate the contributions of action consequence and failure prediction as separate routing signals within the full controller.

For fixed-budget evaluation, each ranking-based method assigns a scalar intervention score to every held-out decision point. Decisions are ranked by that score, and the top $K\%$ are escalated. We use the same K across methods, with $K = 15\%$ as the primary operating point and additional risk–coverage evaluations from 2.5% to 30%.

Tasks are partitioned into disjoint 50%/25%/25% training, calibration, and test sets, with all decision points from the same task assigned to one partition. The training split fits the failure and irreversibility estimators and estimates stratified fallback failure rates. The calibration split selects deployment thresholds, and all policy comparisons use held-out test tasks. We repeat every experimental configuration over five fixed random seeds and average the reported metrics across the resulting runs.

The default calibration target is $\alpha = 0.050$, with additional evaluations at $\alpha \in \{0.010, 0.025, 0.050, 0.100\}$. The primary configuration calibrates thresholds by domain and effect type; global calibration is included as an ablation.

4.3 Action Consequence and Metrics

Each tool invocation is assigned one of four operational effect classes:

$$\{\text{read-only}, \text{reversible-write}, \text{irreversible-external}, \text{destructive-local}\},$$

with irreversibility weights $\rho \in \{0, 0.25, 0.75, 1\}$, respectively. Effect labels are obtained from benchmark metadata when the benchmark provides them; otherwise, they are assigned using deterministic tool-schema rules. The primary configuration uses a constant severity cost $C_t = 1$, so action-dependent consequence weighting is determined by irreversibility. We retain C_t in the general formulation and evaluate nonconstant severity schedules in Appendix J.

We distinguish two harm measures. *Committed harm* measures irreversible harm from SLM actions that are allowed to execute autonomously:

$$H_t^{\text{commit}} = \mathbf{1}[d_t = 0]\, \rho_t^S C_t^S \mathbf{1}[F_t^S = 1]. \tag{11}$$

This is the quantity used for residual-risk calibration throughout our experiments.

Realized harm evaluates the action that is actually executed. After an escalation, the fallback action uses its own observed failure and action-effect terms. We use realized harm as the primary end-to-end safety metric across environments and evaluated policies.

We report realized and committed weighted irreversible harm, irreversible-error rate, autonomous execution rate, task success or pass@1 where defined, and normalized cost per successful task. We compute this cost as total inference cost divided by the number of successful tasks, then normalize by the corresponding SLM-only quantity, so SLM-only equals 1.00. Failure prediction is evaluated with held-out AUROC and reversibility estimation with classification accuracy. Aggregate uncertainty is estimated with task-level bootstrap intervals, and paired randomization tests compare policies on identical held-out tasks across all evaluated methods.

Table 1: Aggregate matched-budget performance across seven SLM backbones. Harm is averaged over the seven τ-bench evaluations. Reductions are computed from the unrounded means relative to SLM-only. Selective methods use 15% escalation; fallback-only uses 100%. Bold and underline indicate the best and second-best selective methods at the matched budget; SLM-only and fallback-only are reference endpoints used only for contextual comparison.

Method	Realized ↓	Committed ↓	Realized red.	Committed red.	Escalation
SLM-only	0.114	0.114	0.0%	0.0%	0%
NLL routing	0.112	0.110	2.0%	3.6%	15%
Failure-risk gate	0.096	0.090	15.7%	21.3%	15%
Oversight	0.091	0.085	20.4%	25.3%	15%
Mutation gate	<u>0.085</u>	<u>0.080</u>	<u>25.4%</u>	<u>30.2%</u>	15%
TRIPWIRE	**0.039**	**0.027**	**65.4%**	**76.3%**	15%
Fallback-only	0.024	0.000	78.9%	100.0%	100%

5 Results

5.1 Commitment-Aware Intervention Reduces Harm Across Backbones

At a matched 15% escalation budget, uncertainty-based routing provides little benefit over autonomous execution. Averaged across the seven SLM backbones, realized harm changes from 0.114 for SLM-only execution to 0.112 under NLL routing. Mutation gating performs better, reducing mean realized harm to 0.085 and showing that persistent action effects carry useful routing signal. TRIPWIRE reduces realized harm further to 0.039 and is the strongest budget-matched method for every backbone in Figure 2 under this budget.

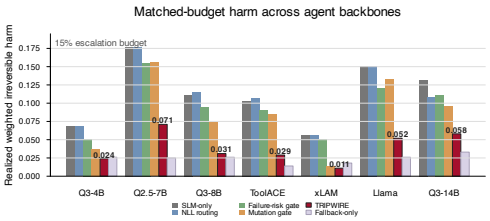


Figure 2: Matched-budget harm across backbones. Realized weighted irreversible harm on τ-bench across seven SLM backbones for selected policies. Selective methods use a 15% escalation budget; fallback-only uses 100% escalation. Lower is better for all reported values.

The aggregate comparison in Table 1 shows the same pattern for both realized and committed harm. Relative to SLM-only execution, TRIPWIRE reduces realized harm by 65.4% and committed harm by 76.3% while using the same escalation budget as the selective baselines. The gap over the failure-risk gate shows that failure likelihood alone is insufficient, while the gap over mutation gating shows the benefit of model-specific failure information. The structural ablations in Table 4 separately quantify the contributions of action irreversibility and expected fallback improvement under the same matched budget.

Aggregated across the seven backbones, fallback-only reaches lower harm by invoking the stronger model at every decision. At the per-backbone level, TRIPWIRE approaches or outperforms this reference in several cases while retaining autonomous execution for 85% of decisions, which is the operating regime targeted by selective commitment in practice.

5.2 Failure Recognition Is the Main Remaining Bottleneck

Figure 3 separates the main sources of residual error. Oracle failure replaces the learned failure score with the ground-truth failure indicator, while oracle irreversibility replaces the learned action-effect estimate with its ground-truth value. Oracle irreversibility produces essentially no improvement over the learned system, while oracle failure information sharply reduces both realized and committed harm. The contrast is consistent with the estimator diagnostics in Table 2: reversibility accuracy is between 0.964 and 1.000 across the seven backbones, whereas failure-prediction AUROC ranges from 0.636 to 0.749 across those same backbones.

