OPERANT CONDITIONING FOR LANGUAGE-MODEL AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Can language-model agents be conditioned by the consequences of their own actions? We introduce Operant Contingency Learning (OCL), which constructs policy targets from resource and burden changes while preserving their identity, direction, context, and current relevance. Across Qwen3-0.6B and Gemma-3-1B-IT with three seeds each, a controlled conditioning assessment recovers all four canonical operant effects. After 960 trials, positive and negative reinforcement increase held-out action probability by 21.50 and 17.75 percentage points relative to exposure-matched yoked controls, while positive and negative punishment decrease it by 10.16 and 9.10 points. All 24 model–seed–effect combinations change in the predicted direction relative to their calibrated initial policies, while yoked changes show no consistent direction. Separate skill-acquisition experiments use nine seeds per model on TextWorld and a simplified Telecom task derived from τ^2-bench. Under stochastic action selection, OCL-C without scalar loss achieves 77.15% error-free success on Telecom versus 59.65% for Scalar RL, and 14.55% on TextWorld versus 5.60% for Scalar RL. Both advantages occur in all 18 paired model–seed runs on each task and survive Holm correction across eight clean-success comparisons. OCL-C combined with scalar loss achieves 65.25% stochastic clean success on Telecom and 18.87% on TextWorld, revealing different observed rankings of the two OCL variants across tasks. In exploratory Telecom comparisons, OCL-C without scalar loss also reduces mean remaining burden by 45.2% relative to Scalar RL and reaches confirmed greedy mastery in 14/18 runs, compared with 11/18 for OCL-C combined with scalar loss and 5/18 for Scalar RL. During skill-acquisition, OCL-C without scalar loss exhibited mean action-probability changes in the predicted direction across all 16 task–model–consequence categories, compared with 14 for OCL-C with scalar loss and 13 for Scalar RL. These findings support typed action consequences as an effective basis for operant conditioning and task acquisition, including policy learning without a scalar task-reward loss.

1 INTRODUCTION

A language-model agent's action can advance its task while consuming a needed resource or creating a new burden. Scalar reward and score-function learning provide effective tools for goal-directed behavior (Williams, 1992; Sutton et al., 2000; Sutton & Barto, 2018), including interactive language tasks (Carta et al., 2023; Zhou et al., 2024; Gaven et al., 2024; Abdulhai et al., 2025). A scalar signal, however, does not explicitly retain which conditions changed or why those changes matter now. Multi-objective and decomposed-reward methods already address related representation questions (Roijers et al., 2013; van Seijen et al., 2017). Here we ask whether typed action consequences can support operant conditioning and task acquisition without a goal-directed scalar task-reward loss.

Operant conditioning relates antecedent context, response, and consequence (Skinner, 1938; 1953). Conventionally, *positive* and *negative* denote addition and removal of a stimulus (Keller & Schoenfeld, 1950), although their interpretation depends on the specified environmental transition (Michael, 1975; Baron & Galizio, 2005). We adopt functional definitions: *reinforcement* and *punishment* require increases and decreases in later responding caused by the response–consequence relation (Cooper et al., 2020; Holth, 2005). We therefore distinguish programmed resource gain, resource loss, burden addition, and burden removal from their measured behavioral effects (Figure 1). Operant conditioning is a behavioral framework; scalar RL can also exhibit operant behavior as a byproduct of training.

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

We introduce **Operant Contingency Learning (OCL)**, a novel policy-target construction mechanism designed to elicit predictable operant behavior while retaining consequence identity, direction, context, and current relevance. Current consequences and relevance determine the selected action's target; contextual memory guides redistribution among alternatives. To our knowledge, online language-model training from typed consequences with controlled tests of all four operant effects remains underexplored. We test unary conditioning against exposure-matched yoked controls and compare contextual OCL (OCL-C), with and without scalar task-reward loss, against Scalar RL in skill acquisition. Both OCL variants retain terminal-outcome credit on decisions with no measured consequence change. All methods share the public policy interface, optimizer, and entropy regularization; OCL receives richer consequence supervision. Memory supplies training targets; evaluation uses only the trained policy and public inputs.

Across Qwen3-0.6B and Gemma-3-1B-IT, a controlled assay with three seeds per model recovers all four operant effects without scalar reward or entropy regularization. All 24 model–seed–effect combinations shift in the predicted direction against calibrated initialization and exposure-matched yoked controls. Separate acquisition experiments use nine seeds per model on TextWorld and a simplified Telecom task derived from τ^2-bench. Without scalar loss, OCL-C achieves 77.15% versus 59.65% stochastic clean success on Telecom and 14.55% versus 5.60% on TextWorld, improving on Scalar RL in all 18 paired runs per task. Both pooled advantages survive Holm correction. Adding scalar loss yields different observed rankings of the two OCL variants across tasks.

The study contributes a target-construction mechanism, a controlled test of all four operant effects, and a skill-acquisition analysis separating success, remaining burden, violations, damage, and observed action-probability changes. These results establish typed consequences as an effective basis for learning without a scalar task-reward loss, while the mixed harm outcomes show why success and error-free execution require separate evaluation.

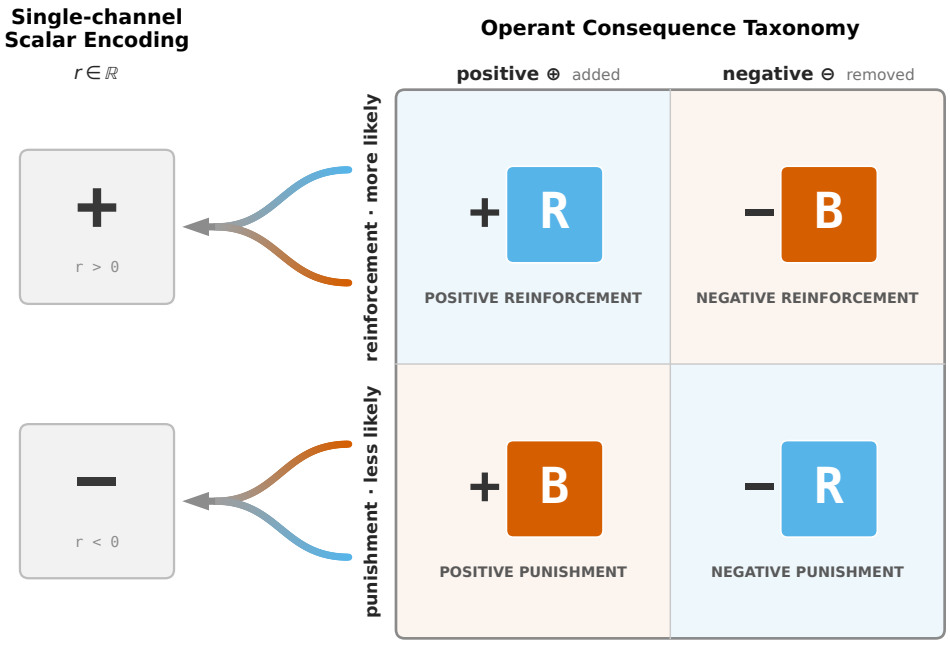

Figure 1: **Scalar feedback and typed consequences.** This conceptual schematic contrasts a scalar signal with resource (R) and burden (B) changes. Positive and negative denote addition and removal of a stimulus; reinforcement and punishment require observed increases and decreases in later responding.

2 Related Work

Scalar-feedback post-training and online reinforcement learning are well established for language models and language agents (Stiennon et al., 2020; Ouyang et al., 2022; Carta et al., 2023; Zhou

et al., 2024; Gaven et al., 2024; Abdulhai et al., 2025). Multi-objective learning and factored policy gradients retain distinct objectives (Roijers et al., 2013; Spooner et al., 2021); successor features and general value functions retain separate predictive quantities (Barreto et al., 2017; Sutton et al., 2011). Hybrid Reward Architecture learns component value functions before aggregation (van Seijen et al., 2017), whereas reward machines expose temporal and logical reward structure (Icarte et al., 2018). MPO separates distribution improvement from policy fitting, and MO-MPO fits a policy to objective-specific action distributions (Abdolmaleki et al., 2018; 2020). OCL constructs targets from identified consequence changes and current relevance, with contextual memory redistributing probability among alternatives. Its distinction and novelty lie in the target construction. OCL combines target fitting with an optional scalar task-reward loss; typed consequences modify policy targets rather than augmenting task rewards through reward shaping (Ng et al., 1999).

Language-agent memories retain reflections, outcome-derived lessons, retrieved successful trajectories, skills, or structured experience for later prompting (Shinn et al., 2023; Zhao et al., 2024; Wang et al., 2024; Majumder et al., 2024). Rememberer already maintains contextual numeric action-value estimates: it estimates cumulative scalar returns and retrieves encouraged and discouraged actions as prompt examples, without updating language-model weights (Zhang et al., 2023). OCL instead records criterion-specific consequence changes and uses memory to construct training targets. Evaluation uses the trained model and adapter with public observations and within-episode history, without consulting or updating consequence memory. Appendix D expands these comparisons.

The behavioral foundation is the antecedent–response–consequence relation (Skinner, 1938; 1953). We adopt functional definitions: reinforcement increases later responding, and punishment decreases it (Cooper et al., 2020; Holth, 2005). Motivating operations alter outcomes' current effectiveness (Michael, 1982; Laraway et al., 2003); homeostatic RL provides a computational precedent by defining scalar reward through reductions in physiological deviation (Keramati & Gutkin, 2014). OCL uses supplied task-need predicates to determine beneficial consequence relevance while retaining punitive relevance for harm. Response dependence is distinct from consequence exposure (Lattal, 1974; Hammond, 1980); our behavioral assessment examines that distinction through exposure-matched yoking.

3 Operant Contingency Learning

Operant conditioning treats behavior as selected by its consequences (Skinner, 1981). It is possible for Scalar RL to exhibit operant behavior, though it is not guaranteed. OCL is one computational realization: our method retains what changed, whether stimuli were added or removed, whether or not that change matters in the current context, and then constructs a target distribution over all possible and available actions. We examine contextual OCL (OCL-C), with and without a scalar task-reward loss, against Scalar RL. Figure 2 summarizes skill-acquisition training.

Actions and consequences. Given public observation and history x, the language model assigns a probability $\pi_\theta(a \mid x)$ to each member of a finite action set $\mathcal{A}(x)$ through a hierarchical distribution over tool keys and their complete argument choices. Training samples from this distribution; greedy evaluation selects the action most probable, while stochastic evaluation chooses an action randomly from the learned probability distribution. For decision i, let a_i be the sampled action and $p_i = \pi_{\theta_g}(\cdot \mid x_i)$ the distribution during collection of group g. Each measured criterion c supplies a signed change d_{ic} and a resource or burden identity. Define

$$e_{ic} = \begin{cases} d_{ic}, & c \text{ is a resource,} \\ -d_{ic}, & c \text{ is a burden.} \end{cases} \tag{1}$$

Resource addition and burden removal therefore favor increasing an action's response in a situation or context, while resource removal and burden addition favor a decrease. Beneficial changes require task-defined current relevance and affordability, while harmful changes retain punitive relevance.

Contextual consequence memory. For each criterion, OCL-C stores action-specific consequence estimates indexed by the ordered action set and public antecedent context, together with a fallback pooled across contexts of that action set. After each observation, both estimates for the chosen action are initialized to e_{ic} or updated as

$$M_{c,x,a} \leftarrow (1-\alpha)M_{c,x,a} + \alpha e_{ic}, \qquad \alpha = 0.5. \tag{2}$$

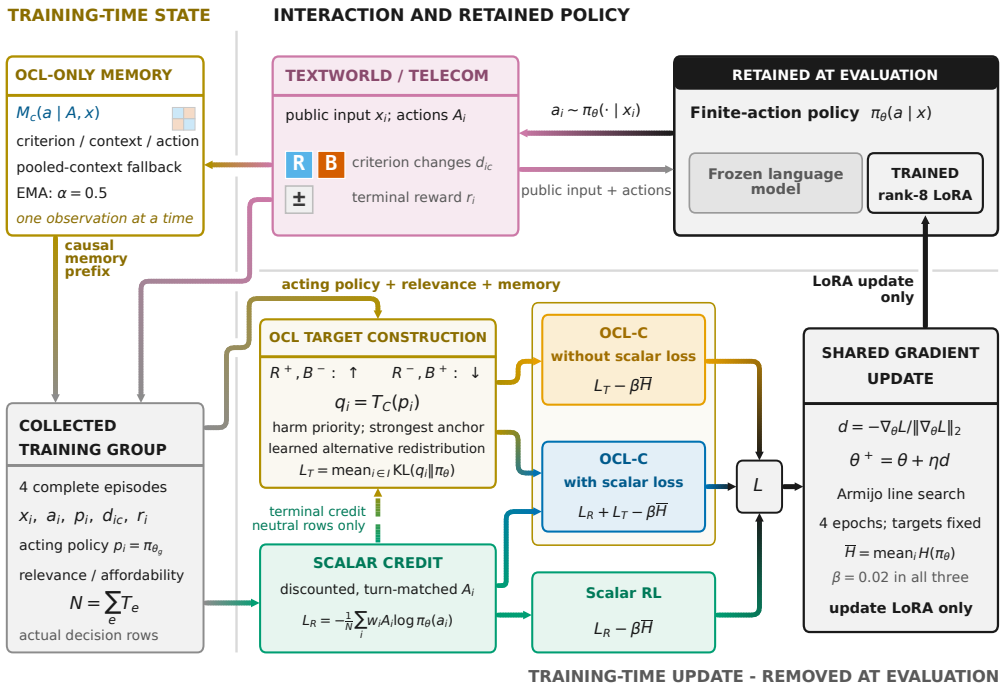

Figure 2: **Skill-acquisition training.** OCL freezes targets anchored in recorded acting probabilities, using contextual memory, relevance, and criterion changes, including derived opportunity losses. Dashed terminal credit applies only when all measured changes are zero, including without scalar loss. All objectives retain entropy. I indexes active targets and w_i denotes detached, clipped importance weights. Evaluation retains only the policy and public interface.

Each decision uses only the chronological prefix through its own observation. After the chosen action yields a benefit, memory downweights the most harmful remembered alternative; after harm, it upweights the most beneficial affordable alternative. Comparisons are criterion-specific, with contextual estimates preceding pooled fallback. The chosen action is always conditioned, and its current consequence determines the direction, even when historical estimates disagree.

From consequences to targets. For each eligible nonzero criterion, let $s_{ic} = \text{sign}(e_{ic})$, let $r_{ic} \in [0, 1]$ encode beneficial relevance (one for harm), and set $u_{ic} = 0.4 s_{ic} r_{ic} |d_{ic}|$; count changes retain their integer magnitude, and Boolean states are encoded as $0/1$. With counterpart b_{ic}, define $v_{icb} = -u_{ic} \mathbf{1}\{b = b_{ic}\}$; without one, all $v_{icb} = 0$. The criterion target is

$$q^c_{ia_i} = \sigma(\text{logit}\, p_{ia_i} + u_{ic}) \,, \tag{3}$$

$$q^c_{ib} = (1 - q^c_{ia_i}) \frac{p_{ib} \exp(v_{icb})}{\sum_{j \neq a_i} p_{ij} \exp(v_{icj})}, \qquad b \neq a_i. \tag{4}$$

Thus the observed action changes relative to all alternatives. Memory redistributes the remaining probability without weakening that action's prescribed change.

When any harmful criterion occurs, OCL retains eligible harmful criterion targets; otherwise OCL retains eligible beneficial targets. Among these, choose the smallest focal-action probability after harm or the largest after benefit, denoted $q^*_{ia_i}$. For their mean target $\bar{q}_i$, combine them as

$$q_{ia_i} = q^*_{ia_i}, \qquad q_{ib} = (1 - q^*_{ia_i}) \frac{\bar{q}_{ib}}{\sum_{j \neq a_i} \bar{q}_{ij}} \quad (b \neq a_i). \tag{5}$$

This prioritizes harm while preserving the strongest focal change and the alternatives preferred across selected criteria. Without an eligible target, $q_i = p_i$. Only when every measured criterion has zero change, terminal-return credit supplies a unary target using $u_i = 0.4\, \text{clip}(A_i, -1, 1)$ and $v_i = 0$. This credit serves as statistical outcome supervision.

