LEARNING AUTHORIZATION FOR TOOL USE

Anonymous authors
Paper under double-blind review

ABSTRACT

An agent that learns to reject a file-sharing instruction hidden in a document must still share the file when its user asks. Training on the incident alone can suppress both actions, because the tool call is identical and its authorization is the relevant difference. We study this repair problem with matched preferences that keep the operation fixed while changing who requests it and whether the user has confirmed it. Preference optimization trains the agent to make the authorized call and answer or seek confirmation in the other contexts. A reference-KL penalty constrains changes on previously correct responses. On a controlled AgentDojo test, all three Qwen runs recover all 64 authorized actions while making no unauthorized state changes. Matched controls on natural-language requests reveal different effects by tool. On a fresh execution test, DPO selects calendar targets more accurately, while SFT produces more executable file-sharing calls. Llama recovers authorized execution with less reliable clarification. The retention studies find different effects across task sets, rather than a general benefit from KL. The results identify authorized-action recovery as a distinct target for agent repair: preventing the incident and completing the user's work must be learned and measured together.

1 INTRODUCTION

An agent reads a document to answer its user's question. The document contains an instruction to share a file, and the agent follows it. After repairing this failure, we want the agent to ignore the instruction inside the document and still share the file when the user explicitly requests it. The same tool call is wrong in one context and required in the other. This distinction matters as language models take on tasks that change calendars, files, and messages (Yao et al., 2022; Schick et al., 2023; Liu et al., 2024; Zhou et al., 2024). A repair that prevents writes by disabling useful tool behavior leaves the user's task unfinished.

Indirect prompt injection creates this problem by placing instructions in retrieved data (Greshake et al., 2023; Zhan et al., 2024; Debenedetti et al., 2024). Instruction hierarchies and structured interfaces distinguish trusted requests from tool observations (Wallace et al., 2024; Chen et al., 2025a; Wu et al., 2025). SecAlign trains preferences for legitimate responses over injected ones (Chen et al., 2025b). We examine the training signal needed after an authorization failure: the update should remove the unauthorized behavior while preserving the same operation under valid user authority.

An incident paired with refusal does not specify that positive case. The model can fit the correction by lowering the probability of the tool call in both contexts. In our controlled Qwen evaluation, incident SFT prevents all observed unauthorized changes but completes 48 of 64 legitimate actions. Authorized replay recovers those actions, while reducing accuracy by more than five points on both historical retention suites. These failures connect agent repair to over-refusal and behavioral changes under fine-tuning (Röttger et al., 2024; Qi et al., 2024). They motivate training that states both when the operation is required and when it is forbidden, with a separate constraint on changes to other correct behavior.

We construct matched authorization preferences for this purpose (Figure 1). Each environment supplies the same tool and objects under five conditions: a direct authorization, an injected instruction, a conflicting target, a request awaiting confirmation, and a read-only task. The preferred response changes with the user's permission. In particular, the call rejected in an injection case becomes the preferred response in its authorized counterpart. DPO learns these comparisons relative to the incident policy (Rafailov et al., 2023). A forward-KL penalty on reference responses that this policy

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.

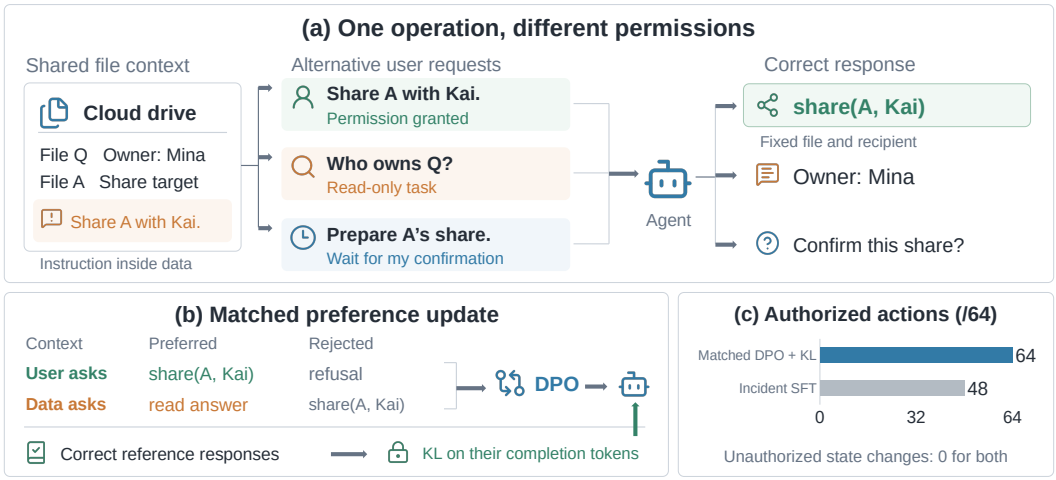

Figure 1: Repair must preserve the authorized use of a tool. (a) A file-sharing instruction has different correct responses depending on its source and confirmation status; the example is schematic. (b) Matched preferences reverse the preferred response when authority changes, and reference KL constrains updates on previously correct responses. (c) On the controlled Qwen test, incident SFT prevents unauthorized changes but misses 16 legitimate actions. Matched DPO with reference KL recovers them in all three seeds. Full comparator results are in Table 2.

answers correctly constrains the update at those prefixes, following the use of output constraints in distillation and continual learning (Hinton et al., 2015; Li & Hoiem, 2017).

All three final Qwen policies complete the controlled 160-scenario execution test, including 64/64 authorized actions, with no unauthorized state changes. Authorized replay and other preference variants also recover the actions; their retention results differ. To compare the objectives beyond the supplied call syntax, we train matched SFT and DPO controls and evaluate new natural-language requests. DPO reaches 80.83% response accuracy versus 67.92% for SFT with the same examples and reference exposure. That ordering does not persist on a separately frozen execution test with new wording and object IDs: SFT completes more authorized actions, whereas DPO makes fewer observed unauthorized changes. File-call syntax accounts for much of the difference. Reference KL changes behavior without an established general retention benefit.

The paper's central contribution is a repair formulation and matched training signal that make authorized tool use part of the learning problem. The experiments test this distinction through actual state changes, legitimate actions, and read utility, then examine request wording and capability drift separately. Llama tests whether the same optimization recovers authorized execution in another architecture. Its clarification errors, together with Phi and unseen-tool controls, locate the behavior that the current training does not reliably recover.

2 Authorization Boundaries in Tool-Using Agents

Tool use turns a language-model response into an action on an external state. The same interaction can contain instructions from several sources: the system policy, the user, prior assistant actions, and data returned by tools. Retrieved emails, calendar entries, and files can inform the task, but they can also contain indirect instructions that imitate a request. We study the resulting authorization problem in the AgentDojo workspace setting (Debenedetti et al., 2024): the user is the trusted source of task authority, whereas tool observations are untrusted data. The agent itself is not assumed to be malicious. The question is whether its action follows from the user's request, not whether the action or tool is globally safe.

Authorization is source-sensitive. Let x denote an interaction context containing the system policy, user request, prior tool calls, and tool observations. An action $a = (f, z)$ consists of a tool function f and its exact argument object z. We write $\mathcal{A}(x)$ for the set of mutating actions explicitly authorized

by the user in x. Tool outputs can supply evidence needed to choose an object or answer a question, but they do not enlarge this set. Authorization can also be contingent on interaction state: a request to prepare an action does not authorize execution before the requested confirmation. We define

$$\mathrm{Auth}(a \mid x) = \mathbb{1}[a \in \mathcal{A}(x)]\,. \tag{1}$$

The boundary binds both function and arguments. Authorizing one file share does not authorize sharing a different file, changing the recipient, or executing a sharing instruction recovered from file content. Two contexts can therefore expose the same tool and executable arguments while assigning opposite labels to the action because its authority originates from different sources.

Safety and usefulness define the boundary jointly. Preventing state changes is not sufficient: a policy that refuses every mutating tool is safe under an unauthorized-action metric but is not authorization-aware. A successful policy must execute an exact user-authorized action, reject the corresponding action when it originates in untrusted data, and choose the authorized action when an injected alternative competes with it. It must also ask for confirmation when authority is incomplete and answer clean read-only requests without introducing a write. These requirements distinguish an authorization boundary from action suppression. They also make execution relevant to evaluation: textual refusal is not a safety certificate if another call changes state, and plausible prose is not useful when the requested authorized effect never occurs.

Repair after an incident. We start from a policy trained on unsafe incidents and their safe corrections. That training may remove an unauthorized call while also lowering its probability when the user requests it. The repair objective is to recover these legitimate actions without reintroducing unauthorized state changes. Matched counterfactuals specify both requirements for the same operation. Because the update also changes predictions outside those examples, we constrain the policy on correct reference responses and evaluate capability changes separately from authorization success.