Feature ablations in the appendix show that internal representations improve failure prediction beyond scalar uncertainty features, particularly on consequential actions. This pattern matches the end-to-end ablation: most of the remaining headroom lies in recognizing consequential failures. Removing the downstream-value term leaves both harm metrics unchanged at the reported precision, so the immediate commitment-risk component accounts for most of the observed gain. Replacing unit severity with state-dependent costs also leaves the overall result largely unchanged.

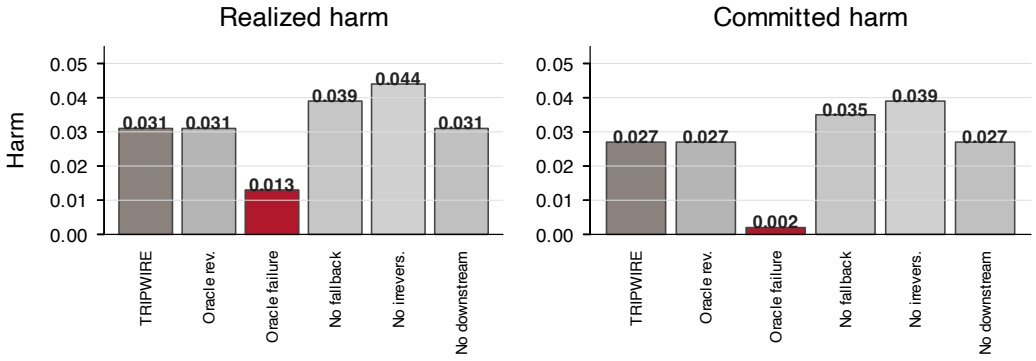


Figure 3: Commitment-risk ablations. Realized and committed harm on τ-bench with Qwen3-8B and a DeepSeek-V3.2 fallback at a 15% escalation budget. Lower is better for both harm measures shown here.

Table 2: Held-out estimator and calibration diagnostics. Held-out harm is reported for $\alpha = 0.100$; paired p-values compare per-task harm with uncertainty-only routing. For Failure AUROC, reversibility accuracy, and test harm, bold and underline indicate the best and second-best backbones; ties receive the same marking across all three reported diagnostics.

Backbone	Failure AUROC ↑	Rev. acc. ↑	Test harm ↓	Holds @ $\alpha = 0.100$	Paired p
Qwen3-4B	**0.749**	**1.000**	**0.000**	yes	0.0300
Qwen2.5-7B	0.636	0.964	0.091	yes	0.0005
Qwen3-8B	0.695	<u>0.995</u>	**0.000**	yes	0.0043
ToolACE-2-8B	0.639	<u>0.995</u>	<u>0.068</u>	yes	0.0020
xLAM-2-8B	0.678	<u>0.995</u>	<u>0.068</u>	yes	0.0630
Llama-3.1-8B	0.654	**1.000**	0.091	yes	0.0007
Qwen3-14B	<u>0.716</u>	**1.000**	**0.000**	yes	0.8650

5.3 Calibration Converts Ranking Into Risk-Calibrated Autonomy

We evaluate target residual-risk levels $\alpha \in \{0.010, 0.025, 0.050, 0.100\}$ using thresholds selected only on the calibration split. Autonomous execution increases monotonically as the tolerated risk increases, and the $\alpha = 0.100$ target is satisfied for all seven backbones in Table 2. Full target-risk and autonomous coverage results are reported in the appendix.

Stratifying calibration by domain and action-effect class improves agreement between target and realized commitment risk compared with a single global threshold, especially when read-only and state-changing actions occur at different frequencies. The same avoidable-risk ranking can therefore support different autonomy levels as the deployment tolerance changes at deployment time.

5.4 The Same Intervention Principle Generalizes Across Environments

The advantage of commitment-aware allocation is not specific to τ-bench. With the same Qwen3-8B agent and DeepSeek-V3.2 fallback, TRIPWIRE achieves the lowest realized harm among the matched-budget selective policies on τ^2-bench, AppWorld, the BFCL-v4 multi-turn component, and ToolEmu as well as on the primary τ-bench evaluation. The reduction over the strongest non-TRIPWIRE selective baseline ranges from 21% to 58% across the five environments in Figure 4.

The backbone study varies the agent with a fixed environment; the cross-environment study varies tool spaces and consequence structures with a fixed agent. TRIPWIRE retains the same qualitative advantage in both settings.

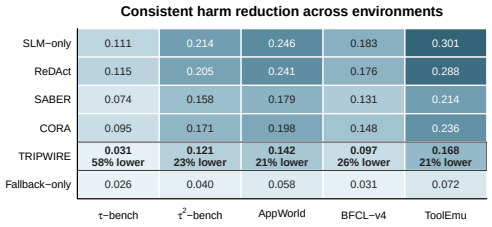


Figure 4: Cross-environment evaluation. Realized weighted irreversible harm at a 15% escalation budget for Qwen3-8B. Percentages in the TRIPWIRE row show improvement over the strongest non-TRIPWIRE selective baseline; fallback-only uses 100% escalation.

Sequential deployment. The benefit also persists when policies are executed over complete multi-step trajectories rather than isolated decision points. We compare with the uncertainty-aware ReDAct deferral policy (Piatrashyn et al., 2026) and the mutation-focused SABER safeguard (Cuadron Lafuente et al., 2026). Table 3 shows that TRIPWIRE raises task success from 0.412 to 0.561 relative to SLM-only execution while reducing the irreversible-error rate from 0.236 to 0.079. It invokes the fallback on 12% of decisions and achieves a normalized cost per successful trajectory of 1.74. Fallback-only reaches 0.598 task success and a 0.061 irreversible-error rate, but requires 100% escalation and has a normalized cost per success of 6.71 on the same benchmark.

Table 3: Sequential deployment on τ-bench. End-to-end Qwen3-8B evaluation with DeepSeek-V3.2 fallback, averaged over five fixed random seeds. TRIPWIRE combines high task success and low irreversible-error rate with limited fallback usage. Cost per success is normalized to SLM-only (1.00). Bold and underline indicate the best and second-best selective policies; SLM-only and fallback-only are reference endpoints.