Three learning objectives. For four collected episodes, discounted returns use $\gamma = 0.97$. Subtract the mean return at the same turn in the other three episodes, then divide advantages by their root-mean-square over actual decisions. For baseline computation, returns are zero beyond each episode's final turn. With N decisions and $\mathcal{I} = \{i : q_i \neq p_i\}$, minimize

$$\mathcal{L}(\theta) = -\frac{\lambda_R}{N} \sum_i w_i A_i \log \pi_\theta(a_i \mid x_i) - \frac{\beta}{N} \sum_i H(\pi_\theta(\cdot \mid x_i)) + \frac{\lambda_T}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \mathrm{KL}(q_i \,\|\, \pi_\theta(\cdot \mid x_i)) \,. \tag{6}$$

The scalar term uses REINFORCE-style score-function learning (Williams, 1992). Advantage normalization, clipped weights, and four-epoch fitting are study-specific choices. The typed term is zero when $\mathcal{I}$ is empty. The settings (λ_R, λ_T) are $(1, 0)$ for Scalar RL, $(1, 1)$ for OCL-C with scalar loss, and $(0, 1)$ for OCL-C without scalar loss. All skill-acquisition conditions use $\beta = 0.02$. Scalar RL constructs no typed targets or memory.

Targets remain fixed across four optimization epochs. At each update start, detached importance weights are $w_i = \min\{10, \pi_{\theta_{\text{start}}}(a_i \mid x_i)/p_{ia_i}\}$, with exact unit weights initially. They remain fixed during step selection. A bounded Armijo search selects a step along the unit-length negative objective gradient, updating only low-rank adapters (LoRA; Hu et al., 2022).

Removing scalar loss therefore removes the goal-directed score-function term, while retaining typed feedback, neutral-decision terminal credit, and acquisition entropy. The comparison tests these complete learning procedures. At evaluation, actions depend only on the trained model and public input, without consulting consequence memory. App. A specifies target selection, numerical optimization, task feedback.

4 Experimental Procedure

Qwen3-0.6B (Yang et al., 2025) and Gemma-3-1B-IT (Gemma Team et al., 2025) share the Colab G4 environment (RTX 6000 PRO Blackwell).

Controlled conditioning. Seeds $3, 7, 11$/model test resource/burden addition/removal in eight training and eight held-out six-action relay contexts/effect, balanced for resource demand/burden sensitivity. Training-only calibration supplies identical parameters, with initial action probabilities within 0.01 of uniform. Each condition samples actions across 16 groups of 60 independently reset single-action trials/effect (960). Contingent packets contain tested realized signed consequences and eligibility. Yoked replay matches antecedents and deliveries, including zeros and eligibility, independently of yoked actions. Scalar reward and entropy regularization are zeroed to isolate unary typed-consequence conditioning. Counterpart redistribution is inactive. Each group receives four fitting epochs. Update-free held-out focal probabilities before/after conditioning test direction against initial preference and yoked exposure. We average the 24 model–seed–effect combinations.

Skill acquisition. Three objectives use seeds $3, 7, 11, 13, 17, 19, 23, 29, 31$ per model per task. Our finite-action, TextWorld-inspired relay task (Côté et al., 2018) requires managing resources and removing burdens through successive gates under tool costs and deadlines. Our locally simulated Telecom task uses contextual material from τ^2-bench (Barres et al., 2025) for configuration verification, fault repair, and explicit completion (Appendix A.6). Each model–seed–task parent, calibrated only on four initial training episodes' supports, supplies copies with fresh learning states and identical instance schedules. Within each group, each learner (1) observes states, independently samples and executes actions, records consequences and acting probabilities for four complete episodes under a fixed policy; (2) constructs terminal advantages and, for OCL, fixed typed targets; and (3) fits its objective for four epochs before starting the next group with the updated policy. Budgets (Telecom: 192; TextWorld: 256 groups) do not match decisions, time, or realized KL. Alongside binary terminal rewards, OCL receives measured criterion-specific consequences and private opportunity-loss estimates in the form of attainable TextWorld gate progress without further violations or damage, as well as Telecom clean completion. TextWorld errors affect clean scoring without suppressing later local feedback.

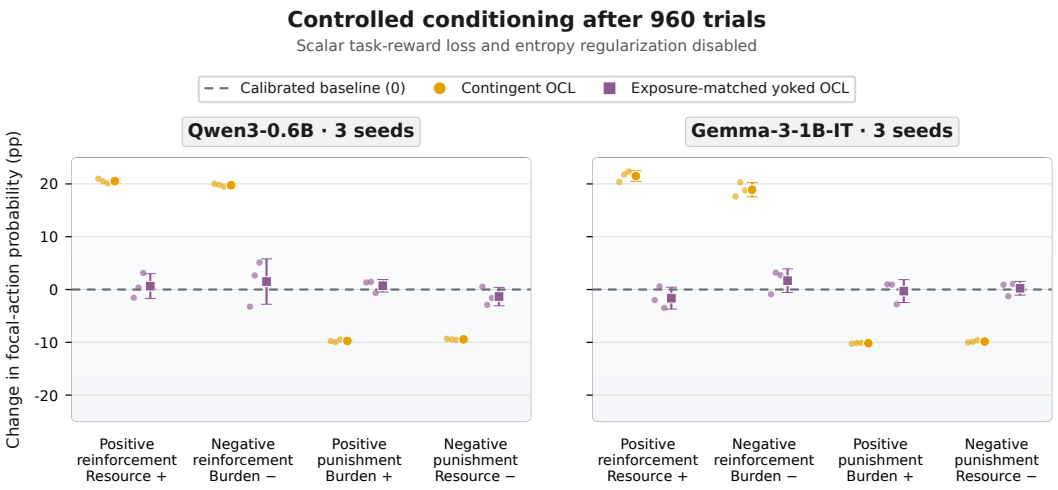

Figure 3: **Controlled conditioning.** Changes from calibrated baseline; small points average eight held-out probes per seed. Large markers means; bars, sample std. dev. over three seeds per model.

Evaluation. Every eight groups, greedy development clean success $\geq 94/98$ (Telecom) or $\geq 92/96$ (TextWorld) triggers an additional independent confirmation. Passing both at this threshold freezes the learned policy. Independent tests at these budgets evaluate final or frozen policies greedily and with four stochastic replicas per instance. Clean success requires TextWorld completion with zero cumulative violations (damage counts), or Telecom valid target attestation, zero residual burden, and successful task completion. Greedy evaluation selects the highest-probability action at each step. Stochastic evaluation samples from the learned action distribution ($T = 1$), making higher-probability actions more frequent. To be clear: greedy evaluation tests the policy's top-ranked choices, while stochastic evaluation tests how reliably its probability distribution produces clean completion. We use one greedy rollout and average four stochastic rollouts per held-out case.

Behavior and inference. Saved scores quantify chosen-action probability changes from collection to epoch four, signed by predicted consequences, which fall into 3 categories: physical, derived opportunity-loss, or mixed. The four-category overview cannot isolate Telecom burden removal from joint resource addition. Our retrospective analysis uses method-specific visited states, unlike held-out conditioning. Average is within matched seeds, and nine blocks. Paired differences have pointwise 95% Student-t intervals and two-sided exact sign-flip tests. Holm covers eight primary comparisons (two tasks $\times$ two selection modes $\times$ each OCL variant versus Scalar RL).

5 Results

Conditioning. Without scalar reward or entropy regularization, 960 trials of unary typed-consequence conditioning produce all four predicted held-out focal-action probability changes: positive reinforcement, $16.83\% \rightarrow 37.84\%$; negative reinforcement, $16.65\% \rightarrow 35.99\%$; positive punishment, $16.85\% \rightarrow 6.90\%$; negative punishment, $16.57\% \rightarrow 6.93\%$ (Figure 3). Exposure-matched contingent-minus-yoked effects are $+21.50$, $+17.75$, -10.16, and -9.10 percentage points (pp). All 24 model–seed–effect means follow predictions against both calibrated baseline and yoked controls; yoked changes lack consistent direction. This supports unary conditioning beyond exposure; Appendix B.1 gives earlier checkpoints and probe-level variation. The reinforcement–punishment asymmetry is largely absent in log odds, where mean paired changes from initialization are $+1.09$, $+0.98$, -1.02, -0.98, respectively. Starting near $1/6$, equal signed log-odds shifts produce larger probability increases than decreases (Appendix B.2).

Acquisition. Without scalar loss, OCL-C beats Scalar RL in all 18 paired model–seed runs per task: stochastic clean success is 77.15% versus 59.65% on Telecom and 14.55% versus 5.60% on TextWorld (Figure 4). Both have Holm-adjusted $p = 0.03125$ over eight primary tests. All four no-scalar comparisons, including greedy, survive correction (Table 1).

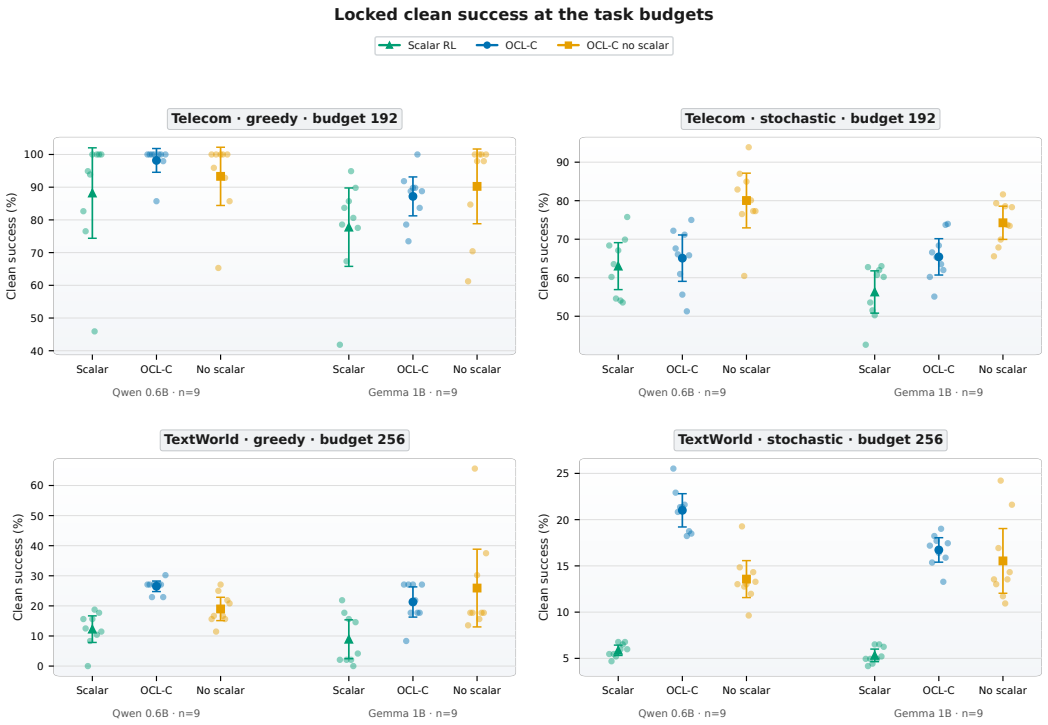

Figure 4: **Endpoint clean success.** Points: seed results; markers: means; bars: pointwise 95% Student-t intervals (9/model). "OCL-C" and "OCL-C no scalar" include and exclude scalar loss.

With scalar loss, stochastic clean success is 65.25% on Telecom and 18.87% on TextWorld, reversing OCL rankings across tasks. Both Scalar RL contrasts survive primary correction. TextWorld improves in all 18 pairs. Telecom's no-scalar mean remaining burden is 0.345 versus Scalar RL's 0.630 (45.2% lower); confirmed greedy mastery is 14/18 versus 11/18 with scalar loss and 5/18 for Scalar RL. These gains accompany more decisions and recorded time (Appendix B.5).

Clean execution. At the fixed TextWorld endpoint, no-scalar Gemma seed 13 achieves **65.6% greedy clean success** (63/96), versus 17.7% with scalar loss and 2.1% for Scalar RL, exceeding the no-scalar Gemma mean (25.9%). Its stochastic clean success is 24.2%. Across models, greedy clean success is 22.45% without scalar loss, 23.90% with it, and 10.59% for Scalar RL. Both OCL variants complete 100% of greedy episodes, mostly with violations. No method confirms mastery. Stochastic mean violations are respectively 2.224, 1.653, and 3.780, but damage is 0.721, 0.911, and 0.665.

Table 1: Eight pooled clean-success contrasts. Pointwise intervals; Holm corrects the eight tests.

Task	Mode	Contrast	Difference (pp), 95% CI	Seed blocks	Sign-flip p	Holm p
Telecom	greedy	OCL-C vs Scalar RL	9.69 [-0.44, 19.83]	9	0.0547	0.0547
Telecom	greedy	OCL-C no scalar vs Scalar RL	8.79 [1.05, 16.52]	9	0.0195	0.0391
Telecom	stochastic	OCL-C vs Scalar RL	5.60 [2.45, 8.75]	9	0.0078	0.0312
Telecom	stochastic	OCL-C no scalar vs Scalar RL	17.50 [12.23, 22.78]	9	0.0039	0.0312
TextWorld	greedy	OCL-C vs Scalar RL	13.31 [8.08, 18.54]	9	0.0039	0.0312
TextWorld	greedy	OCL-C no scalar vs Scalar RL	11.86 [2.41, 21.32]	9	0.0078	0.0312
TextWorld	stochastic	OCL-C vs Scalar RL	13.27 [11.57, 14.97]	9	0.0039	0.0312
TextWorld	stochastic	OCL-C no scalar vs Scalar RL	8.96 [6.53, 11.38]	9	0.0039	0.0312

Skill-acquisition behavior. Without scalar loss, all 16 task–model–consequence means and every eligible seed mean follow the predicted direction; all 16 pointwise 95% intervals lie above zero. Corresponding category-mean counts are 14/16 with scalar loss and 13/16 for Scalar RL (Figure 5). These retrospective summaries use each method's own states. Telecom burden removal co-occurs with resource addition; negative punishment measures derived opportunity loss. Four causal effects are not isolated.

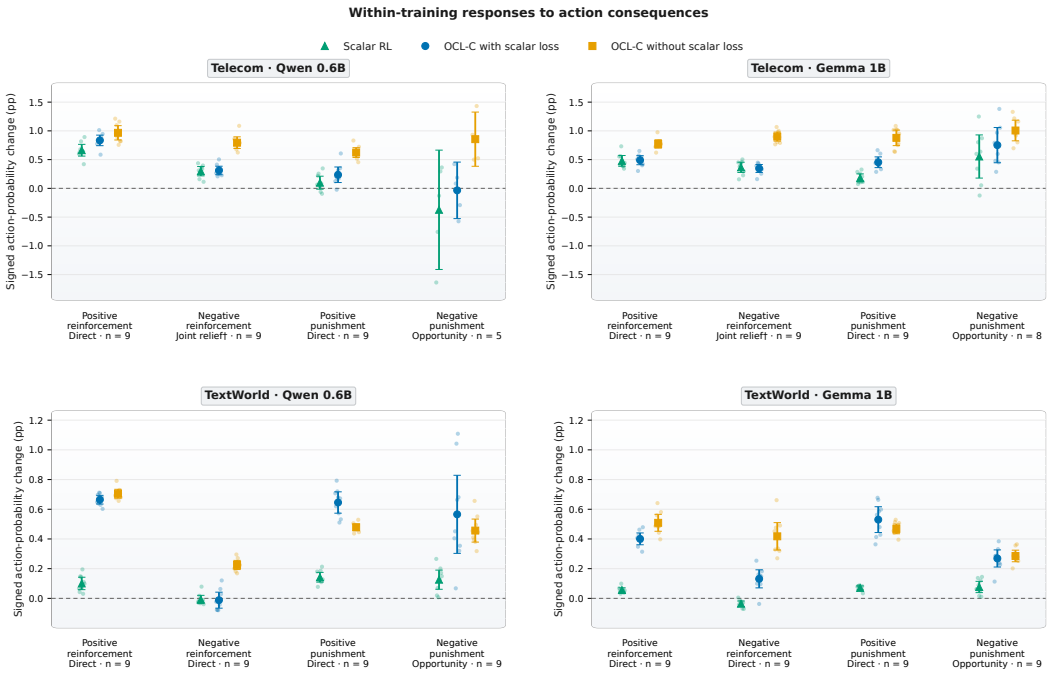

Figure 5: **Observed acquisition responses.** No-scalar OCL shows predicted mean response directions in all 16 task–model–consequence categories. Points summarize decisions/seed. Markers/bars show means/pointwise 95% seed-level intervals. n counts seeds with qualified observations in all 3.

