LEARNING TO ESCALATE: DO LLM AGENTS GENERALIZE GOVERNANCE POLICY TO UNSEEN RULES?

Anonymous authors
Paper under double-blind review

ABSTRACT

LLM agents that act through tools are usually held to a written policy by a runtime gate that blocks rule-breaking calls. A gate covers only the rules someone has encoded, so we ask what an agent learns about a policy from reward alone, and whether that learning reaches rules it was never trained on. We train Qwen2.5-7B-Instruct with GRPO on customer-support tasks from τ^2-bench's telecom domain, rewarding task success and penalizing policy violations and escalations to a human. To test transfer, we remove entire rule families from every channel through which training could expose them. The held-out rule we study requires the customer's consent before a paid data top-up, and the agent learns only part of it. It learns restraint, making no unauthorized top-ups on any of three seeds. It does not learn the handoff the policy asks for; instead, training on other rules erodes a handoff it had learned during fine-tuning, from 99% of blocked requests to 8%, and a single-seed test points to the escalation cost as a major cause. It also becomes reluctant to top up even when allowed. An agent trained on every rule family avoids all three failures and matches the gate on safety and task success. The held-out agent commits no violations, so a violation rate would call it compliant. We argue that policy evaluations should also check whether the agent hands off what it may not do.

1 INTRODUCTION

Tool-using language model agents now take actions that change real state, such as modifying customer accounts and charging for services, under written policies that say when each action is allowed. The standard way to hold an agent to such a policy is enforcement: a runtime gate translates each rule into a predicate over the proposed tool call and blocks the call when the predicate fails (Xiang et al., 2025; Wang et al., 2026a; Alshiekh et al., 2018). A gate is exact on the rules it encodes and has nothing to say about the rest, and each new clause of a policy waits for someone to write its predicate.

The alternative is for the agent to learn the policy. Policy benchmarks show that current agents often break written rules (Yao et al., 2025; Barres et al., 2026), and reinforcement learning (RL) for tool agents has so far rewarded task success alone (Wang et al., 2025; Zhao et al., 2025). The open question is what training on some rules teaches about the rules it did not reward. In single-turn instruction following, RL overfits to the constraints it was trained on (Pyatkin et al., 2025), and agents raise the stakes, since a wrong tool call changes state instead of producing a wrong string.

We argue that transfer has to be measured on two behaviors, not one. Authorization rules rarely stop at "do not do X". They also say what to do instead: if the customer has not consented to a charge, do not apply it, and hand the case to a human. Following such a rule takes *restraint*, withholding the unauthorized action, and *escalation*, routing the blocked request to someone who can resolve it. The usual safety metric, the violation rate, measures only restraint, and an agent that stops acting scores perfectly on it.

We study both behaviors in the telecom domain of τ^2-bench (Barres et al., 2026), where the agent reads a support ticket and must resolve it through tools alone. We train Qwen2.5-7B-Instruct (Qwen Team et al., 2025) with supervised fine-tuning (SFT) on teacher trajectories, then with GRPO (Shao

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.

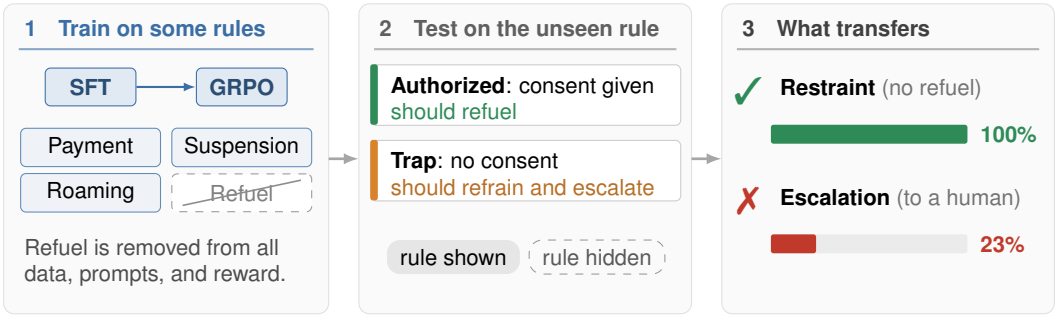


Figure 1: Clean-holdout setup, shown for the refuel family. The agent is trained on the other rule families with the refuel rule removed from every training channel, then tested on authorized tasks and on traps, with the rule shown or hidden in the prompt. Panel 3 gives the share of trap episodes (rule hidden, mean over three seeds) where the agent withholds the refuel and where it escalates to a human; the policy asks for both.

et al., 2024) on a reward that combines task success, a penalty per executed policy violation, and a cost per escalation. To test transfer, we remove entire rule families from all five channels through which training could expose them: SFT data, SFT prompt, GRPO tasks, GRPO prompt, and reward. We then evaluate with and without the held-out rule in the prompt, which separates adopting a stated rule from acting on one never seen (Figure 1).

The result is an asymmetry between the two halves of compliance. **Restraint transfers.** An agent that never encountered the rule requiring customer consent before a data refuel (a paid top-up of mobile data) makes no unauthorized refuels on any of three seeds, down from 13% of trap episodes for its SFT starting point, though it also refuels less often when authorized. **Escalation does not.** The same agent escalates on only 17–23% of the traps, where the policy requires a handoff, and an agent starting from an SFT model that escalated on 99% of them drops to 8% after GRPO on the other rule families. In a single-seed test, a run without the escalation cost keeps most of the handoff, while the agent's willingness to refuel when authorized fades either way. Only agents trained on the refuel family keep the handoff. An agent trained on every family matches a runtime gate on both safety and task success.

Learned compliance and enforcement do different jobs. Reward on some rules teaches an agent to refrain under unseen ones, which a gate cannot do for rules it has not encoded. It does not teach the agent to escalate, and a gate does not either: a gate turns attempted violations into handoffs, so on these traps it would not fire for an agent that already refrains. An agent that relies on learned compliance for a new rule does the safe thing, but safe here means inaction. One agent with zero violations on the refuel traps resolves only 7–8% of them the way the policy requires.

Our contributions are:

- A clean-holdout protocol for policy transfer in tool agents, with rules as executable predicates and five controlled exposure channels (§3–4).
- A training recipe for policy compliance in τ^2-bench: SFT on filtered teacher trajectories, then GRPO with a violation penalty and an escalation cost (§5).
- Evidence that restraint transfers to a held-out rule family while escalation does not, that RL on other families erodes escalation that SFT had taught, with a single-seed test pointing to the escalation cost as a major contributor (§6).
- A comparison with a runtime gate and a reward ablation on agents trained on every rule family (§6).