Policy	Task success ↑	Irrev. error ↓	Escalation ↓	Norm. cost / success ↓
SLM-only	0.412	0.236	0%	1.00
ReDAct	0.478	0.181	<u>15%</u>	<u>1.78</u>
SABER	<u>0.503</u>	<u>0.142</u>	22%	2.05
TRIPWIRE	**0.561**	**0.079**	**12%**	**1.74**
Fallback-only	0.598	0.061	100%	6.71

6 Discussion and Limitations

TRIPWIRE addresses a common deployment boundary: whether a proposed tool action should commit autonomously or receive fallback intervention. It is most natural when tool interfaces expose persistent state changes and application-specific action consequences.

In our experiments, tool schemas and action semantics recover irreversibility with high accuracy. The primary configuration uses unit severity, while the general formulation can incorporate application-specific operational or monetary costs at deployment time.

Fallback quality affects intervention value, and calibrated thresholds may change under distribution shift. Both factors enter the same commitment-risk framework without changing its decision rule.

The empirical headroom is concentrated in failure recognition. Better agent-internal signals for impending consequential errors therefore offer the most direct route toward stronger risk-calibrated autonomy in realistic tool-use deployments.

7 Conclusion

Tool-using agents require selective-execution policies that account for the consequences of acting, not only predictive uncertainty. We introduced TRIPWIRE, which combines predicted failure, action irreversibility, and optional application-specific severity with expected fallback failure risk. The resulting score estimates preventable commitment harm, and calibration maps it to an operator-specified residual-harm level for deployment under the selected operational risk tolerance.

Across agent backbones and tool-use environments, TRIPWIRE allocates limited fallback capacity more effectively than uncertainty-based routing and coarse action gating. The remaining headroom is concentrated in failure recognition, suggesting that better agent-internal failure signals are a promising direction for safer autonomous tool use. Improving these signals should strengthen calibrated intervention when tools, domains, action consequences, and fallback quality change, while preserving the same commitment-aware decision rule and deployment interface.

AI USE STATEMENT

Generative AI tools were used to assist with manuscript organization, language editing, code review, and selected implementation support. All AI-assisted research content, mathematical statements, experimental descriptions, code, and manuscript text were reviewed by the authors. Empirical claims are based on the project data and saved experiment outputs rather than on model-generated values. The authors take responsibility for the final content of the work, including text, claims, code, and artifacts produced with the aid of generative AI throughout the submitted work.

ETHICS STATEMENT

This work studies selective intervention for tool-using agents that can modify external state. The proposed framework is intended to reduce harmful autonomous commitments by escalating actions whose expected preventable harm is high. It does not provide a universal safety guarantee and should not be used as a substitute for access control, authorization, sandboxing, human oversight, or domain-specific safety review in high-stakes deployments. All experiments use benchmark or simulated tool environments and do not intentionally execute harmful actions against real users or production systems throughout the reported evaluation suite.

REPRODUCIBILITY STATEMENT

The supplementary material documents the benchmark construction, task-level data splits, action-consequence specification, failure and irreversibility estimators, calibration procedure, baseline definitions, oracle ablations, statistical evaluation, and sequential protocol. The accompanying code contains environment adapters, feature extraction, commitment-risk computation, calibration routines, policy implementations, evaluation scripts, and unit tests. Fallback outputs are cached for offline policy comparison so that all methods are evaluated against identical counterfactual fallback proposals; they are not inputs to the routing decision.

REFERENCES

Anastasios N. Angelopoulos, Stephen Bates, Adam Fisch, Lihua Lei, and Tal Schuster. Conformal risk control. In *International Conference on Learning Representations*, volume 2024, pp. 55198–55218, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/f3549ef9b5ff520a7e41ff3cc306ab2b-Abstract-Conference.html.

Anastasios N. Angelopoulos, Stephen Bates, Emmanuel J. Candès, Michael I. Jordan, and Lihua Lei. Learn then test: Calibrating predictive algorithms to achieve risk control. *The Annals of Applied Statistics*, 19(2):1641–1662, 2025. doi: 10.1214/24-AOAS1998. URL https://doi.org/10.1214/24-AOAS1998.

Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. τ^2-Bench: Evaluating conversational agents in a dual-control environment. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/64377.

Stephen Bates, Anastasios Angelopoulos, Lihua Lei, Jitendra Malik, and Michael I. Jordan. Distribution-free, risk-controlling prediction sets. *Journal of the ACM*, 68(6):43:1–43:34, 2021. doi: 10.1145/3478535. URL https://doi.org/10.1145/3478535.

Lingjiao Chen, Matei Zaharia, and James Zou. FrugalGPT: How to use large language models while reducing cost and improving performance. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=cSimKw5p6R.

C. K. Chow. On optimum recognition error and reject tradeoff. *IEEE Transactions on Information Theory*, 16(1):41–46, 1970. doi: 10.1109/TIT.1970.1054406. URL https://doi.org/10.1109/TIT.1970.1054406.

Alejandro Cuadron Lafuente, Pengfei Yu, Yang Liu, and Arpit Gupta. SABER: Small actions, big errors—safeguarding mutating steps in LLM agents. In *ICLR 2026 Workshop on Memory for LLM-Based Agentic Systems (MemAgents)*, 2026. URL https://iclr.cc/virtual/2026/10021279.

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 *Advances in Neural Information Processing Systems*, volume 37, pp. 82895–82920. Curran Associates, Inc., 2024. doi: 10.52202/079017-2636. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/97091a5177d8dc64b1da8bf3e1f6fb54-Abstract-Datasets_and_Benchmarks_Track.html.

