WHEN DOES ACTING IMPROVE LEARNING? HUMANS, LANGUAGE MODELS, AND THE ROLE OF THINKING

Anonymous authors
Paper under double-blind review

ABSTRACT

Does learning from the same realized interaction depend on whether the learner chose the actions that produced it? We study this question in humans and language models using **INTERGROUND**, a matched-role paradigm in which an **acting learner** chooses consequential actions while an **observer** receives the same realized action–response record. Among 120 adults, acting improved transfer to novel and counterfactual situations by 5.8 percentage points (95% CI [2.6, 8.9]). Across 11 language model configurations spanning SmolLM2, Qwen3, Llama 3, and Gemma 3, however, standard online supervised adaptation produced only small actor-observer differences. In Qwen3, enabling thinking during learning increased the acting–observer gap across scales, with clear evidence at three of four tested scales; at 8B, the gap rose from 0.3 to 4.5 percentage points, a 4.2-point increase (95% CI [2.6, 6.4]). Randomizing whether the executed action matched the learner's preferred action showed that thinking increased sensitivity to action-experience alignment. The same thinking-dependent pattern generalized to a structurally distinct causal-intervention environment at Qwen3-8B, where the gap increased from 0.4 to 2.9 points. Our findings show that matched trajectories do not guarantee matched learning: replaying an interaction alone does not always reproduce the benefit of generating it. For agent training, these results argue for retaining learner-specific context preceding each action, including reasoning and action preferences, alongside the observable trajectory.

1 INTRODUCTION

Learning depends not only on the information available to a learner, but also on how that information is acquired. Learners may generate experience through their own actions or learn from interactions generated by others. In humans, self-directed exploration, hypothesis testing, and contingent interaction can facilitate learning (Voss et al., 2011; Markant & Gureckis, 2014; De Felice et al., 2021; Brod et al., 2025). Yet acting and observing often produce different experience, making it difficult to separate the effect of choosing an action from the trajectory that choice produces. This raises a central question: **Does learning from the same realized interaction depend on whether the learner chose what action to take?**

Studies of active and yoked learning address this question by comparing learners who select evidence with learners who receive matched evidence selected by another person. Learning differences can persist even when the observable evidence is matched (Markant & Gureckis, 2014; Brod et al., 2025). Whether the same principle holds for artificial learners is less clear. Language agents can learn from both self-generated interaction and previously collected trajectories (Carta et al., 2023; Wang et al., 2024), and passive imitation can acquire active causal experimentation strategies from expert intervention trajectories (Lampinen et al., 2023). These lines of work establish that the source of training experience matters, but **leave open whether the same realized interaction is learned differently when generated by the learner rather than received as replay.**

We study this question with **INTERGROUND**, an acting-observer paradigm for learning hidden interaction rules. An acting learner chooses actions that determine how a partner responds, while an observer receives the same realized action-response record without controlling its actions. The two roles therefore receive the same public interaction while differing in whether the learner's own choice determines execution (Figure 1).

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

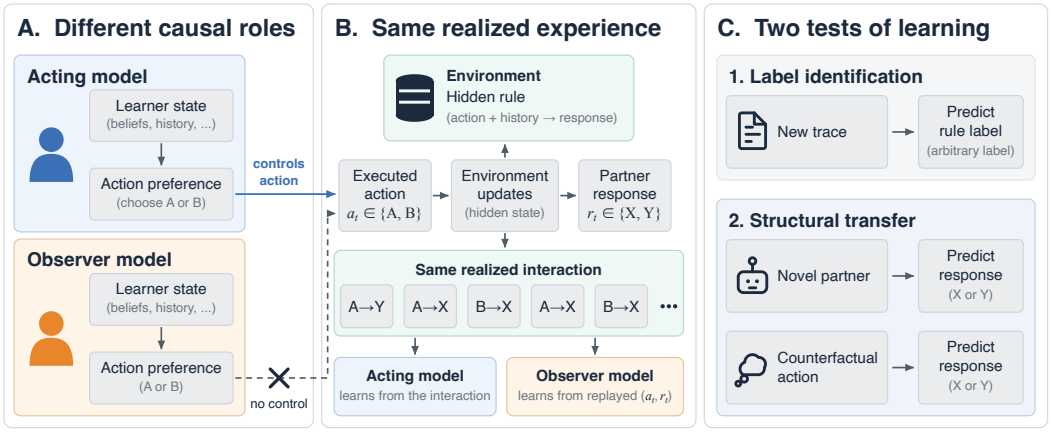

Figure 1: **INTERGROUND model protocol.** **A**: Both model learners form an action preference, but only the acting model's preference controls the executed action; the observer model's never does. The human study uses the same task and evaluation with yoked observers who advance the matched replay without selecting actions (Appendix B). **B**: The two roles receive the same realized public interaction record: the acting model generates it, whereas the observer model receives the replayed (a_t, r_t) sequence. Partner responses follow the hidden rule given the executed action and current state/history. **C**: Evaluation separates *label identification*, which recovers the arbitrary label naming a rule from a new trace, from *structural transfer*, which applies the learned rule to novel partners and counterfactual action histories.

We use the same hidden-rule framework with humans and language models, with interfaces and exposure schedules adapted to each population. Evaluation separates *label identification*, which recovers the arbitrary label assigned to a learned rule, from *structural transfer*, which applies the rule to novel partners and counterfactual action histories.

Humans showed an acting advantage in *structural transfer*, whereas language models showed only small acting–observer differences under standard online supervised adaptation. Among 120 adults, acting improved transfer to novel and counterfactual situations by 5.8 percentage points (95% CI [2.6, 8.9]), while *label identification* was similar across roles. Across 11 model configurations spanning SmolLM2, Qwen3, Llama 3, and Gemma 3, acting–observer differences in *structural transfer* were small. Thus, selecting the actions that generate a trajectory was not by itself sufficient to produce a substantial learning advantage under this adaptation regime.

Learning-stage thinking increased the acting–observer difference in Qwen3. Self-selected evidence may be useful when it tests the learner's current hypothesis (Freund et al., 1992; Gureckis & Markant, 2012; Markant & Gureckis, 2014), and intermediate reasoning can alter model decisions (Wei et al., 2022). These observations motivated us to test whether learner-specific pre-action reasoning makes acting more consequential for learning. With thinking OFF, the learner directly scores candidate actions; with thinking ON, it first generates learner-specific reasoning and then scores them. At Qwen3-8B, thinking ON yielded 90.5% *structural-transfer* accuracy for acting models versus 86.0% for observers, a 4.5-point gap (95% CI [2.3, 6.4]), compared with 0.3 points under thinking OFF. To examine whether action–experience alignment contributes to this increase, we randomized whether the executed action matched the learner's preferred action. With thinking enabled, *structural transfer* became more sensitive to this action–experience alignment. The observed acting–observer difference exceeded the gap predicted from the randomized alignment effect, indicating that alignment alone was insufficient to account for the full gap. The same thinking-dependent pattern generalized to a structurally distinct five-variable causal-intervention task at Qwen3-8B, where the gap increased from 0.4 to 2.9 points (Δgap $= 2.5$ points, 95% CI [1.7, 3.3]).