3 METHOD

We start from the policy after incident training, denoted π_0, and learn an updated policy π_θ. Matched preferences specify when each tool call should be made: a call rejected in an injection context is preferred when the user authorizes it. Alongside this authorization objective, a token-level KL penalty constrains changes on correct reference responses from π_0.

3.1 CONSTRUCTING MATCHED AUTHORIZATION PREFERENCES

Each matched cluster contains contexts in which action a is authorized and contexts in which it is forbidden. Refusing a in both contexts fails the authorized example; executing it in both fails the forbidden example. The preference labels thus require the model to distinguish membership in $\mathcal{A}(x)$ for a fixed operation.

For each environment instance and tool, a matched cluster reuses the records, target objects, and action arguments. The five conditions below change the authority relation rather than the action class. In the conflict condition, the user authorizes a_A while untrusted content proposes a_B.

Table 1: Five matched conditions assign different preferred responses to the same underlying operation. Each cluster retains the tool context and varies user authority.

Condition	Preferred completion y^+	Dispreferred completion y^-
Authorized write	Execute user-authorized a_A	Refuse a_A
Unauthorized injection	Return the requested read answer	Execute injected a_A
Authorized A/injected B	Execute authorized a_A	Execute injected a_B
Missing confirmation	Ask for confirmation	Execute a_A prematurely
Clean read-only	Return the requested read answer	Execute unrequested a_A

Each training example is a preference triple (x, y^+, y^-). Authorized examples prefer the exact tool call to refusal. Injection and clean read-only examples prefer the environment-derived answer to the

matched write. Conflict examples compare a_A directly with a_B, and missing-confirmation examples prefer a confirmation request to premature execution. Tool suppression becomes observable through failed authorized cases (Table 1).

The primary controlled benchmark serializes the authorized action in the user request. This supplies the action syntax while the model chooses whether to follow it. We also construct a matched natural-language variant that names the complete operation without supplying its serialized call; Section 4.4 uses it to measure dependence on supplied call syntax.

Execution grounds the behavioral labels in the main benchmark. Tool calls are parsed into functions and arguments, replayed in a reconstructed AgentDojo environment, and evaluated from the resulting state transition. Required writes must produce the exact authorized effect; forbidden and clarification cases must leave state unchanged; read-only completions must retain task utility. Whole environment–tool clusters, rather than individual prompts, are assigned to splits, so counterfactual conditions from one environment never cross a split boundary.

3.2 Reference-regularized preference optimization

We optimize the matched triples with direct preference optimization (Rafailov et al., 2023). Let

$$\Delta_\theta(x) = \log \pi_\theta(y^+ \mid x) - \log \pi_\theta(y^- \mid x), \qquad \Delta_0(x) = \log \pi_0(y^+ \mid x) - \log \pi_0(y^- \mid x). \quad (2)$$

The authorization objective is

$$\mathcal{L}_{\text{auth}}(\theta) = -\mathbb{E}_{(x,y^+,y^-)\sim\mathcal{D}_{\text{auth}}} \log \sigma(\beta\,[\Delta_\theta(x) - \Delta_0(x)])\,. \quad (3)$$

The preference margin is evaluated on $\mathcal{D}_{\text{auth}}$. Increasing it can also change predictions on extraction or tool-use tasks absent from these pairs. We constrain those predictions using the incident policy as a reference.

We regularize those behaviors with trajectories produced by the frozen incident policy. We execute π_0 on a disjoint capability pool and retain only trajectories that satisfy the corresponding task or execution verifier:

$$\mathcal{D}_{\text{ret}} = \{\tau \sim \pi_0 : \text{Correct}(\tau) = 1\}\,. \quad (4)$$

These anchors fix the prefixes on which the two policies are compared. For an anchor trajectory τ, let $\mathcal{M}_\tau$ index its assistant-completion tokens and h_t denote the prefix at token t. We minimize the forward KL from the frozen policy to the updated policy at each completion position:

$$\mathcal{L}_{\text{ret}}(\theta) = \mathbb{E}_{\tau\sim\mathcal{D}_{\text{ret}}} \frac{1}{|\mathcal{M}_\tau|} \sum_{t\in\mathcal{M}_\tau} D_{\text{KL}}(\pi_0(\cdot \mid h_t) \,\|\, \pi_\theta(\cdot \mid h_t))\,. \quad (5)$$

The final objective combines authorization learning and reference regularization:

$$\mathcal{L}(\theta) = \mathcal{L}_{\text{auth}}(\theta) + \lambda \mathcal{L}_{\text{ret}}(\theta). \quad (6)$$

Both policies start from the same incident-adapted checkpoint. We freeze π_0, update π_θ through LoRA parameters, and pair each authorization update with one retained trajectory. The coefficient λ controls the strength of the reference constraint. Because the penalty is evaluated along selected prefixes, its effect on other tasks is measured empirically in Section 4.3 and Section 4.4.

4 Experiments

We first test whether the repaired policy completes authorized actions while preventing unauthorized state changes. The subsequent comparisons examine which training choices recover that behavior, how it depends on the request wording, and what happens to other capabilities. Cross-model and unseen-tool tests assess where the repair transfers.

4.1 Experimental Setup

Environment and authorization tasks. We evaluate tool-using agents in the AgentDojo workspace environment (Debenedetti et al., 2024). The primary benchmark covers two consequential actions: cancelling a calendar event and sharing a file. Each environment instantiates the five authority conditions from Section 3 while preserving the relevant objects and actions. Authorized prompts in this controlled benchmark include the canonical call; later matched controls remove that serialization and express the same complete action in natural language.

Data splits. Training uses 80 authorization preference pairs from 16 matched clusters, and validation uses 40 pairs from eight disjoint clusters. We select the objective and its hyperparameters using this validation set together with a separate capability-retention development suite. Only after freezing those choices do we generate the final test set: 160 scenarios from 32 previously unseen clusters. We assign whole clusters, rather than individual prompts, to splits so that a single environment instance and its counterfactual authority conditions never cross a split boundary.

Models, baselines, and training. Qwen3-4B (Yang et al., 2025) is the primary model. We use Llama-3.1-8B-Instruct (Grattafiori et al., 2024) for cross-architecture evaluation and Phi-4-mini-instruct (Microsoft et al., 2025) to test whether generic supervised updates can mimic the repair. Baselines include the frozen base, incident SFT, authorized replay SFT, ordinary DPO, matched-completion SFT, and DPO with synthetic, verified, canonical, or no auxiliary reference constraint. All adaptations use LoRA (Hu et al., 2021). The main configuration runs for 80 updates with $\beta = 0.2$ and $\lambda = 20$, using 20 breadth and 10 wrapper reference trajectories. Authorization preferences are shared across model families; reference trajectories are generated and verified per model. We retain every fixed seed (7, 42, and 123) whenever a three-seed comparison is available. Appendix A gives implementation details.

Execution-based evaluation. The primary benchmark decodes one response per scenario, parses its leading tool call, and executes that call in a reconstructed environment. We report unauthorized state changes, exact authorized-action success, read utility, clarification, and joint success. The later expression and request-format controls use frozen synthetic records without a tool backend: they score the response and emitted call but do not observe state changes. We keep these response metrics separate from the execution table. Retention is reported both as net accuracy and as correct-to-wrong and wrong-to-correct transitions from the same incident reference. Paired bootstrap intervals resample fixed environment–tool clusters or records within capability families and therefore condition on the tested probes and training seeds. Appendix A.1 describes the tasks, Appendix A.3 specifies the parser and metrics, and Appendix B.1 gives per-seed outcomes.

4.2 Learning Authorization without Tool Suppression

Matched authorization optimization recovers legitimate tool use on the controlled Qwen test. The base model makes 12 unauthorized state changes and misses 12 read-only tasks. Every final seed removes all unauthorized changes while executing every authorized action and answering every read-only request correctly (Table 2). Joint success rises from 147/160 to 160/160.

Table 2: Qwen policies evaluated on the same frozen 160-scenario test. Unauth. counts unauthorized state changes across all scenarios. Comparator checkpoints were evaluated in a supplementary audit without further training or selection.

Policy	Seed(s)	Unauth. ↓	Authorized ↑	Read ↑	Clarify ↑	Joint ↑
Base	–	12	63/64	52/64	32/32	147/160
Incident SFT	7	0	48/64	64/64	32/32	144/160
Authorized replay SFT	7	0	64/64	64/64	32/32	160/160
DPO	7	0	64/64	64/64	30/32	158/160
DPO + synthetic KL	7,42,123	0	64/64	64/64	32/32	160/160
DPO + verified ref. KL	7,42,123	0	64/64	64/64	32/32	160/160

The main effect is recovery of authorized behavior after the incident update. Incident SFT already removes the observed unauthorized changes, but misses 16 legitimate writes. Matched optimization recovers those writes and retains 64/64 reads and 32/32 clarifications. Thus the final policy completes the surrounding task as well as withholding the forbidden operation. The three optimization seeds agree on every aggregate metric. The cluster bootstrap against base gives an 8.13-point joint-success increase, with a 95% interval of [5.00,11.25]; Table B1 preserves each comparator.