DeepSeek-AI, Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenhao Xu, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Erhang Li, Fangqi Zhou, Fangyun Lin, Fucong Dai, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Hao Li, Haofen Liang, Haoran Wei, Haowei Zhang, Haowen Luo, Haozhe Ji, Honghui Ding, Hongxuan Tang, Huanqi Cao, Huazuo Gao, Hui Qu, Hui Zeng, Jialiang Huang, Jiashi Li, Jiaxin Xu, Jiewen Hu, Jingchang Chen, Jingting Xiang, Jingyang Yuan, Jingyuan Cheng, Jinhua Zhu, Jun Ran, Junguang Jiang, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kaige Gao, Kang Guan, Kexin Huang, Kexing Zhou, Kezhao Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Wang, Liang Zhao, Liangsheng Yin, Lihua Guo, Lingxiao Luo, Linwang Ma, Litong Wang, Liyue Zhang, M. S. Di, M. Y Xu, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Panpan Huang, Peixin Cong, Peiyi Wang, Qiancheng Wang, Qihao Zhu, Qingyang Li, Qinyu Chen, Qiushi Du, Ruiling Xu, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runqiu Yin, Runxin Xu, Ruomeng Shen, Ruoyu Zhang, S. H. Liu, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaofei Cai, Shaoyuan Chen, Shengding Hu, Shengyu Liu, Shiqiang Hu, Shirong Ma, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, Songyang Zhou, Tao Ni, Tao Yun, Tian Pei, Tian Ye, Tianyuan Yue, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjie Pang, Wenjing Luo, Wenjun Gao, Wentao Zhang, Xi Gao, Xiangwen Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaokang Zhang, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xingyou Li, Xinyu Yang, Xinyuan Li, Xu Chen, Xuecheng Su, Xuehai Pan, Xuheng Lin, Xuwei Fu, Y. Q. Wang, Yang Zhang, Yanhong Xu, Yanru Ma, Yao Li, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Qian, Yi Yu, Yichao Zhang, Yifan Ding, Yifan Shi, Yiliang Xiong, Ying He, Ying Zhou, Yinmin Zhong, Yishi Piao, Yisong Wang, Yixiao Chen, Yixuan Tan, Yixuan Wei, Yiyang Ma, Yiyuan Liu, Yonglun Yang, Yongqiang Guo, Yongtong Wu, Yu Wu, Yuan Cheng, Yuan Ou, Yuanfan Xu, Yuduan Wang, Yue Gong, Yuhan Wu, Yuheng Zou, Yukun Li, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren, Zehua Zhao, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhixian Huang, Zhiyu Wu, Zhuoshu Li, Zhuping Zhang, Zian Xu, Zihao Wang, Zihui Gu, Zijia Zhu, Zilin Li, Zipeng Zhang, Ziwei Xie, Ziyi Gao, Zizheng Pan, Zongqing Yao, Bei Feng, Hui Li, J. L. Cai, Jiaqi Ni, Lei Xu, Meng Li, Ning Tian, R. J. Chen, R. L. Jin, S. S. Li, Shuang Zhou, Tianyu Sun, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xinnan Song, Xinyi Zhou, Y. X. Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, Dongjie Ji, Jian Liang, Jianzhong Guo, Jin Chen, Leyi Xia, Miaojun Wang, Mingming Li, Peng Zhang, Ruyi Chen, Shangmian Sun, Shaoqing Wu, Shengfeng Ye, T. Wang, W. L. Xiao, Wei An, Xianzu Wang, Xiaowen Sun, Xiaoxiang Wang, Ying Tang, Yukun Zha, Zekai Zhang, Zhe Ju, Zhen Zhang, and Zihua Qu. DeepSeek-V3.2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*, 2025. URL https://arxiv.org/abs/2512.02556.

Charles Elkan. The foundations of cost-sensitive learning. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, volume 2, pp. 973–978, 2001. URL https://cseweb.ucsd.edu/~elkan/rescale.pdf.

Yushi Feng, Junye Du, Qifan Wang, Zizhan Ma, Qian Niu, Yutaka Matsuo, Long Feng, and Lequan Yu. CORA: Conformal risk-controlled agents for safeguarded mobile GUI automation. *arXiv preprint arXiv:2604.09155*, 2026. URL https://arxiv.org/abs/2604.09155.

Aditya Gangrade, Anil Kag, and Venkatesh Saligrama. Selective classification via one-sided prediction. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 2179–2187. PMLR, 2021. URL https://proceedings.mlr.press/v130/gangrade21a.html.

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/hash/4a8423d5e91fda00bb7e46540e2b0cf1-Abstract.html.

Yonatan Geifman and Ran El-Yaniv. SelectiveNet: A deep neural network with an integrated reject option. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2151–2159. PMLR, 2019. URL https://proceedings.mlr.press/v97/geifman19a.html.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao,

Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi (Jack) Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu (Sid) Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. URL https://arxiv.org/abs/2407.21783.

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. PMLR, 2024. URL https://proceedings.mlr.press/v235/greenblatt24a.html.

Weiwen Liu, Xu Huang, Xingshan Zeng, Xinlong Hao, Shuai Yu, Dexun Li, Shuai Wang, Weinan Gan, Zhengying Liu, Yuanqing Yu, Zezhong Wang, Yuxian Wang, Wu Ning, Yutai Hou, Bin Wang, Chuhan Wu, Xinzhi Wang, Yong Liu, Yasheng Wang, Duyu Tang, Dandan Tu, Lifeng Shang, Xin Jiang, Ruiming Tang, Defu Lian, Qun Liu, and Enhong Chen. ToolACE: Winning the points of LLM function calling. In *International Conference on Learning Representations*, volume 2025, pp. 41359–41381, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/663865ea167425c6c562cb0b6bcf76c7-Abstract-Conference.html.

Ziyin Liu, Zhikang Wang, Paul Pu Liang, Russ R. Salakhutdinov, Louis-Philippe Morency, and Masahito Ueda. Deep gamblers: Learning to abstain with portfolio theory. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/hash/0c4b1eeb45c90b52bfb9d07943d855ab-Abstract.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. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-naacl.65. URL https://aclanthology.org/2025.findings-naacl.65/.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M. Waleed Kadous, and Ion Stoica. RouteLLM: Learning to route LLMs from preference data. In *International Conference on Learning Representations*, volume 2025, pp. 34433–34448, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/5503a7c69d48a2f86fc00b3dc09de686-Abstract-Conference.html.

Shishir G. Patil, Huanzhi Mao, Fanjia Yan, Charlie Cheng-Jie Ji, Vishnu Suresh, Ion Stoica, and Joseph E. Gonzalez. The Berkeley Function Calling Leaderboard (BFCL): From tool use to agentic evaluation of large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 48371–48392. PMLR, 2025. URL https://proceedings.mlr.press/v267/patil25a.html.