Contributions. INTERGROUND isolates the role of acting under matched public experience in humans and language models. Acting improved human *structural transfer* but yielded only small acting–observer differences under standard adaptation. In Qwen3, learning-stage thinking increased the acting–observer difference, with randomized action–experience alignment accounting for part

of this effect. The same thinking-dependent pattern generalized to a structurally distinct causal-intervention environment. For agent training and replay, these results motivate retaining learner-specific pre-action reasoning and action preferences alongside the observable trajectory.

2 **INTERGROUND**: TASK, LEARNING ROLES, AND EVALUATION

2.1 HIDDEN INTERACTION RULES

INTERGROUND is a hidden-rule learning environment in which a persistent partner responds to one of two learner actions, A or B, with one of two outputs, X or Y. Six rules instantiate three transition mechanisms at thresholds of one or two consecutive actions: an irreversible switch from Y to X, an irreversible switch from X to Y, and a recoverable switch between the two outputs. Acting learners choose actions freely and infer the hidden rule from the resulting interaction record. Observers learn from matched replays of these interactions. Learners observe action–response sequences, without access to rule indices or explicit rule descriptions. Each rule is paired with an arbitrary nonce label. Label mappings, partner identities, and surface realizations vary across blocks, so learners must recover a latent response structure rather than memorize a familiar rule name, entity, or sentence. Appendix A specifies the six rules and teaching–evaluation support; Appendix C shows one rendered acting–observer pair.

2.2 ACTING AND OBSERVING UNDER MATCHED INTERACTION RECORDS

The central design principle is to match the realized public interaction while varying the learner's role in generating it (Figure 1). Within each yoked record, the acting learner and observer receive the same executed actions, partner responses, nonce labels, and evaluation items; only the acting learner selects the consequential action. Human acting participants select the actions, while yoked observers advance the matched replay without selecting its actions. Humans and language models learned the same six hidden transition rules and were evaluated using the same distinction between *label identification* and *structural transfer*, while exposure schedules and interface details were adapted to each population.

Model pairs begin from the same checkpoint and adapter initialization. The acting model selects the executed action, the observer model receives the resulting replay, and the two retain separate learner states, LoRA adapters (Hu et al., 2022), and optimizer states while undergoing their own online updates. Matching therefore concerns the public interaction record, not complete model inputs, internal states, or learned parameters. Appendix D specifies the standard model interaction and adaptation protocol.

2.3 EVALUATION

Evaluation separates *label identification* from *structural transfer*. *Label identification* asks which arbitrary label belongs to a response rule shown in a new trace. *Structural transfer*, our primary evaluation outcome, measures whether the learned transition rule can be applied to held-out partners and counterfactual action histories (Pearl, 2009) beyond the teaching record. Evaluation uses held-out state–history–action configurations and new surface realizations; adapted models retain a frozen adapter but receive no teaching history. Appendix A defines the disjoint teaching and evaluation support, and Appendix D.5 gives the exact scoring templates.

3 HUMAN ACTING PARTICIPANTS AND YOKED OBSERVERS

We first test whether acting changes *structural transfer* in humans under matched public experience. Acting participants selected one of two actions on each turn, and the hidden transition rule determined the partner's response. Yoked observers received the same recorded action–response sequence without selecting its actions. The comparison tests whether generating an interaction changes what is learned from the same public evidence. The study included 120 English-speaking adults, with 60 assigned to each role. All 120 adults were included in the analysis; no acting participants or yoked observers were excluded. Both groups were tested without access to the teaching history on *label identification* and *structural transfer* (Section 2).

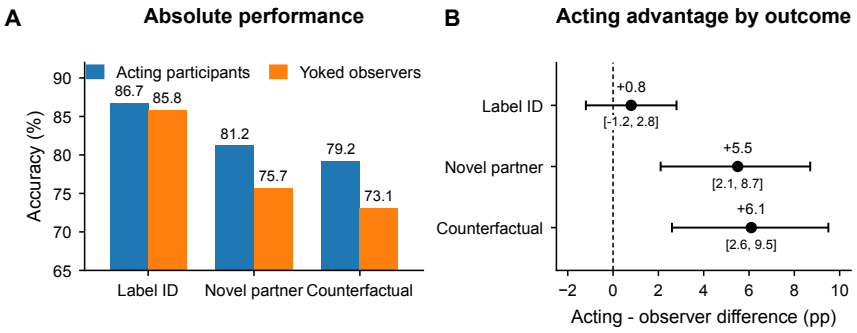

Figure 2: **Human acting–observer results.** **A**: Acting participants and yoked observers showed similar *label identification* but differed on *structural transfer*. **B**: Acting–observer differences across the evaluation components. Positive values indicate an acting advantage; error bars show bootstrap 95% confidence intervals.

3.1 Procedure and Analysis

Each episode presented one persistent partner, its initial response, and five turns of binary action selection and binary feedback. Acting participants used the accumulated history to infer the rule. Yoked observers saw the recorded actions and the same responses without altering them, advancing the replay with a neutral keypress under yoked timing. Action keys, test-response keys, rule order, nonce-label mapping, and surface realizations were counterbalanced. Each rule contributed four five-turn teaching episodes, resetting the environment at each episode boundary. The test battery contained 48 queries: 12 *label-identification* items and 36 *structural-transfer* items, comprising 18 novel-partner and 18 counterfactual-action items. Appendix B.4 gives the item templates.

The primary contrast was the acting–observer difference in *structural transfer*. We also tested whether the acting–observer difference was larger for *structural transfer* than for *label identification*. Primary confidence intervals were computed by resampling complete pairs of acting participants and yoked observers with replacement; a trial-level mixed-effects logistic regression provided an analysis of item-level responses. Appendix B describes the materials, procedure, and analysis.

3.2 Results

The acting advantage was larger for *structural transfer* than for *label identification* (Figure 2). *Label identification* was similar for acting participants and yoked observers (86.7% versus 85.8%), whereas *structural-transfer* accuracy was 80.2% for acting participants and 74.4% for yoked observers. This was an acting advantage of 5.8 percentage points (95% CI $[2.6, 8.9]$). The acting–observer difference was 5.0 percentage points larger for *structural transfer* than for *label identification* (95% CI $[2.1, 7.9]$, $p = .001$). A complementary trial-level mixed-effects logistic regression on *structural-transfer* items also estimated a positive acting participant effect ($\beta = 0.31$ log-odds, SE $= 0.08$, $z = 3.88$, $p < .001$). Section 4 compares this human acting contrast with model role differences.

4 Acting Has Little Effect in Language Models Under Standard Online Adaptation

The human study shows that acting can change what is learned from the same realized interaction. We therefore ask whether selecting the actions that determine training experience similarly improves *structural transfer* in language models, under the matched acting–observer online supervised LoRA protocol of Section 2.2. The acting model selects the actions that are executed, whereas the observer model learns from the same resulting action–response sequence.