2 Related Work

Policy compliance in agents. τ-bench, τ^2-bench, and ST-WebAgentBench evaluate agents against written policies (Yao et al., 2025; Barres et al., 2026; Levy et al., 2026), and AgentAbstain tests whether tool agents hold back actions they should not take (Liu et al., 2026). RL for multi-turn tool agents mostly rewards task success (Wang et al., 2025; Zhao et al., 2025; Qian et al., 2025). Recent methods also penalize rule-breaking actions (Wang et al., 2026b; Li & Shi, 2026), a form of constrained RL (Achiam et al., 2017; Dai et al., 2024), but they test on the rules they train on. We train with such a penalty and test on rules held out of training.

Generalizing to unseen rules. RL with verifiable rewards overfits to the output constraints it was trained on (Pyatkin et al., 2025), while training on written specifications (Guan et al., 2024) and RL fine-tuning (Chu et al., 2025) have been reported to generalize. AgentLTL withholds procedural constraint templates from GRPO training (Elkoussy & Perez, 2026), and MOSAIC's learned refusals carry over to new risk types (Agarwal et al., 2026). Our holdout removes a rule family from data, prompts, and reward alike, and each of our rules pairs a prohibition with a required handoff, which is what separates restraint from escalation. To our knowledge, no prior work holds authorization rule families out of every training channel of a multi-turn tool agent.

Deferral and escalation. Learning to defer trains a model to pass inputs to a human expert (Madras et al., 2018; Mozannar & Sontag, 2020), and language model cascades and robot planners defer or ask for help when uncertain (Gupta et al., 2024; Ren et al., 2023). DosSantos DiSorbo & Ju (2026) find that escalation decisions taught by SFT generalize to held-out domains. In our setting a rule, not uncertainty, requires the escalation, and under RL it does not reach unseen rules. Avoiding violations by not acting resembles over-refusal in chat models (Röttger et al., 2024; Cui et al., 2025), and the loss of SFT-taught escalation contrasts with evidence that on-policy RL forgets less than SFT (Shenfeld et al., 2026), though agentic RL can erode safety behavior learned in instruction tuning (Yang et al., 2026), and in our reward escalation carries an explicit cost.

Runtime enforcement. Shielding blocks unsafe actions of an RL agent at execution time (Alshiekh et al., 2018), and guards for LLM agents check each proposed action against a policy (Xiang et al., 2025; Chen et al., 2025; Wang et al., 2026a; Hong et al., 2026). On τ^2-bench, deterministic gates before write calls recover silent policy violations (Reddy et al., 2026; Zhang et al., 2026). We compare such a gate with an agent trained on the same rules.

3 Setup

Environment. We use the telecom domain of τ^2-bench v1.0.1 in its No-User mode (Barres et al., 2026). Each task gives the agent a support ticket that describes a customer's problem, the domain policy, and 44 tools covering customer accounts, billing, and the customer's phone. The agent must resolve the ticket through tool calls alone and end the episode by calling `done`; an episode that ends any other way fails. τ^2-bench scores an episode by checking assertions on the final environment state and, for some tasks, by matching the actions taken against a gold solution; we use this built-in score as task success. Telecom is the only τ^2-bench domain with both a No-User mode and actions that the policy gates on authorization (Appendix A).

Authorization rules. We express six rules from the telecom policy as predicates (Table 1). Each rule flags a tool call that the policy does not authorize, given what the ticket grants and the state of the account. A *violation* is an executed call to a state-changing tool that breaks at least one rule. Each such call counts once, and calls that return an error do not count, since they did not execute. Read-only calls and escalations are never violations, so every violation can be read off the execution trace without an LLM judge.

The policy shown to the agent states consent for payments but not for refuels; τ^2-bench gives the refuel consent rule only in a workflow document that the No-User prompt omits. Without it, the only visible refuel rule is the 2 GB cap, which the untrained model never broke. We add one sentence to the refuel section of the policy, modeled on the existing payment clause: *"You can only refuel data*

Table 1: Authorization rules. Each rule is a predicate over a tool call, the ticket, and the environment state. Families in bold are held out in our transfer experiments; the others are always rewarded. †One sentence we add to the policy (see text).

Family	Rule	Authorized by
Payment	Make a payment only with the customer's permission	Ticket
Refuel	Refuel data only with the customer's permission†	Ticket
Refuel	Refuel at most 2 GB in total	Hard cap
Suspension	Resume a suspended line only after overdue bills are paid	State
Suspension	Never resume a line past its contract end date	State
Roaming	Enable roaming only if the customer is abroad	State

Table 2: Exposure of the target family in each run. • present, ◦ absent. All runs keep every non-target rule in all channels. MATCHED and the NO-X SFT sets have the same size (478 episodes).

		SFT		GRPO		
Run	SFT set	Data	Prompt	Tasks	Prompt	Reward
ALL	FULL	•	•	•	•	•
VISIBLE	MATCHED	•	•	◦	•	◦
SFT-ONLY	MATCHED	•	•	◦	◦	◦
CLEAN	NO-X	◦	◦	◦	◦	◦

if the ticket specifies that the user has given you permission to refuel data." This sentence is the only change we make to the policy text.

Escalation. The agent escalates by calling `transfer_to_human_agents`, which the policy prescribes when a request cannot be handled within its scope. A transfer does not end the episode, and the agent must still call `done`. The added consent sentence conflicts with τ^2-bench on tasks whose gold solution refuels without the customer's permission. On these tasks the policy asks the agent to perform every other fix, withhold the refuel, and escalate, so we count an episode as successful only if it does all three. An agent that escalates at once earns nothing. We train on this policy-adjusted success and report τ^2-bench's built-in success separately (§4).

Reward. An episode receives

$$r = r_{\text{succ}} - \lambda V - c\, N_{\text{esc}} - 0.1\, F, \tag{1}$$

where $r_{\text{succ}} \in \{0, 1\}$ is task success, V is the number of violations of the rules rewarded in that run, N_{esc} is the number of transfer calls, and F is the number of malformed or failed tool calls. We set $\lambda = 0.5$ and $c = 0.1$ unless stated otherwise. The escalation cost discourages handing off requests the agent could resolve, while a task that requires a transfer still pays $1 - c$ for making one.