Mechanism use. All 72 OCL skill-acquisition runs exercised contextual redistribution, opportunity-loss targeting, and neutral-decision terminal credit (Table 7). Without scalar loss, contextual redistribution turned on for 18.22% of active local-consequence targets on Telecom and 18.27% on TextWorld. Opportunity loss was the sole selected criterion in 575 and 1,956 decisions, respectively.

6 Discussion

Typed consequences support controlled conditioning and task acquisition without a goal-directed scalar task-reward loss. **Operant Contingency Learning** implements one version of the operant framework. Scalar RL, while not prioritizing its outcome, can also exhibit operant behavior. OCL's contribution is retaining resource and burden identities when constructing policy targets.

OCL-C without scalar loss has the highest observed stochastic clean-success mean on Telecom, while the combined objective leads TextWorld. Direct comparisons between the two are not significant after exploratory Holm correction (Appendix B.3). Neutral-decision terminal credit supplies 58.99% and 15.31% of active no-scalar targets, respectively (Table 7); learners therefore retain substantial outcome supervision. Scalar loss is unnecessary for these gains and not uniformly detrimental. With scalar loss, Qwen's mean action probability increases after Telecom opportunity loss and decreases after TextWorld burden removal; both intervals include zero. These departures show that combined updates can oppose prescribed local targets, without identifying the scalar term's contribution.

TextWorld exposes a gap between completion and reliable clean execution. Under stochastic selection, both OCL variants have fewer violations but greater mean damage than Scalar RL. Explicit consequence categories do not guarantee uniform improvement across criteria.

Penalizing incorrect responses can improve reasoning by redistributing probability toward alternatives favored by the model's prior (Zhu et al., 2025). This begs the question: which useful alternatives should be strengthened? Experiments with rats and pigeons provide a precedent: frequent reinforcement of an alternative response enhanced suppression of a response whose reinforcement had been withdrawn (Leitenberg et al., 1975). Behavioral research motivates examining the wider effects of

training: rats acquired avoidance responses that reduced shock frequency (Herrnstein & Hineline, 1966), while aversive dog-training was associated with more stress indicators than reward-based training (Vieira de Castro et al., 2020). School selection was nonrandom and individual contingencies were not isolated. These findings motivate testing whether reinforcing useful alternatives in agents reduces unwanted avoidance, without equating model penalties with animal distress.

Rewards can also select unintended behavior. OpenAI traced GPT-5.1's use of its browser as a calculator while behaving as though it had searched to a bug rewarding superficial web-tool use (Williams et al., 2025). This motivates distinguishing an action from its useful consequences. OCL makes consequence identity and relevance explicit, but depends on valid measurements. Neutral-decision terminal credit can still favor redundant actions. Reward-hacking robustness is untested.

We hypothesize that explicitly reinforcing useful intermediate actions while tracking their benefits and costs can make agents more predictable and safer. Behavioral psychology would teach us to specify context, response, and consequence, then test predicted changes in responding. Our behavioral assessment supports directional control under specified contingencies. Further tests should measure independent harms, unnecessary refusal, and transfer across contexts, using scalar and typed learners with matched consequence information. Positive reinforcement can also produce adverse effects, so safety must be assessed independently of contingency labels (Perone, 2003).

Testing this hypothesis requires benchmarks that expose intermediate consequences alongside success. TextWorld and τ^2-bench support interaction (Côté et al., 2018; Barres et al., 2025), while we added consequence and relevance support. Future benchmarks should, where appropriate, record action-linked state changes, consequence identity and direction, contextual need, delayed effects, simultaneous benefits and costs, and distinguish zero, unavailable, and unobserved feedback. Contingency changes and exposure-matched controls can test whether behavior follows intended consequences.

7 Limitations

Evidence covers two small models on adapted six-action, short-horizon tasks. Larger agents, full τ^2-bench, unrestricted generation, and deployment remain untested; low-rank adaptation and the action interface may constrain behavior.

The yoked behavioral assessment isolates contingency for the unary operator. Skill-acquisition components—contextual consequence memory, relevance, target construction, opportunity feedback, and terminal credit—lack ablations. Richer OCL feedback prevents attributing gains to consequence identity alone. No-scalar acquisition retains outcome supervision and entropy regularization. Comparisons with additional scalar algorithms and equally informative supervision are necessary.

Four epochs per four-episode group fix reuse but were neither ablated nor established optimal. Shared fitting controls update opportunities and does not ensure equal policy movement. For Scalar RL, OCL-C with scalar loss, and OCL-C without scalar loss, mean local KL per accepted update is respectively $(1.217, 1.182, 0.894) \times 10^{-4}$ nats on Telecom and $(2.411, 2.157, 2.424) \times 10^{-4}$ on TextWorld. No-scalar gains accompany lower Telecom and similar TextWorld KL (Appendix C). Evaluation every eight groups limits cost but potentially misses earlier or transient competence. Unconfirmed runs are censored at their training cap. Finer evaluation, greater TextWorld budget, and epoch-schedule sensitivity would strengthen efficiency conclusions.

8 Conclusion

OCL provides one realization of operant conditioning, constructing policy targets that preserve the identity, direction, context, and current relevance of resource and burden changes. The behavioral assessment recovers all four canonical effects, with every model–seed–effect mean shifting as predicted against the controls. In Telecom and TextWorld, OCL-C without scalar loss improves stochastic clean success over Scalar RL in every paired run. Combining typed targets with scalar loss also supports skill acquisition, but does not yield the predicted mean response direction in every consequence category. Task-dependent rankings leave open when these objectives are complementary. These findings motivate applying behavioral psychology's experimental methods to agent alignment: supervise the consequences of intermediate actions alongside terminal outcomes and test the resulting behavior. Whether this improves safety across contexts remains an empirical question.

Ethics Statement

Discussion of operant conditioning evokes imagery of live animal test subjects. Our experiments use computational models and simulated tasks, with no live human or animal test subjects. Nevertheless, there exists a risk that a reader may incorrectly surmise from our work, or feel we suggest, that language models possess the ability to feel the emotional impacts of punishing and reinforcing consequences akin to those of living organisms. In the broader scientific community's effort to make language-model agents and artificial intelligence safer and more predictable, it is important that we find ways to communicate new works in ways that make plain the findings without treating observed changes in action probability as evidence that models experience desire, fear, or other emotions. When misspecified or trained adversarially, attaching the wrong consequences to the right actions (or vice versa), one would expect to see harmful behavior increase. As such, and as holds true with all training philosophies, correct implementation of operant conditioning is necessary to see the desired change in agent behavior. Training with operant conditioning does not guarantee a safer model.

AI Use Statement

Language models were used to create this work throughout multiple levels of involvement. Core hypotheses, mechanism design, experiments, and measurements were proposed by us. Language models were used to implement these ideas in code. They also provided feedback on methodology and mathematical formulations, assisted with statistical analysis, and supported interpretation of results. We were responsible for executing the code and collecting the results, drawing conclusions, and running statistical tests to determine significance. Language models were employed to draft broad language for the paper, while we read, checked language-to-code alignment, revised wording, statements, conclusions, and generally edited the paper to reflect our own ideas and views. We take responsibility for the final paper.

Reproducibility Statement

Sections 3 and 4 describe the method and experimental procedure. Appendix A specifies implementation, task construction, training, evaluation, and statistical analysis. Appendix C defines the reported policy-movement measurements. The anonymous supplement provides the task rosters, initial prompts, source contexts, bank assignments, and training schedules described in Appendix A.6, together with the controlled-conditioning context rosters and trial schedule. Source code will be released upon acceptance.

References

Abbas Abdolmaleki, Jost Tobias Springenberg, Yuval Tassa, Remi Munos, Nicolas Heess, and Martin Riedmiller. Maximum a posteriori policy optimisation. In *International Conference on Learning Representations*, 2018. URL https://arxiv.org/abs/1806.06920.

Abbas Abdolmaleki, Sandy Huang, Leonard Hasenclever, Michael Neunert, Francis Song, Martina Zambelli, Murilo Martins, Nicolas Heess, Raia Hadsell, and Martin Riedmiller. A distributional view on multi-objective policy optimization. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 11–22. PMLR, 2020. URL https://proceedings.mlr.press/v119/abdolmaleki20a.html.

Marwa Abdulhai, Isadora White, Charlie Victor Snell, Charles Sun, Joey Hong, Yuexiang Zhai, Kelvin Xu, and Sergey Levine. LMRL Gym: Benchmarks for multi-turn reinforcement learning with language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 126–153, 2025. URL https://proceedings.mlr.press/v267/abdulhai25a.html.

Alan Baron and Mark Galizio. Positive and negative reinforcement: Should the distinction be preserved? *The Behavior Analyst*, 28(2):85–98, 2005. doi: 10.1007/BF03392107.

Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. τ^2-bench: Evaluating conversational agents in a dual-control environment, 2025. URL https://arxiv.org/abs/2506.07982.

André Barreto, Will Dabney, Rémi Munos, Jonathan J. Hunt, Tom Schaul, Hado P. van Hasselt, and David Silver. Successor features for transfer in reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 30, pp. 4055–4065, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/hash/350db081a661525235354dd3e19b8c05-Abstract.html.

Zacharie Bugaud. CheeseBench: Evaluating large language models on rodent behavioral neuroscience paradigms. *arXiv preprint arXiv:2604.10825*, 2026. URL https://arxiv.org/abs/2604.10825.

Thomas Carta, Clément Romac, Thomas Wolf, Sylvain Lamprier, Olivier Sigaud, and Pierre-Yves Oudeyer. Grounding large language models in interactive environments with online reinforcement learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 3676–3713, 2023. URL https://proceedings.mlr.press/v202/carta23a.html.

Ruth M. Colwill and Robert A. Rescorla. Associative structures in instrumental learning. In Gordon H. Bower (ed.), *The Psychology of Learning and Motivation*, volume 20, pp. 55–104. Academic Press, Orlando, FL, 1986. doi: 10.1016/S0079-7421(08)60016-X.

John O. Cooper, Timothy E. Heron, and William L. Heward. *Applied Behavior Analysis, Global Edition*. Pearson, third edition, 2020. ISBN 9781292324654. URL https://www.pearson.com/en-gb/subject-catalog/p/applied-behavior-analysis-global-edition/P200000003973/9781292324654.

Marc-Alexandre Côté, Ákos Kádár, Xingdi Yuan, Ben Kybartas, Tavian Barnes, Emery Fine, James Moore, Ruo Yu Tao, Matthew Hausknecht, Layla El Asri, Mahmoud Adada, Wendy Tay, and Adam Trischler. TextWorld: A learning environment for text-based games. *arXiv preprint arXiv:1806.11532*, 2018. URL https://arxiv.org/abs/1806.11532.

Nathaniel D. Daw, Yael Niv, and Peter Dayan. Uncertainty-based competition between prefrontal and dorsolateral striatal systems for behavioral control. *Nature Neuroscience*, 8(12):1704–1711, 2005. doi: 10.1038/nn1560.

Anthony Dickinson and Bernard W. Balleine. Motivational control of goal-directed action. *Animal Learning & Behavior*, 22(1):1–18, 1994. doi: 10.3758/BF03199951.

Yujie Feng, Hao Wang, Jian Li, Xu Chu, Zhaolu Kang, Yiran Liu, Yasha Wang, Philip S. Yu, and Xiao-Ming Wu. FOREVER: Forgetting curve-inspired memory replay for language model continual learning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 24945–24965. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.1144. URL https://aclanthology.org/2026.acl-long.1144/.

Loris Gaven, Clément Romac, Thomas Carta, Sylvain Lamprier, Olivier Sigaud, and Pierre-Yves Oudeyer. SAC-GLAM: Improving online RL for LLM agents with soft actor-critic and hindsight relabeling. *arXiv preprint arXiv:2410.12481*, 2024. URL https://arxiv.org/abs/2410.12481.

Gemma Team, Aishwarya Kamath, Johan Ferret, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025. URL https://arxiv.org/abs/2503.19786.

John Gkountouras, Matthias Lindemann, Phillip Lippe, Efstratios Gavves, and Ivan Titov. Language agents meet causality—bridging LLMs and causal world models. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=y9A2TpaGsE.

Lynn J. Hammond. The effect of contingency upon the appetitive conditioning of free-operant behavior. *Journal of the Experimental Analysis of Behavior*, 34(3):297–304, 1980. doi: 10.1901/jeab.1980.34-297.

Richard J. Herrnstein and Phillip N. Hineline. Negative reinforcement as shock-frequency reduction. *Journal of the Experimental Analysis of Behavior*, 9(4):421–430, 1966. doi: 10.1901/jeab.1966.9-421.

Per Holth. Two definitions of punishment. *The Behavior Analyst Today*, 6(1):43–47, 2005. doi: 10.1037/h0100049.

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*, 2022. URL https://arxiv.org/abs/2106.09685.

Rodrigo Toro Icarte, Toryn Klassen, Richard Valenzano, and Sheila McIlraith. Using reward machines for high-level task specification and decomposition in reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2107–2116, 2018. URL https://proceedings.mlr.press/v80/icarte18a.html.

Robert Johansson. Machine psychology: Integrating operant conditioning with the non-axiomatic reasoning system for advancing artificial general intelligence research. *arXiv preprint arXiv:2405.19498*, 2024. URL https://arxiv.org/abs/2405.19498.

Fred S. Keller and William N. Schoenfeld. *Principles of Psychology: A Systematic Text in the Science of Behavior*. Appleton-Century-Crofts, New York, 1950.

Mehdi Keramati and Boris Gutkin. Homeostatic reinforcement learning for integrating reward collection and physiological stability. *eLife*, 3:e04811, 2014. doi: 10.7554/eLife.04811. URL https://elifesciences.org/articles/04811.

Sean Laraway, Susan Snycerski, Jack Michael, and Alan Poling. Motivating operations and terms to describe them: Some further refinements. *Journal of Applied Behavior Analysis*, 36(3):407–414, 2003. doi: 10.1901/jaba.2003.36-407.

Kennon A. Lattal. Combinations of response-reinforcer dependence and independence. *Journal of the Experimental Analysis of Behavior*, 22(2):357–362, 1974. doi: 10.1901/jeab.1974.22-357.

Harold Leitenberg, Richard A. Rawson, and James A. Mulick. Extinction and reinforcement of alternative behavior. *Journal of Comparative and Physiological Psychology*, 88(2):640–652, 1975. doi: 10.1037/h0076418.

Yuheng Lu, Bingshuo Qian, Caixia Yuan, Huixing Jiang, and Xiaojie Wang. Controlled low-rank adaptation with subspace regularization for continued training on large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 19165–19181. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.940. URL https://aclanthology.org/2025.acl-long.940/.

Yijun Ma, Zehong Wang, Yiyang Li, Ziming Li, Xiaoguang Guo, Weixiang Sun, Chuxu Zhang, and Yanfang Ye. ProPlay: Procedural world models for self-evolving LLM agents. *arXiv preprint arXiv:2606.12780*, 2026. URL https://arxiv.org/abs/2606.12780.

Bodhisattwa Prasad Majumder, Bhavana Dalvi Mishra, Peter Jansen, Oyvind Tafjord, Niket Tandon, Li Zhang, Chris Callison-Burch, and Peter Clark. CLIN: A continually learning language agent for rapid task adaptation and generalization. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=xS6zx1aBI9.