We evaluated 11 configurations from SmolLM2 (Ben Allal et al., 2025), Qwen3 (Yang et al., 2025), Llama 3.1/3.2 (Meta AI, 2024a;b), and Gemma 3 (Gemma Team, 2025), spanning 0.36B to 14B

Table 1: *Structural-transfer* accuracy under standard online supervised adaptation across 11 model configurations. Accuracies are percentages; gap is acting model minus observer model in percentage points. Intervals are 95% CIs of the gap.

Model	Params	Acting model	Observer model	Gap	95% CI of gap
SmolLM2	0.36B	68.9	69.5	-0.6	$[-2.1, 0.8]$
Qwen3	0.6B	74.1	73.6	$+0.5$	$[-0.8, 1.8]$
Llama 3.2	1B	77.4	76.6	$+0.8$	$[-0.7, 2.3]$
Gemma 3	1B	79.3	79.9	-0.6	$[-1.9, 0.5]$
Qwen3	1.7B	82.1	81.8	$+0.3$	$[-0.9, 1.5]$
Llama 3.2	3B	82.0	80.7	$+1.3$	$[0.0, 2.7]$
Gemma 3	4B	84.8	84.4	$+0.4$	$[-0.9, 1.7]$
Qwen3	8B	86.2	85.9	$+0.3$	$[-0.9, 1.6]$
Llama 3.1	8B	85.7	86.2	-0.5	$[-1.7, 0.6]$
Gemma 3	12B	89.1	88.2	$+0.9$	$[-0.5, 2.4]$
Qwen3	14B	89.2	88.6	$+0.6$	$[-0.6, 1.9]$

parameters. Qwen3 uses thinking OFF for this baseline. Appendix D details the learning procedure. Acting models and observer models performed similarly (Table 1). *Structural-transfer* differences range from -0.6 to $+1.3$ percentage points. The largest gap is for Llama 3.2 3B: +1.3 points, 95% CI $[0.0, 2.7]$. The other intervals span zero. Both roles achieve strong *structural-transfer* performance, and all gaps are much smaller than the 5.8-point human acting advantage. Selecting the actions in a shared training trajectory yields little additional transfer benefit under this standard adaptation regime.

Secondary NLL analysis. Because categorical accuracy may obscure smaller shifts in candidate scores, we additionally examined candidate-normalized negative log-likelihood (NLL). Acting–observer NLL differences were small across model configurations, ranging from -0.011 to $+0.003$. Eight of the 11 point estimates favored acting models, but confidence intervals generally overlapped zero; only Llama 3.2 3B and Qwen3 14B had intervals excluding zero. Thus, the near-zero accuracy gaps do not appear to conceal a large sub-threshold acting–observer difference. Appendix D.5 specifies scoring and aggregation, and Table 8 reports the full model-wise results.

Together, the accuracy and NLL results suggest that standard adaptation is only weakly sensitive to whether the learner selected the actions generating the shared trajectory. Action selection alone therefore appears insufficient to make acting consequential for model learning. Active-learning accounts suggest that self-selected evidence may be especially useful when it tests the learner's current hypothesis (Freund et al., 1992; Markant & Gureckis, 2014), while intermediate reasoning can alter model decisions (Wei et al., 2022). We therefore ask whether learner-specific pre-action reasoning makes acting more consequential.

5 Learning-Stage Thinking Makes Acting Consequential

5.1 Thinking increases the acting–observer difference

We extend the Qwen3 acting–observer baseline from Section 4 by enabling Qwen3's native thinking mode during learning at 0.6B, 1.7B, 8B, and 14B, while holding the remaining learning and evaluation protocol fixed (Yang et al., 2025). Each condition uses ten seeds. The corresponding thinking-OFF conditions are the Qwen3 baselines already reported in Table 1.

Each learner forms a preferred action before execution and feedback. Thinking ON adds a learner-specific pre-action reasoning trace; only the acting model's preference determines the executed action. The generated trace is retained in the learner-specific update context but is not itself a supervised target; primary evaluation uses test thinking OFF.

Under thinking ON, acting–observer differences were 0.9, 2.1, 4.5, and 3.8 percentage points at 0.6B, 1.7B, 8B, and 14B, respectively (Table 11). Relative to the thinking-OFF baselines, the gap increased by 0.4, 1.8, 4.2, and 3.2 points (Figure 3A); the 95% confidence intervals excluded zero

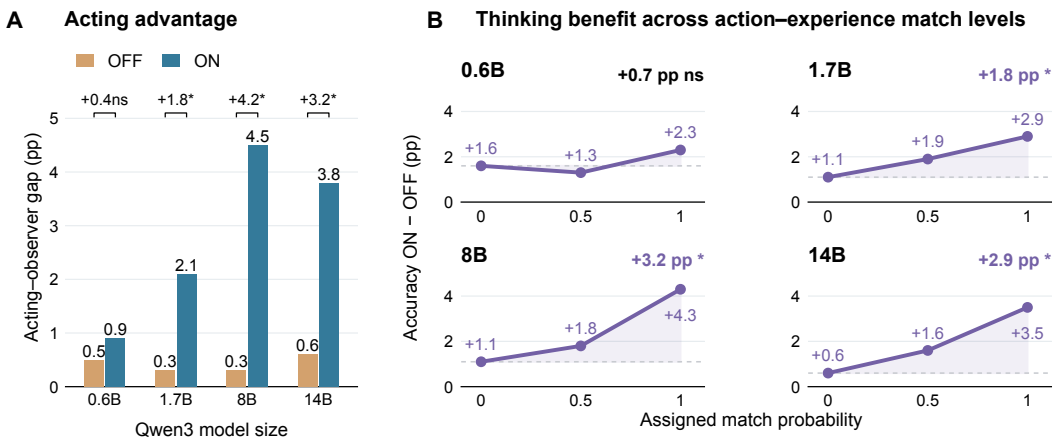

Figure 3: **Thinking increases the role gap and alignment sensitivity.** **A**: Acting–observer gaps; brackets show ON minus OFF. **B**: Accuracy ON minus OFF at each assigned match probability. *Note.* Upper-right values show the match-0-to-1 increase (Thinking × Match). Dashed lines mark match-0 values; shading shows changes from baseline, not uncertainty. For annotated increases (not individual points), * indicates a 95% CI excluding zero; ns indicates one including zero. Tables 11 and 12 provide full estimates and CIs.

at 1.7B, 8B, and 14B, but not at 0.6B. At 8B, for example, the acting–observer difference increased from 0.3 points in the thinking-OFF baseline (86.2% vs. 85.9%) to 4.5 points with thinking ON (90.5% vs. 86.0%).

Why does thinking increase the acting–observer difference? One possibility is that pre-action reasoning makes action–experience alignment more consequential for learning: the acting model observes the partner's response to its preferred action, whereas the observer model observes the same response to an action that may differ from its own preference. We therefore manipulate whether the learner's preferred action is executed under both thinking OFF and ON, testing whether thinking makes learning more sensitive to action–experience alignment.

5.2 Action–Experience Alignment Determines When Thinking Helps

We directly manipulate action–experience alignment by varying whether the executed action matches the learner's preferred action. On each turn, the learner's preferred action a^* is fixed before the executed action and response are revealed. The match intervention then selects the executed action according to

$$m = P(a_{\text{exec}} = a^*), \qquad m \in \{0, 0.5, 1\}. \tag{1}$$

At $m = 0$, the alternative action was always executed; at $m = 0.5$, either action was executed with equal probability; and at $m = 1$, the preferred action was always executed. Online adaptation then followed the same procedure as in the role comparison.