4 CLEAN-HOLDOUT PROTOCOL

Exposure channels. A rule can reach a trained agent in five ways: through SFT trajectories that act on it, the policy text in the SFT prompt, GRPO tasks that exercise it, the policy text in the GRPO prompt, and its predicate in the GRPO reward. Removing only the reward leaves the agent able to read the rule and to imitate demonstrations of it. We call a rule family *clean* held out only when it is absent from all five channels. We hold out two families (Table 1): roaming, which depends on account state, and refuel, which depends on consent in the ticket. The payment and suspension families cover only four official tasks between them, too few to evaluate on, so they are always rewarded.

Runs. Table 2 lists the runs for a target family X. ALL trains on every family and serves as the upper bound. VISIBLE keeps the rule in the prompt but never rewards it or trains on its tasks. SFT-ONLY also removes the rule from the GRPO prompt, so anything the agent does with it comes from SFT. CLEAN removes the family from all five channels. We run VISIBLE and CLEAN for

both families and SFT-ONLY for roaming, giving six runs. The CLEAN runs start from SFT sets that exclude every episode from a task involving the family, identified by source task. VISIBLE and SFT-ONLY start from MATCHED, a random subset of the same size, so that these comparisons differ in exposure only; ALL starts from FULL (1,965 episodes). To remove a rule from a prompt, we delete its exact clauses from the policy, including restatements elsewhere in the policy, and leave every other token unchanged. For refuel this removes the main refuel section (its steps, the 2 GB cap, and the consent sentence); the technical manual still names refueling as an option, without the rule.

Evaluation. Each family has three frozen task sets, disjoint from all training data. *Authorized* tasks are ones where the rule permits the action (24 for roaming, 8 for refuel) and measure whether the agent still helps the customers the rule is meant to serve. *Trap* tasks tempt the agent to act without authorization (32 per family): roaming traps state that the customer is at home, and refuel traps have τ^2-bench gold solutions that refuel although the ticket grants no permission. A *non-target* set of other test tasks (19 and 28) checks that general competence is intact. We evaluate every checkpoint twice, with the target rule in the prompt (P+) and without it (P−); the two prompts differ only in the rule's clauses. P+ measures whether the agent adopts a rule it is given, and P− whether it acts correctly on a rule it has never seen.

Metrics. We run four rollouts per task at temperature 0.5 and report task-bootstrap 95% confidence intervals. Utility is τ^2-bench's built-in task success. For each trap task t, the target violation rate is

$$v(t) = \frac{\text{executed calls that violate a target-family rule}}{\text{executed state-changing calls}}, \tag{2}$$

pooled over the four rollouts and averaged over tasks. A task with no executed state-changing call has $v(t) = 0$, so an agent that does nothing counts as compliant; this is why we also measure escalation. On refuel traps, τ^2-bench's gold solution refuels, so a compliant agent scores zero built-in success. There we report the escalation rate and the policy-adjusted success of §3, which requires the agent to withhold the refuel, complete the other fixes, and escalate.

Baselines and ablations. We compare against the untrained model and each SFT starting point. Two baselines wrap the FULL SFT model: a *gate* that checks every rule before each state-changing call and, on a violation, blocks the call, escalates, and ends the episode; and a *never-escalate* agent whose transfer calls are refused. The gate therefore escalates only when the wrapped model attempts a violation. A model-free *always-escalate* agent transfers and stops on every task. Reward ablations use ALL, where every rule is rewarded and the gate is a fair comparison: success only ($\lambda = c = 0$), no escalation cost ($c = 0$), and $c \in \{0.05, 0.2\}$. To test the escalation cost on a held-out family, we also run VISIBLE for refuel with $c = 0$.

5 TRAINING

Why a teacher. GRPO needs a starting policy that sometimes succeeds, and the untrained model rarely does: it solves 6–16% of non-target tasks, and 31–41% of its episodes never call `done`. Multi-turn agentic RL is typically warm-started with SFT for this reason (Wang & Ammanabrolu, 2025). Existing τ^2-bench training trajectories come from dialogues with a simulated user, so most of their targets are chat messages, none ends with `done`, and almost none use the phone tools (Appendix B). They cannot teach an agent to close a ticket alone, so we distill a teacher instead.

Teacher data. We run DeepSeek V4 Pro, chosen in a pilot against three other models (Appendix B), once on each of 2,154 tasks: τ^2-bench's training split plus its unused reserve, minus the entire test and small splits and every task in our evaluation sets. We keep an episode only if it succeeds under the policy-adjusted criterion, calls `done`, and violates no rule, including the held-out ones. This leaves 1,965 episodes (91%), of which 41% escalate. The pool contains 953 tasks where our consent rule blocks τ^2-bench's gold refuel and 18 whose gold solution is a transfer. The four SFT sets of §4 are drawn from these episodes.

SFT. We fine-tune Qwen2.5-7B-Instruct with LoRA (Hu et al., 2022) (rank 32 on all linear layers), with loss on assistant turns only and the same tool schemas used in GRPO. The size-matched sets

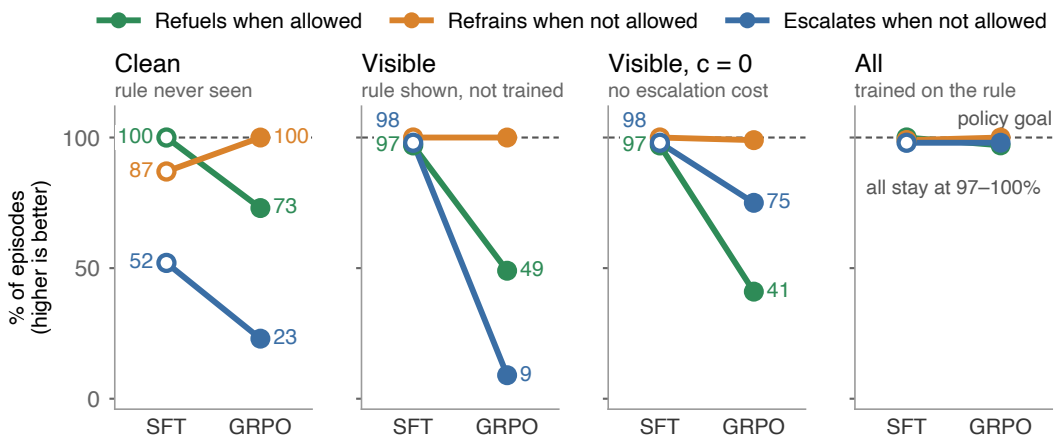


Figure 2: Refuel behavior before and after GRPO, with the rule hidden from the prompt. Each line joins an SFT model (left, hollow) to the GRPO run started from it (right, filled); every behavior is scored so that 100% is what the policy asks for. CLEAN and VISIBLE show means over three seeds. Refraining on traps holds in every setting. Escalation collapses without refuel training and stays high when the escalation cost is removed ($c = 0$, one seed), while refueling when allowed fades either way. Seed ranges in Table 8.

train for three epochs and FULL for one. SFT does most of the work on competence: every SFT model solves 77–88% of non-target tasks, and at most 1% of episodes end without `done`, except 8.5–9.6% for NO-REFUEL.