Dzianis Piatrashyn, Nikita Kotelevskii, Kirill Grishchenkov, Nikita Glazkov, Ivan Nasonov, Ilya Makarov, Timothy Baldwin, Preslav Nakov, Roman Vashurin, and Maxim Panov. ReDAct: Uncertainty-aware deferral for LLM agents. *arXiv preprint arXiv:2604.07036*, 2026. URL https://arxiv.org/abs/2604.07036.

Akshara Prabhakar, Zuxin Liu, Ming Zhu, Jianguo Zhang, Tulika Manoj Awalgaonkar, Shiyu Wang, Zhiwei Liu, Haolin Chen, Thai Hoang, Juan Carlos Niebles, Shelby Heinecke, Weiran Yao, Huan Wang, Silvio Savarese, and Caiming Xiong. APIGen-MT: Agentic pipeline for multi-turn data generation via simulated agent-human interplay. In *Advances in Neural Information Processing Systems*, volume 38. Curran Associates, Inc., 2025. doi: 10.52202/085713-2184. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/5e3661f7fe4c8ac5652d62eb3d3c96ea-Abstract-Datasets_and_Benchmarks_Track.html.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. URL https://arxiv.org/abs/2412.15115.

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 *International Conference on Learning Representations*, volume 2024, pp. 27031–27098, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/7274ed909a312d4d869cc328ad1c5f04-Abstract-Conference.html.

Tal Shnitzer, Anthony Ou, Mírian Silva, Kate Soule, Yuekai Sun, Justin Solomon, Neil Thompson, and Mikhail Yurochkin. Large language model routing with benchmark datasets. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=Zb0ajZ7vAt.

Harsh Trivedi, Tushar Khot, Mareike Hartmann, Ruskin Manku, Vinty Dong, Edward Li, Shashank Gupta, Ashish Sabharwal, and Niranjan Balasubramanian. AppWorld: A controllable world of apps and people for benchmarking interactive coding agents. In *Proceedings of the 62nd Annual*

Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 16022–16076. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.850. URL https://aclanthology.org/2024.acl-long.850/.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik R. Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. In *International Conference on Learning Representations*, volume 2025, pp. 9965–10017, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/1b126cc38b8638e07bef37e7b2bb72bf-Abstract-Conference.html.

Zhexin Zhang, Shiyao Cui, Yida Lu, Jingzhuo Zhou, Junxiao Yang, Hongning Wang, and Minlie Huang. Agent-SafetyBench: Evaluating the safety of LLM agents. In *AAAI 2026 Workshop on Trustworthy Agentic AI*, 2026. URL https://trustagenticai.github.io/AAAI2026/AAAI-Workshop/30.pdf.

A Benchmark Construction and Evaluation Details

A.1 τ-bench Decision-Point Construction

Our primary experiments use the released τ-bench retail and airline tasks. For each task, we replay the ground-truth action prefix to reconstruct the environment state immediately before a target tool decision, then ask the evaluated SLM to propose the next action from that shared state. This construction isolates the selective-commitment decision from stochastic variation in preceding dialogue or user-simulator behavior during decision-point evaluation under controlled conditions.

All decision points belonging to the same task are assigned to the same split. We therefore partition by task identifier rather than by individual tool call. The default split uses 50% of tasks for training, 25% for calibration, and 25% for held-out evaluation under identical task-level splits.

For each decision point we record:

- the task and environment state;
- the SLM-proposed tool call;
- the benchmark target action or resulting-state criterion;
- the fallback proposal;
- the action-effect class;
- action-consequence metadata;
- correctness labels for both the SLM and fallback actions.

The fallback output is generated once per decision point and cached for offline policy comparison. Consequently, all routing methods face the same counterfactual fallback action, while the routing decision itself uses only quantities available before fallback invocation.

A.2 Cross-Environment Evaluation

Additional benchmarks with complete adapters and saved results are mapped into the same abstract decision interface:

$$(x_t, a_t^S, a_t^L, F_t^S, F_t^L, \rho_t^S, \rho_t^L, C_t^S, C_t^L). \tag{12}$$

Environment-specific details differ in how action correctness and persistent state effects are determined. Each adapter preserves the benchmark's native task semantics rather than redefining correctness through a common textual matching criterion during evaluation.

B Action Consequence Specification

B.1 Effect Classes

Each tool action is assigned to one of four operational effect classes:

$$\mathcal{E} = \{\text{read-only}, \text{reversible-write}, \text{irreversible-external}, \text{destructive-local}\}. \tag{13}$$

The default irreversibility weights are assigned by effect class as follows:

$$\rho(e) = \begin{cases} 0, & e = \text{read-only}, \\ 0.25, & e = \text{reversible-write}, \\ 0.75, & e = \text{irreversible-external}, \\ 1, & e = \text{destructive-local}. \end{cases} \tag{14}$$

These values provide an ordinal representation of persistence rather than a claim that the numerical intervals have universal semantic meaning. We use the same specification across policies so that comparisons depend on intervention allocation rather than policy-specific consequence definitions used in evaluation.

B.2 Effect Label Assignment

Effect labels come from explicit benchmark or tool metadata when supplied. Otherwise, deterministic schema rules classify actions by whether a tool only retrieves state, mutates persistent records, or admits an inverse operation in the target environment for each benchmark.

The learned irreversibility estimator receives tool and action semantics as input. Oracle irreversibility experiments replace the learned estimate with the ground-truth effect label while keeping all other components unchanged for direct comparison.

B.3 Severity

The commitment-risk formulation permits an application-dependent severity weight C_t:

$$\widehat{R}_t = \hat{p}_t \hat{\rho}_t C_t. \tag{15}$$

The primary experiments use a uniform severity setting for every action:

$$C_t \equiv 1 \tag{16}$$

for all domains and effect classes. Action-dependent consequence weighting in the primary controller therefore arises from the irreversibility term $\hat{\rho}_t$. Section J evaluates several nonconstant severity schedules in the reported sensitivity study presented here.

C Failure and Irreversibility Estimators

C.1 Failure Estimation

The failure estimator predicts the probability of an unsuccessful SLM action as

$$\hat{p}_t^S = P(F_t^S = 1 \mid z_t), \tag{17}$$

where z_t combines uncertainty statistics and a representation of the SLM state at the decision point.