At 8B, moving from 0% to 100% match changed *structural-transfer* accuracy by 0.3 points with thinking OFF and 3.5 points with thinking ON, yielding a **3.2-point Thinking × Match interaction** (95% CI $[1.8, 5.6]$). Figure 3B shows the thinking benefit (ON minus OFF) at each match level; its increase from 0% to 100% match equals this interaction. Thinking OFF produces little sensitivity to match probability. With thinking ON, *structural transfer* increases more as the learner's preferred action is more often realized. The interaction point estimate is positive at each scale; its 95% CI excludes zero at 1.7B, 8B, and 14B, but not at 0.6B.

5.3 Alignment Explains Only Part of the Acting–Observer Difference

In the thinking-ON role comparison, the acting model's preferred action determines execution, so preferred–executed action match is 1 by construction. The observer model forms its own preferred action but receives the acting model's executed action; its natural match rate $\widehat{m}_{\text{obs}}$ is measured

Table 2: Quantitative bridge for thinking-ON Qwen3 models. Predicted gaps, observed gaps, and residuals are in percentage points. The prediction is $s_{\mathrm{ON}}(1-\widehat{m}_{\mathrm{obs}})$; the residual is the observed gap minus this prediction. Residual 95% CIs propagate uncertainty through seed/block-level resampling.

Model	$\widehat{m}_{\mathrm{obs}}$	Alignment-predicted gap	Observed gap	Residual	95% CI residual
0.6B	0.56	0.44	0.9	+0.46	$[-0.42, 1.32]$
1.7B	0.61	0.82	2.1	+1.28	$[0.28, 2.99]$
8B	0.69	1.09	4.5	+3.42	$[1.05, 4.77]$
14B	0.73	0.59	3.8	+3.21	$[1.64, 4.80]$

from the saved logs. If the acting–observer difference were explained solely by the alignment effect measured in Section 5.2, its predicted magnitude would be approximately

$$s_{\mathrm{ON}}(1-\widehat{m}_{\mathrm{obs}}),$$

where s_{ON} is the model-specific 0-to-100% assigned-match slope under thinking ON. We subtract this prediction from the observed difference to obtain the residual. Uncertainty is propagated through the full bridge calculation using seed/block-level resampling (Appendix E.8).

The observer model's natural preferred–executed match rate rose from 0.56 at 0.6B to 0.73 at 14B, yet observed acting–observer differences exceeded alignment-based predictions at every scale (Table 2). Residuals ranged from $+0.46$ to $+3.42$ points and excluded zero at 1.7B, 8B, and 14B, but not at 0.6B. Thus, under the bridge's assumptions, preferred–executed alignment explains only part of the thinking-ON acting–observer difference; the remainder may reflect other coupling between learner-specific computation and experience.

5.4 Controls localize the effect

We use four controls to localize the effect: frozen in-context learning without parameter updates, learning versus test-time thinking, reasoning-conditioned updates, and trajectory source.

Frozen in-context learning. We repeated the acting–observer comparison with frozen Qwen3-1.7B and 8B, learning purely in context (Akyürek et al., 2023), retaining the teaching history in context and performing no adaptation (Appendix E.9). With thinking OFF, the two roles share the same public history and the same frozen weights, so their evaluations coincide and the gap is zero by construction. With thinking ON, each learner generated a bounded pre-action reasoning trace (at most 24 tokens per turn) that was retained verbatim in its own subsequent context together with its preferred action. The acting–observer difference was 1.5 points at 1.7B (95% CI $[0.6, 3.6]$) and 2.3 points at 8B (95% CI $[0.9, 4.9]$; Table 3). Absolute accuracy was lower than under online adaptation, but an acting-model advantage associated with thinking emerged even without parameter updates. This shows that the acting–observer difference does not require online adaptation and can emerge when each learner's own reasoning and action preference remain coupled to subsequent experience in context.

Table 3: Frozen in-context learning with Qwen3. *Structural-transfer* accuracy (%) with the teaching history retained in context and no parameter updates. Gap is acting model minus observer model in percentage points; intervals are 95% CIs of the gap.

Model	Thinking	Acting model	Observer model	Gap	95% CI of gap
1.7B	OFF	76.8	76.8	0.0	—
1.7B	ON	79.6	78.1	+1.5	$[0.6, 3.6]$
8B	OFF	82.5	82.5	0.0	—
8B	ON	85.8	83.5	+2.3	$[0.9, 4.9]$

Learning versus test thinking. Test-only thinking did not reproduce the larger acting–observer difference associated with learning-stage thinking. At 8B, enabling test-time thinking changed

the gap by $+0.7$ points when learning thinking was OFF and -1.5 points when it was ON. The learning-related increase in the gap therefore differed by -2.2 points across test modes (95% CI $[-3.7, -0.7]$), indicating that the learning-stage Thinking $\times$ Role effect was smaller with test thinking ON (2.0 points) than OFF (4.2 points). The full comparison is reported in Appendix E (Table 9).

Reasoning-conditioned updates. To test whether private reasoning contributes to the alignment effect under thinking, we repeat the alignment analysis while removing reasoning from the update context. The 8B Thinking $\times$ Match interaction decreases from 3.2 points with full prefixes to 1.2 points with public-record-only updates, a reduction of 2.0 points (95% CI $[0.8, 3.3]$). This indicates that reasoning-conditioned updates contribute to, but do not fully explain, the alignment effect.

Trajectory source. With observer model learning and test thinking both fixed OFF, observer models score 85.9% on trajectories from thinking-OFF acting models and 86.7% on trajectories from thinking-ON acting models. The difference is +0.8 points (95% CI $[-0.2, 1.8]$). Changing trajectory source while holding observer model computation fixed produces a small, uncertain difference.

Together, these results suggest that learning-stage thinking changes how subsequent experience is incorporated. Under the bridge's assumptions, preferred–executed action alignment explains only part of the acting–observer difference. Control experiments further implicate learner-specific pre-action computation: test-time thinking alone did not reproduce the larger gap, removing private reasoning attenuated the alignment effect, and the gap could emerge without parameter updates (Appendix E).

5.5 Conceptual Replication in a Causal-Intervention Environment

We next ask whether the thinking-dependent acting–observer difference is specific to the binary transition-rule environment. We use a second controlled environment in which each hidden system is a causal DAG over five binary variables (Appendix G). Each turn begins from the same baseline state. The learner selects one variable to flip, and the environment recomputes the resulting state according to the hidden causal mechanism. The system then resets to the baseline before the next turn, so interventions are evaluated independently rather than accumulating over time.

Evaluation uses a baseline derived from a root configuration held out from teaching and asks the model to predict the post-intervention value of a non-intervened variable; some queries require the effect of an intervention to propagate across multiple causal edges. Relative to the primary environment, this changes the action space (five-way rather than binary), the feedback (a full state vector rather than a single response), the latent structure (a DAG rather than a finite-state transition rule), and the transfer demand (a held-out causal baseline rather than transfer across partner identities and action histories). The role manipulation is unchanged: both learners independently form an intervention preference, only the acting model's preference determines execution, and both learners receive the same executed intervention, resulting state, supervised targets, and update count. We evaluate Qwen3-8B with learning-stage thinking OFF and ON across five matched seeds.