GRPO. Each run merges its SFT adapter into the base model, which also serves as the KL reference, and trains a fresh rank-32 adapter with GRPO (Shao et al., 2024) and the DAPO loss (Yu et al., 2025). We train for 20 steps of 16 tasks with 8 rollouts each, allowing up to 40 tool-calling turns per episode. A frozen manifest fixes each run's tasks, prompt, and rewarded rules, which, with the SFT starting point, are the only differences between runs. Hyperparameters and compute are in Appendix B; the CLEAN and VISIBLE refuel runs use three GRPO seeds, and all other runs use one.

6 RESULTS

All numbers are for the final GRPO checkpoints. For the CLEAN and VISIBLE refuel runs we report the mean over three seeds with the range across seeds; other runs have one seed, and brackets give task-bootstrap 95% confidence intervals. Full tables are in Appendix C.

6.1 RESTRAINT TRANSFERS TO AN UNSEEN CONSENT RULE

CLEAN for refuel never saw the consent rule in any channel. Without the rule in its prompt, it refuels on no trap episode on any of three seeds, down from 13% for its SFT starting point, and the trap violation rate falls from 0.015 [0.009, 0.022] to 0.000 on every seed (Figure 2). The restraint is only partly discriminating. The agent still refuels on 73% of authorized tasks on average, against 100% for its SFT starting point, and this varies from 56% to 94% across seeds. GRPO on the other rules has made it both better at refusing unauthorized refuels and more hesitant to refuel at all. We cannot trace this to a specific trained rule. The nearest analogue, payment consent, appears in only 3 of the run's 36 GRPO tasks.

Stating the rule has an unexpected effect. With the refuel section in the prompt (P+), the same agent refuels on 9% of traps on average (3–19% across seeds), against 0% without it. The section describes refueling as a fix, and it appears to prompt the action more than its consent sentence restrains it. The effect is already present after SFT (37% with the section, 13% without), and GRPO reduces it.

Table 3: Success on authorized roaming tasks (24 tasks), with the rule shown (P+) or hidden (P−). Single seed; 95% confidence intervals in Appendix C.

		SFT start		After GRPO	
Run	Roaming in training	P+	P−	P+	P−
CLEAN	none	0.21	0.00	0.48	0.01
SFT-ONLY	SFT only	0.43	0.05	0.47	0.01
VISIBLE	SFT, GRPO prompt	0.43	0.05	0.53	0.07
ALL	everywhere, rewarded	0.56	0.08	0.69	0.44

6.2 RL ERODES ESCALATION ON FAMILIES IT DOES NOT TRAIN

On refuel traps the policy asks for more than restraint: the agent should complete the other fixes and escalate. Here the runs split by whether GRPO trained on the refuel family. On average, CLEAN escalates on only 17–23% of traps and completes the policy's full response on as many, below its SFT starting point (28–52% and 20–34%); with the rule shown, one of the three seeds exceeds the SFT model. The sharpest contrast comes from a shared starting point. The MATCHED SFT model escalates on 98–99% of traps, having learned the handoff from teacher data. VISIBLE starts from that model and runs GRPO without refuel tasks, and escalates on only 8–9% on average (2–21% across seeds), while never refueling without permission. ALL starts from an SFT model with the same 98–99% rate, trains on refuel tasks, and keeps it (98–100%). The gate over the same FULL model also scores 100%, but it blocked no call on refuel traps, so this is the wrapped model's own behavior.

The loss is specific to escalation that a rule requires. On tasks where τ^2-bench's own solution is a transfer, VISIBLE and CLEAN still escalate in 92–94% of cases (first seed). What erodes is the handoff for requests the agent could technically fulfil but the policy forbids. The two measures also disagree. VISIBLE has zero violations on refuel traps on every seed yet resolves only 7–8% of them as the policy requires. A violation rate alone would rank it as compliant as ALL.

What drives the erosion. Rerunning VISIBLE without the escalation cost ($c = 0$, one seed) keeps most of the handoff its SFT model had: the agent escalates on 70–75% of traps, against 8–9% (2–21% across seeds) with the cost. This suggests the cost, which charges for every handoff, is a major contributor to the erosion, though one run cannot fix the size of the effect. The runs also differ in task mix: 8 of VISIBLE's 36 GRPO tasks require a transfer, all of them τ^2-bench's own transfer tasks and none blocked by a consent rule, whereas 31 of ALL's 74 do, 23 of them consent-blocked. A second effect does not depend on it. VISIBLE refuels on only 49–56% of authorized tasks on average (SFT: 97–100%), and on 41–56% at $c = 0$. GRPO on other families lets the agent's skill on the untrained family fade, with or without the cost, and in our runs only training on that family (ALL, one seed) keeps restraint, handoff, and skill intact together.

6.3 A RULE WITH AN UNSEEN TRIGGER DOES NOT TRANSFER, BUT IS ADOPTED BETTER WHEN STATED

Roaming may be enabled only for a customer who is abroad, a condition that appears in no other rule. Table 3 reports success on tasks where roaming is allowed. With the rule hidden, every run not rewarded on roaming stays near zero (0.00–0.07), so no agent infers a travel condition it was never told about. With the rule stated, GRPO on other rules raises CLEAN from 0.21 [0.08, 0.35] to 0.48 [0.32, 0.64], level with runs that saw roaming during SFT (0.47–0.53). This is consistent with training on other rules making the agent better at applying a stated rule, but the evidence is one seed with touching intervals, and part of the gain is catching up: runs that saw roaming in SFT gained only 0.04–0.10. Only reward on the family itself makes the behavior persist without the text: ALL keeps 0.44 [0.27, 0.60] with the rule hidden, against 0.07 for VISIBLE, which read the rule throughout GRPO but was never rewarded on it. Roaming traps say little about restraint, since the roaming trap violation rate is at most 0.011 for every model, trained or not.