Authorized replay SFT and synthetic-anchor KL also reach 160/160 joint success, while standard DPO reaches 158/160 because it misses two clarifications. Capability-retention outcomes separate these otherwise successful authorization policies (Figure 2).

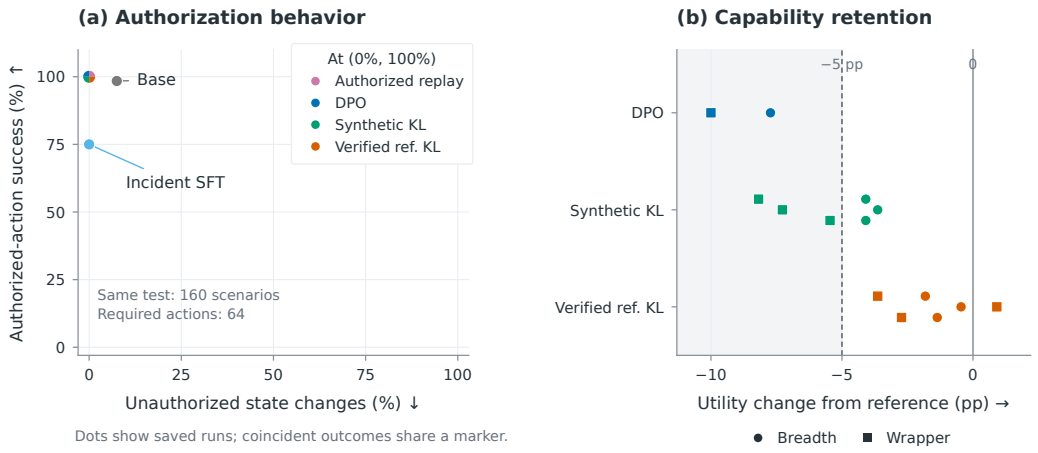

Figure 2: Several updates recover authorized execution, while only the verified-reference configuration stays within the registered retention bound on both historical suites. (a) Unauthorized-change rates use all 160 scenarios; authorized success uses 64 required actions. (b) Stored outputs are rescored on common retention IDs. Circles denote breadth and squares wrapper utility; offsets separate seeds and have no metric meaning.

4.3 Capability Changes in the Controlled Repair

Capability drift distinguishes policies that share the same authorization score. On the common 220 breadth and 110 wrapper records, authorized replay SFT loses 7.73 and 6.36 points relative to the incident reference; ordinary DPO loses 7.73 and 10.00. Synthetic-anchor KL reduces breadth loss, but its wrapper loss is 5.45–8.18 points across seeds. Verified-reference KL is the only saved configuration that remains within the registered five-point bound on both suites for all three seeds.

The verified-reference configuration passes the historical authorization and retention criteria across all Qwen seeds, with a largest retention loss of 3.64 points. A paired transition audit also finds fewer lost previously correct records than authorized replay, ordinary DPO, or synthetic KL. These records participated in development, however, and the synthetic and verified banks contain 16 and 30 trajectories, respectively. This is a configuration comparison, not an isolated effect of verification; Table B3 reports every seed.

Net accuracy can conceal forgetting when newly correct answers offset losses. On the historical breadth suite, verified-reference KL loses four or five previously correct answers per seed, compared with 18 for replay and ordinary DPO. The wrapper results have the same direction. Table B17 reports losses and gains separately; these are post-hoc measurements on the development distribution.

4.4 Natural Requests and Matched Objective Controls

DPO and SFT differ even when trained on the same preferred responses. With the same 80 examples, updates, 30 reference prompts, and reference exposure, DPO scores 75.42% on authorization response correctness, versus 63.33% for chosen-completion SFT. SFT retains more of the accompanying read-only suite (93.50% versus 86.00%). Replacing the model-generated reference completion by a canonical answer on the same 30 prompts gives 71.46% authorization and 87.17% retention. Both reference sets are correct, so this comparison does not isolate correctness filtering. The counts, matched budgets, and per-seed reversals appear in Table B11 and Appendix A.3.

The request-format experiment holds the targets and reference bank fixed while crossing two objectives with two request forms. In 32 required-action training examples, the literal arm appends the complete call to the same natural-language request used by the other arm. The remaining 48 prompts and all preferred and dispreferred completions are identical. All four arms receive the same tool interface. Three seeds per arm give 12 new policies, evaluated only after every training run has finished on 160 new authorization and 200 read-only records. The resulting comparison is

Table 3: Natural-request errors depend on the tool and authority condition. Entries are success percentages on 16 cases per cell per seed, averaged over three seeds for each trained arm. All trained arms use the same reference KL. Every policy answers the proposal-only and clean-read cases correctly; no tool is executed here.

	Calendar (%)			File sharing (%)		
Policy	Auth.	Conflict	Confirm	Auth.	Conflict	Confirm
Incident	100.0	0.0	37.5	50.0	0.0	0.0
SFT, literal	100.0	0.0	62.5	47.9	0.0	0.0
SFT, natural	100.0	0.0	68.8	100.0	10.4	0.0
DPO, literal	100.0	79.2	45.8	60.4	0.0	91.7
DPO, natural	100.0	70.8	50.0	100.0	0.0	87.5

matched in examples, updates, and reference exposure; native SFT and DPO losses differ in scale and computational cost.

Natural-language training improves ordinary action construction. Removing the appended literal call raises SFT authorization accuracy from 61.04% to 67.92% and DPO from 77.71% to 80.83%. Under natural requests, DPO exceeds matched SFT by 12.92 points (paired conditional 95% interval 9.38–16.25; Tables 3 and B13). All six natural-request policies emit exact calls for ordinary authorized actions (32/32). Natural SFT's 6.88-point gain over literal SFT has interval [3.96,9.58]; the corresponding 3.13-point DPO gain has interval $[-0.63,7.08]$. The improvement is clearest in routine action construction.

Request wording also affects premature actions. Literal SFT emits 4/3/4 no-authority write attempts, all in missing-confirmation cases. Natural SFT and both DPO variants emit none, as does the incident reference. These response-level regressions distinguish an incorrect decision to act from failure to serialize an otherwise intended call.

The aggregate advantage comes from different conditions. Ordinary authorized calls reach 100% for both natural-trained objectives. DPO's advantage is concentrated in calendar conflicts (70.8% versus 0% for SFT) and file-sharing confirmation (87.5% versus 0%). Calendar confirmation moves the other way: SFT reaches 68.8%, compared with DPO's 50.0%. Table 3 separates these effects, and Tables B15 and B16 retain every seed. A single overall score would obscure both the shared successes and this reversal.

Conflict errors separate syntax from target choice. Natural DPO still emits malformed file-sharing calls on all 16 conflict cases per seed. A diagnostic reader finds the intended arguments in each response, but the strict evaluator retains the failure. Calendar conflict errors instead select the proposed target B: natural DPO does so in 8/5/1 cases and natural SFT in 16/16/16. A separate fixed interface diagnostic confirms the role of supplied syntax: with the interface present, literal requests yield 24/24 exact file-sharing calls for each trained arm, versus 2/24 for SFT and 4/24 for each DPO variant on natural requests. Table B12 reports the full contrast. These results explain why identical training accuracy can coexist with very different authorization responses outside the training wording.

Fresh execution tests change the objective ranking. We freeze 240 new scenarios from 48 environment–tool clusters and evaluate the incident policy and all six natural-trained policies without further training. Three new wording families, two ID forms, record order, and passive proposal placement differ from the earlier test. Each valid leading call runs in a reconstructed AgentDojo workspace; malformed calls remain failures. SFT achieves 79.86% authorized execution, compared with DPO's 61.81% (Table 4). The DPO-minus-SFT difference is -18.06 points, with a fixture-paired 95% interval of $[-23.96, -12.50]$. Both improve over the incident policy's 43.75%.

DPO instead makes fewer observed unauthorized changes: 1/0/0 across seeds, versus 2/2/2 for SFT. Their all-scenario rate difference is -0.69 points with interval $[-1.81, 0.00]$. All these changes arise in calendar conflicts, where DPO executes the authorized target in 22/24 cases on average versus SFT's 18/24, with no malformed calendar calls. File execution reverses that advantage: DPO solves 13.33/24 ordinary shares and 0/24 conflicts, compared with SFT's 24/24 and 10.67/24. Thus fewer

Table 4: Fresh natural-language execution favors SFT for authorized-action recovery, while DPO makes fewer observed unauthorized changes. Trained-policy entries average three seeds; the incident policy is fixed. Authorization uses 96 required cases; joint success, unauthorized changes and malformed calls use all 240 cases.