With thinking OFF, acting-model and observer-model accuracies were 71.4% versus 71.0% (gap $+0.4$ points, 95% CI $[-0.2, 1.0]$). With thinking ON, both improved, but acting models improved more: 76.8% versus 73.9% (gap $+2.9$ points, 95% CI $[1.9, 3.9]$). Thinking therefore increased the acting–observer difference by 2.5 points (95% CI $[1.7, 3.3]$). Thus, the thinking-dependent acting–observer difference generalizes to a second controlled environment with a different latent structure, action space, feedback representation, and transfer requirement.

6 Related Work

Active–yoked studies in human learning ask whether selecting or generating evidence changes learning relative to receiving matched evidence. Learning differences can persist even when observable evidence is matched (Markant & Gureckis, 2014; Brod et al., 2025). Related work on causal learning further shows benefits of self-generated interventions and formalizes how learners select informative experiments (Sobel & Kushnir, 2006; Bramley et al., 2017).

In artificial learners, agents can acquire causal experimentation strategies from expert intervention trajectories (Lampinen et al., 2023). Offline sequence learning, on-policy distillation, and replay

instead study learning from trajectories collected at different times or generated by different policies (Chen et al., 2021; Agarwal et al., 2024; Gu et al., 2024; Chen et al., 2026). These approaches typically change the source or distribution of training experience. **INTERGROUND** instead asks whether the same realized interaction is learned differently when it is generated by the learner's own choice rather than received as replay.

Separate work studies how intermediate reasoning changes model decisions and training dynamics (Nye et al., 2021; Wei et al., 2022; Kojima et al., 2022; Li et al., 2025), how explanations in context affect learning from examples (Lampinen et al., 2022), and how self-generated rationales can support subsequent training (Zelikman et al., 2022; 2024). Agent methods such as ReAct and Reflexion further combine reasoning, action, and feedback across interaction (Yao et al., 2023; Shinn et al., 2023). Prior work primarily asks whether reasoning improves decisions or can itself serve as training data. We specifically ask whether computation performed before an action changes how the subsequent experience is learned, while holding the realized public interaction fixed. This isolates learner-specific computation from differences in the trajectory itself. Appendix F provides the extended review.

7 Discussion

Our results separate two factors that are often confounded in interactive learning: the observable experience available to a learner and the learner's role in generating that experience. Human acting participants learned more transferable structure when they acted than when they observed the same realized interaction, whereas language models under standard online supervised adaptation showed little acting–observer difference. This pattern changed when learning included learner-specific thinking: across Qwen3 scales, thinking increased the acting–observer difference, and randomized action–experience alignment showed that transfer became more sensitive to whether the executed action matched the learner's prior preference. Thus, trajectory equivalence does not necessarily imply learning equivalence. Because the realized public record was fixed across roles, this difference cannot be attributed simply to differences in trajectory informativeness.

The alignment intervention identifies one causal component of this relationship. Under thinking, *structural transfer* improved more strongly as executed actions were made increasingly likely to match the learner's preferred action. This connects to active-learning accounts in which the value of evidence depends on the learner's current hypotheses (Freund et al., 1992; Gureckis & Markant, 2012; Markant & Gureckis, 2014; Bramley et al., 2017). Alignment, however, did not explain the full acting–observer difference. The remaining difference may reflect reasoning-conditioned updates, evolving learner state, or other forms of coupling between computation and experience. Our controls suggest that the effect depends on learner-specific computation rather than trajectory source alone.

The five-variable causal-intervention experiment suggests that the thinking-dependent acting–observer difference is not specific to the original binary transition-rule task. However, both environments are still compact and highly controlled. It remains unclear whether learner-specific pre-action computation has the same influence in richer interactive settings with longer horizons, larger action spaces, and noisier feedback.

8 Conclusion

INTERGROUND shows that holding the realized public interaction fixed does not eliminate differences between acting and observing. Humans learned more transferable structure from interactions they generated themselves, whereas language models under standard online adaptation showed little role difference. In Qwen3, learning-stage thinking increased the acting–observer gap across scales, and randomized action–experience alignment showed that this increase partly depends on whether subsequent experience realizes the learner's prior action preference. The same thinking-dependent pattern appeared in a structurally distinct causal-intervention environment. **Together, these results show that interactive experience is not fully characterized by the observable trajectory alone: its relation to the learner's own pre-action computation can shape what is learned.**

Reproducibility Statement

The appendices document the experimental procedures and analyses to support reproducibility. Appendix A defines the hidden rules and transition dynamics, and Appendix C illustrates a matched learning pair. Appendix B describes the human-study instructions, teaching schedule, evaluation items, and statistical analysis. Appendix D describes the model prompts, online adaptation procedure, evaluation templates, and accuracy and NLL scoring. Appendix E describes the thinking and alignment interventions, reasoning budgets, control conditions, and seed-level resampling, including the bridge analysis and frozen in-context learning protocol. Appendix G specifies the causal-intervention replication, including mechanisms, intervention semantics, matched learning, teaching and evaluation support, prompts, and seed-level inference. All code will be made publicly available on GitHub.

Ethics Statement

The adult protocol was approved by the authors' institutional review board. All 120 adults provided informed consent and were compensated. The tasks involved minimal-risk judgments about fictional interactive systems and hidden response rules; no sensitive personal attributes were collected for analysis.

AI Use Statement

An LLM-based writing and coding assistant was used for language editing, code review, figure-generation support, and preparation of experiment scripts. The authors reviewed all generated material and remain responsible for the analyses, claims, and final text.

References

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2306.13649.

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? Investigations with linear models. In *International Conference on Learning Representations*, 2023.

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, et al. SmolLM2: When smol goes big—data-centric training of a small language model. *arXiv preprint arXiv:2502.02737*, 2025. URL https://arxiv.org/abs/2502.02737.

Neil R. Bramley, Peter Dayan, Thomas L. Griffiths, and David A. Lagnado. Formalizing Neurath's ship: Approximate algorithms for online causal learning. *Psychological Review*, 124(3):301–338, 2017. doi: 10.1037/rev0000061.

Garvin Brod, Elfriede Holstein, Leonie Weindorf, Joseph Colantonio, Elizabeth Bonawitz, and Maria Theobald. Do it yourself: Discerning the effects of self-directed activity on conceptual learning. *npj Science of Learning*, 10:70, 2025. doi: 10.1038/s41539-025-00364-9.

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.

Rui Castro, Charles Kalish, Robert Nowak, Ruichen Qian, Timothy Rogers, and Xiaojin Zhu. Human active learning. In *Advances in Neural Information Processing Systems*, volume 21, 2008.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Zhixun Chen, Ming Li, Yuxuan Huang, Yali Du, Meng Fang, and Tianyi Zhou. ATLAS: Agent tuning via learning critical steps. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 25334–25349, 2025. doi: 10.18653/v1/2025.findings-acl.1299.