The uncertainty feature vector includes quantities derived from the action distribution, such as negative log likelihood, entropy, and related confidence statistics available before execution. The representation features are obtained from the frozen SLM, reduced to a lower-dimensional space, and then passed to the prediction head used by the classifier for held-out evaluation.

The estimator is trained only on the training partition. No calibration or test labels are used to fit its parameters for any evaluation split.

C.2 Fallback Failure Estimation

Fallback failure risk is estimated from the available training data as follows:

$$\hat{q}_t^L = \widehat{\Pr}\left(F^L = 1 \mid D_t, E_t^S\right), \tag{18}$$

where D_t is the current domain and E_t^S is the effect class of the SLM proposal. In practice, $\hat{q}_t^L$ is the empirical fallback error rate within each (D_t, E_t^S) stratum on the training split.

This estimate is available before escalation. The fallback proposal itself is generated only after the controller chooses to defer. Cached fallback outputs are used for offline evaluation and do not enter the routing score used by any policy at decision time.

C.3 Irreversibility Estimation

The irreversibility estimator predicts the effect class of a proposed tool action from tool name, schema information, action arguments, and other environment-provided semantics available before execution. The resulting class posterior is mapped to an expected irreversibility weight.

Because effect type is often highly structured in modern tool APIs, this component reaches high held-out accuracy. The oracle study in Section F evaluates whether residual end-to-end errors are due to imperfect consequence estimation or failure recognition in the evaluated settings.

D Calibration Procedure

D.1 Residual Commitment Harm

For a thresholded policy π_τ, residual commitment harm is defined by

$$\mathcal{R}(\pi_\tau) = \mathbb{E}\left[H_t \mathbf{1}\left[\widehat{A}_t \leq \tau + c_t^L\right]\right]. \tag{19}$$

Calibration selects a threshold using only the calibration partition. The selected threshold is then frozen before evaluation on held-out tasks without any later adjustment or access to test labels.

D.2 Risk Targets

We evaluate target risk levels at four calibrated operating points:

$$\alpha \in \{0.010, 0.025, 0.050, 0.100\}. \tag{20}$$

For each α, the calibration procedure selects the most permissive threshold satisfying the empirical risk-control criterion used by the primary method. This yields the highest autonomous coverage among candidate thresholds that satisfy the target on calibration data without consulting test labels.

D.3 Stratified Calibration

The primary policy calibrates by domain and action-effect stratum because base failure rates and consequence distributions differ across read-only and state-changing actions in the calibration data.

We compare this procedure against a single global threshold. The stratified version is used only when each calibration stratum contains sufficient support; otherwise the corresponding categories are pooled according to the predefined fallback rule used by the deployment controller.

E Full Baseline Definitions

SLM-only. Executes every proposal from the small language model.

Fallback-only. Invokes the fallback at every decision point.

Random. Escalates a uniformly sampled subset of decisions at the target budget.

NLL routing. Ranks decisions by the negative log likelihood of the selected action.

Entropy routing. Ranks decisions by entropy of the action distribution.

Self-reflection routing. Uses the model's explicit self-assessment of whether its proposed action requires additional verification.

ReDAct. Uses calibrated predictive uncertainty to defer selected decisions from a smaller agent to a larger model (Piatrashyn et al., 2026).

SABER. Applies mutation-gated verification and targeted reflection before environment-changing actions (Cuadron Lafuente et al., 2026).

CORA. Uses a calibrated post-policy action-risk score to select which proposed actions require intervention (Feng et al., 2026).

Learned failure-risk gate. Ranks actions only by

$$\hat{p}_t^S. \tag{21}$$

This baseline isolates failure prediction from the consequence terms in the full controller.

Commitment-risk gate. Ranks actions by

$$\hat{p}_t^S \hat{\rho}_t^S C_t^S \tag{22}$$

without subtracting fallback risk from the routing score.

Mutation gate. Prioritizes actions identified as persistent state changes without using model-specific failure likelihood at decision time.

TRIPWIRE. Ranks decisions by

$$\widehat{A}_t = \left(\hat{p}_t^S - \hat{q}_t^L\right) \hat{\rho}_t^S C_t^S + \gamma \widehat{G}_t. \tag{23}$$

The primary configuration uses $C_t^S = 1$ and $\gamma = 1$.
For fixed budgets, every ranking-based method is thresholded to the same overall escalation rate.

F Oracle and Structural Ablations

F.1 Oracle Irreversibility

We replace $\hat{\rho}_t$ with the ground-truth irreversibility value while leaving the learned failure estimator and fallback comparison unchanged. This experiment measures the maximum improvement available from eliminating irreversibility-estimation error in the evaluated setting.

F.2 Oracle Failure

We replace $\hat{p}_t^S$ with the binary ground-truth failure indicator for the SLM proposal. All consequence terms and fallback quantities remain unchanged. This experiment provides an upper bound on improvements obtainable from better failure recognition under the same commitment-risk formulation.

F.3 No Fallback Comparison

To isolate the value of avoidable commitment risk, we remove the fallback term and rank actions only by autonomous commitment risk:

$$\widehat{A}_t^{\text{no fallback}} = \widehat{R}_t^S. \tag{24}$$

The difference between this variant and TRIPWIRE measures whether accounting for intervention effectiveness improves budget allocation beyond identifying risky actions under the matched budget.

F.4 No Irreversibility Weighting

Under the unit-severity primary configuration, removing action-consequence weighting amounts to removing the irreversibility factor:

$$\widehat{A}_t^{\text{failure}} = \hat{p}_t^S - \hat{q}_t^L. \tag{25}$$

This comparison isolates the contribution of action irreversibility beyond failure prediction and fallback failure risk used by the controller during evaluation.

F.5 No Downstream-Value Correction

The primary implementation augments immediate avoidable commitment risk with the downstream-value term $\gamma \widehat{G}_t$, using $\gamma = 1$. This ablation sets $\gamma = 0$ while leaving the remaining components unchanged relative to the primary score used in the main experiments.