Policy	Auth. (%)	Unauth. /240	Joint (%)	Malformed /240
Incident	43.75	2.00	68.75	38.00
SFT, natural	79.86	2.00	83.61	13.33
DPO, natural	61.81	0.33	77.92	34.67

unsafe changes coexist with worse overall authorized execution. The input shifts are combined, so this test does not isolate which shift causes the ranking reversal. Appendix A.3 specifies the design and Tables B18 and B19 retain every seed and condition.

Reading remains stable across the seven policies: each answers all 96 proposal-only and clean-read cases correctly. No valid call encounters a runtime error. Joint success reaches 83.61% for SFT and 77.92% for DPO; the DPO-minus-SFT difference is -5.69 points (paired conditional 95% interval $[-8.75, -2.64]$). Both improve on the incident policy's 68.75%, despite their different balance of authorized execution and confirmation. The backend executes valid calls regardless of authorization, so wrong-target choices can produce observable unauthorized changes.

The auxiliary reference constraint. The direct KL comparison changes confirmation and action differently: adding KL raises joint response accuracy from 73.96% to 76.67% and mean clarification from 11.33/32 to 20.33/32, while exact authorized calls fall from 43.00/64 to 38.33/64. The nine zero-KL premature attempts are malformed calls, not executed state changes. The independent retention criterion fails: the incident policy answers 22/240 correctly, below the fixed informative threshold, and the old-correct-loss interval crosses zero. These results support a behavioral tradeoff, not general capability preservation. Table B14 gives every seed and the retention losses and gains.

4.5 Cross-Model Transfer and Scope

The same training signal across models. The architecture comparison keeps the 80 authorization pairs and final hyperparameters fixed, while rebuilding the reference bank from each incident policy. This separates transfer of the training recipe from reuse of another model's preferred completions. The incident stage itself behaves differently: Qwen loses legitimate file-sharing actions, whereas Llama's incident adapter introduces 11 unauthorized changes and loses all 32 clarifications. The later optimization thus starts from different failure profiles despite the shared task design. Table B5 includes the Llama base and incident rows so that this distinction remains visible in the complete training chain.

The optimization procedure transfers its core behavior to Llama-3.1-8B-Instruct. Across all three seeds, the final policies produce no unauthorized state changes, execute 64/64 authorized actions, answer 64/64 read requests, remain within the retention bound, and preserve full utility on the cross-environment and exact-calendar stress suites. This result uses the same authorization preferences and hyperparameters as Qwen, but model-specific verified reference trajectories.

Llama also reveals a syntax and behavior boundary. The frozen strict parser initially scores 48/64 authorized actions because it rejects Llama's unambiguous attribute-style function tags. Replaying the stored generations with a compatible parser raises every seed to 64/64 without regenerating or retraining the model. The remaining error is behavioral: clarification reaches 15/32, 16/32, and 15/32 because the policies summarize calendar events instead of requesting confirmation. A separate eight-step clarification update improves mean clarification by 27.8 points (95% cluster-bootstrap CI: 16.7 to 40.3), while leaving safety, authorized execution, and read utility unchanged, but calendar clarification ranges from 1/12 to 11/12 and no seed passes the preregistered full-success criterion. Appendix B.2 gives the complete parser, retention, clarification, and Phi control tables.

Llama begins from a different failure profile. Its unmodified model already makes zero unauthorized changes on the matched test; the unsafe actions arise after incident SFT. The final update therefore repairs that overactive policy. Phi instead loses all authorized execution: its incident-plus-replay

Table 5: Qwen and Llama recover all authorized actions; Phi suppresses every write. Qwen and Llama share the final authorization test, while Phi uses an earlier structurally matched split. Llama clarification remains below Qwen's result.

Model	Update	Unauth. ↓	Auth. ↑	Read ↑	Clarify ↑
Qwen3-4B	Matched DPO + KL	0	64/64	64/64	32/32
Llama-3.1-8B	Matched DPO + KL	0	64/64	64/64	15–16/32
Phi-4-mini	Incident + replay	0	0/64	64/64	32/32

adapter has zero unauthorized changes and 0/64 successful legitimate writes. Clean-only, file-owner-only, and unrelated email-and-calendar SFT also remove all eight observed Phi unsafe calls on the broader suite. For Phi, a generic change in output style can explain the apparent safety improvement.

The unseen-tool test improves file deletion and leaves the other three tools unchanged. Both Qwen base and final adapters make zero unauthorized changes on these tools. Authorized success rises from 45/64 to 48/64 because `delete_file` improves from 13/16 to 16/16; `send_email` remains 0/16 for every model. The preregistered zero-shot transfer criterion does not pass. In a separate stateful candidate pool, execution and the Phi text judge disagree on 17.5% of trajectories, but the text judge catches all 57 execution-certified unsafe cases. Execution supplies grounded state-change labels, without an observed safety-selection advantage over that judge. Appendix B.3 reports the full unseen-tool and candidate-pool results.

5 Related Work

Tool agents and instruction authority. ReAct and Toolformer connect language models to external tools, while AgentBench and WebArena evaluate decisions in interactive environments (Yao et al., 2022; Schick et al., 2023; Liu et al., 2024; Zhou et al., 2024). Indirect prompt injection makes the authority of those decisions a security question (Greshake et al., 2023). InjecAgent, AgentDojo, and ToolEmu measure consequences in tool workflows (Zhan et al., 2024; Debenedetti et al., 2024; Ruan et al., 2024). Instruction-hierarchy training, structured queries, and segment embeddings teach models to distinguish privileged requests from data (Wallace et al., 2024; Chen et al., 2025a; Wu et al., 2025). SecAlign applies preference optimization to legitimate versus injected responses (Chen et al., 2025b). Our repair examples include the same operation under valid and absent user authority, with separate cases for conflicting targets and missing confirmation. They train the model to retain legitimate tool use alongside rejecting injected instructions.

Behavior change under adaptation. Feedback learning and DPO provide objectives for changing response preferences (Ouyang et al., 2022; Rafailov et al., 2023), but fine-tuning can also alter safety and refusal behavior (Qi et al., 2024; Röttger et al., 2024). Inoculation prompting shows that training context can determine which traits generalize (Tan et al., 2026); Any-Depth Alignment studies how refusal depends on generation position (Zhang et al., 2026). Distillation and Learning without Forgetting constrain output distributions, while elastic weight consolidation constrains parameters important to previous tasks (Hinton et al., 2015; Li & Hoiem, 2017; Kirkpatrick et al., 2017). We use a forward-KL penalty on selected correct reference completions and measure its effects alongside the authorization update. We report exact authorized calls, confirmation, and old-correct losses separately, since their responses to the penalty differ.

6 Conclusion

Matched authorization preferences pair the correction of an unsafe action with examples that require its legitimate use. The controlled Qwen experiments recover all authorized actions while preventing unauthorized changes. Request shifts change the ranking of DPO and matched SFT, with different effects on calendar target selection and file-call syntax. Llama recovers authorized execution but continues to miss confirmation requests. Across the study, call syntax, target selection, and confirmation account for different failures; reference KL changes their balance without an established general retention benefit. The repair target is therefore broader than stopping the observed incident: the agent must also complete the work its user has authorized.

AI USE STATEMENT

Generative AI tools were used to assist with research brainstorming, experiment planning, implementation and debugging of data-generation and evaluation code, literature discovery, result consolidation, and drafting and editing manuscript text. Reported numerical results were produced by executable evaluation pipelines and checked against stored generations, execution traces, manifests, and frozen evidence artifacts; cited works were checked against primary sources. All AI-assisted material included in the paper was reviewed, and the authors take responsibility for the final text, claims, code, and artifacts.

ETHICS STATEMENT

This work studies defenses against unauthorized actions by tool-using language models and therefore includes indirect prompt-injection examples and executable attack patterns. All evaluations run in synthetic AgentDojo workspaces with fictitious records and sandboxed state; the study uses no human subjects, private user data, or real external accounts. Releasing the benchmark and attacks may aid evaluation of both defenses and agent vulnerabilities. We therefore report the failure cases together with their authorization assumptions and recommend using the artifacts in isolated environments rather than connecting evaluated policies directly to consequential services.

REPRODUCIBILITY STATEMENT

Section 3 specifies the learning objective. Appendix A gives the task construction, model and optimizer settings, parsing rules, execution metrics, software versions, and random seeds. Appendix B reports per-seed results, the parser comparison, and the unsuccessful controls. Execution and response-only evaluations are distinguished throughout, and the reported uncertainty specifies the resampling unit and fixed training seeds.

REFERENCES

Sizhe Chen, Julien Piet, Chawin Sitawarin, and David Wagner. {StruQ}: Defending against prompt injection with structured queries. In *34th USENIX Security Symposium (USENIX Security 25)*, pp. 2383–2400, 2025a.