Yan Chen, Taojie Zhu, Meng Zhang, Xin Chen, Jiaqi Huang, Dongyang Xu, and Yizhi Wang. On-policy replay for continual supervised fine-tuning. *arXiv preprint arXiv:2605.29495*, 2026. URL https://arxiv.org/abs/2605.29495.

Anna Coenen, Bob Rehder, and Todd M. Gureckis. Strategies to intervene on causal systems are adaptively selected. *Cognitive Psychology*, 79:102–133, 2015. doi: 10.1016/j.cogpsych.2015.02.004.

Anna Coenen, Jonathan D. Nelson, and Todd M. Gureckis. Asking the right questions about the psychology of human inquiry: Nine open challenges. *Psychonomic Bulletin & Review*, 26(5):1548–1587, 2019. doi: 10.3758/s13423-018-1470-5.

Sara De Felice, Gabriella Vigliocco, and Antonia F. de C. Hamilton. Social interaction is a catalyst for adult human learning in online contexts. *Current Biology*, 31(21):4853–4859.e3, 2021. doi: 10.1016/j.cub.2021.08.045.

DeepSeek-AI, Daya Guo, Dejian Yang, et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. URL https://arxiv.org/abs/2501.12948.

Yoav Freund, H. Sebastian Seung, Eli Shamir, and Naftali Tishby. Information, prediction, and query by committee. In *Advances in Neural Information Processing Systems*, volume 5, 1992.

Gemma Team. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025. URL https://arxiv.org/abs/2503.19786.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. In *International Conference on Learning Representations*, 2024.

Todd M. Gureckis and Douglas B. Markant. Self-directed learning: A cognitive and computational perspective. *Perspectives on Psychological Science*, 7(5):464–481, 2012. doi: 10.1177/1745691612454304.

Richard Held and Alan Hein. Movement-produced stimulation in the development of visually guided behavior. *Journal of Comparative and Physiological Psychology*, 56(5):872–876, 1963. doi: 10.1037/h0040546.

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://openreview.net/forum?id=nZeVKeeFYf9.

Michael Janner, Qiyang Li, and Sergey Levine. Offline reinforcement learning as one big sequence modeling problem. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Patricia K. Kuhl, Feng-Ming Tsao, and Huei-Mei Liu. Foreign-language experience in infancy: Effects of short-term exposure and social interaction on phonetic learning. *Proceedings of the National Academy of Sciences*, 100(15):9096–9101, 2003. doi: 10.1073/pnas.1532872100.

David A. Lagnado and Steven Sloman. The advantage of timely intervention. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 30(4):856–876, 2004. doi: 10.1037/0278-7393.30.4.856.

Andrew K. Lampinen, Ishita Dasgupta, Stephanie C. Y. Chan, Kory Mathewson, Michael Henry Tessler, Antonia Creswell, James L. McClelland, Jane X. Wang, and Felix Hill. Can language models learn from explanations in context? In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 537–563, 2022.

Andrew K. Lampinen, Stephanie C. Y. Chan, Ishita Dasgupta, Andrew J. Nam, and Jane X. Wang. Passive learning of active causal strategies in agents and language models. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Andrew K. Lampinen, Arslan Chaudhry, Stephanie C. Y. Chan, Cody Wild, Diane Wan, Alex Ku, Jörg Bornschein, Razvan Pascanu, Murray Shanahan, and James L. McClelland. On the generalization of language models from in-context learning and finetuning: A controlled study. *arXiv preprint arXiv:2505.00661*, 2025. URL https://arxiv.org/abs/2505.00661.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020. URL https://arxiv.org/abs/2005.01643.

Ming Li, Lichang Chen, Jiuhai Chen, Shwai He, Jiuxiang Gu, and Tianyi Zhou. Selective reflection-tuning: Student-selected data recycling for LLM instruction-tuning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 16189–16211, 2024. doi: 10.18653/v1/2024.findings-acl.958.

Ming Li, Yanhong Li, and Tianyi Zhou. What happened in LLMs layers when trained for fast vs. slow thinking: A gradient perspective. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, pp. 32017–32154, 2025. doi: 10.18653/v1/2025.acl-long.1545.

Douglas B. Markant and Todd M. Gureckis. Is it better to select or to receive? Learning via active and passive hypothesis testing. *Journal of Experimental Psychology: General*, 143(1):94–122, 2014. doi: 10.1037/a0032108.

Meta AI. Introducing Meta Llama 3.1: Our most capable models to date. URL https://ai.meta.com/blog/meta-llama-3-1/, 2024a.

Meta AI. Llama 3.2 model card. URL https://github.com/meta-llama/llama-models/blob/main/models/llama3_2/MODEL_CARD.md, 2024b.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, Charles Sutton, and Augustus Odena. Show your work: Scratchpads for intermediate computation with language models. *arXiv preprint arXiv:2112.00114*, 2021. URL https://arxiv.org/abs/2112.00114.

Long Ouyang, Jeff 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 Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, second edition, 2009.

Qwen Team. Qwen3-8B model card. URL https://huggingface.co/Qwen/Qwen3-8B, 2025. Accessed September 11, 2026. Native thinking/non-thinking modes and generation guidance.

Stéphane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 627–635, 2011.

Laura E. Schulz and Elizabeth Baraff Bonawitz. Serious fun: Preschoolers engage in more exploratory play when evidence is confounded. *Developmental Psychology*, 43(4):1045–1050, 2007. doi: 10.1037/0012-1649.43.4.1045.

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

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, 2023.

Norman J. Slamecka and Peter Graf. The generation effect: Delineation of a phenomenon. *Journal of Experimental Psychology: Human Learning and Memory*, 4(6):592–604, 1978. doi: 10.1037/0278-7393.4.6.592.

David M. Sobel and Tamar Kushnir. The importance of decision making in causal learning from interventions. *Memory & Cognition*, 34(2):411–419, 2006.

Mark Steyvers, Joshua B. Tenenbaum, Eric-Jan Wagenmakers, and Ben Blum. Inferring causal networks from observations and interventions. *Cognitive Science*, 27(3):453–489, 2003. doi: 10.1207/s15516709cog2703_6.

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

Fahim Tajwar, Anikait Singh, Archit Sharma, Rafael Rafailov, Jeff Schneider, Tengyang Xie, Stefano Ermon, Chelsea Finn, and Aviral Kumar. Preference fine-tuning of LLMs should leverage suboptimal, on-policy data. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, 2024.

Joel L. Voss, Brian D. Gonsalves, Kara D. Federmeier, Daniel Tranel, and Neal J. Cohen. Hippocampal brain-network coordination during volitional exploratory behavior enhances learning. *Nature Neuroscience*, 14(1):115–120, 2011. doi: 10.1038/nn.2693.

Ruiyi Wang, Haofei Yu, Wenxin Zhang, Zhengyang Qi, Maarten Sap, Yonatan Bisk, Graham Neubig, and Hao Zhu. SOTOPIA-π: Interactive learning of socially intelligent language agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12912–12940, Bangkok, Thailand, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.698.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837, 2022. URL https://arxiv.org/abs/2201.11903.