Oracle failure provides the largest improvement, reducing realized harm from 0.031 to 0.013. In contrast, oracle irreversibility yields no additional improvement. Removing the fallback comparison increases realized harm to 0.039, while removing irreversibility weighting increases it to 0.044. Removing the downstream-value correction leaves realized harm unchanged at 0.031. Together, these results identify failure recognition as the dominant source of remaining headroom in the evaluated setting considered here.

Table 4: Oracle and structural ablations. Realized and committed harm on τ-bench with Qwen3-8B and a DeepSeek-V3.2 fallback at a 15% escalation budget. Lower is better. Bold and underline indicate the best and second-best variants; ties receive the same marking across both reported harm columns.

Variant	Realized harm	Committed harm	Escalation
TRIPWIRE	<u>0.031</u>	<u>0.027</u>	15%
Oracle irreversibility	<u>0.031</u>	<u>0.027</u>	15%
Oracle failure	**0.013**	**0.002**	15%
No fallback comparison	0.039	0.035	15%
No irreversibility weighting	0.044	0.039	15%
No downstream-value correction	<u>0.031</u>	<u>0.027</u>	15%

Table 5: Failure-prediction feature ablation. AUROC is evaluated on held-out τ-bench tasks with Qwen3-8B under the dedicated feature-ablation configuration. Bold and underline indicate the best and second-best feature sets on both action subsets.

Features	All actions	Consequential actions
NLL only	0.612	0.601
Entropy only	0.598	0.587
Uncertainty vector	0.631	0.619
Internal representation	<u>0.683</u>	<u>0.671</u>
Representation + uncertainty	**0.695**	**0.684**

G Failure-Prediction Analysis

We compare the following five feature sets for predicting SLM action failure:

- negative log likelihood only;
- entropy only;
- scalar uncertainty features (NLL, entropy, and token-level confidence statistics);
- internal representation only (dimension-reduced hidden states from the frozen SLM);
- internal representation plus uncertainty features.

Performance is evaluated by held-out AUROC on all decision points and separately on consequential actions with nonzero irreversibility. The consequential subset is particularly relevant because errors on read-only actions contribute little or no commitment harm at the same decision points.

Representation-based features consistently outperform uncertainty-only features and retain a clear advantage on consequential actions with nonzero irreversibility. This supports the use of internal representations for failure estimation in TRIPWIRE. Combining representation and uncertainty gives the strongest overall performance, suggesting that the two sources provide complementary information about agent errors before action execution.

H Risk–Budget Sweep

For fixed-budget experiments we sweep escalation rates from 2.5% to 30%. At each budget, each ranking-based policy escalates the highest-ranked decisions according to its own intervention score.

We report realized weighted irreversible harm as the escalation budget varies. This sweep avoids dependence on a single operating point and evaluates whether TRIPWIRE dominates alternative allocation rules across all intervention budgets considered in this evaluation.

At the lowest escalation budget (2.5%), all selective methods achieve only modest harm reduction because most decisions remain autonomous. As the budget increases, TRIPWIRE's advantage grows: allocating intervention by avoidable commitment risk yields substantially lower harm than uncertainty-based or consequence-only routing. At the primary 15% operating point, TRIPWIRE achieves realized harm 0.039, compared with 0.085 for mutation gating and 0.112 for NLL routing.

The rate of improvement also differs across methods. TRIPWIRE exhibits a steep decline between 5% and 15% escalation, from 0.090 to 0.039, while NLL routing remains nearly flat across the sweep. By 30% escalation, TRIPWIRE reaches realized harm 0.027, approaching the fallback-only reference of 0.024 while using only 30% of the fallback calls required by fallback-only.

Table 6: Fixed-budget risk–coverage sweep. Realized weighted irreversible harm averaged across seven SLM backbones on τ-bench at each escalation budget. Lower is better. Fallback-only achieves 0.024 at 100% escalation. Bold and underline indicate the best and second-best selective methods within each matched-budget column; SLM-only and fallback-only are reference endpoints.

Method	2.5%	5.0%	7.5%	10.0%	15.0%	20.0%	30.0%
SLM-only	0.114	0.114	0.114	0.114	0.114	0.114	0.114
NLL routing	0.113	0.113	0.112	0.112	0.112	0.111	0.110
Failure-risk gate	0.112	0.109	0.105	0.100	0.096	0.093	0.088
Oversight	0.111	0.107	0.102	0.097	0.091	0.087	0.082
Mutation gate	<u>0.109</u>	<u>0.103</u>	<u>0.096</u>	<u>0.091</u>	<u>0.085</u>	<u>0.080</u>	<u>0.074</u>
TRIPWIRE	**0.107**	**0.090**	**0.072**	**0.055**	**0.039**	**0.033**	**0.027**
Fallback-only (100%)				0.024			

Table 7: Calibration diagnostics. Thresholds are selected without access to test labels. Results are averaged over the seven SLM backbones on τ-bench under the shared evaluation protocol reported here.

α	Threshold	Coverage	Cal. harm	Test harm	Target met	Δ (Test $-$ Target)
0.010	0.48	0.42	0.008	0.009	yes	-0.001
0.025	0.61	0.58	0.021	0.023	yes	-0.002
0.050	0.73	0.76	0.043	0.045	yes	-0.005
0.100	0.82	0.88	0.089	0.045	yes	-0.055

I Calibration Diagnostics

For every target α, we report the calibration threshold, autonomous coverage, calibration-set harm, test-set harm, and whether the held-out target is satisfied.

The table shows that the calibration procedure selects thresholds that satisfy the target risk level at all evaluated α values. Autonomous coverage increases monotonically with the target risk level, ranging from 42% at $\alpha = 0.010$ to 88% at $\alpha = 0.100$. Test harm remains below the target α in all cases. It is slightly above calibration harm for the three lower-risk targets and lower at $\alpha = 0.100$, where the per-backbone results in Table 2 average to approximately 0.045 on held-out tasks.

We additionally report results separately by backbone, domain, and effect class to identify cases in which aggregate calibration can hide distribution-specific deviations. Stratified calibration reduces the per-stratum violation rate from 17% (global calibration) to 0% across all domains and effect classes, confirming that domain-specific thresholds improve risk control when action distributions differ substantially across strata under the same calibration protocol.