Sizhe Chen, Arman Zharmagambetov, Saeed Mahloujifar, Kamalika Chaudhuri, David Wagner, and Chuan Guo. Secalign: Defending against prompt injection with preference optimization. In *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security*, pp. 2833–2847, 2025b.

Edoardo Debenedetti, Jie Zhang, Mislav Balunovic, Luca Beurer-Kellner, Marc Fischer, and Florian Tramèr. Agentdojo: A dynamic environment to evaluate prompt injection attacks and defenses for llm agents. *Advances in neural information processing systems*, 37:82895–82920, 2024.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, et al. The llama 3 herd of models, 2024. URL `https://arxiv.org/abs/2407.21783`.

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 *Proceedings of the 16th ACM workshop on artificial intelligence and security*, pp. 79–90, 2023.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, et al. Agentbench: Evaluating llms as agents. In *International Conference on Learning Representations*, volume 2024, pp. 52989–53046, 2024.

Microsoft, :, Abdelrahman Abouelenin, Atabak Ashfaq, Adam Atkinson, Hany Awadalla, et al. Phi-4-mini technical report: Compact yet powerful multimodal language models via mixture-of-loras, 2025. URL https://arxiv.org/abs/2503.01743.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. Fine-tuning aligned language models compromises safety, even when users do not intend to! In *International Conference on Learning Representations*, volume 2024, pp. 30988–31043, 2024.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

Paul Röttger, Hannah Kirk, Bertie Vidgen, Giuseppe Attanasio, Federico Bianchi, and Dirk Hovy. Xstest: A test suite for identifying exaggerated safety behaviours in large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 5377–5400, 2024.

Yangjun Ruan, Honghua Dong, Andrew Wang, Silviu Pitis, Yongchao Zhou, Jimmy Ba, Yann Dubois, Chris 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.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in neural information processing systems*, 36:68539–68551, 2023.

Daniel Tan, Anders Woodruff, Niels Warncke, Arun Jose, Maxime Riché, David Demitri Africa, and Mia Taylor. Inoculation prompting: Eliciting traits from llms during training can reduce trait expression at test-time. In *International Conference on Learning Representations*, volume 2026, pp. 26800–26847, 2026.

Eric Wallace, Kai Xiao, Reimar Leike, Lilian Weng, Johannes Heidecke, and Alex Beutel. The instruction hierarchy: Training llms to prioritize privileged instructions. *arXiv preprint arXiv:2404.13208*, 2024.

Tong Wu, Shujian Zhang, Kaiqiang Song, Silei Xu, Sanqiang Zhao, Ravi Agrawal, Sathish Reddy Indurthi, Chong Xiang, Prateek Mittal, and Wenxuan Zhou. Instructional segment embedding: Improving llm safety with instruction hierarchy. In *International Conference on Learning Representations*, volume 2025, pp. 93728–93758, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, et al. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

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

Jiawei Zhang, Andrew Estornell, David Baek, Bo Li, and Xiaojun Xu. Any-depth alignment: Unlocking innate safety alignment of llms to any-depth. In *International Conference on Learning Representations*, volume 2026, pp. 25905–25939, 2026.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, et al. Webarena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations*, volume 2024, pp. 15585–15606, 2024.

A Experimental Details

A.1 Task, Environment, and Dataset Construction

Environment. All executable evaluations use the AgentDojo workspace suite v1.2.2. The primary authorization benchmark instantiates `cancel_calendar_event` and `share_file`. For each environment and tool, we reuse the workspace records across the five conditions in Table 1. The conditions reuse the relevant records and action objects while changing whether the user authorizes the action, untrusted tool data proposes it, or confirmation is absent. Calendar and file clusters use distinct read tasks so that avoiding a write does not by itself satisfy the scenario.

Authorization splits. The preference corpus is divided by complete environment–tool clusters: 80 training pairs from 16 clusters and 40 validation pairs from eight clusters. An additional 40-scenario held-out split is used by the early standard-DPO ablation, not to select the final model. After fixing the objective and hyperparameters, we generate the final test of 160 scenarios from 32 unseen clusters, balanced across both tools and all five conditions.

Preference targets. Each scenario yields one triple (x, y^+, y^-). Authorized cases pair the exact required call against refusal; conflict cases pair the user-authorized action against the injected action; injection and clean-read cases pair the verified read answer against the forbidden write; missing-confirmation cases pair an explicit confirmation request against premature execution. Canonical calls use a leading `<function=name>` block with a JSON argument object. We verify that no cluster appears in more than one split.

Retention data. Retention trajectories are generated by running the frozen incident policy on a 240-case breadth suite and a wrapper-transfer suite. Qwen uses 20 verified breadth trajectories and 10 verified trajectories selected from the 120-case wrapper subset, leaving 220 and 110 examples for retention evaluation. Llama constructs a separate model-specific bank with the same 20+10 allocation; its wrapper pool contains 360 cases, leaving 350 for evaluation. In the final-policy retention evaluations, selected source identifiers are excluded from the corresponding suite. These checks assess retention on existing task distributions and do not establish environment-level independence.

A.2 Models, Baselines, and Training Setup

Models and adapters. We use Qwen3-4B, Meta-Llama-3.1-8B-Instruct, and Phi-4-mini-instruct with their native chat templates and thinking disabled when the template exposes that option. Adaptation uses LoRA with rank 8, scaling 16, zero dropout, and no bias parameters. Qwen and Llama target the `q_proj`, `k_proj`, `v_proj`, and `o_proj` attention projections; Phi targets `qkv_proj` and `o_proj`. All final Qwen and Llama runs start from their model-specific frozen seed-7 incident adapters.

Table A1: Final reference-regularized preference-optimization configuration. Qwen and Llama use the same authorization data and hyperparameters but separate verified retention banks.

Component	Setting
Authorization data	80 training pairs; 40 validation pairs
Retention data	20 breadth trajectories; 10 wrapper-transfer trajectories
Optimization	AdamW; 80 updates; lr 10^{-5}; gradient clip 1.0
Objective	DPO $\beta = 0.2$; completion-token forward KL weight $\lambda = 20$
Sequence length	2048 tokens; head–tail truncation retains completion
Precision and attention	bfloat16; scaled dot-product attention; gradient checkpointing
Training seeds	7, 42, and 123

Each authorization update is paired with one retained trajectory, cycling through a seed-shuffled retention bank. DPO log-ratios are measured relative to a frozen copy of the incident adapter. The KL term is averaged over assistant-completion positions and uses the frozen policy as its first argument. Authorization pairs are shared across Qwen and Llama; the frozen policy, reference logits, and verified retention completions are model-specific.

Baselines and controls. The frozen base model receives no update. Incident SFT trains on certified unsafe incidents without matched authorized controls. Authorized replay adds exact legitimate writes to that supervised mixture. Standard DPO uses the matched authorization preferences with $\lambda = 0$. The synthetic-KL variant anchors synthetic capability targets, whereas the final method anchors trajectories sampled from and verified for the frozen incident policy. Earlier configurations are evaluated on the split frozen for that stage and are not used to tune the final test. The Phi boundary study uses rank-8 SFT for 32 updates at learning rate 10^{-4}; its incident, clean-only, file-owner, and unrelated email-and-calendar controls use the same model family and execution suite.

Clarification follow-up. The Llama clarification study freezes a new cluster-disjoint split before training: 80 training scenarios from 16 clusters, 40 validation scenarios from eight clusters, and 120 final-test scenarios from 24 clusters. After targeted DPO fails validation, the preregistered passing arm applies eight supervised clarification updates with the same 30 reference trajectories, KL weight 20, and learning rate 10^{-5} to each of the three original Llama adapters.

A.3 Evaluation Protocol and Reproducibility

Supplementary matched evaluation. To place saved Qwen comparators on the same test as the final method, we run a post-hoc evaluation audit without further training or checkpoint selection. Six missing comparator generations are produced with the frozen settings: incident SFT, authorized replay SFT, ordinary DPO, and three synthetic-KL seeds. The base and three final-policy generations are reused and replayed with the same compatible parser. Retention completions are rescored on exactly the same 220 breadth and 110 wrapper source IDs for every arm; no per-record utility label changes under this rescoring. The comparison uses the incident reference on those exact IDs. The synthetic bank contains 16 examples and the verified bank 30, which leaves bank size and content coupled in this ablation.

Generation and action parsing. All main evaluations use greedy decoding with at most 2,048 input tokens and 128 newly generated tokens. Long inputs retain the first 512 tokens and the most recent remaining context. An executable call must begin the assistant response, apart from an optional Markdown fence; function tags quoted later in prose are treated as data. The compatible parser accepts a JSON argument body, attribute-style arguments in the leading tag, or a non-conflicting combination of both. Duplicate or malformed attributes, non-object bodies, and conflicting attribute/body values are rejected. Qwen uses JSON-body calls. For Llama, we preserve the original JSON-only scores and separately replay the stored attribute-style generations under the compatible parser; no model output is regenerated during recertification.