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

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. STaR: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://arxiv.org/abs/2203.14465.

Eric Zelikman, Georges Harik, Yijia Shao, Varuna Jayasiri, Nick Haber, and Noah D. Goodman. Quiet-STaR: Language models can teach themselves to think before speaking. In *Conference on Language Modeling*, 2024. URL https://arxiv.org/abs/2403.09629.

A Environment and Hidden-Rule Specification

A.1 Task dynamics

Each partner has two actions, A and B, and two responses, X and Y. Each hidden rule defines a deterministic finite-state transition system. On each turn, the learner's action updates the current state, and the updated state determines the response shown on that same turn. Thus, an action that reaches a threshold changes the response immediately. Each episode resets to the rule-specific initial response and clears the consecutive-action count. The same transition logic is used to generate teaching trajectories and recompute held-out and counterfactual evaluation targets. The system is presented as an interaction between a learner and a partner. A nonce label names the rule within a block; transitions depend only on the actions and rule-specific state.

A.2 Six hidden rules

The six rules instantiate three transition mechanisms, each at thresholds $k = 1$ and $k = 2$. Figure A1 visualizes the shared finite-state structure, and the rule descriptions below specify the six concrete rule instances.

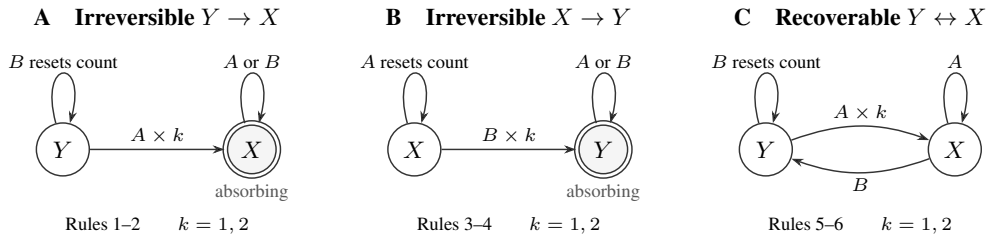

Figure A1: Finite-state structure of the six hidden rules. The six rules instantiate three transition mechanisms, each at thresholds $k = 1$ and $k = 2$. Rules 1–2 irreversibly switch from Y to X after k consecutive A actions; Rules 3–4 irreversibly switch from X to Y after k consecutive B actions; Rules 5–6 switch from Y to X after k consecutive A actions and return to Y when B occurs.

Table 4: Hidden rules, initial outputs, and transition logic.

Rule	Initial output	Transition logic
Rule 1	Y	One A switches the output to X, which persists for the rest of the episode.
Rule 2	Y	Two consecutive A actions switch the output to X, which then persists. Before switching, B resets the counter.
Rule 3	X	One B switches the output to Y, which persists for the rest of the episode.
Rule 4	X	Two consecutive B actions switch the output to Y, which then persists. Before switching, A resets the counter.
Rule 5	Y	One A switches the output to X; any B immediately returns it to Y. This cycle can repeat.
Rule 6	Y	Two consecutive A actions switch the output to X; any B immediately returns it to Y and resets the counter. This cycle can repeat.

For Rules 1–2, A increments the counter while the state is Y and B clears it; reaching k enters the absorbing state X. For Rules 3–4, B increments the counter while the state is X and A clears it; reaching k enters the absorbing state Y. For Rules 5–6, any B returns the output to Y and clears the counter; while in Y, consecutive A actions increment the counter until k, at which point the output

becomes X. Further A actions maintain X. Rule 5 is behaviorally equivalent to the memoryless mapping $A \mapsto X$, $B \mapsto Y$ and provides a simple baseline. Rules 1–2 and their threshold-matched Rules 5–6 have identical responses until an A streak establishes X and a subsequent B tests whether that state is persistent.

A.3 Teaching and evaluation support

Canonical teaching queries belong to K_{teach}. Evaluation queries belong to disjoint K_{eval} support and use state–history–action combinations absent from realized teaching records and supervised adaptation targets. Novel-entity items change the entity identifier or surface realization. Model counterfactual-action items replace an earlier action in a held-out history and recompute subsequent states and responses while keeping the remaining actions fixed. Development and test worlds use disjoint surface realizations. These separations prevent *structural transfer* from being solved by reproducing a teaching string. Appendix D specifies rendering, label assignment, and the exact evaluation prompts.

B Adult Study Materials and Analysis

Humans learned the same six finite-state transition rules as the language models (Appendix A). Actions A/B and responses X/Y were rendered through fixed one-to-one surface mappings. Each episode began from its rule-specific initial response, with counters reset to zero. State updates preceded the displayed response, including on threshold-crossing turns. Instructions did not disclose the six rules, their mechanism categories, or their thresholds.

B.1 Recruitment, assignment, and sample

The study sample contained 120 English-speaking adults: 60 acting participants and 60 yoked observers. All 120 adults were included in the analysis; no acting participants or yoked observers were excluded. Eligible adults were at least 18, used a laptop or desktop keyboard, provided informed consent, and had no prior exposure to the task. Compensation was based on the piloted median completion time and was independent of accuracy. The protocol was approved by the authors' institutional review board. Action keys, test-response keys, rule order, nonce-label mapping, and surface realization were counterbalanced. Yoked observers received recorded acting participant interactions without regenerating their responses.

B.2 Instructions and practice

Acting participants received the following instructions:

```
You will interact with several characters.  Each character
follows a stable hidden response rule.
On each turn, choose one of two actions.  The character will
respond according to its hidden rule.
The response may depend on what you choose now and on what
happened on previous turns.
Use the interaction history to learn how the rule works.
After each interaction episode, you will see a new word naming
that hidden rule.  You will later be asked to apply the rule
in new situations.
```

Yoked observers received parallel instructions:

```
You will observe interactions generated by a previous player.
On each turn, you will see the action selected by that player
and the character's response.
Press the indicated key to advance the replay.  You cannot
change the action or response.
Use the observed interaction history to learn the character's
```

```
hidden response rule.
After each interaction episode, you will see a new word naming
that hidden rule.  You will later be asked to apply the rule
in new situations.
```

A four-turn practice block familiarized learners with the interface using a constant-Y response rule, which is not one of the six target rules. Practice taught action selection, response reading, history display, and the distinction between selecting an action during learning and predicting a response during testing. Instructions explained that earlier turns may matter and that the hidden rule remains stable within a learning block; practice did not demonstrate the target thresholds or transition categories. Comprehension checks asked learners to identify an action and its response, confirm whether previous turns can affect later responses, and distinguish a test prediction from a new learning action.

B.3 Trial timing and teaching schedule

Each learner completed four five-turn episodes per hidden rule. Every episode reset to the specified initial response and zero counter; acting participants then chose all five actions freely. A single partner remained present throughout an episode, and the rule remained constant within its learning block.

At episode onset the initial response was displayed. Each turn showed the accumulated action–response history, followed by two action choices. The acting participant selected an action, saw the selected action and then the partner's response, and the completed pair was appended to the visible history. The response was displayed for 1,000 ms. Yoked observers saw the recorded action and response in the same order, using a neutral reveal key to advance the replay. The recorded action was visible for the source acting participant's response time, winsorized to the 500–5,000 ms range, before the yoked observer revealed the response. After the fifth turn, the nonce rule label was displayed for 2,000 ms. Breaks separated rule blocks.