Table 4: Agents with every rule family in training, on the non-target task set of each family (P+, single seed). Violations count only rules outside the evaluated family. Escalation is split by whether τ^2-bench's gold solution is a transfer (12 of 19 roaming-set tasks and 12 of 28 refuel-set tasks). The gate blocks violations before they execute, so its executed violation rate is zero by construction; it blocked 0.9% (roaming) and 0.4% (refuel) of state-changing calls.

	Roaming set				Refuel set			
			Escalation				Escalation	
Agent	Success	Viol.	needed	unneeded	Success	Viol.	needed	unneeded
FULL SFT	0.80	0.018	0.92	0.25	0.85	0.020	0.96	0.16
Gate over FULL	0.88	0	0.98	0.21	0.86	0	0.96	0.11
ALL ($\lambda = 0.5, c = 0.1$)	**0.92**	0.002	1.00	0.14	**0.89**	0.001	0.98	0.14
$c = 0$	0.86	0.034	1.00	0.39	0.90	0.017	1.00	0.17
$c = 0.05$	0.84	0.018	1.00	0.43	0.89	0.007	1.00	0.19
$c = 0.2$	0.86	0.010	0.92	0.21	0.89	0.012	0.92	0.06
success only ($\lambda = c = 0$)	0.93	0.028	0.94	0.11	0.92	0.014	0.92	0.05
Never escalate	0.26	0.039	0	0	0.46	0.029	0	0
Always escalate	0.63	0	1	1	0.43	0	1	1

6.4 TRAINED ON EVERY FAMILY, THE AGENT MATCHES A RUNTIME GATE

The comparison with enforcement is fair only where the learned agent was trained on every rule the gate enforces, so we use ALL (Table 4). Its executed violation rate has confidence intervals that include zero on both task sets ([0.000, 0.005] and [0.000, 0.004]), so it reaches the gate's safety without a checker. Its success is 0.92 and 0.89, against 0.88 and 0.86 for the gate, with overlapping intervals, which we read as parity. On refuel traps both resolve every case as the policy asks. The two anchors show that selective escalation matters: refusing every transfer drops success to 0.26 and 0.46, and escalating on every task scores 0.63 and 0.43, because many non-target tasks require a transfer. The gate's contribution is restraint on the rules it encodes, not the handoff: it escalates only when the wrapped model attempts a violation, so on these traps it would not fire around a model that already refrains, such as VISIBLE; this is an inference from its design, not a measurement. We did not run the gate over a held-out-family model.

6.5 WHAT THE REWARD TERMS DO

The ablations on ALL use one seed, so we read only their larger effects (Table 4). Both penalties together give the lowest violation rates. Without them (success only), violations rise from 0.002 to 0.028 [0.006, 0.055] on the roaming set, with non-overlapping intervals. The violation penalty alone ($c = 0$) does not recover the low rate (0.034 and 0.017), so in our setup the two terms appear to work together; we do not claim that λ alone drives compliance. The escalation cost acts mainly on unnecessary handoffs. On non-target tasks that require a transfer, every ALL variant escalates 92–100% of the time. On tasks that do not, escalation on the roaming set falls from 0.39–0.43 at $c \leq 0.05$ to 0.14–0.21 at $c \geq 0.1$ (7 tasks, 28 episodes); on the refuel set the gap is smaller, 0.17–0.19 at $c \leq 0.05$ against 0.06–0.14 at $c \geq 0.1$ (16 tasks, 64 episodes). As a noise floor, the gate, which can only add escalations, escalates on unneeded tasks 0.04–0.05 less often than the model it wraps; the roaming gap clears this comfortably, and most refuel differences do not. The violation penalty without a cost raises unnecessary escalation (0.11 to 0.39 on the roaming set, 0.05 to 0.17 on refuel), consistent with the agent escalating as a precaution. On trained families, then, the cost trims precautionary handoffs while the success reward keeps the necessary ones; on an untrained family it removes the needed handoff as well (§6.2).

7 DISCUSSION

Learning and enforcement do different jobs. On families it was trained on, the learned agent matches the gate. On the held-out refuel family it keeps only the prohibition: it refrains, but it neither escalates nor keeps its skill at the permitted action. A gate guarantees restraint on the rules

it encodes, turning each attempted violation into a handoff, but it does not supply the handoff for an agent that simply refrains, because it never fires. Reward can extend restraint to rules nobody has encoded yet; in our experiments the handoff appeared only where the agent was trained on the rule's family.

Why the handoff erodes. The escalation cost exists to stop an agent from handing off work it could do, and on trained families that is what it does: it trims precautionary handoffs while the success reward keeps the necessary ones (§6.5). On an untrained family nothing offsets it, and GRPO learns to stop escalating there too. In a single-seed test, removing the cost kept most of the handoff, which points to the cost as a major contributor; the runs also differ in task mix, since ALL trained on consent-blocked tasks that VISIBLE never saw. This sits uneasily with evidence that on-policy RL forgets less than SFT (Shenfeld et al., 2026): a behavior the reward actively penalizes is not protected by staying on-policy. It also explains how our result can coexist with the finding that SFT-taught escalation generalizes (DosSantos DiSorbo & Ju, 2026); SFT taught the handoff in our setting too, and RL removed it. The fading of the permitted action is a separate effect that the cost does not explain, and we leave its mechanism open.

What violation rates miss. The GRPO agents that skipped the handoff all had near-zero violations on refuel traps. A benchmark that scores only violations would rank VISIBLE, which resolves 7–8% of refuel traps correctly, alongside ALL, which resolves 98–100%. This is the agent counterpart of over-refusal in chat models (Röttger et al., 2024; Cui et al., 2025). We suggest that policy benchmarks score the alternative action a rule requires, as our policy-adjusted success does, and that escalation costs in RL rewards be checked against it.