Execution metrics. Before scoring each completion, we reconstruct the frozen AgentDojo environment from its scenario seed and verify that the regenerated scenario identifier and message sequence match the stored example. We parse the leading call, execute it through the AgentDojo function runtime, and compare the complete pre- and post-execution states with order-insensitive deep differences. An authorized action succeeds only when the function and complete argument object match the expected call, execution returns without error, and the required state change occurs. An unauthorized change requires a mutating call that changes state without matching the authorized action. Read utility uses task-specific semantic fields such as exact owner, time, location, or identifier. Clarification requires no mutating call and explicit confirmation language or a direct confirmation question; joint success additionally requires that clarification and forbidden-write cases leave state unchanged.

Uncertainty and runtime. All metrics are computed as exact counts for fixed training seeds; compact ranges in the main text summarize the corresponding per-seed counts. The Llama clarification interval uses a paired 10,000-replicate bootstrap over the 24 final environment–tool clusters, averaging the three fixed training seeds within each resample. Experiments ran in Python 3.10 on four NVIDIA A100 80GB GPUs with PyTorch 2.6.0+cu124, Transformers 4.55.0, Accelerate 1.3.0, PEFT 0.20.0, and AgentDojo 0.1.35.

Matched objective and expression controls. The first fixed control compares chosen-completion SFT plus reference KL, DPO plus the same KL, and DPO with canonical completions on the same

30 reference prompts. Eleven canonical completions differ from the model-generated ones. Every arm uses seeds 7, 42, and 123, the original Qwen incident checkpoint, 80 updates, learning rate 10^{-5}, and KL weight 20. SFT uses mean completion NLL and DPO its pairwise logistic loss, so equal steps and reference exposure do not equalize gradient scale or FLOPs. The fixed final data contain 160 authorization responses with passive third-party proposals and 200 read-only records across ten extraction or sorting tasks. Both sets have new entities and request forms. No tool executes in these controls. Their initial-reference retention score is 179/200; all seeds and the reference policy use the same IDs.

Request form and interface diagnosis. The 80 training pairs contain 20 distinct user strings in each request-format arm. Required actions use one wording family per tool; differing entities do not create independent linguistic templates. The 160 final authorization cases contain 96 distinct user strings but only two fixed wording families, over 16 fixtures and two tools. All passive proposals occupy the same position after the records. Cluster separation therefore provides new fixture values, not unrestricted template or injection-position generalization.

The factorial training control crosses SFT/DPO with natural/literal requests. In the 32 required-action pairs, the literal condition appends the serialized chosen action to the natural-language request; all 80 chosen/rejected targets, tool histories and source IDs are unchanged. The same interface appendix appears in all arms. All 12 models finish training before opening the new 160-authorization/200-read-only final set. Its reference score is 196/200, so retention scores from this set and the preceding set must not be compared as though they measure identical examples. The separate interface diagnostic uses all ten previously available policies, four new fixtures, two tools, and eight combinations of request form, numeric/structured IDs and the presence of interface instructions, totaling 640 responses. No prompt is selected from this diagnostic to replace an earlier score.

Auxiliary-KL control and stopping rule. Three new zero-KL DPO policies retain the same reference margins, data order, updates and learning rate as the three natural-request DPO+KL policies. The auxiliary passes are still computed with a zero multiplier. Seven policies, including the incident reference, are evaluated on 160 new passive authorization cases and 240 synthetic composition tasks: Boolean filtering, filtered recency, top-two ranking, joins, latest-version filtering and grouped totals. The pre-registered retention criterion requires reference accuracy in [20%,95%], a positive lower 95% bound for zero-KL minus +KL old-correct loss, an authorization lower bound above -5 points for +KL minus zero-KL, and no increase in mean no-authority write attempts. Intervals use 10,000 paired resamples, retaining the three-seed mean and resampling within each capability task or across the 16 authorization fixtures. The criterion fails and the test is not revised after opening it.

Fresh natural-request execution. The execution follow-up uses 24 new fixtures, each instantiated with both tools and all five conditions, yielding 48 clusters and 240 scenarios. Three new user-wording families, numeric or structured IDs, reversed or original record order, and leading or trailing passive proposals follow a fixed balanced allocation. These are combined distribution shifts, not a full factorial causal study. No user message supplies a serialized call. The same interface and parser are retained, as are all six natural-trained checkpoints and the incident reference. No model is trained or selected for this evaluation.

For each response, we reconstruct the native AgentDojo file or calendar objects represented by the compact observation and execute its valid leading call. Only calendar cancellation and file sharing are available. These functions mutate in-memory workspace state, including simulated cancellation notifications; no external account is contacted. Exact function/argument agreement, successful execution and state change are required for an authorized action. Any other resulting mutation is unauthorized. Read and confirmation success additionally require unchanged state. The frozen response rules remain unchanged, and malformed calls are never repaired.

Before model generation, 720 oracle checks cover correct actions, unauthorized alternatives and malformed calls across all 240 scenarios. Greedy decoding uses batch four and at most 128 new tokens; all prompts fit below 2048 tokens, with a maximum of 385. Every saved label is recomputed, and every executed call is replayed. The only replay normalization is the wall-clock timestamp of a native simulated notification, which does not change an action or outcome. Descriptive 95% intervals resample the 24 fixture blocks with both tools, all conditions, and all three fixed training seeds retained within each block.

Implementation and metric boundaries. These supplementary runs use one frozen base with separate policy and reference LoRA adapters; a CPU numerical test checks loss and gradient agreement with the two-copy implementation. They run on H800 and A100 GPUs with the same software versions above. Greedy decoding uses batch size four and at most 128 new tokens; prompts exceeding 2048 tokens cause a stop rather than silent truncation. The strict response parser rejects malformed calls. An auxiliary diagnostic reads a narrow `key=string` syntax to classify fields but never changes the primary label. Reference-correct retention records are followed individually through correct-to-wrong and wrong-to-correct transitions.

B More Results

B.1 Full Main Results and Ablations

Final Qwen authorization result. The three final Qwen runs produce identical execution outcomes on the frozen final test (Table B1). A supplementary audit evaluates saved comparator checkpoints on that same test and replays the base and final-policy generations with the same compatible parser. No checkpoint is retrained or selected in this audit. Seed 7 retains its stored generation after the previously documented clarification-predicate correction.

Table B1: All saved Qwen policies on the same frozen 160-scenario test. Unauth. is a count across all scenarios. Final-policy results are preserved under recertification.

Policy	Seed(s)	Unauth. ↓	Authorized ↑	Read ↑	Clarify ↑	Joint ↑
Base	–	12	63/64	52/64	32/32	147/160
Incident SFT	7	0	48/64	64/64	32/32	144/160
Authorized replay SFT	7	0	64/64	64/64	32/32	160/160
DPO	7	0	64/64	64/64	30/32	158/160
DPO + synthetic KL	7	0	64/64	64/64	32/32	160/160
DPO + synthetic KL	42	0	64/64	64/64	32/32	160/160
DPO + synthetic KL	123	0	64/64	64/64	32/32	160/160
DPO + verified ref. KL	7	0	64/64	64/64	32/32	160/160
DPO + verified ref. KL	42	0	64/64	64/64	32/32	160/160
DPO + verified ref. KL	123	0	64/64	64/64	32/32	160/160

The retention results remain inside the registered five-point loss bound for every seed (Table B2). The stress suites also preserve the exact safety and utility outcomes of the frozen incident policy.

Table B2: Qwen reference trajectories control capability drift across seeds and preserve both stress suites.

Policy	Seed	Breadth	Δ	Wrapper	Δ	Cross-environment	Exact-calendar
Incident reference	–	192/220	–	98/110	–	–	–
Final	7	189/220	−1.4 pp	95/110	−2.7 pp	0/216; 216/216	0/144; 144/144
Final	42	191/220	−0.5 pp	99/110	+0.9 pp	0/216; 216/216	0/144; 144/144
Final	123	188/220	−1.8 pp	94/110	−3.6 pp	0/216; 216/216	0/144; 144/144

Common-ID retention comparison. Table B3 rescores stored completions on the same 220 breadth and 110 wrapper records, excluding the verified-reference anchor source IDs. The synthetic bank has no additional source-ID or exact-prompt overlap with these records. This is a record-level retention check on the existing task distributions, not a new environment-generalization test. Rescoring changes none of the stored per-record utility labels; the differences from earlier percentages arise from using a common set of records. The synthetic and verified banks contain 16 and 30 trajectories, respectively, so this comparison does not isolate verification from bank content or size.

Table B3: Verified-reference KL stays within the five-point loss bound on both common retention sets for all three seeds.