Jack Michael. Positive and negative reinforcement, a distinction that is no longer necessary; or a better way to talk about bad things. *Behaviorism*, 3(1):33–44, 1975. URL https://www.jstor.org/stable/27758829.

Jack Michael. Distinguishing between discriminative and motivational functions of stimuli. *Journal of the Experimental Analysis of Behavior*, 37(1):149–155, 1982. doi: 10.1901/jeab.1982.37-149.

Andrew Y. Ng, Daishi Harada, and Stuart J. Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the Sixteenth International Conference on Machine Learning*, pp. 278–287. Morgan Kaufmann, 1999. URL https://ai.stanford.edu/~ang/papers/shaping-icml99.pdf.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744, 2022. doi: 10.52202/068431-2011. URL https://proceedings.neurips.cc/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract-Conference.html.

Deokgun Park, Md Ashaduzzaman Rubel Mondol, SM Mazharul Islam, and Aishwarya Pothula. Hippocampus-inspired cognitive architecture (HICA) for operant conditioning. *arXiv preprint arXiv:2212.08626*, 2022. URL https://arxiv.org/abs/2212.08626.

Jiangweizhi Peng, Yuanxin Liu, Ruida Zhou, Charles Fleming, Zhaoran Wang, Alfredo Garcia, and Mingyi Hong. HiPER: Hierarchical reinforcement learning with explicit credit assignment for large language model agents. *arXiv preprint arXiv:2602.16165*, 2026. URL https://arxiv.org/abs/2602.16165.

Michael Perone. Negative effects of positive reinforcement. *The Behavior Analyst*, 26(1):1–14, 2003. doi: 10.1007/BF03392064.

Jan Peters, Katharina Mülling, and Yasemin Altun. Relative entropy policy search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 24, pp. 1607–1612, 2010. doi: 10.1609/aaai.v24i1.7727.

Diederik M. Roijers, Peter Vamplew, Shimon Whiteson, and Richard Dazeley. A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research*, 48:67–113, 2013. doi: 10.1613/jair.3987. URL https://www.jair.org/index.php/jair/article/view/10836.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1889–1897, 2015. URL https://proceedings.mlr.press/v37/schulman15.html.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 8634–8652, 2023. doi: 10.52202/075280-0377. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/1b44b878bb782e6954cd888628510e90-Abstract-Conference.html.

B. F. Skinner. *The Behavior of Organisms: An Experimental Analysis*. Appleton-Century, New York, 1938.

B. F. Skinner. *Science and Human Behavior*. Macmillan, New York, 1953.

B. F. Skinner. Selection by consequences. *Science*, 213(4507):501–504, 1981. doi: 10.1126/science.7244649.

Thomas Spooner, Nelson Vadori, and Sumitra Ganesh. Factored policy gradients: Leveraging structure for efficient learning in MOMDPs. In *Advances in Neural Information Processing Systems*, volume 34, pp. 5481–5493, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/2ba8698b79439589fdd2b0f7218d8b07-Abstract.html.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F. Christiano. Learning to summarize from human feedback. In *Advances in Neural Information Processing Systems*, volume 33, pp. 3008–3021, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/1f89885d556929e98d3ef9b86448f951-Abstract.html.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, second edition, 2018.

Richard S. Sutton, David A. McAllester, Satinder P. Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, volume 12, pp. 1057–1063, 2000. URL https://proceedings.neurips.cc/paper/1999/hash/464d828b85b0bed98e80ade0a5c43b0f-Abstract.html.

Richard S. Sutton, Joseph Modayil, Michael Delp, Thomas Degris, Patrick M. Pilarski, Adam White, and Doina Precup. Horde: A scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction. In *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems*, pp. 761–768, 2011.

Edward L. Thorndike. *Animal Intelligence: Experimental Studies*. Macmillan, New York, 1911. URL https://archive.org/details/animalintelligen00thor.

David S. Touretzky and Lisa M. Saksida. Operant conditioning in skinnerbots. *Adaptive Behavior*, 5 (3–4):219–247, 1997. doi: 10.1177/105971239700500302.

Harm van Seijen, Mehdi Fatemi, Joshua Romoff, Romain Laroche, Tavian Barnes, and Jeffrey Tsang. Hybrid reward architecture for reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 30, pp. 5392–5402, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/1264a061d82a2edae1574b07249800d6-Abstract.html.

Ana Catarina Vieira de Castro, Danielle Fuchs, Gabriela Munhoz Morello, Stefania Pastur, Liliana de Sousa, and I. Anna S. Olsson. Does training method matter? evidence for the negative impact of aversive-based methods on companion dog welfare. *PLOS ONE*, 15(12):e0225023, 2020. doi: 10.1371/journal.pone.0225023.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=ehfRiF0R3a.

Haomiaomiao Wang, Tomás E. Ward, and Lili Zhang. Comparative reversal learning reveals rigid adaptation in LLMs under non-stationary uncertainty. In *Information Processing and Management of Uncertainty in Knowledge-Based Systems*, volume 3020 of *Communications in Computer and Information Science*, pp. 452–466. Springer, 2026. doi: 10.1007/978-3-032-28997-1_32. URL https://doi.org/10.1007/978-3-032-28997-1_32.

Xiao Wang, Yuansen Zhang, Tianze Chen, Songyang Gao, Senjie Jin, Xianjun Yang, Zhiheng Xi, Rui Zheng, Yicheng Zou, Tao Gui, Qi Zhang, and Xuanjing Huang. TRACE: A comprehensive benchmark for continual learning in large language models. *arXiv preprint arXiv:2310.06762*, 2023. URL https://arxiv.org/abs/2310.06762.

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. *arXiv preprint arXiv:2504.20073*, 2025. URL https://arxiv.org/abs/2504.20073.

Marcus Williams, Cameron Raymond, and Micah Carroll. Sidestepping evaluation awareness and anticipating misalignment with production evaluations. OpenAI Alignment Research Blog, December 2025. URL https://alignment.openai.com/prod-evals/.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992. doi: 10.1007/BF00992696.

An Yang et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Lu Yang, Zelai Xu, Minyang Xie, Jiaxuan Gao, Zhao Shok, Yu Wang, and Yi Wu. MAGE: Meta-reinforcement learning for language agents toward strategic exploration and exploitation. *arXiv preprint arXiv:2603.03680*, 2026. URL https://arxiv.org/abs/2603.03680.

Xiao Yu, Baolin Peng, Ruize Xu, Yelong Shen, Pengcheng He, Suman Nath, Nikhil Singh, Jiangfeng Gao, and Zhou Yu. Reinforcement world model learning for LLM-based agents. *arXiv preprint arXiv:2602.05842*, 2026. URL https://arxiv.org/abs/2602.05842.

Danyang Zhang, Lu Chen, Situo Zhang, Hongshen Xu, Zihan Zhao, and Kai Yu. Large language models are semi-parametric reinforcement learning agents. In *Advances in Neural Information Processing Systems*, volume 36, pp. 78227–78239, 2023. doi: 10.52202/075280-3419. URL https://papers.neurips.cc/paper_files/paper/2023/hash/f6b22ac37beb5da61efd4882082c9ecd-Abstract-Conference.html.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. ExpeL: LLM agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19632–19642, 2024. doi: 10.1609/aaai.v38i17.29936.

Yifei Zhou, Andrea Zanette, Jiayi Pan, Sergey Levine, and Aviral Kumar. ArCHer: Training language model agents via hierarchical multi-turn RL. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 62178–62209, 2024. URL https://proceedings.mlr.press/v235/zhou24t.html.

Xinyu Zhu, Mengzhou Xia, Zhepei Wei, Wei-Lin Chen, Danqi Chen, and Yu Meng. The surprising effectiveness of negative reinforcement in LLM reasoning. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-4220. URL https://proceedings.nips.cc/paper_files/paper/2025/hash/b8026fe01f5f8f7a80a1a9bf01d08cad-Abstract-Conference.html.

A REPRODUCIBILITY

A.1 POLICY PARAMETERIZATION AND ACTION PROBABILITIES

The model snapshots are:

Model
`Qwen/Qwen3-0.6B`
`google/gemma-3-1b-it`

The pretrained weights are frozen. Both models use LoRA (Hu et al., 2022) with rank eight, scaling parameter sixteen, and zero dropout on the query, key, value, and output projections of attention and the gate, up, and down projections of the feed-forward layers. The experiment seed initializes the adapters and the corresponding sampling streams. Qwen's thinking mode is disabled. Model and adapter computation uses 32-bit floating point, with autocasting and TensorFloat-32 disabled and mathematical scaled dot-product attention. Joint-distribution renormalization for teaching and fitting, target construction, objective reductions, and acceptance calculations use 64-bit arithmetic. Each scoring call contains one public decision context; gradients are accumulated with the global objective denominators unchanged.

Actions are complete members of an explicitly enumerated support, represented by a key k and arguments z. TextWorld has six actions: four tool uses, advance, and wait. Its four tool arguments share one key; advance and wait have separate keys. Telecom has six actions over five keys, including two opposing settings within the relevant intervention channel. Every supported action is scored; an unsuccessful or unaffordable action remains available for selection.

Let $S_k(x)$ be the sum of autoregressive token log probabilities of a key followed by a delimiter consisting of a newline, the word `ARGUMENT`, and another newline. Let $S_{z|k}(x)$ be the corresponding sum for the complete argument string followed by the tokenizer's end-of-sequence token, conditional on that key prefix. Arguments use compact JSON with sorted keys. Scores are sums, without length normalization. Define, for a vector with m entries,

$$F_m(s)_j = \epsilon + (1 - m\epsilon)\frac{\exp(s_j)}{\sum_l \exp(s_l)}, \qquad \epsilon = 10^{-8}. \tag{7}$$

The finite policy is

$$\pi_\theta(k, z \mid x) = F_{|\mathcal{K}|}(S(x))_k \, F_{|\mathcal{Z}_k|}(S_{\cdot|k}(x))_z. \tag{8}$$

Thus normalization occurs separately across keys and across arguments within each key. The complete-action probabilities are not obtained by a single softmax over complete-string scores. Sampling uses this distribution at temperature one without top-k or nucleus truncation. Greedy evaluation chooses the largest joint probability. The teacher introduces no additional probability floor.

A.2 CONSEQUENCE MEMORY AND COUNTERPART SELECTION

Each criterion maintains separate action-wise estimates and counts for its exact ordered action support and public antecedent, and for the same criterion and support pooled across antecedents. A measured action updates both cells once. Its first value is the observed oriented change e_{ic}; subsequent values use the exponential average in Section 3. Other actions are unchanged. Zero measurements are observations, whereas an absent criterion is not imputed. Counts start at zero, and one observation makes an alternative eligible. Memory is empty at the beginning of each learning condition and persists across its training groups.

The TextWorld antecedent comprises the next gate type, charge stock, contamination stock, activation of the charge and contamination carriers, and the affordable-action mask. Telecom uses profile family and status; intent family; subscriber location, service goal, and interaction style; Wi-Fi availability and plan-change permission; latest user-response status; diagnostic availability, signal count, and multiple-signal indicator; membership indicators for ticket constraints and diagnostic subsystems; and the currently exposed active-fault count. Neither antecedent includes instance identity, hidden solution, decision number, or future outcome. These variables index training memory; policy inputs are described with the tasks below.

For an eligible benefit, choose an observed alternative with negative estimated effect, minimizing its estimate and then its position in the ordered action support. For harm, choose an affordable observed alternative with positive estimated effect, maximizing its estimate, then minimizing cost and action position. Test the matching antecedent first; only if it supplies no eligible counterpart, test the pooled cell. With neither, use the unary target. The chosen action is excluded from counterpart selection. No progress action is protected from conditioning.

The current measured effect fixes the sign even when the chosen action's historical estimate has the opposite sign. Benefit requires a currently relevant criterion and an affordable chosen action. Harm has strength one. An underlying effect with magnitude at most 10^{-12} produces no criterion target. All current acquisition criteria have one-count units; magnitude scales both the chosen-action log-odds shift and counterpart log-weight shift. Within a row, all measured criterion memories are updated, including criteria whose targets are subsequently suppressed by harm priority. Criterion targets are combined exactly as in Section 3; ties for the strongest anchor use criterion-name order. Targets that equal the collection distribution contribute no typed loss.

Neutral-row terminal credit is allowed only when every measured criterion has exactly zero change. It cannot replace a nonzero but irrelevant beneficial outcome, and it does not enter consequence memory. The same rule is used by both acquisition OCL conditions. Consequently, disabling the scalar loss does not remove all terminal-outcome information.

A.3 Scalar credit and frozen epoch objectives

Let T_e denote the length of episode $e \in \{1, 2, 3, 4\}$ and let $r_{e,t}$ be its unchanged native task reward. Intermediate rewards are zero. The return and matched-turn baseline are

$$G_{e,t} = \sum_{u=t}^{T_e-1} 0.97^{u-t} r_{e,u}, \tag{9}$$

$$D_{e,t} = G_{e,t} - \frac{1}{3} \sum_{e' \neq e} G_{e',t}, \tag{10}$$

$$s = \sqrt{\frac{1}{N} \sum_{e=1}^{4} \sum_{t=0}^{T_e-1} D_{e,t}^2}, \qquad A_{e,t} = \begin{cases} D_{e,t}/s, & s > 10^{-12}, \\ D_{e,t}, & s \leq 10^{-12}. \end{cases} \tag{11}$$

For the baseline only, set $G_{e,t} = 0$ when $t \geq T_e$. The normalization uses the $N = \sum_e T_e$ real decisions; padding creates no additional training rows. The same advantages supply the scalar policy-gradient term and, when applicable, neutral-row target construction.

Collection probabilities and typed targets remain fixed through four optimization epochs. At the beginning of epoch h, compute

$$w_{i,h} = \min\left\{10, \frac{\pi_{\theta_h}(a_i \mid x_i)}{p_i(a_i)}\right\}, \qquad w_{i,1} = 1. \tag{12}$$

Weights are detached and held fixed throughout that epoch's search. They multiply only scalar advantages, not typed divergence or entropy. Scalar and entropy losses are averaged over all real decisions; typed divergence is averaged over active targets. Four epochs need not produce four parameter changes because an unsuccessful search leaves the parameters unchanged.

A.4 Numerical optimization

For each learning epoch, compute $g = \nabla_\theta \mathcal{L}(\theta_h)$ and, when its norm is finite and positive, $d = -g/\|g\|_2$. Every trial is placed at $\theta_h + \eta d$ from the same saved epoch-start parameters. There is no momentum or optimizer-state update. The first acceptable trial is retained; otherwise restore θ_h. The maximum is twelve candidate trials per epoch.

Write $\Delta\mathcal{L} = \mathcal{L}(\theta_h + \eta d) - \mathcal{L}(\theta_h)$. Acceptance requires finite scores and both

$$\Delta\mathcal{L} < -\varepsilon_{\text{arith}}, \qquad \Delta\mathcal{L} \leq -10^{-4}\eta\|g\|_2. \tag{13}$$

Acquisition starts each group with $\eta = 10^{-3}$. Pure typed conditioning starts with $\eta = 0.1\mathcal{L}(\theta_h)/\|g\|_2$. Both cap η at one and reuse the most recent accepted scale for later epochs of the same group. A

scored-policy plateau raises the lower bound and doubles the scale when there is no upper bound. Nonfinite scores, worsening, or insufficient decrease impose an upper bound; without a lower bound the scale is halved, and with both bounds it is bisected. Zero gradients and exhausted searches produce no update.