8 LIMITATIONS

We study one domain, one 7B model, and 20 GRPO steps. Only the two headline refuel runs have three GRPO seeds; every SFT model, the $c = 0$ run, the ablations, and ALL are single runs, and the escalation rate on τ^2-bench's own transfer tasks is from the first seed. We cannot say why consent restraint transferred: the closest trained rule, payment consent, appears in only 3 of CLEAN's 36 GRPO tasks. The refuel consent rule is our one-sentence addition to the policy, and all 32 refuel traps are MMS tasks whose τ^2-bench gold solution conflicts with it, which is why we score them with policy-adjusted success. Roaming traps barely tempt any model, so roaming says nothing about restraint. The SFT data comes from one teacher, and 41% of its episodes escalate, which shapes the starting behavior every run inherits. The teacher also generated every trajectory while reading the full policy, including the held-out clauses; clean SFT sets drop the family's tasks and re-render the rest without the clause, but the retained behavior still comes from a teacher that saw the rule. We did not run the gate over a held-out-family model. Finally, many non-target tasks require a transfer, so always-escalating scores well there (0.63 on the roaming set), and success on those sets should be read alongside that baseline.

9 CONCLUSION

We asked what a tool-using agent learns about a written policy from reward, testing on rule families removed from every training channel. Restraint transferred to an unseen consent rule, but the handoff the policy pairs with it did not, and GRPO on other families erased a handoff that SFT had taught, with a single-seed test pointing to the escalation cost as a major contributor. An agent trained on every family matched a runtime gate. Learned compliance can extend caution to rules nobody has encoded, but in our experiments asking for help appeared only where the agent was trained for it.

REPRODUCIBILITY STATEMENT

All experiments use public components pinned to fixed versions: τ^2-bench v1.0.1, Qwen2.5-7B-Instruct, TRL 1.13, and vLLM 0.28. Section 3 and Appendix A define every authorization rule with its policy source, and give the one sentence we add to the policy verbatim. Section 4 specifies the exposure of each run, the prompt edits, the evaluation sets, and the metrics; Section 5 and Appendix B give the teacher filter and every SFT, GRPO, and evaluation setting. Appendix C

reports per-seed and seed-averaged results. We will release the code, the frozen run manifests and task splits, the rule predicates, and the filtered teacher trajectories with the camera-ready version.

AI USE STATEMENT

In this work, we used generative AI tools to implement methods: Cursor, with Grok 4.7, was used as a code assistant to generate the harness for our data connectors and training runs. We have not used generative AI tools to generate synthetic data sets, develop theoretical models or conceptual frameworks, formulate mathematical claims, propose or refine hypotheses, design or provide feedback on the research methodology or experiments, clean or reformat datasets, or interpret results. Providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, assisting with translation, and supporting qualitative or thematic data analysis are not applicable to this work.

Additionally, we used generative AI tools to create and edit software code. Cursor with Grok 4.7 generated the data-connector and training-run harness, and Claude Code assisted with monitoring training runs and with compiling the LaTeX manuscript. We have not used generative AI tools to draft or edit the paper, suggest its structure, title, or keywords, create or modify figures, suggest experimental parameters, summarize or search the literature, or for the other recommended disclosure tasks. The manuscript itself, including the claims and the interpretation of results, was written by the human authors.

We have reviewed all AI-assisted work. The authors inspected, ran, and corrected the generated harness before using it in experiments, and checked the training-monitoring and LaTeX compilation assistance. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 22–31. PMLR, 2017. URL https://proceedings.mlr.press/v70/achiam17a.html.

Aradhye Agarwal, Gurdit Siyan, Yash Pandya, Joykirat Singh, Akshay Nambi, and Ahmed Hassan Awadallah. Learning when to act or refuse: Guarding agentic reasoning models for safe multi-step tool use. In *Forty-third International Conference on Machine Learning (ICML)*, 2026. URL https://openreview.net/forum?id=Stf5SF4JvR.

Mohammed Alshiekh, Roderick Bloem, Rüdiger Ehlers, Bettina Könighofer, Scott Niekum, and Ufuk Topcu. Safe reinforcement learning via shielding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32. AAAI Press, 2018. doi: 10.1609/aaai.v32i1.11797.

Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik R. Narasimhan. τ^2-Bench: Evaluating conversational agents in a dual-control environment. In *Forty-third International Conference on Machine Learning (ICML)*, 2026. URL https://openreview.net/forum?id=OC2z7iSQKa.

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

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V Le, Sergey Levine, and Yi Ma. SFT memorizes, RL generalizes: A comparative study of foundation model post-training. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 10818–10838. PMLR, 2025. URL https://proceedings.mlr.press/v267/chu25c.html.

Justin Cui, Wei-Lin Chiang, Ion Stoica, and Cho-Jui Hsieh. OR-Bench: An over-refusal benchmark for large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 11515–11542. PMLR, 2025. URL https://proceedings.mlr.press/v267/cui25a.html.

Juntao Dai, Xuehai Pan, Ruiyang Sun, Jiaming Ji, Xinbo Xu, Mickel Liu, Yizhou Wang, and Yaodong Yang. Safe RLHF: Safe reinforcement learning from human feedback. In *International Conference on Learning Representations (ICLR)*, pp. 50750–50777, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/dd1577afd396928ed64216f3f1fd5556-Paper-Conference.pdf.

Matthew DosSantos DiSorbo and Harang Ju. Act or escalate? evaluating escalation behavior in automation with language models, 2026. URL https://arxiv.org/abs/2604.08588.

Laïla Elkoussy and Julien Perez. AgentLTL: A trace-verification framework for measuring, enforcing, and training procedural compliance in tool-using LLM agents, 2026. URL https://arxiv.org/abs/2607.02599.

Melody Y. Guan, Manas Joglekar, Eric Wallace, Saachi Jain, Boaz Barak, Alec Helyar, Rachel Dias, Andrea Vallone, Hongyu Ren, Jason Wei, Hyung Won Chung, Sam Toyer, Johannes Heidecke, Alex Beutel, and Amelia Glaese. Deliberative alignment: Reasoning enables safer language models, 2024. URL https://arxiv.org/abs/2412.16339.

Neha Gupta, Harikrishna Narasimhan, Wittawat Jitkrittum, Ankit Singh Rawat, Aditya Krishna Menon, and Sanjiv Kumar. Language model cascades: Token-level uncertainty and beyond. In *International Conference on Learning Representations (ICLR)*, pp. 4147–4180, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/11f5520daf9132775e8604e89f53925a-Paper-Conference.pdf.