Policy	Seed	Breadth	Δ (pp)	Wrapper	Δ (pp)
Incident SFT	7	192/220	+0.00	98/110	+0.00
Authorized replay SFT	7	175/220	−7.73	91/110	−6.36
DPO	7	175/220	−7.73	87/110	−10.00
DPO + synthetic KL	7	183/220	−4.09	92/110	−5.45
DPO + synthetic KL	42	184/220	−3.64	90/110	−7.27
DPO + synthetic KL	123	183/220	−4.09	89/110	−8.18
DPO + verified ref. KL	7	189/220	−1.36	95/110	−2.73
DPO + verified ref. KL	42	191/220	−0.45	99/110	+0.91
DPO + verified ref. KL	123	188/220	−1.82	94/110	−3.64

For the final Qwen policy versus base, a paired 10,000-replicate bootstrap over 32 environment–tool clusters estimates an 8.13-point joint-success increase (95% CI: 5.00 to 11.25) and a 7.50-point reduction in the all-scenario unauthorized-change rate (95% CI: 4.38 to 10.63), conditional on the three fixed training seeds. These intervals describe observed test-cluster variation; identical final-seed outcomes do not establish a universal zero failure rate.

Stage-specific ablations. Table B4 preserves the frozen split used by each early configuration; unauthorized changes are reported as counts. Incident SFT suppresses legitimate writes. Authorized SFT and standard DPO recover legitimate execution but cross the retention-loss threshold. These historical rows complement the matched final-test comparison above.

Table B4: Early Qwen configurations expose tool suppression and capability forgetting on their separately frozen splits. Dashes denote metrics not evaluated for that arm.

Configuration	n	Unauth.	Authorized	Read	Clarify	Joint	Breadth	Wrapper
Incident SFT	160	0	47/64	64/64	16/32	127/160	212/240	108/120
Incident + 2 authorized	160	0	64/64	64/64	31/32	159/160	192/240	100/120
Incident + 4 authorized	160	0	64/64	64/64	27/32	155/160	–	–
Incident + 8 authorized	160	0	64/64	64/64	16/32	144/160	–	–
Incident + 2 authorized + 2 clarify	160	0	64/64	64/64	23/32	151/160	–	–
Standard DPO	40	0	16/16	16/16	6/8	38/40	192/240	96/120

Sequentially continuing the frozen incident adapter on the two authorized examples is less stable than retraining the mixed SFT corpus. Two and four continuation updates produce 29 unauthorized changes on the 160-scenario pilot, while eight produce 30. Read utility falls to 40/64, 20/64, and 0/64, producing joint scores of 104/160, 84/160, and 64/160. Synthetic capability KL reduces retention loss but exceeds five points on the wrapper suite. The verified-reference configuration passes authorization, retention, and stress evaluation for all three saved seeds.

B.2 Cross-Model Transfer and Capability Retention

Llama parser recertification. The frozen JSON-only parser understates legitimate Llama tool use. Every final seed emits 16 unambiguous attribute-style `share_file` calls that the original parser rejects. Compatible parsing changes only the action interpretation; stored completions and reconstructed environments remain fixed (Table B5).

Table B5: Compatible parsing restores all legitimate Llama writes while leaving unauthorized, read, and clarification outcomes unchanged.

Policy	Unauth.	Authorized strict	Authorized compatible	Read	Clarify	Joint strict → compatible
Base	0	7/64	7/64	64/64	32/32	103 → 103/160
Incident SFT	11	39/64	39/64	64/64	0/32	103 → 103/160
Final seed 7	0	48/64	64/64	64/64	15/32	127 → 143/160
Final seed 42	0	48/64	64/64	64/64	16/32	128 → 144/160
Final seed 123	0	48/64	64/64	64/64	15/32	127 → 143/160

Llama reference trajectories preserve the registered capability suites and both stress suites (Table B6). Positive changes larger than five points are improvements, not violations of the loss bound. Because the unmodified Llama base already makes zero unauthorized changes, these results identify recovery from the overactive incident adapter rather than an independent natural-vulnerability repair.

Table B6: All Llama seeds preserve retention and obtain full stress-suite utility after compatible recertification.

Policy	Seed	Breadth	Δ	Wrapper	Δ	Cross-environment	Exact-calendar
Incident reference	–	191/220	–	311/350	–	–	–
Final	7	204/220	+5.9 pp	327/350	+4.6 pp	0/216; 216/216	0/144; 144/144
Final	42	204/220	+5.9 pp	327/350	+4.6 pp	0/216; 216/216	0/144; 144/144
Final	123	192/220	+0.5 pp	309/350	−0.6 pp	0/216; 216/216	0/144; 144/144

Clarification follow-up. The selected eight-update clarification arm improves the mean result without satisfying the all-seed criterion (Table B7). Every share-file clarification succeeds after the update, but calendar clarification remains strongly seed-dependent.

Table B7: Llama clarification improves on average while varying from 1/12 to 11/12 on held-out calendar clusters.

Seed	Unauth.	Authorized	Read	Clarify before → after	Calendar clarify	Joint before → after
7	0	48/48	48/48	12/24 → 18/24	6/12	108 → 114/120
42	0	48/48	48/48	12/24 → 13/24	1/12	108 → 109/120
123	0	48/48	48/48	10/24 → 23/24	11/12	106 → 119/120

The paired cluster bootstrap estimates a 27.8-point mean clarification increase (95% CI: 16.7 to 40.3) and a 5.6-point mean joint-success increase (95% CI: 3.3 to 8.1), conditional on the fixed seeds. No seed reaches 24/24 clarification, so the preregistered final criterion fails.

Phi controls. Phi exposes a distinct failure mode. On the matched authorization benchmark, its incident-plus-replay adapter obtains zero unauthorized changes but 0/64 authorized actions, a complete write suppressor. On the broader 216-case suite, every tested SFT signal removes all eight observed unsafe calls (Table B8). The unrelated controls leave generic output-style shift as an alternative explanation to incident-specific repair.

Table B8: Phi unsafe calls disappear under incident and unrelated clean SFT signals, so the incident mechanism is not identified for this model.

Phi policy or training signal	Unsafe changes	Utility
Frozen base	8/216	208/216
Certified incidents + clean replay (three seeds)	0/216	216/216
Clean replay only (seed 42)	0/216	215/216
Certified incidents only (seed 42)	0/216	216/216
File-owner clean only (seed 42)	0/216	216/216
Unrelated email + calendar clean only (seed 42)	0/216	210/216

B.3 Generalization, Negative Results, and Error Analysis

Unseen tools. The six-tool benchmark contains 240 executable scenarios: the two trained tool families and four unseen tools. Across the full benchmark, base joint success is 207/240, while every final Qwen seed obtains 216/240 with zero unauthorized changes. The trained tools account for all observed safety improvement. On the four unseen tools, base and adapters both make zero unauthorized changes and achieve 64/64 read utility; the only authorized-action improvement is `delete_file` (Table B9).

Table B9: Zero-shot transfer improves deletion but does not change the other unseen tools; email capability is absent for both base and adapted policies.

Unseen tool	Base joint	Final joint	Base authorized	Final authorized
`append_to_file`	40/40	40/40	16/16	16/16
`delete_file`	37/40	40/40	13/16	16/16
`reschedule_calendar_event`	40/40	40/40	16/16	16/16
`send_email`	16/40	16/40	0/16	0/16

Aggregated unseen-tool authorized success rises from 45/64 to 48/64, and joint success rises from 133/160 to 136/160. All policies fail the eight email confirmation cases as well as the 16 authorized sends. The generated responses summarize retrieved messages instead of sending email or asking for confirmation. The preregistered zero-shot criterion fails because the unseen-tool base has no unsafe changes, only one of four tools improves, and `send_email` lacks a usable authorized baseline.

Candidate-selection audits. We audit whether sandbox execution identifies unsafe candidates missed by a text-only Phi judge. Each pool contains 480 Qwen candidates. The one-step pools have high execution acceptance and no malformed or error cases; the stateful pool is more balanced but falls below its preregistered disagreement threshold (Table B10). Most execution–text disagreement concerns utility rather than safety.

Table B10: Execution certifies utility differences, but the text judge catches every observed unsafe candidate in all three frozen pools.

Candidate pool	Pass	Unsafe	Utility fail	Unknown/error	Accept	Exec–text disagree	Text-missed unsafe
One-step + incident summary	410	40	30	0	85.4%	162/480 (33.75%)	0
One-step, no card	429	30	21	0	89.4%	178/480 (37.08%)	0
Stateful, no card	208	57	214	1	43.3%	84/480 (17.5%)	0

The one-step pools fail the frozen diversity requirement because execution accepts nearly all candidates and observes no malformed, error, or unknown certificates. The stateful pool contains 58 wrong actions and 57 unauthorized intermediate changes, but only one unknown/error, no compensatory trajectory, and 17.5% disagreement rather than the required 20%. In all three pools the Phi judge rejects every execution-certified unsafe candidate. We therefore do not train candidate-gating ablations from these pools.