Loss differences are evaluated directly from changes in log probabilities rather than by subtracting complete loss values. For example, the scalar contribution is $-\lambda_R w_i A_i(\ell'_{ia_i} - \ell_{ia_i})/N$, and the typed contribution is $-\lambda_T q_{ij}(\ell'_{ij} - \ell_{ij})/|\mathcal{I}|$. Entropy differences use exponential-minus-one arithmetic for nearby log probabilities. Here ℓ and ℓ' denote the normalized epoch-start and trial log probabilities. The arithmetic allowance is eight times the sum of operand and summation unit-in-the-last-place estimates. Operand allowances are

$$E_R = \frac{\lambda_R}{N} \sum_i |w_i A_i|[\mathrm{ulp}(\ell_{ia_i}) + \mathrm{ulp}(\ell'_{ia_i})], \tag{14}$$

$$E_H = \frac{\beta}{N} \sum_{i,j} \left[e^{\ell_{ij}} (|\ell_{ij}| + 1)\, \mathrm{ulp}(\ell_{ij}) + e^{\ell'_{ij}} (|\ell'_{ij}| + 1)\, \mathrm{ulp}(\ell'_{ij}) \right], \tag{15}$$

$$E_T = \frac{\lambda_T}{|\mathcal{I}|} \sum_{i \in \mathcal{I}, j} q_{ij}[\mathrm{ulp}(\ell_{ij}) + \mathrm{ulp}(\ell'_{ij})]. \tag{16}$$

Set $E_T = 0$ for an empty active set. If t_s are the individual weighted scalar, entropy, and typed difference summands, then

$$\varepsilon_{\mathrm{arith}} = 8 \left[E_R + E_H + E_T + \sum_s \mathrm{ulp}(t_s) + \mathrm{ulp}(\Delta\mathcal{L}) \right]. \tag{17}$$

This is an arithmetic allowance, not an empirical bound on model-scoring variation; no additional repeated-score noise estimate is assumed. Candidate scoring is checked against gradient-mode scoring. Any otherwise acceptable candidate scored without gradient tracking is confirmed using the gradient-mode scoring path before acceptance. Teacher cancellation, an unrepresentable positive target, or a target rounded to identity does not manufacture a nonzero target. Realized policy KL is recorded after an accepted update and is not used as a target budget.

A.5 Calibration and shared initialization

Calibration minimizes the mean divergence from a uniform distribution over each complete training action support. It uses no scalar reward, typed consequence, development context, or test context. A parent is accepted only after the acting scorer confirms a maximum absolute deviation of at most 0.01 for every calibration action. This guarantees uniformity only on the fitted supports, not every reachable state.

Calibration uses limited-memory BFGS with five secant pairs. With committed displacement s and gradient change y, retain a curvature pair only if $s^\top y > 10^{-10}\|s\|\|y\|$; initialize the inverse-Hessian scale with $(s^\top y)/(y^\top y)$. Normalize the resulting descent direction and use the search above with slope $-g^\top d$. The initial quasi-Newton scale is the smaller of one and the unnormalized direction norm. Invalid or non-descent directions fall back to normalized steepest descent. A rejected quasi-Newton search clears its curvature history. Rejected searches count toward the calibration attempt limit.

The controlled assay uses its fixed training-context roster and permits 128 calibration attempts. Acquisition instead deduplicates the supports encountered in four initial training episodes sampled at the original model parameters. Nine retained acquisition parents had a 128-attempt ceiling and twenty-seven had a 256-attempt ceiling; all thirty-six met the tolerance before their respective ceilings. The three acquisition conditions then receive copies of their model–seed–task parent with fresh consequence memory and learning state. They do not inherit calibration curvature. Calibration data collection and fitting precede the reported acquisition groups. The assay's contingent and yoked conditions share their own calibrated parent, separately from acquisition.

Table 2: Context-dependent tool effects in the acquisition task. Actual stock changes respect clipping, affordability, and carrier availability.

Tool	Next gate: POWER	Next gate: SAFETY
Amber	Charge $+1$	Charge -1
Cobalt	Contamination -1	Contamination $+1$
Jade	Charge -1	Charge $+1$
Ivory	Contamination $+1$	Contamination -1

A.6 Acquisition tasks and data

Shared interface. Both tasks have six admissible actions and at most 15 decisions per episode. The policy receives the task instructions, current public observation, available actions, at most three preceding public-state snapshots, and the latest sanitized public receipt. The three training conditions use the same policy-visible information and instance order. Consequence memories and private outcome annotations are not appended to the policy input. Each method nevertheless collects its own on-policy trajectories, so its realized feedback and episode lengths can differ. The additional consequence supervision available to OCL is specified below.

TextWorld dynamics. Our task is a controlled relay-station environment with two ordered gates, one POWER and one SAFETY, four colored tools, an `advance` action, and `wait`. Charge and contamination are integer stocks clipped to $[0, 2]$. Each episode begins with 15 credits, charge in $\{0, 1\}$, and contamination in $\{0, 1, 2\}$. Each tool has a displayed price of one or two credits. An affordable tool use pays its price even if its stock effect is clipped or its carrier is absent. An unaffordable use changes neither stock nor credits. Advance and wait cost no credits; every action consumes one decision.

POWER advance clears its gate only when the charge carrier is active and charge is positive, consuming one charge; otherwise it is blocked. SAFETY advance always clears its gate and incurs damage equal to current contamination when that carrier is active, and zero otherwise. The episode ends after both gates are cleared or the decision limit is reached. Task instructions explain these mechanics and that tool effects reverse with the next gate, but the agent must learn the color–effect mapping from interaction.

Three schedules govern carrier availability: no intervention; charge-carrier removal after decisions 2 and 3, followed by reinstatement after decision 4; or the corresponding contamination-carrier intervention. Scheduled changes occur after the action. An absent carrier disables tool effects on its stock; absent contamination also prevents SAFETY damage. Stocks are retained across carrier removal. Current availability is public, while the future schedule is hidden. Exogenous changes are excluded from action-attributed consequences.

Registered violations are tool-caused charge loss, tool-caused contamination increase, an unaffordable tool attempt, a blocked POWER advance, or a damaging SAFETY advance. Each such decision contributes one irreversible violation, even when damage exceeds one unit. Ordinary prices, successful POWER charge consumption, and exogenous changes are exempt. *Completion* means clearing both gates. *Clean success* additionally requires zero violations. The scalar task reward is one at a clean terminal completion and zero otherwise; diagnostic shaped scores do not enter the scalar objective. Damage is the cumulative contamination incurred while crossing SAFETY, rather than the final contamination stock.

TextWorld instances. The candidate space crosses six initial stock pairs, two gate orders, sixteen tool-price vectors, and three schedules, producing 576 cases. A public initial-state template fixes stocks, gate order, and prices, excluding the hidden schedule. The training set contains 64 templates under all three schedules (192 cases); development, confirmation, and final evaluation each contain 32 different templates under all three schedules (96 cases). Templates are disjoint across these banks. Training has 96 cases per gate order, 64 per schedule, and 12 per price vector; each evaluation bank has 48, 32, and 6, respectively. In the stock order $(0,0), (0,1), (0,2), (1,0), (1,1), (1,2)$, the counts are $(33, 30, 30, 33, 33, 33)$ for training, $(18, 15, 15, 15, 18, 15)$ for development, $(18, 18, 15, 15, 15, 15)$ for confirmation, and $(15, 18, 18, 15, 15, 15)$ for final evaluation. Case allocation is fixed before

training; the accompanying task rosters specify every case, bank membership, initial prompt, and training-group order.

The training schedule contains 256 distinct unordered groups of four cases, reusing individual cases. Every 16 groups visit all 64 training templates once. Schedules rotate across successive visits; each group contains four different initial-stock profiles and all three schedules. Group membership and order are shared across methods, models, and seeds and do not depend on observed outcomes.

Simplified Telecom. Telecom is a deterministic profile-conformance task derived from the contextual material in τ^2-bench, not an evaluation of its full native environment. Seven families concern roaming, mobile data, Data Saver, VPN, airplane mode, Wi-Fi calling, and network mode. Each has two opposing target settings: on/off, except network mode, whose alternatives are modern (4G/5G preferred) and legacy (2G only). Roaming is changed directly by the agent; the other settings are changed through a deterministic user executor responding to the agent's request. The six available actions are inspecting the profile, requesting an observation, finishing, applying either setting through the relevant channel, and a neutral acknowledgment or refresh through the other channel.

Half of each bank uses configuration-verification cases; half starts with one active service fault. A correct setting verifies conformance and, in fault cases, removes one fault when present. An incorrect setting invalidates conformance and registers a violation; in fault cases it also adds one fault. Repeated correct settings do not repeatedly increase the verification criterion. Inspection and refresh may return information without changing these criteria. Finishing terminates the episode. The terminal scalar reward is one only after explicit correct attestation, completed validation, and no registered violation; it is zero otherwise. Premature or incorrect finishing adds a terminal violation. The episode also ends upon reaching 15 decisions.

There are 192 training cases and 98 cases in each development, confirmation, and final bank. Training family counts, in the order above, are $(96, 18, 12, 12, 18, 18, 18)$; each evaluation bank uses $(49, 9, 7, 7, 9, 9, 8)$. Each bank is balanced between direct and requested-user changes and between verification and fault cases. The source contexts originate from 2,285 task descriptions matched to 114 official base contexts; ambiguous matches and contexts outside the designated profile families are excluded. The 74 official training anchors supply training and development roles, while the 40 official test anchors are exclusive to final evaluation. Public contextual material supplies task grounding rather than the hidden correct target. Semantic families recur across partitions, but source-context identities are separated.

Training makes four passes over the 192 fixed cases, grouped into quartets. After the first pass, the case indices are reordered as $(73i + 19) \bmod 192$, $(151i + 47) \bmod 192$, and $(5i + 101) \bmod 192$. The first-pass quartets cross verification/fault cases with direct/requested changes. The schedule is fixed independently of policy outcomes, with exact source contexts and group membership provided in the accompanying task rosters.

Telecom completion and clean success coincide under its evaluator. Remaining burden is an aggregate stock initialized to one in fault cases and zero otherwise; harmful events, including wrong terminal responses, increase it, while fault removals decrease it. Consequently this measure includes unresolved task penalties and is not simply the public service-fault count.

A.7 Consequence supervision during acquisition

Each measured criterion has a fixed identity, resource/burden type, and unit. A zero change is an observed zero, rather than missing data. All criteria use one physical or indicator unit. Simultaneous criteria remain separate until the target construction resolves conflicting directions.

In TextWorld, resource criteria are charge, cleared gates, clean completion, and attainable gate opportunity. Burden criteria are contamination, unaffordable attempts, damage, and registered violations. Tool-caused charge changes are separated from charge productively consumed by POWER advance. Charge gain is relevant when its carrier is active and the number of remaining POWER gates exceeds current charge. Contamination relief is relevant when that carrier is active, contamination is positive, and a SAFETY gate remains. Gate progress remains relevant after an earlier mistake; past violations do not suppress subsequent useful local feedback. Actual harmful changes retain their negative direction independently of current resource need.

TextWorld opportunity feedback uses additional private environmental supervision. For state s, let $V(s)$ be the greatest final gate index attainable within the remaining decisions and credits, under the actual future carrier schedule, without any *new* violation or damage. Historical violation and damage counters are ignored only for this private reachability calculation; they remain in the actual episode and clean-success evaluation. The resource loss assigned to a transition is $-\max\{0, V(s) - V(s')\}$, measured in lost gates. This detects loss of achievable partial progress without imposing a blanket penalty on elapsed time or waiting. The calculation is applied to every action and does not supply an action recommendation to the policy.

In Telecom, resource criteria are verified configuration, clean completion, and the opportunity for clean completion; burden criteria are active service faults and accumulated task violations. Beneficial changes are relevant only while clean completion remains possible. With no previous violation, a verified fault-free state requires one remaining decision to finish; otherwise clean completion requires at least $\max(1, f) + 1$ decisions, where f is the current fault count. A violation or unsuccessful termination makes clean completion impossible. A transition that changes this possibility from true to false receives one unit of opportunity loss. These private criterion annotations are training supervision, not policy inputs.

For both tasks, terminal task outcomes additionally provide statistical credit to decisions with no nonzero measured criterion change, using the same discounted, turn-matched leave-one-episode-out estimator as Scalar RL. This component remains active when OCL's scalar loss is removed. Accordingly, that condition removes the scalar task-reward loss, not all goal supervision. The comparison holds policy-visible information fixed while changing the training signal: OCL receives the specified typed and opportunity feedback in addition to terminal outcomes.

A.8 Acquisition initialization and evaluation

Each model–seed–task combination uses its own training-only calibrated parent, as detailed in Appendix A.5. The three methods inherit identical parameters and have separate subsequent learner states.

Acquisition uses seeds $3, 7, 11, 13, 17, 19, 23, 29, 31$ for each model. Four complete episodes form a group and receive four optimization epochs. Training ends at 192 groups for Telecom or 256 for TextWorld unless mastery is confirmed earlier. Every eight groups, greedy evaluation screens all development cases. Reaching at least 94/98 clean successes in Telecom or 92/96 in TextWorld triggers a second greedy evaluation on the independent confirmation bank at unchanged parameters. Meeting the same threshold there freezes that method's policy. Unconfirmed methods continue to their cap.

At the declared final budget, the final bank is evaluated with four stochastic replicas and one greedy rollout per case: 392 stochastic and 98 greedy Telecom episodes, or 384 stochastic and 96 greedy TextWorld episodes, per model–seed–method. Stochastic selection samples the normalized action policy at temperature one without top-k or top-p truncation; greedy selection chooses its maximum-probability action. A policy frozen at confirmed mastery is carried forward unchanged. Otherwise the endpoint is the policy at the maximum budget. Final evaluations never select a checkpoint or control stopping; their random draws are common across methods and do not advance training randomness. Reported endpoint quantities are episode means, with the model–seed run, rather than the episode, serving as the replication unit for comparisons.

A.9 Controlled conditioning assay

Purpose and experimental units. This assay isolates the unary response-conditioning operator, separating the predicted effects of action-contingent consequences from exposure to the same consequences. We use Qwen3-0.6B and Gemma-3-1B-IT, each with seeds $3, 7, 11$. Every model–seed combination has separate contingent and yoked learners for resource addition, resource removal, burden removal, and burden addition. Thus, there are 48 trained branches. The assay does not test the additional criterion-combination and terminal-credit rules used during acquisition, and contains no Scalar RL comparison.

States and actions. Training and measurement contexts are selected from separate TextWorld partitions before observing policy scores. For each consequence class, each partition supplies eight initial states: four whose next requirement is POWER and four whose next requirement is SAFETY. States are distinct within each class and requirement; different classes may reuse a state. Training excludes every initial public state appearing in the development, confirmation, or final-evaluation partitions. Eligible states require, respectively, insufficient charge for remaining power requirements, positive charge, positive contamination, or contamination below its maximum of two. Every possible action in a training state is checked to ensure that eligible nonzero focal feedback corresponds to an actual physical change. The labeled measurement action must also produce its assigned physical consequence and satisfy its relevance and affordability conditions.

Within each class–requirement stratum, states are ordered by the four-tool cost vector, then a deterministic identifier of the public state, then instance identity; the first four distinct states are retained. The public-state identifier is the SHA-256 digest of its canonical JSON representation. This selection never uses model probabilities. The six available commands are the amber, cobalt, jade, and ivory tools, advance, and wait. At POWER requirements these tools respectively add charge, remove contamination, remove charge, and add contamination. At SAFETY requirements their respective effects are charge removal, contamination addition, charge addition, and contamination removal. The focal action is the tool producing the tested consequence in that state; it is not always the same tool across contexts.

Training-only calibration. Before branching, fit the trainable policy parameters to a uniform distribution over all six actions on the distinct training supports:

$$L_{\mathrm{cal}}(\theta) = \frac{1}{N} \sum_{i=1}^{N} D_{\mathrm{KL}}(U_i \,\|\, \pi_\theta(\cdot \mid x_i)) \,. \tag{18}$$

Stop when every training-action probability differs from uniform by at most 0.01, confirmed using the same scoring procedure as action selection, or after 128 attempted updates. Failed calibration does not initiate conditioning. Calibration uses limited-memory BFGS with five curvature pairs, at most twelve candidate parameter evaluations per attempt, and Armijo coefficient 10^{-4}. Curvature pairs require $s^\top y > 10^{-10}\|s\|\|y\|$; the initial inverse-Hessian scale is $(s^\top y)/(y^\top y)$. Nonfinite or non-descent proposals revert to steepest descent. A quasi-Newton search starts at the unnormalized direction norm, capped at one. Neither consequences nor held-out probabilities enter calibration or its stopping rule. All branches within a model–seed run inherit the same calibrated parameters and start with empty consequence memory; calibration optimizer history is discarded. The one-percentage-point tolerance constrains training probabilities, not consequence magnitudes or held-out baselines.