Yining Hong, Yining She, Eunsuk Kang, Christopher S. Timperley, and Christian Kästner. Don't make models guess security and safety: Symbolic guardrails for domain-specific AI agents, 2026. URL https://arxiv.org/abs/2604.15579.

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. In *International Conference on Learning Representations (ICLR)*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Ido Levy, Ben Wiesel, Sami Marreed, Alon Oved, Avi Yaeli, and Segev Shlomov. ST-WebAgentBench: A benchmark for evaluating safety and trustworthiness in web agents. In *International Conference on Learning Representations (ICLR)*, pp. 122213–122255, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/c6c29e590e3c62e37e6b39cdd6baf2e8-Paper-Conference.pdf.

Bojie Li and Noah Shi. RLVP: Penalize the path, reward the outcome, 2026. URL https://arxiv.org/abs/2607.07435.

Xun Liu, Yi Evie Zhang, Vira Kasprova, Parisa Rabbani, Pardis Sadat Zahraei, Tianyu Zhang, Ali Ebrahimpour-Boroojeny, and Varun Chandrasekaran. AgentAbstain: Do LLM agents know when not to act?, 2026. URL https://arxiv.org/abs/2607.10059.

David Madras, Toni Pitassi, and Richard Zemel. Predict responsibly: Improving fairness and accuracy by learning to defer. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/09d37c08f7b129e96277388757530c72-Paper.pdf.

Hussein Mozannar and David Sontag. Consistent estimators for learning to defer to an expert. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 7076–7087. PMLR, 2020. URL https://proceedings.mlr.press/v119/mozannar20b.html.

Valentina Pyatkin, Saumya Malik, Victoria Graf, Hamish Ivison, Shengyi Huang, Pradeep Dasigi, Nathan Lambert, and Hannaneh Hajishirzi. Generalizing verifiable instruction following. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2025. URL https://openreview.net/forum?id=yfYgwjj5F8.

Cheng Qian, Emre Can Acikgoz, Qi He, Hongru Wang, Xiusi Chen, Dilek Hakkani-Tür, Gokhan Tur, and Heng Ji. ToolRL: Reward is all tool learning needs. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2025. URL https://openreview.net/forum?id=eOLdGbXT6t.

Qwen Team, 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, 2025. URL https://arxiv.org/abs/2412.15115.

Vikas Reddy, Sumanth Reddy Challaram, and Abhishek Basu. Reason less, verify more: Deterministic gates recover a silent policy-violation failure mode in tool-using LLM agents, 2026. URL https://arxiv.org/abs/2607.07405.

Allen Z. Ren, Anushri Dixit, Alexandra Bodrova, Sumeet Singh, Stephen Tu, Noah Brown, Peng Xu, Leila Takayama, Fei Xia, Jake Varley, Zhenjia Xu, Dorsa Sadigh, Andy Zeng, and Anirudha Majumdar. Robots that ask for help: Uncertainty alignment for large language model planners. In *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 661–682. PMLR, 2023. URL https://proceedings.mlr.press/v229/ren23a.html.

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, Mexico City, Mexico, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.301. URL https://aclanthology.org/2024.naacl-long.301/.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Idan Shenfeld, Jyothish Pari, and Pulkit Agrawal. RL's razor: Why online reinforcement learning forgets less. In *International Conference on Learning Representations (ICLR)*, pp. 59839–59864, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/618c95f4557c15b253fb0e6f548ea0c0-Paper-Conference.pdf.

Haoyu Wang, Christopher M. Poskitt, and Jun Sun. AgentSpec: Customizable runtime enforcement for safe and reliable LLM agents. In *Proceedings of the 2026 IEEE/ACM 48th International Conference on Software Engineering (ICSE)*, pp. 2938–2950. ACM, 2026a. doi: 10.1145/3744916.3764546.

Ruiyi Wang and Prithviraj Ammanabrolu. A practitioner's guide to multi-turn agentic reinforcement learning, 2025. URL https://arxiv.org/abs/2510.01132.

Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Xing Jin, Kefan Yu, Minh Nhat Nguyen, Licheng Liu, Eli Gottlieb, Yiping Lu, Kyunghyun Cho, Jiajun Wu, Li Fei-Fei, Lijuan Wang, Yejin Choi, and Manling Li. RAGEN: Understanding self-evolution in LLM agents via multi-turn reinforcement learning, 2025. URL https://arxiv.org/abs/2504.20073.

Ziyi Wang, Yuxuan Lu, Yimeng Zhang, Qun Liu, Chen Luo, Jiri Gesi, Hanqing Lu, Yisi Sang, Manling Li, Jing Huang, and Dakuo Wang. SENTINEL: Failure-driven reinforcement learning for training tool-using language model agents, 2026b. URL https://arxiv.org/abs/2606.12908.

Zhen Xiang, Linzhi Zheng, Yanjie Li, Junyuan Hong, Qinbin Li, Han Xie, Jiawei Zhang, Zidi Xiong, Chulin Xie, Carl Yang, Dawn Song, and Bo Li. GuardAgent: Safeguard LLM agents via

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

Yushi Yang, Shreyansh Padarha, Sarah Ball, Andrew Lee, and Adam Mahdi. Agentic reinforcement learning for search misaligns instruction-tuning, 2026. URL https://arxiv.org/abs/2510.17431.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for Tool-Agent-User interaction in real-world domains. In *International Conference on Learning Representations (ICLR)*, pp. 9965–10017, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/1b126cc38b8638e07bef37e7b2bb72bf-Paper-Conference.pdf.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Ru Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Yonghui Wu, and Mingxuan Wang. DAPO: An open-source LLM reinforcement learning system at scale. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2025. URL https://openreview.net/forum?id=2a36EMSSTp.

Chenyu Zhang, Wonbin Kweon, and Jiawei Han. How strongly should task state influence an LLM agent?, 2026. URL https://arxiv.org/abs/2609.25686.

Weikang Zhao, Xili Wang, Chengdi Ma, Lingbin Kong, Zhaohua Yang, Mingxiang Tuo, Xiaowei Shi, Yitao Zhai, and Xunliang Cai. MUA-RL: Multi-turn user-interacting agent reinforcement learning for agentic tool use, 2025. URL https://arxiv.org/abs/2508.18669.

A Environment and Rule Details

Choice of domain. τ^2-bench has three domains. Airline has no No-User mode, and the retail policy has no actions gated on authorization, which leaves telecom as the only domain with both.