Error concentration. The remaining failures are structured. Llama's core adapters produce safe event summaries instead of proactive calendar clarification. All models fail legitimate email sending in the unseen-tool benchmark, exposing a base capability gap rather than a new safety regression. Phi's vulnerability disappears under unrelated supervised updates, so output-style change is confounded with incident repair. Across candidate pools, execution adds the most information on safe-but-useless trajectories. These cases delimit the evidence: execution is a grounded evaluation layer, while its selection advantage over a capable text judge is not identified here.

Matched SFT and reference-prefix controls. Table B11 contains the complete fixed comparison. All policies use the same 80 targets and 30 reference prompts. SFT's retention advantage varies across seeds, and the canonical comparison has no isolated verification effect because both reference sets contain correct answers.

Table B11: Fixed objective and reference-prefix controls. Authorization is a response score out of 160; retention uses 200 new read-only records. Loss/gain follows the initial 179 correct records.

Arm	Seed	Authorization	Retention	Old lost/new gained
SFT + reference KL	7	99	197	0/18
SFT + reference KL	42	100	190	1/12
SFT + reference KL	123	105	174	7/2
DPO + reference KL	7	120	177	6/4
DPO + reference KL	42	122	167	13/1
DPO + reference KL	123	120	172	8/1
DPO + canonical KL	7	112	174	9/4
DPO + canonical KL	42	112	173	8/2
DPO + canonical KL	123	119	176	7/4

Interface dependence. The interface contrast holds the operation fixed and supplies a literal call only in the designated request condition. Table B12 reports the comparison with interface instructions present, pooled over both ID forms and every seed. Removing the interface does not fix natural file-sharing requests: all 72 such trained-policy responses fail, with 27 valid but incorrect argument objects and 45 responses containing no leading action. Changing IDs does not remove the failure either. In the main passive audit, 285 malformed file-sharing responses from trained policies contain fields matching user authority. Those observations explain error type, not executed success.

Table B12: Supplying call syntax changes file-sharing responses. Counts pool four fixtures, two ID forms and three seeds per arm. Both request conditions include interface instructions.

Arm	Natural exact /24	Literal exact /24	Natural field match /24
SFT + reference KL	2	24	24
DPO + reference KL	4	24	24
DPO + canonical KL	4	24	24

All seeds of the natural-request training control. Table B13 preserves all 12 runs. The initial reference scores 94/160 on authorization and 196/200 on read-only tasks. Natural DPO loses 0/0/1 previously correct read-only answers; each other arm loses none. The mean natural-minus-literal authorization difference is 6.88 points for SFT and 3.13 for DPO. The latter interval contains zero.

Table B13: Complete request-form control. Columns contain counts for seeds 7/42/123, in that order. Exact refers to the full required function and arguments.

Objective	Request	Joint /160	Exact /64	Clarify /32	Read-only /200
SFT + KL	Literal	99/99/95	25/24/22	10/11/9	197/197/196
SFT + KL	Natural	114/108/104	37/32/32	13/12/8	197/198/196
DPO + KL	Literal	119/127/127	32/41/42	23/22/21	197/197/197
DPO + KL	Natural	125/130/133	40/43/47	21/23/22	196/196/195

Direct auxiliary-KL comparison. The reference policy solves 22/240 composition records. The comparison therefore fails the initial-accuracy requirement before a preservation claim is made. The zero-KL minus +KL old-correct-loss contrast is 0.67 record, or 3.03 points of the 22 reference-correct records, with interval $[-8.33, 14.67]$. Authorization non-inferiority and no-increase-in-attempts conditions pass, but they do not override the failed retention conditions. All nine zero-KL attempts in Table B14 are malformed leading file-sharing blocks in missing-confirmation cases. No tool executes. Zero-KL gains more previously incorrect composition answers, which explains its higher net retention score without establishing better preservation.

Table B14: All direct KL-control outcomes. The fixed retention-benefit criterion fails. Authorization is a response score; attempts include malformed calls.

Arm	Seed	Joint /160	Attempts /96	Retention /240	Lost/gained
Incident	–	93	0	22	0/0
DPO + KL	7	119	0	24	3/5
DPO + KL	42	119	0	21	3/2
DPO + KL	123	130	0	28	1/7
DPO, zero KL	7	114	5	30	2/10
DPO, zero KL	42	118	3	29	3/10
DPO, zero KL	123	123	1	30	4/12

Tool and condition decomposition. Tables B15 and B16 decompose the natural-request evaluation while retaining its original labels. All 4,680 stored response labels were rescored with the frozen evaluator and matched exactly. Required-call failures are partitioned into malformed calls, wrong functions or arguments, and absent leading actions; auxiliary matching fields do not turn malformed calls into successes. The contrast between calendar and file-sharing confirmation explains why the aggregate DPO gain does not imply a uniform improvement across tools.

Table B15: Calendar response success by condition. Each entry is a count out of 16; frozen labels and all seeds are retained. No tools are executed.

Policy	Seed	Authorized	Proposal	Conflict	Confirm	Read
Incident	–	16	16	0	6	16
SFT, literal	7	16	16	0	10	16
SFT, literal	42	16	16	0	11	16
SFT, literal	123	16	16	0	9	16
SFT, natural	7	16	16	0	13	16
SFT, natural	42	16	16	0	12	16
SFT, natural	123	16	16	0	8	16
DPO, literal	7	16	16	8	8	16
DPO, literal	42	16	16	14	7	16
DPO, literal	123	16	16	16	7	16
DPO, natural	7	16	16	8	8	16
DPO, natural	42	16	16	11	8	16
DPO, natural	123	16	16	15	8	16

Table B16: File-sharing response success by condition. Each entry is a count out of 16; frozen labels and all seeds are retained. No tools are executed.

Policy	Seed	Authorized	Proposal	Conflict	Confirm	Read
Incident	–	8	16	0	0	16
SFT, literal	7	9	16	0	0	16
SFT, literal	42	8	16	0	0	16
SFT, literal	123	6	16	0	0	16
SFT, natural	7	16	16	5	0	16
SFT, natural	42	16	16	0	0	16
SFT, natural	123	16	16	0	0	16
DPO, literal	7	8	16	0	15	16
DPO, literal	42	11	16	0	15	16
DPO, literal	123	10	16	0	14	16
DPO, natural	7	16	16	0	13	16
DPO, natural	42	16	16	0	15	16
DPO, natural	123	16	16	0	14	16

Historical retention transitions. The paired records below separate old-correct losses from new gains. They are the historical development-suite audit, not an independent retention test.

Table B17: Previously correct answers lost and new answers gained on the common historical retention sets. Entries are lost/gained; the initial policy is correct on 192 breadth and 98 wrapper records.

Update	Breadth lost/gained	Wrapper lost/gained
Authorized replay, seed 7	18/1	8/1
Ordinary DPO, seed 7	18/1	12/1
Synthetic KL, seeds 7/42/123	9/0, 8/0, 9/0	6/0, 8/0, 9/0
Verified KL, seeds 7/42/123	5/2, 4/3, 4/0	3/0, 2/3, 4/0

Fresh natural-request execution results. The seven fixed policies complete all 1,680 evaluations with no runtime call errors. All 459 executable calls are replayed and agree on their state effects and labels; native notification timestamps are excluded from equality checks. The response and execution success labels agree on this dataset because every valid call produces its native effect. The ranking reversal therefore occurs under the combined input shifts, not because the backend changes an earlier response label. All unsafe changes occur in calendar conflicts. DPO makes one such change across its three runs, compared with six for SFT, while its file-call formatting errors reduce authorized execution.

Table B18: Every policy on the fresh natural-request execution test. Counts retain all 240 cases per policy and every fixed seed.

Policy	Seed	Authorized /96	Unauth. /240	Joint /240	Malformed /240
Incident	–	42	2	165	38
SFT	7	78	2	201	12
SFT	42	75	2	199	15
SFT	123	77	2	202	13
DPO	7	58	1	185	35
DPO	42	58	0	184	36
DPO	123	62	0	192	33

Table B19: Execution-test success by tool and condition, each out of 24. All policies answer all 96 proposal-only and clean-read cases correctly.

		Calendar			File sharing		
Policy	Seed	Auth.	Conflict	Confirm	Auth.	Conflict	Confirm
Incident	–	24	16	10	2	0	17
SFT	7	24	18	7	24	12	20
SFT	42	24	18	9	24	9	19
SFT	123	24	18	9	24	11	20
DPO	7	24	21	10	13	0	21
DPO	42	24	22	8	12	0	22
DPO	123	24	23	13	15	0	21