Trials and exposure matching. Each branch receives 16 groups of 60 single-decision trials, giving 960 trials per effect and condition. Within an effect, training contexts cycle in their fixed order: zero-indexed slot j in group g uses context $(60g + j) \bmod 8$. Each trial resets the physical environment, history, and preceding receipt; parameters and consequence memory persist. The learner samples one action from its complete policy distribution, with temperature one and no probability truncation. Conditions have independent random streams, while scheduled contexts are identical. No action is forced.

The contingent condition receives only the focal signed component of its own action's consequence, retaining eligibility and setting scalar reward to zero. The yoked condition receives the exact donor sequence of focal changes, zeros, and eligibility indicators at corresponding trial positions, irrespective of its own sampled action. Its independently realized consequences are excluded from supervision. Consequently, external feedback exposure is matched, while active targets, accepted updates, and realized policy change can differ.

Isolated target and fitting. Let p_i denote the distribution during collection, a_i the sampled action, and $s_i \in \{-1, +1\}$ the predicted punishment or reinforcement direction. An eligible nonzero focal event gives shift $u_i = 0.4 s_i r_i$, using binary pre-action relevance r_i. Reinforcement additionally requires active need and affordability; punishment uses its own relevance indicator. Zero or ineligible

events provide no active target. For an active event,

$$q_i(a_i) = \sigma(\operatorname{logit} p_i(a_i) + u_i)\,, \tag{19}$$

$$q_i(j) = \frac{1 - q_i(a_i)}{1 - p_i(a_i)} p_i(j), \qquad j \neq a_i. \tag{20}$$

Consequence memory uses an exponential update with coefficient 0.5, directly initializes the first observation, and includes eligible zeros. However, fresh memory exposed only to one signed consequence cannot identify an opposite-effect comparison action, so learned redistribution among alternatives is inactive. Consequence magnitude is not used to scale this assay's shift.

Targets are frozen through four fitting epochs per group. With c_i indicating active targets, minimize

$$L_{\text{cond}}(\theta) = \frac{\sum_i c_i D_{\text{KL}}(q_i \| \pi_\theta(\cdot \mid x_i))}{\sum_i c_i}. \tag{21}$$

Skip fitting when the denominator is zero. Scalar and entropy terms are zero. Each epoch searches along normalized negative gradient, starting at $\min(1, 0.1L/\|\nabla L\|)$, with at most twelve candidates and Armijo coefficient 10^{-4}. The preceding accepted scale initializes later epochs within that group. Plateaus double the scale up to one; failed sufficient decrease halves it, or bisects the interval when both bounds are known. Failed searches restore parameters. There is no prescribed policy-KL dose. Computation uses FP32, math attention, disabled autocast and TF32, and one support per scoring call.

Readout and estimands. At groups $0, 4, 8, 16$, score the focal-action probability on all eight held-out contexts without updating parameters or memory. Group 16 is the fixed endpoint. For each model, seed, and effect, average paired contingent-minus-baseline and contingent-minus-yoked probability differences over contexts, then average equally over seeds within models and over models. Reinforcement predicts positive differences and punishment negative differences. Dispersion within a model is sample standard deviation across its three seeds. Contexts and checkpoints are repeated measurements; the 24 model–seed–effect direction checks are descriptive and are not independent replications.

A.10 Outcome definitions and statistical analysis

Episode outcomes. TextWorld completion means reaching the final gate. Clean success requires completion with zero cumulative registered violations. A violation is one executed action that decreases charge through a tool, increases contamination through a tool, attempts an unaffordable tool, attempts an advance without sufficient charge, or advances while incurring damage. Each such action contributes one violation, whereas damage records the actual cumulative damage magnitude. Productive charge consumption during advance, ordinary tool costs, waiting, and exogenous state changes are not registered violations. Damage-causing advances are therefore excluded by the clean-success criterion without adding a separate damage predicate. Completing an episode after a violation remains completion, but cannot recover clean success.

Telecom clean success requires an attested correct target configuration, followed by termination within the episode horizon, with zero remaining burden. A subsequent incorrect configuration invalidates the preceding attestation. Remaining burden begins with the initial service fault, when present, and adds the magnitudes of new fees or faults, policy violations, invalid or rejected actions, and unsuccessful termination; removal of a corresponding burden subtracts its magnitude. The task evaluator defines completion and clean success identically. Thus a Telecom completion rate is not a separate measure of completion despite unresolved burden. The two tasks' burden and damage measurements have different meanings and are not pooled into a common harm score.

For each trained policy, greedy evaluation produces one episode per held-out context; stochastic evaluation produces four. The respective evaluation denominators are 98 and 392 for Telecom, and 96 and 384 for TextWorld. Clean-success and completion rates are episode proportions. Burden, damage, and violations are episode means. All replicas contribute to their run's estimate; they do not increase the number of independent training runs. A policy frozen after confirmed mastery is evaluated unchanged at its task's endpoint. Supporting training quantities count actual episodes and decisions; confirmed mastery is a binary run outcome. Comparisons of recorded training time describe the runs and do not control hardware or contention.

Acquisition estimands. Let Y_{msk} denote a run estimate for model m, seed s, and method k, at a specified task and action-selection mode. The models are the two fixed models studied, not a random sample of language models. Each method receives equal weight within a model's complete common seed set. For pooled estimates, first form

$$Z_{sk} = \frac{Y_{\mathrm{Qwen},s,k} + Y_{\mathrm{Gemma},s,k}}{2}, \qquad \widehat{\mu}_k = \frac{1}{n}\sum_s Z_{sk}. \tag{22}$$

For methods k and ℓ, use paired seed differences $D_s = Z_{sk} - Z_{s\ell}$ and estimate $\widehat{\Delta} = n^{-1}\sum_s D_s$. The acquisition cohort has $n = 9$ blocks, corresponding to 18 model–seed runs. This grouping preserves dependence that may arise from matched seed schedules across the two models. Rates are expressed in percent and their differences in percentage points. Counts of favorable individual model–seed comparisons are descriptive and use a different denominator from the inferential analysis. Relative burden reduction is $100(1 - \widehat{\mu}_k/\widehat{\mu}_\ell)$, a ratio of cohort means rather than a mean of within-seed ratios.

For every metric and comparison, require observations for all three methods in the same seed block; pooled blocks additionally require both models. Missing outcomes or missing exposure are excluded, not imputed as zero. Report the number of eligible blocks. Complete-case estimates apply to that common cohort and can be affected by nonrandom missingness.

Intervals and tests. For an estimate based on values V_s—either run-block outcomes or paired differences—the pointwise 95% interval is

$$\overline{V} \pm t_{0.975,n-1}\frac{S_V}{\sqrt{n}}, \qquad S_V^2 = \frac{1}{n-1}\sum_s (V_s - \overline{V})^2. \tag{23}$$

These intervals are approximate, use seed-level variation, and are not adjusted for multiplicity. Bounds are not silently truncated to the range of a probability. A single block has no interval. An observed mastery rate of zero or one does not imply certainty.

The primary two-sided paired test enumerates all 2^n sign vectors:

$$p = 2^{-n}\sum_{\boldsymbol{\epsilon}\in\{-1,1\}^n} \mathbf{1}\left\{\left|n^{-1}\sum_s \epsilon_s D_s\right| \geq |\widehat{\Delta}| - \delta\right\}, \qquad \delta = \max\{10^{-12}, 10^{-12}\max_s |D_s|\}. \tag{24}$$

There are 512 sign vectors for the acquisition cohort. This test requires symmetric paired differences or equivalent exchangeability under the null; matched seeds alone do not establish randomized treatment assignment.

Holm adjustment applies to eight primary reporting comparisons: the two task endpoints, greedy and stochastic evaluation, and each OCL variant versus Scalar RL. For ordered unadjusted values $p_{(1)} \leq \cdots \leq p_{(8)}$, the adjusted value at rank j is

$$\widetilde{p}_{(j)} = \min\left\{1, \max_{i\leq j}(9-i)p_{(i)}\right\}. \tag{25}$$

The family retains all eight intended comparisons if a result is unavailable. This is a reporting definition, not a claim of preregistration. Supporting burden, mastery, and execution measures are presented descriptively rather than promoted to additional primary endpoints. Exploratory corrected comparisons do not belong to this primary family and are not used to expand its claims.

A.11 Retrospective behavioral analysis of acquisition

Measurement. For training decision i, retain the selected action a_i, its public input x_i, its collection probability $p_i(a_i)$, and its probability after each of the four optimization epochs on that group. Consequence categories are assigned from the measured environmental changes, without using the learner's target or target direction. For an eligible benefit set $s_i = +1$; for harm set $s_i = -1$. The primary response is

$$B_i = 100 s_i \left[\pi_{\theta_{g,4}}(a_i \mid x_i) - p_i(a_i)\right]. \tag{26}$$

The post-update probabilities are the shared training scorer's measurements on the stored inputs, not an independently executed native-policy evaluation. Positive B_i denotes change in the predicted

direction. Changes after epochs one through three use the same collection baseline; four group-relative changes are repeated measurements and are never summed. Direction-adherence proportions classify signed changes above 10^{-12} as positive, below -10^{-12} as opposite, and the remainder as ties, with this tolerance applied in probability units. This is a numerical convention, not a significance threshold.

Consequence classification. Resource addition and burden removal predict increased action probability; resource removal and burden addition predict decreased probability. An eligible beneficial event must have positive current relevance. Harmful events remain eligible. A single-quadrant decision has eligible nonzero events in exactly one quadrant, even if several criteria change within that quadrant; it contributes once. Direct physical changes, derived opportunity or feasibility changes, and mixtures of these sources are kept separate. Decisions with several quadrants are classified as jointly beneficial or jointly harmful when their directions agree, and as opposing when they disagree. Irrelevant benefits, neutral observations, and terminal-credit-only decisions are separate categories. Statistical terminal credit is not counted as a direct physical consequence, and an action producing opposing consequences is not forced to satisfy incompatible probability directions.

The four-category overview uses isolated direct resource addition and burden addition, isolated derived resource loss, and isolated direct burden removal where available. Telecom supplies no isolated relevant burden-removal observations. Its marked burden-removal category therefore uses decisions that jointly remove a relevant service fault and establish relevant configuration verification, without any harmful nonzero consequence. Each qualifying decision contributes once to that joint summary and is not duplicated into isolated resource addition. This subset remains part of joint-benefit coverage; it does not increase the total number of recorded decisions. It cannot identify the independent contribution of burden removal when resource addition occurs simultaneously.

Aggregation and interpretation. Within each model–seed–task–method–category, average B_i over that run's eligible encountered decisions; then give the resulting run means equal seed weight. Methods generally encounter different states, event frequencies, and episode lengths. A larger event count does not give that run greater weight or create additional independent replications. Paired summaries require the same complete three-method seed set for each category. Pooled comparisons average the models within each common seed as above.

The overview contains 2 tasks $\times$ 2 models $\times$ 4 categories. Its descriptive tally separately counts positive category means, pointwise intervals above zero, and categories for which every eligible seed mean is positive. These are different statements. Fourteen categories have all nine seeds. Telecom's derived resource-loss category has five common Qwen seeds, $3, 11, 13, 17, 19$, and eight common Gemma seeds, $3, 11, 13, 17, 19, 23, 29, 31$. Missing exposure is not an absent behavioral effect. A positive tally across all sixteen categories would not establish sixteen independent effects, sixteen multiplicity-corrected results, or an isolated Telecom burden-removal effect.

This retrospective overview is descriptive and separate from the eight primary clean-success comparisons. The joint Telecom category was added after examining the isolated categories and is explicitly post hoc. Interpretation emphasizes magnitudes, pointwise intervals, coverage, and observed directions; the category tally is not an inferential sample size. Since each method is measured on its own visited states, these comparisons do not identify causal effects on a matched state distribution or prove that an individual consequence caused a recorded policy change.

B Additional Results and Interpretation

B.1 Controlled conditioning: replication and diagnostics

Replication across models and checkpoints. All six model–seed conditioning runs completed all four effects. For Qwen3-0.6B, the contingent-minus-yoked changes after 960 trials were $+19.88 \pm 2.79$, $+18.26 \pm 4.54$, -10.46 ± 1.42, and -8.11 ± 1.67 pp for positive reinforcement, negative reinforcement, positive punishment, and negative punishment, respectively. For Gemma-3-1B-IT, they were $+23.12 \pm 2.42$, $+17.23 \pm 1.24$, -9.87 ± 2.23, and -10.09 ± 1.27 pp. These uncertainties are sample standard deviations across three seeds, not confidence intervals or significance tests.

The pooled changes from the calibrated initial policy were $+21.01$, $+19.34$, -9.96, and -9.65 pp in the same order. At the earlier fixed checkpoints of 240 and 480 trials, all 24 model–seed–effect means also had the predicted sign relative to both baseline and yoked controls. The direction-corrected contingent-minus-yoked means increased from $(4.94, 3.65, 3.34, 3.18)$ pp at 240 trials to $(10.03, 8.10, 6.34, 6.19)$ pp at 480 trials and $(21.50, 17.75, 10.16, 9.10)$ pp at 960 trials. These sampled checkpoints show progressive separation; they do not establish monotonic change at every update.

Run averages do not imply uniform probe responses. The 24/24 result averages eight held-out contexts within each model–seed–effect combination. At the endpoint, 188/192 individual probe changes had the predicted sign relative to baseline and 190/192 relative to the yoked policy. All four baseline exceptions occurred under negative reinforcement for Gemma seed 3 in power-requiring contexts; two also changed oppositely relative to the yoked policy. The training calibration criterion was met in every run. Individual held-out focal probabilities nevertheless ranged from 15.54% to 17.69% initially, because the uniformity tolerance applied to training contexts. Baseline-referenced contrasts use measured initial probabilities rather than assuming exact uniformity.

Yoked changes relative to baseline split equally between the predicted and opposite directions across the 24 combinations. Their raw probability increases and decreases were 14/10; the 12/12 statement applies only after reversing the sign for punishment. This absence of a consistent direction is descriptive and is not an equivalence test or evidence that yoked policies did not learn.

Exposure matching does not equalize policy change. Every contingent/yoked pair received 960 trials and matching counts of eligible and nonzero consequence packets. Active-target counts and accepted fitting steps also matched within every pair. Most branches accepted 64 fitting steps; the Qwen seed 11 positive-punishment pair accepted 60. The sum of successive measured policy divergences was larger for the yoked learner in every pair. Thus, the contingent advantage cannot be described as an effect at equal realized KL, nor can weaker yoked responding be attributed to an absence of updates. The assay tests the association between a sampled action and its delivered consequence, with delivered exposure held fixed.

Productive resource consumption and negative-punishment feedback. Passing a power requirement consumes charge but constitutes task progress, so this productive use receives zero negative-punishment feedback. A broader diagnostic compares actual stock changes with each learner's own task-generated focal feedback, including for yoked learners; it does not compare recipient stocks with donor-assigned feedback. This diagnostic also flags physical consumption with zero punitive feedback. The negative-punishment branches recorded 667 physical/typed flags for contingent learners and 466 for yoked learners, without separating mismatch types in the aggregate counts. Exhaustive reconstruction of the specified training states identifies successful power advancement as the only possible discrepancy, while designated resource-removing tools retain physically grounded punitive feedback. The retained aggregate records do not permit an independent historical recount of each flag. These observer diagnostics do not enter learning or held-out probability estimation. The result therefore concerns designated typed consequences, not punishment of every physical resource decrease.

B.2 Probability-scale asymmetry

Uniform calibration over six actions gives a nominal focal-action probability of $1/6$. The 0.01 tolerance bounds absolute probability error on training supports only; held-out probes are not constrained by this stopping rule. For baseline probability p and signed log-odds displacement $\pm\delta$, define $q_\pm = \sigma(\operatorname{logit} p \pm \delta)$. Then

$$\frac{q_+ - p}{p - q_-} = \frac{p + (1-p)e^{\delta}}{1 - p + pe^{\delta}} > 1 \qquad \text{for } 0 < p < \tfrac{1}{2},\ \delta > 0. \tag{27}$$

At $p = 1/6$, the assay's active log-odds margin $\delta = 0.4$ gives target changes of $+6.31$ and -4.85 percentage points.