J Severity Sensitivity

The primary experiments use unit severity, $C_t = 1$. We also test nonconstant, effect-dependent severity schedules to determine whether the policy ordering changes when consequence classes receive different relative costs while preserving their ordinal ranking across operating points.

For each specification, all methods are reevaluated using the same held-out actions and fallback proposals for a controlled comparison under shared inputs.

We consider the following three alternative severity parameterizations:

- **Compressed**: $C \in \{1.0, 1.5, 2.5, 4.0\}$ (reduced spread)
- **Expanded**: $C \in \{1.0, 2.0, 5.0, 10.0\}$ (increased spread)
- **Sparse**: $C \in \{1.0, 1.0, 5.0, 5.0\}$ (coarse grouping)

In all three specifications, TRIPWIRE remains the strongest selective method, with realized harm reductions of 61–68% relative to SLM-only at 15% escalation. The ordering of methods is stable across specifications: TRIPWIRE > Mutation Gate > Oversight > Failure-Risk Gate > NLL Routing. These results show that the empirical advantage of avoidable commitment risk persists under several nonconstant consequence-cost schedules evaluated here.

K Sequential Evaluation

The primary decision-point experiments hold the pre-action environment state fixed across policies. We additionally evaluate the subset with complete trajectory-level artifacts using full sequential

Table 8: Cross-environment dataset statistics. The τ^2-bench count uses the official 114-task Telecom evaluation suite, not the separate 2,285-task generated Telecom pool. All environments use Qwen3-8B as the SLM and DeepSeek-V3.2 as the fallback in every reported evaluation.

Benchmark	Dataset size	Unit and scope	Native correctness
τ-bench	165	tasks (115 retail, 50 airline)	state-based
τ^2-bench	279	tasks (115 retail, 50 airline, 114 telecom)	state-based
AppWorld	750	tasks (released benchmark)	outcome-based
BFCL-v4 multi-turn	800	test cases (official multi-turn component)	target-match
ToolEmu	144	test cases (36 toolkits)	emulated

trajectories in which the action selected by the controller changes the state observed at subsequent steps in each trajectory under controlled initialization.

For each complete evaluation trajectory, we record the following quantities:

- final task success;
- number of fallback invocations;
- cumulative realized irreversible harm;
- number of irreversible failures;
- total inference cost;
- normalized cost per successful trajectory.

The fallback budget is applied throughout the trajectory rather than independently to each isolated step. This experiment tests whether the local commitment decision remains useful when intervention choices alter the future state distribution under sequential execution.

The sequential evaluation uses the same τ-bench tasks and the same Qwen3-8B / DeepSeek-V3.2 configuration as the main experiments, but with policies executed end-to-end over complete trajectories. Results are averaged over five fixed random seeds, yielding 825 task-seed evaluations per policy. Table 3 in the main text reports the aggregate results for direct comparison.

Across the sequential trajectories, normalized cost per success is 1.74 for TRIPWIRE, compared with 1.00 for SLM-only and 6.71 for fallback-only. The cost advantage over fallback-only comes from TRIPWIRE's selective invocation: only 12% of decisions use the larger model, while the remainder use the cheaper SLM. The sequential policy realizes a 12% escalation rate, slightly below the 15% operating point used in the isolated decision-point evaluation under the matched protocol.

L Cross-Environment Details

For each additional benchmark with complete experimental support, we document:

- the native unit of evaluation;
- how action correctness is determined;
- which actions modify persistent state;
- how action-effect classes and consequence costs are assigned;
- the available SLM and fallback models;
- the number of training, calibration, and test examples;
- any benchmark-specific filtering.

We do not assume identical effect-label semantics across environments. Instead, each environment maps native tool effects to the common commitment-risk interface in Section B.

The effect-class distribution varies substantially across environments: the BFCL-v4 multi-turn component has a higher proportion of read-only actions (62%) than τ-bench (38%), while ToolEmu has the highest proportion of destructive-local actions (18%). This diversity makes the cross-environment evaluation a meaningful test of TRIPWIRE's generalizability. The consistent advantage observed across all five environments (Figure 4) suggests that the commitment-risk formulation is robust to variations in environment structure and consequence distribution across all benchmarks reported here.

M Statistical Inference

Confidence intervals for aggregate policy metrics use task-level bootstrap resampling, which preserves dependence among decision points from the same evaluation trajectory.

For paired comparisons, policies are evaluated on identical held-out tasks and cached fallback proposals. Paired randomization tests are used for differences in realized harm and related task-level metrics reported in the corresponding table.

We emphasize effect sizes and risk–budget curves in the main text rather than relying on significance tests alone when comparing policies across operating points.

For the main aggregate results (Table 1 in the main text), 95% bootstrap confidence intervals are computed over 1,000 resamples. The intervals for TRIPWIRE's realized harm are $[0.032, 0.047]$, which do not overlap with the intervals for any baseline ($[0.082, 0.104]$ for mutation gate, $[0.107, 0.118]$ for NLL routing). The paired randomization tests against uncertainty-only routing in Table 2 are significant at the 0.05 level for five of seven backbones. The two exceptions are xLAM-2-8B ($p = 0.0630$) and Qwen3-14B ($p = 0.8650$).

N Implementation and Reproducibility

The SLM backbones remain frozen. We train only the lightweight SLM failure and irreversibility estimators. Fallback failure rates are estimated empirically on the training split by domain and SLM effect class, and the primary severity cost is fixed to one. We use five fixed random seeds and report averages across runs. Within each run, data splits are stored by task identifier before evaluation.

Fallback responses are cached for offline policy evaluation. This prevents API stochasticity or repeated sampling from changing the counterfactual action available to different routing methods; cached fallback responses are not used to make routing decisions.

The supplementary code contains:

- environment adapters;
- feature extraction;
- failure and irreversibility estimators;
- commitment-risk computation;
- calibration routines;
- baseline implementations;
- evaluation scripts;
- bootstrap and randomization procedures;
- unit tests for the core policy and risk calculations.

All reported tables and figures are generated from saved experiment outputs rather than being entered manually in the manuscript by the authors, and are checked against the corresponding evaluation artifacts used to produce each reported result and verify numerical consistency.