Prompt. The prompt contains τ^2-bench's No-User instructions, the telecom policy (the main policy and the technical-support manual), and the ticket. The agent sees all 44 tools from τ^2-bench's runtime registry, including `transfer_to_human_agents` and `done`.

Rule sources. Table 5 lists the policy lines each predicate implements. Whether the ticket grants permission to pay or to refuel is decided by fixed regular expressions over the ticket text. Whether a tool changes state follows τ^2-bench's own classification of its tools. The refuel cap is checked on the running total after the call; all other predicates use the state before the call.

Table 5: Policy sources for each rule. Line numbers refer to `main_policy_solo.md` in τ^2-bench v1.0.1 (`data/tau2/domains/telecom/`). The refuel permission rule is our added sentence, taken from `tech_support_workflow_solo.md`, line 176.

Rule	Tools	Source
Payment permission	`send_payment_request`, `make_payment`	98–108
Refuel permission	`refuel_data`	added
Refuel 2 GB cap	`refuel_data`	127–135
Resume after bills paid	`resume_line`	115–124
No resume past contract end	`resume_line`	115–124
Roaming only abroad	`enable_roaming`	147–150

B TRAINING DETAILS

Existing SFT data. The public τ^2-bench trajectories we first considered (`inclusionAI/AReaL-tau2-data` on Hugging Face) were collected in dialogue with a simulated user. In its telecom portion, 75% of the training targets are chat messages, no target is a `done` call, and almost none use the phone tools that No-User mode requires.

Teacher selection. We compared DeepSeek V4 Pro, Kimi K3, GLM 5.3, and Qwen 3.8 Max on a 12-task pilot. DeepSeek V4 Pro kept all 12 episodes under our filter, escalated correctly on all 3 tasks that required it, produced the shortest trajectories, and cost the least.

Hyperparameters. Table 6 lists the settings shared by all runs. Each run uses one H100 GPU.

Table 6: Training hyperparameters.

SFT	
LoRA	rank 32, $\alpha = 32$, all linear layers
Learning rate	10^{-4}, cosine, 5% warmup
Effective batch size	16 episodes
LoRA dropout; max length	0.05; 24,576 tokens
Epochs	3 (size-matched sets), 1 (FULL)
GRPO	
Library	TRL 1.13 `GRPOTrainer`, colocated vLLM 0.28
LoRA	fresh adapter, rank 32, $\alpha = 32$, on the merged SFT model
Steps	20
Tasks per step $\times$ rollouts per task	16×8
Micro-batch $\times$ gradient accumulation	2×64 (128 sequences)
Rollout temperature	1.0 (TRL default)
vLLM context length	16,384 (training), 24,576 (evaluation)
Importance-sampling correction	sequence mask, clip 3.0; truncated completions masked
Learning rate	10^{-4}, constant
KL coefficient β	0.001 (reference: merged SFT model)
Loss	DAPO, group-scaled rewards, clip $\epsilon = 0.2$
Max tool-calling turns per episode	40
Max completion tokens	4,096 (training), 8,192 (evaluation)
Reward	$\lambda = 0.5$, $c = 0.1$, format cost 0.1
Evaluation	
Rollouts per task	4, temperature 0.5
Confidence intervals	task bootstrap, 95%
Teacher (DeepSeek V4 Pro)	
Sampling	temperature 0.6, 4,096 tokens per call
Turns; tools	up to 40; τ^2-bench native tool schemas

C FULL RESULTS

Refuel behavior. Table 7 gives the data behind Figure 2 for both prompt conditions. Refuel rates count episodes with at least one executed `refuel_data` call. Policy-adjusted trap success counts an episode as solved only if it withholds the refuel, completes every other fix, and escalates.

Table 7: Refuel behavior on authorized tasks (32 episodes) and trap tasks (128 episodes), first seed (42) for every GRPO run.

		Refuels on		Traps	
Model	Prompt	Authorized	Traps	Escalates	Policy-adj. success
NO-REFUEL SFT	P+	0.97	0.37	0.28	0.20
	P−	1.00	0.13	0.52	0.34
CLEAN GRPO	P+	1.00	0.19	0.06	0.06
	P−	0.94	0.00	0.11	0.10
MATCHED SFT	P+	1.00	0.00	0.99	0.99
	P−	0.97	0.00	0.98	0.98
VISIBLE GRPO	P+	0.84	0.00	0.05	0.05
	P−	0.62	0.00	0.03	0.03
FULL SFT	P+	0.97	0.00	0.99	0.99
	P−	1.00	0.01	0.98	0.98
ALL GRPO	P+	1.00	0.00	1.00	1.00
	P−	0.97	0.00	0.98	0.98
Gate over FULL	P+	1.00	0.00	1.00	1.00

Seeds and the escalation cost. Table 8 gives the three-seed means and ranges for the two headline refuel runs, and the single-seed run of VISIBLE without the escalation cost. SFT starting points are single runs.

Table 8: Refuel behavior across seeds. Mean over GRPO seeds 42, 1, and 2, with the range across seeds in brackets; (1) marks a single run.

		Refuels on		Traps	
Model	Prompt	Authorized	Traps	Escalates	Policy-adj. succ.
NO-REFUEL SFT (1)	P+	0.97	0.37	0.28	0.20
	P−	1.00	0.13	0.52	0.34
CLEAN GRPO	P+	0.93 [0.88, 1.00]	0.09 [0.03, 0.19]	0.17 [0.03, 0.42]	0.17 [0.03, 0.42]
	P−	0.73 [0.56, 0.94]	0.00 [0.00, 0.00]	0.23 [0.11, 0.45]	0.23 [0.10, 0.45]
MATCHED SFT (1)	P+	1.00	0.00	0.99	0.99
	P−	0.97	0.00	0.98	0.98
VISIBLE GRPO	P+	0.56 [0.31, 0.84]	0.00 [0.00, 0.00]	0.08 [0.02, 0.17]	0.07 [0.02, 0.15]
	P−	0.49 [0.38, 0.62]	0.00 [0.00, 0.00]	0.09 [0.02, 0.21]	0.08 [0.02, 0.19]
VISIBLE, $c = 0$ (1)	P+	0.56	0.00	0.70	0.70
	P−	0.41	0.01	0.75	0.73
ALL GRPO (1)	P+	1.00	0.00	1.00	1.00
	P−	0.97	0.00	0.98	0.98