At 960 trials, we compute $\operatorname{logit}(p_{960}) - \operatorname{logit}(p_0)$ separately for each paired held-out probe, average the eight probes within each model–seed run, then average seeds and models equally. Mean changes

are $+1.087$, $+0.985$, -1.019, and -0.984 for positive reinforcement, negative reinforcement, positive punishment, and negative punishment, respectively. These descriptive checks show similar magnitudes on the log-odds scale.

B.3 Complete acquisition endpoints and paired inference

The following existing tables report absolute outcomes for each model and their equal-weight average at 192 groups for Telecom and 256 groups for TextWorld. "OCL-C" denotes the variant with scalar loss throughout these tables and the acquisition figures. "OCL-C no scalar" denotes its removal. Each model has nine seeds. The pooled inferential unit is a matched seed block, averaging the two fixed models within seed; it is not an individual episode or one of 18 independent model–seed units. The endpoint banks contain 98 Telecom and 96 TextWorld instances, with one greedy rollout and four stochastic replicas per instance. Policies that met the development-and-confirmation criterion earlier were frozen and evaluated at the declared endpoint; test outcomes did not choose checkpoints.

Table 1 gives the eight primary reporting contrasts. Intervals are pointwise 95% Student-t intervals on paired differences over nine blocks. The two-sided sign-flip tests enumerate all $2^9 = 512$ assignments; Holm adjustment covers the eight contrasts jointly. Seven survive at $\alpha = 0.05$. The exception is OCL-C with scalar loss versus Scalar RL under greedy Telecom selection: $+9.69$ pp, interval $[-0.44, 19.83]$, adjusted $p = 0.05469$. This uncertainty does not extend to either stochastic OCL-versus-Scalar comparison, both of which survive correction on both tasks. Pointwise intervals in the absolute endpoint tables are untruncated and can extend outside an outcome's natural bounds.

Direct comparisons between OCL variants, consequence outcomes, mastery, and compute belong to a separate family of 694 exploratory comparisons. The observed task rankings and supporting burden, harm, and mastery differences reported here do not survive that family's Holm correction. They contextualize the primary outcomes rather than establish additional multiplicity-controlled advantages.

Table 3: Telecom locked stochastic, budget 192. Means [pointwise 95% t intervals], common three-method seeds. Both models uses paired seed blocks.

Model	Method	Seed blocks	Clean success (%)	Completion (%)	Burden
Qwen 0.6B	Scalar RL	9	63.01 [56.91, 69.11]	63.01 [56.91, 69.11]	0.58 [0.49, 0.67]
Qwen 0.6B	OCL-C	9	65.08 [59.06, 71.10]	65.08 [59.06, 71.10]	0.54 [0.45, 0.62]
Qwen 0.6B	OCL-C no scalar	9	80.05 [72.94, 87.15]	80.05 [72.94, 87.15]	0.30 [0.19, 0.42]
Gemma 1B	Scalar RL	9	56.29 [50.80, 61.78]	56.29 [50.80, 61.78]	0.68 [0.58, 0.78]
Gemma 1B	OCL-C	9	65.42 [60.70, 70.14]	65.42 [60.70, 70.14]	0.54 [0.47, 0.62]
Gemma 1B	OCL-C no scalar	9	74.26 [69.95, 78.58]	74.26 [69.95, 78.58]	0.39 [0.33, 0.45]
Both models	Scalar RL	9	59.65 [55.90, 63.40]	59.65 [55.90, 63.40]	0.63 [0.57, 0.69]
Both models	OCL-C	9	65.25 [61.17, 69.33]	65.25 [61.17, 69.33]	0.54 [0.48, 0.60]
Both models	OCL-C no scalar	9	77.15 [72.17, 82.14]	77.15 [72.17, 82.14]	0.34 [0.27, 0.42]

B.4 The exceptional TextWorld endpoint and completion quality

The Gemma seed 13 no-scalar policy achieves 63/96 clean greedy episodes (65.625%) after 256 groups, with 96/96 completed episodes, mean damage 0.073, and mean violations 0.375. It is the only greedy TextWorld endpoint above 60% among the 54 model–seed–method runs; the next highest is 37.5%, also a no-scalar Gemma run. Its same-seed greedy comparators achieve 17/96 clean episodes

Table 4: Telecom locked greedy, budget 192. Means [pointwise 95% t intervals], common three-method seeds. Both models uses paired seed blocks.

Model	Method	Seed blocks	Clean success (%)	Completion (%)	Burden
Qwen 0.6B	Scalar RL	9	88.21 [74.38, 102.04]	88.21 [74.38, 102.04]	0.19 [-0.02, 0.39]
Qwen 0.6B	OCL-C	9	98.19 [94.55, 101.82]	98.19 [94.55, 101.82]	0.00 [-0.01, 0.01]
Qwen 0.6B	OCL-C no scalar	9	93.31 [84.41, 102.21]	93.31 [84.41, 102.21]	0.08 [-0.06, 0.23]
Gemma 1B	Scalar RL	9	77.78 [65.79, 89.76]	77.78 [65.79, 89.76]	0.28 [0.02, 0.54]
Gemma 1B	OCL-C	9	87.19 [81.24, 93.14]	87.19 [81.24, 93.14]	0.13 [0.03, 0.24]
Gemma 1B	OCL-C no scalar	9	90.25 [78.83, 101.67]	90.25 [78.83, 101.67]	0.06 [-0.08, 0.19]
Both models	Scalar RL	9	82.99 [74.44, 91.54]	82.99 [74.44, 91.54]	0.23 [0.08, 0.39]
Both models	OCL-C	9	92.69 [89.37, 96.01]	92.69 [89.37, 96.01]	0.07 [0.02, 0.12]
Both models	OCL-C no scalar	9	91.78 [86.03, 97.53]	91.78 [86.03, 97.53]	0.07 [-0.02, 0.16]

Table 5: TextWorld locked stochastic, budget 256. Means [pointwise 95% t intervals], common three-method seeds. Both models uses paired seed blocks.

Model	Method	Seed blocks	Clean success (%)	Completion (%)	Damage	Violations
Qwen 0.6B	Scalar RL	9	5.87 [5.32, 6.43]	49.94 [46.16, 53.73]	0.65 [0.62, 0.68]	3.75 [3.61, 3.89]
Qwen 0.6B	OCL-C	9	21.01 [19.21, 22.81]	99.97 [99.90, 100.04]	0.92 [0.89, 0.94]	1.45 [1.39, 1.52]
Qwen 0.6B	OCL-C no scalar	9	13.57 [11.57, 15.57]	91.58 [86.55, 96.61]	0.75 [0.67, 0.82]	2.37 [2.07, 2.66]
Gemma 1B	Scalar RL	9	5.32 [4.64, 6.00]	50.38 [48.43, 52.32]	0.68 [0.65, 0.70]	3.81 [3.72, 3.90]
Gemma 1B	OCL-C	9	16.72 [15.40, 18.05]	98.61 [97.65, 99.58]	0.90 [0.87, 0.94]	1.85 [1.79, 1.92]
Gemma 1B	OCL-C no scalar	9	15.54 [12.04, 19.04]	96.73 [93.81, 99.65]	0.70 [0.60, 0.79]	2.08 [1.87, 2.29]
Both models	Scalar RL	9	5.60 [5.01, 6.18]	50.16 [47.62, 52.70]	0.67 [0.64, 0.69]	3.78 [3.67, 3.88]
Both models	OCL-C	9	18.87 [17.59, 20.14]	99.29 [98.81, 99.78]	0.91 [0.88, 0.94]	1.65 [1.59, 1.71]
Both models	OCL-C no scalar	9	14.55 [12.37, 16.74]	94.16 [90.42, 97.89]	0.72 [0.65, 0.79]	2.22 [1.99, 2.46]

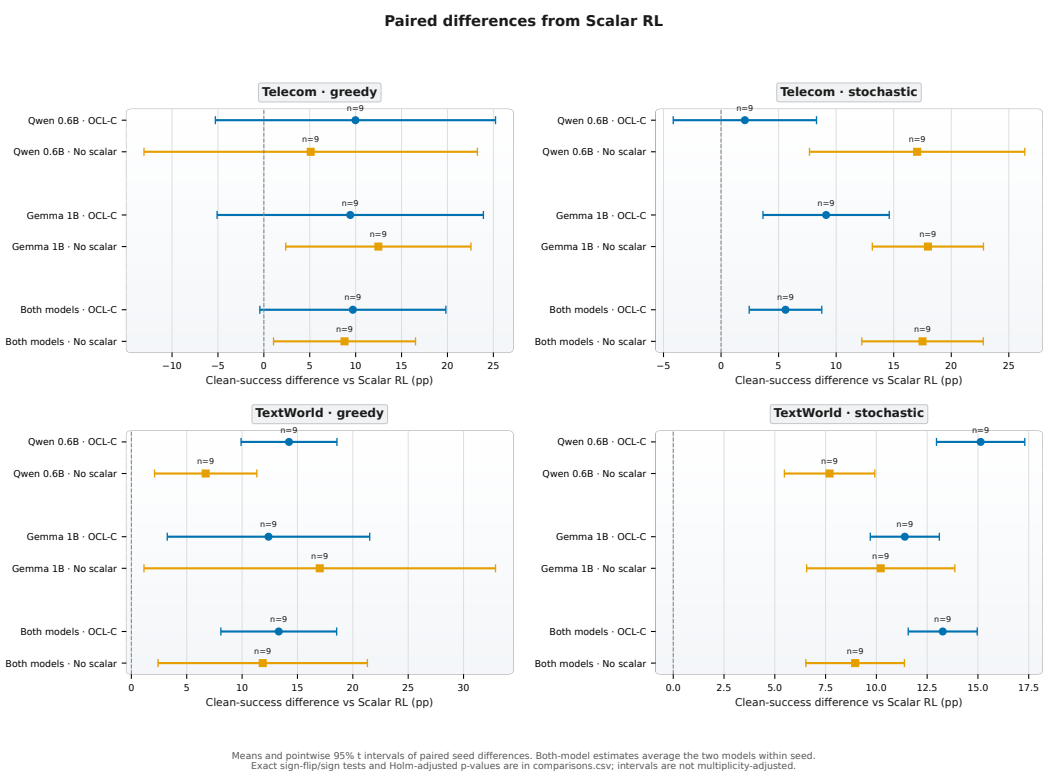

Figure 6: **Paired clean-success differences from Scalar RL.** Model-specific estimates use nine matched seeds; pooled estimates first average the two model differences within seed. Markers show mean paired differences; intervals are pointwise 95% Student-t intervals across matched seed blocks. Pooled corrected tests are given in Table 1; a pointwise interval alone does not establish significance after correction.

with scalar loss and 2/96 under Scalar RL. Across all nine no-scalar Gemma seeds, however, greedy clean success averages 25.93%. The exceptional run is a post hoc descriptive observation at the declared endpoint, not a selected development checkpoint or evidence of reliable mastery.

The same no-scalar policy achieves 93/384 clean stochastic episodes (24.21875%), versus 68/384 with scalar loss and 17/384 for Scalar RL. The exceptional score is therefore selection-mode dependent: the same trained policy has lower clean success when actions are sampled. The replicated stochastic advantage is a separate observation: all 18 no-scalar model–seed runs exceed Scalar RL, so its directional consistency does not depend on this extreme greedy value.

Figure 7 separates clean completion, unclean completion, and failure to complete. On TextWorld, OCL-C with scalar loss completes 99.29% of stochastic episodes and the no-scalar variant completes 94.16%, compared with 50.16% for Scalar RL. Under greedy selection, both OCL variants complete every episode, but approximately three quarters remain unclean. All methods record zero confirmed TextWorld mastery runs. Higher completion therefore does not establish reliable adherence to the clean criterion.

The consequence outcomes further separate forms of failure; Figure 8 shows the greedy endpoints. Under stochastic selection, as reported in the endpoint tables, removing scalar loss yields less mean damage than retaining it (0.721 versus 0.911), but more violations (2.224 versus 1.653) and lower clean success. Scalar RL records the least damage (0.665) while completing substantially fewer episodes. These aggregates cannot determine whether completion differences explain the damage differences. On Telecom, the no-scalar mean burden of 0.345 is 45.2% below Scalar RL's 0.630 and 36.0% below the with-scalar mean of 0.539. This task-specific burden measure includes task-error and terminal penalties, not only unresolved physical faults.

Table 6: TextWorld locked greedy, budget 256. Means [pointwise 95% t intervals], common three-method seeds. Both models uses paired seed blocks.

Model	Method	Seed blocks	Clean success (%)	Completion (%)	Damage	Violations
Qwen 0.6B	Scalar RL	9	12.27 [7.85, 16.69]	41.78 [30.80, 52.76]	0.66 [0.58, 0.73]	5.23 [4.15, 6.32]
Qwen 0.6B	OCL-C	9	26.50 [24.75, 28.25]	100.00 [100.00, 100.00]	0.93 [0.88, 0.98]	1.14 [1.11, 1.17]
Qwen 0.6B	OCL-C no scalar	9	18.98 [15.10, 22.86]	100.00 [100.00, 100.00]	0.94 [0.89, 1.00]	1.15 [1.03, 1.26]
Gemma 1B	Scalar RL	9	8.91 [2.46, 15.37]	49.19 [29.75, 68.63]	0.66 [0.51, 0.80]	3.69 [2.51, 4.87]
Gemma 1B	OCL-C	9	21.30 [16.28, 26.31]	100.00 [100.00, 100.00]	0.90 [0.86, 0.94]	1.37 [1.23, 1.51]
Gemma 1B	OCL-C no scalar	9	25.93 [13.01, 38.84]	100.00 [100.00, 100.00]	0.56 [0.30, 0.82]	1.22 [0.93, 1.50]
Both models	Scalar RL	9	10.59 [7.03, 14.15]	45.49 [34.04, 56.93]	0.66 [0.58, 0.73]	4.46 [3.55, 5.38]
Both models	OCL-C	9	23.90 [21.18, 26.62]	100.00 [100.00, 100.00]	0.91 [0.89, 0.94]	1.26 [1.18, 1.33]
Both models	OCL-C no scalar	9	22.45 [14.59, 30.32]	100.00 [100.00, 100.00]	0.75 [0.61, 0.90]	1.18 [1.01, 1.36]

B.5 Mastery, training exposure, and cost

Confirmed Telecom mastery occurs in 14/18 no-scalar runs (6/9 Qwen, 8/9 Gemma), 11/18 with-scalar runs (5/9 Qwen, 6/9 Gemma), and 5/18 Scalar RL runs (1/9 Qwen, 4/9 Gemma). These counts refer to passing both development and independent confirmation thresholds, not a single high test score. Mean trained groups are 163.1, 171.1, and 187.1, respectively. The higher no-scalar mastery frequency therefore accompanies fewer groups on average, but not lower measured trajectory or time cost.

On Telecom, the no-scalar variant averages 4,194.5 training decisions and 132.8 minutes, compared with 3,328.1 decisions and 101.2 minutes with scalar loss, and 3,404.0 decisions and 102.5 minutes for Scalar RL. On TextWorld, all runs use 256 groups; the corresponding decision counts are 9,147.6, 7,910.1, and 12,079.4, with recorded times of 222.4, 191.4, and 299.3 minutes. Episode lengths vary across methods. Recorded times are not hardware-controlled measurements. These results support task-dependent costs and do not establish an advantage under equal decisions, wall time, or realized policy KL.

B.6 Interpreting response changes during acquisition

Figure 5 contains 16 cells: two tasks, two models, and four consequence categories. Within each run and category, action-probability changes are averaged from the collection policy to the policy after the fourth fitting epoch, then summarized across seeds. The sign is reversed for punishment, so positive values indicate the predicted response direction. OCL-C without scalar loss has positive means in all 16 cells, OCL-C with scalar loss in 14, and Scalar RL in 13. Every eligible seed mean is positive in each no-scalar cell, and all 16 corresponding pointwise intervals lie above zero. These are dependent descriptive summaries, not 16 independent or multiplicity-corrected discoveries.

Fourteen cells include all nine seeds. Telecom opportunity-loss cells use five common Qwen seeds and eight common Gemma seeds with the required category exposure. Missing exposure is not a zero response, and event counts do not increase the replication sample size. The underlying training runs all have complete extraction coverage.

The four categories are not homogeneous physical-event interventions. Positive reinforcement and positive punishment use isolated direct consequences. Negative punishment uses isolated derived opportunity losses. TextWorld negative reinforcement uses isolated direct burden removal, whereas

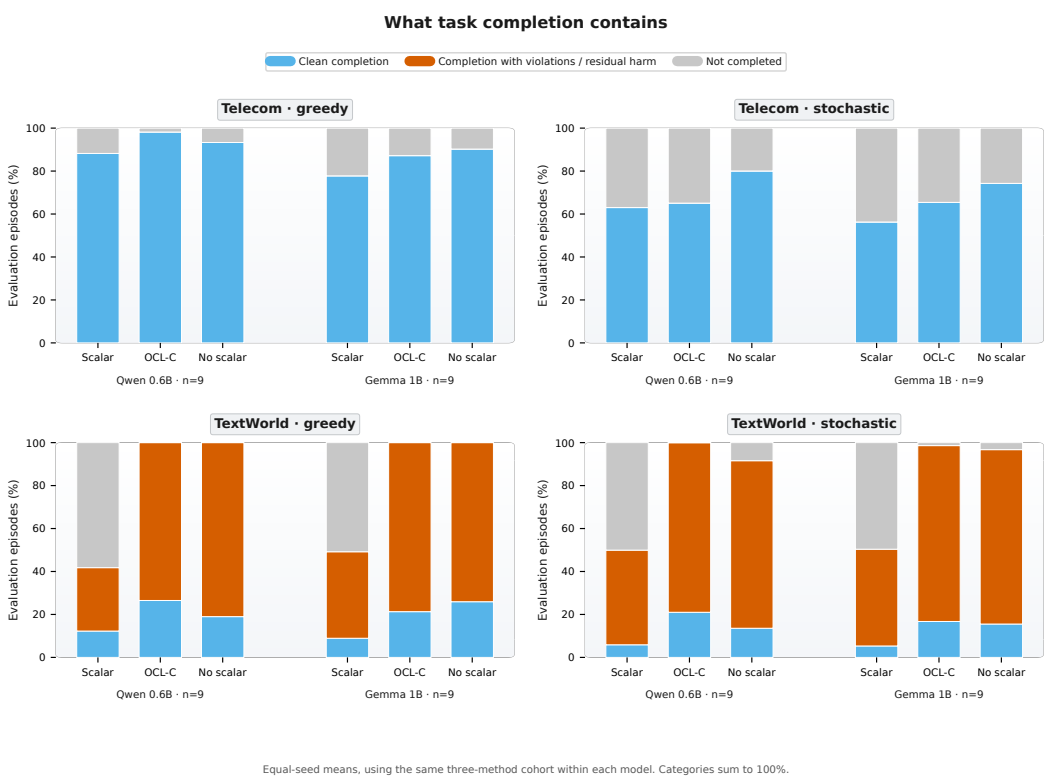

Figure 7: **Completion and clean execution are distinct.** Episode outcomes at the declared endpoints are partitioned into clean completion, unclean completion, and noncompletion, then averaged across seeds. Telecom completion and clean success coincide by definition; TextWorld permits completion after registered violations.

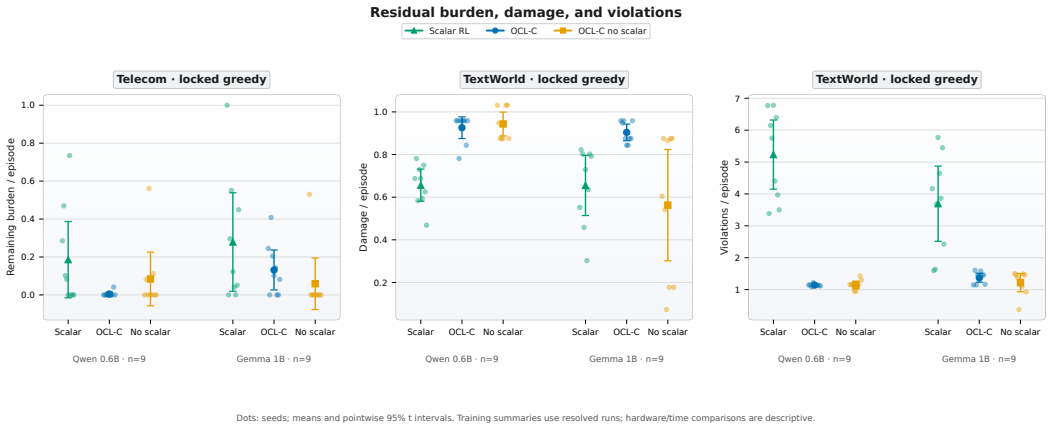

Figure 8: **Task-specific consequence outcomes.** Remaining Telecom burden and TextWorld damage and violations under greedy selection at the declared endpoints. Small points are individual seed estimates; larger markers and bars show model-specific means and pointwise 95% Student-t intervals across nine seeds. These outcomes support a differentiated account of failure rather than uniform harm reduction.

Telecom relief jointly removes a service fault and adds verified configuration resources. The latter cannot identify an isolated burden-removal effect. Moreover, each method contributes its own visited states and action distribution. The probability changes reflect the complete update, including the

Table 7: **Mechanism use during acquisition.** Each column pools 18 OCL-C runs (nine seeds per model). Counts include each collected decision once across the four fitting epochs. Rates are pooled count ratios with denominators specified below. Contextual redistribution uses context-conditioned counterparts; any redistribution also includes pooled fallback. Opportunity counts active targets containing opportunity loss; terminal counts active targets on locally neutral decisions. Categories overlap. Invocation frequency does not measure net redistribution or causal contribution.

	Telecom		TextWorld	
OCL-C scalar loss	On	Off	On	Off
Collected decisions (N)	59,906	75,501	142,381	164,656
Active local-consequence targets (L)	23,652	24,508	78,788	91,784
All active typed targets (T)	51,360	59,760	96,553	108,373
Share of all collected decisions (N), %				
Contextual redistribution invoked	7.07	5.91	7.37	10.18
Any redistribution invoked	20.63	18.24	14.73	19.36
Opportunity-loss target	9.51	6.47	1.58	2.07
Neutral-decision terminal target	46.25	46.69	12.48	10.07
Conditional shares and counts				
Contextual redistribution / L (%)	17.91	18.22	13.31	18.27
Any redistribution / L (%)	52.26	56.18	26.61	34.74
Terminal targets / T (%)	53.95	58.99	18.40	15.31
Opportunity sole selected criterion (count)	131	575	1,381	1,956

method's other active learning terms, and cannot be causally attributed to one event in isolation. These response patterns complement the controlled assay without implying that operant behavior is unique to OCL.

B.7 Use of acquisition target components

We audited all 108 acquisition runs, covering all 23,208 training groups, and reconstructed chronological consequence memory for all 72 OCL runs. Each collected decision contributes once, irrespective of four-epoch reuse. Table 7 pools counts across 18 runs per task and objective; percentages are ratios of pooled counts, so runs with more decisions contribute more weight.

An active target differs from its collection distribution. Active local-consequence targets use nonzero measured criteria, including derived opportunities; they exclude neutral-decision terminal targets. A redistribution invocation requires a retained criterion with a selected counterpart, a nonzero comparison shift, more than two supported actions, and an active final target. Context-specific and pooled-fallback invocations are counted after criterion selection; different criteria can invoke both in one decision. Opportunity-containing targets retain clean-completion opportunity loss on Telecom or gate-opportunity loss on TextWorld. Active terminal targets use episode-return credit only when every measured criterion change is zero. These categories overlap except for local-consequence and terminal targets.

Opportunity loss was selected into an active target whenever observed. Without scalar loss, it was the sole selected criterion in 11.78% of opportunity-containing targets on Telecom and 57.26% on TextWorld. Neutral denotes no measured criterion change, not an unnecessary action. Counterpart invocation records construction operations; criterion mixing can cancel their net redistribution. Full action distributions were not retained, so the final target's distance from its unary counterpart cannot be quantified. Usage frequencies establish component participation, not gradient contribution or causal improvement in acquisition.

C Interpreting recorded policy movement

For run r, let x_{rgi} be the ith visited decision context in training group g, with N_{rg} decision rows. At fitting epoch e, the recorded local divergence is

$$K_{rg,e} = \frac{1}{N_{rg}} \sum_{i=1}^{N_{rg}} D_{\mathrm{KL}}\left(\pi_{\theta_{rg,e-1}}(\cdot \mid x_{rgi}) \,\|\, \pi_{\theta_{rg,e}}(\cdot \mid x_{rgi})\right). \tag{28}$$

Both distributions use the same complete six-action support. The reference is the policy immediately before that epoch, not necessarily the policy that collected the group. Contexts remain fixed during its four epochs. Natural logarithms give units of nats, and every visited decision row receives equal weight within an epoch.

The recorded total is $S_r = \sum_g \sum_e K_{rg,e}$. It includes committed acquisition updates and excludes calibration, evaluation, and rejected candidate steps. All 108 acquisition runs accepted all four epochs in every collected group. Consequently, $U_r = 4G_r$, where U_r counts accepted updates and G_r counts groups. The main-text summaries first compute S_r/U_r for each run and then average the two models and nine seeds equally. Averaging these run-level ratios differs from dividing the cohort's total KL by its total update count when run lengths differ.

Mean local KL per group is four times the per-update quantity: $(4.868, 4.727, 3.577) \times 10^{-4}$ for Telecom and $(9.644, 8.628, 9.695) \times 10^{-4}$ for TextWorld, ordered as Scalar RL, OCL-C with scalar loss, and OCL-C without scalar loss. Total-divided-by-episode is numerically equal to total-divided-by-update only because there are four episodes and four accepted updates per group. It is not an episode-trajectory divergence. Dividing S_r by decisions likewise gives an exposure-normalized index, not a decision-weighted mean KL across all decision–update pairs.

Successive divergences use changing reference policies and visited contexts, and do not telescope into an endpoint distance. The sum also depends on how updates are subdivided. Neither a small difference between aggregate means nor overlapping uncertainty intervals establishes equivalence. A prospective KL-controlled comparison should specify a common context distribution, a reference policy, a measurement schedule, and the intervention used to match policy movement. These choices distinguish measured movement from both a reference-policy regularization coefficient and OCL's target-to-policy fitting divergence.

D Additional comparison with earlier work

Earlier work has studied online RL for language models, agent memory, computational operant conditioning, and behavioral adaptation. Skinnerbots learns temporal predictors and reward-rate estimates for operant chaining and changing contingencies (Touretzky & Saksida, 1997). HICA proposes a cognitive architecture for operant learning, while Machine Psychology demonstrates symbolic operant and reversal learning with NARS (Park et al., 2022; Johansson, 2024). Experiments varying response-dependent and response-independent consequence delivery motivate the question of contingency beyond exposure (Lattal, 1974; Hammond, 1980). Our controlled assay addresses this question with exposure-matched yoked controls. The separate acquisition experiments compare Scalar RL with OCL-C with and without scalar loss; they do not include reversal or yoked acquisition conditions.

The behavioral foundations include the law of effect and operant analysis (Thorndike, 1911; Skinner, 1938), with later accounts separating response–outcome knowledge from current outcome value (Colwill & Rescorla, 1986; Dickinson & Balleine, 1994). Homeostatic RL already makes outcome value depend on current need through a scalar measure of physiological drive reduction (Keramati & Gutkin, 2014). OCL instead uses supplied relevance variables when constructing targets from distinct consequence criteria. Computational accounts of habitual and goal-directed control are adjacent to this question (Daw et al., 2005); our experiments do not test outcome devaluation or distinguish these forms of control.

Constructing an action distribution and fitting a parametric policy to it has precedents in MPO (Abdolmaleki et al., 2018). OCL constructs targets directly from recorded acting probabilities and typed consequences, using memory to redistribute probability among alternatives; the scalar

score-function term is included or omitted in the joint training objective. REPS and TRPO use relative-entropy constraints in policy optimization (Peters et al., 2010; Schulman et al., 2015). Our bounded Armijo search instead selects steps by objective decrease, and records realized policy KL after acceptance. It neither targets a requested KL nor establishes KL-matched comparisons (Appendix C). Recent language-agent work studies strategic exploration, hierarchical credit assignment, and causal environment models (Yang et al., 2026; Peng et al., 2026; Gkountouras et al., 2025). These approaches address questions complementary to preserving direct consequence categories.

Persistent memory and parameter updates also connect OCL to continual-learning questions. TRACE measures continual language-model learning, while CLoRA and FOREVER address interference through regularized adaptation and replay (Wang et al., 2023; Lu et al., 2025; Feng et al., 2026). Our fixed-task acquisition experiments do not test cross-task retention or separate interference in model weights from interference in consequence memory.

Table 8: Related methods. Entries describe information retained across interactions and whether the acting language model's weights change during the method's learning procedure. OCL consequence memory is used during training only.

Work	LM weights updated	Information carried forward	How it is used	Main difference from this paper
GLAM (Carta et al., 2023)	yes	sampled trajectories and scalar returns	online PPO updates	Does not use OCL's typed consequence memory.
ArCHer (Zhou et al., 2024)	yes	utterance- and token-level scalar credit	hierarchical value and policy updates	Separates temporal credit rather than consequence categories.
SAC-GLAM (Gaven et al., 2024)	yes	replayed transitions and hindsight relabels	off-policy actor–critic updates	Reuses scalar-reward experience.
RAGEN (Wang et al., 2025)	yes	multi-turn trajectories and scalar scores	multi-turn RL updates	Studies stability without OCL's consequence estimates.
Reflexion (Shinn et al., 2023)	no	natural-language reflections	later prompting	Keeps LM weights fixed.
ExpeL (Zhao et al., 2024)	no	natural-language lessons and successful trajectories	prompting with lessons and retrieved demonstrations	Uses outcome feedback to revise prompt memory, without LM-weight updates.
CLIN (Majumder et al., 2024)	no	structured textual abstractions	later prompting	Keeps LM weights fixed.
Voyager (Wang et al., 2024)	no	executable skills	later planning and action	Uses a skill library rather than typed consequence targets.
Rememberer (Zhang et al., 2023)	no	contextual scalar action-value estimates	encouraged/ discouraged prompt examples	Uses numeric memory at inference, without LM-weight updates.
ProPlay (Ma et al., 2026)	no	procedure graph and reward-based transition reliability	procedural guidance in the acting prompt	Updates inference-time memory rather than LM weights.
RWML (Yu et al., 2026)	yes	action-conditioned next-observation predictions in model weights	RL on next-state prediction; optional subsequent task RL	Trains prediction of environment changes rather than typed action targets.

Work	LM weights updated	Information carried forward	How it is used	Main difference from this paper
Skinnerbots (Touretzky & Saksida, 1997)	n/a	learned temporal predictors, reward-rate estimates, and working-memory predicates	computational operant chaining	Classical robotic learner, not an LM-weight update.
Machine Psychology (Johansson, 2024)	n/a	symbolic experience statements	NARS inference and learning	Symbolic operant and reversal experiments.
Comparative reversal (Wang et al., 2026)	no	in-context interaction history	behavioral evaluation	Evaluates rigidity without online LM-weight changes.
CheeseBench (Bugaud, 2026)	no	history and textual working memory	zero-shot LM evaluation	Behavioral-neuroscience benchmark without LM fine-tuning.
Scalar RL (this work)	yes	model parameters; no consequence memory	scalar score-function loss and entropy	Scalar-only acquisition comparator.
OCL-C with scalar loss (this work)	yes	model parameters and contextual typed estimates	typed-target KL, scalar loss, and entropy	Current changes set focal targets; memory redistributes alternatives.
OCL-C without scalar loss (this work)	yes	model parameters and contextual typed estimates	typed-target KL and entropy	Omits the scalar loss; retains terminal credit on zero-change decisions.