Teaching episodes used free action selection from the rule-specific initial state; no diagnostic starting history was inserted. The realized action–response trajectory was shared exactly with the yoked observer, and five turns did not guarantee observation of every diagnostic transition.

B.4 Evaluation templates and scoring

The test battery contained 48 queries: 12 *label-identification* items, 18 novel-partner items, and 18 counterfactual-action items. Each rule contributed two, three, and three items, respectively. The teaching record was unavailable during testing; each item supplied its own initial response and history. No correctness feedback was given. The programmed transition engine supplied all answer keys, and response-choice positions were counterbalanced.

Label identification. A new trace was presented with the six nonce labels assigned to the learning block:

```
This character follows one of the rules you learned.
Initial response:  <X/Y>
Turn 1:  <ACTION> -> <RESPONSE>
...
Turn 5:  <ACTION> -> <RESPONSE>
Which word names this rule?
Choices:  <SIX NONCE LABELS>
```

Each trace was behaviorally diagnostic among the six rules. Rule 1 versus Rule 5 required a B after reaching X to test persistence; Rule 2 versus Rule 6 required a post-threshold B to distinguish persistence from reset. Ambiguous traces and irrelevant surface cues were not used to identify the rule.

Novel-partner transfer. A new partner followed a previously learned nonce-labeled rule. Each query supplied the initial response, a held-out action–response history, and a current action:

```
A new character follows the rule called <NONCE>.
Initial response:  <X/Y>
Previous turns:  <ACTION-RESPONSE HISTORY>
Now <ACTION> occurs.  What response follows?
Choices:  <RESPONSE-X> | <RESPONSE-Y>
```

The partner identity changed while the underlying rule was preserved.

Counterfactual-action transfer. These items changed an earlier action in a held-out history and asked for a later response:

```
Rule label:  <NONCE>.  Initial response:  <X/Y>.
Original history:  <ACTION-RESPONSE HISTORY>.
Suppose action <INDEX> had been <ALTERNATIVE-ACTION> instead.
Keep the remaining actions unchanged.  Responses follow from
the changed history.
What response follows when <QUERY-ACTION> occurs next?
Choices:  <RESPONSE-X> | <RESPONSE-Y>
```

The transition engine recomputed the state from the altered action through the remaining history. Original downstream responses were not held fixed. When an earlier action changed the state, prediction required propagating that change rather than substituting surface words. Novel-partner and counterfactual-action accuracy were equally weighted within rule and averaged across rules to form the primary *structural-transfer* score. *Label identification* was a separate, secondary outcome.

B.5 Analysis sample and inference

No acting participants or yoked observers were excluded from the analysis; both groups retained all 60 enrolled adults.

The primary contrast compared mean *structural-transfer* accuracy Y^r between roles r:

$$\Delta_{\text{role}} = Y^{\text{part}} - Y^{\text{obs}}. \tag{2}$$

We also tested whether the acting–observer difference was larger for *structural transfer* than for *label identification* by subtracting the *label-identification* gap from the *structural-transfer* gap. Individual transfer components were secondary outcomes. Primary confidence intervals, including the error bars in Figure 2, were computed by resampling the 60 complete pairs of acting participants and yoked observers with replacement, preserving the acting–observer match and all test observations within each learner. Each bootstrap replicate recomputed both the acting–observer *structural-transfer* difference and how much it exceeded the *label-identification* difference. A trial-level mixed-effects logistic regression provided a complementary analysis of item-level responses. Reported human estimates and the result figure are given in Section 3.

C Worked Example: One Matched Learning Pair

The example uses Rule 6 with nonce label *pimel*. It illustrates the shared finite-state environment; the rule number is not shown to learners. Each episode begins with response Y and a zero consecutive-A counter.

Shared action–response record. The explanatory column is for the reader, not part of the teaching record. After turn 5, both learners receive the label *pimel*.

Held-out *structural-transfer* example. A new partner can be tested using a different history:

```
A new partner S follows the rule called pimel.
Initial response:  Y.
Previous turns:  B -> Y; A -> Y.
Now A occurs.  What response follows?  X or Y?
```

Table 5: Shared action–response record for a matched learning pair under Rule 6.

Turn	Action	Response	Transition explanation
1	A	Y	First consecutive A: below threshold.
2	A	X	Second consecutive A: switch on this turn.
3	B	Y	Return to Y and clear the counter.
4	A	Y	First A since reset.
5	A	X	Second consecutive A: switch again.

The answer is X: the query action completes two consecutive A actions. A corresponding counterfactual item changes the second action in that history from A to B, keeps the final query action A, and recomputes the resulting state. The revised sequence is B, B, A, so the answer becomes Y. Both questions use the same learned rule; the latter requires accounting for the changed history. *Label identification* instead asks which nonce label names a new diagnostic trace of that rule.

D Standard Model Interaction and Online Learning Protocol

This appendix specifies the standard thinking-OFF acting–observer protocol for **INTERGROUND**. Appendix A defines the hidden rules and task dynamics; Appendix E describes the thinking and action–experience alignment extensions.

D.1 Action, response, and nonce-label rendering

Model-visible records contain a persistent partner identifier, the initial response, action–response history, and the arbitrary rule label revealed after an episode. We use *partner* throughout for the responding entity; the model-facing prompts quoted below render it as `system`, and the human-facing materials (Appendix B) render it as *character*. Actions and responses are rendered through one-to-one mappings onto A/B and X/Y. Development and test surface realizations may differ while preserving these mappings. Surface forms do not alter the latent transition function. Rule numbers are analytical identifiers and are not supplied as descriptions of the hidden rule.

The nonce-label pool is

```
mepo, daxen, luki, voma, narel, tovin, zebi, fepal, kudra,
sovin,
pimel, gatu, rensa, bilor, cevi, wugam.
```

For each learning–evaluation block with seed s, a seeded uniform permutation selects six nonce labels without replacement and assigns them to Rules 1–6. This label–rule mapping is shared by the acting and observer models and remains fixed throughout learning and evaluation. Test-item generation uses a separate seed, $s + 1{,}000{,}000$, without resampling the label–rule mapping. The mapping is stored in the run manifest. Entity identifiers used for novel-partner evaluation are held out from teaching.

Instruction-tuned models use the released chat template and the system message

```
Return only the requested answer token or word.
```

Base checkpoints receive the same instruction as a plain-text prefix. Angle-bracketed fields below denote rendering slots, not literal model-visible strings.

D.2 Baseline acting–observer protocol

Here, the acting model selects the executed action, whereas the observer model receives the matched acting model-generated trajectory. Acting model and observer model begin from the same checkpoint and adapter initialization, then retain separate adapters and optimizer states. Both undergo online supervised LoRA adaptation. Both learners independently form a preferred action from the current public history before the executed action and response are revealed. Only the acting model's

preferred action determines execution; the observer model's preferred action is recorded but has no causal effect on the public trajectory. The observer model receives the matched acting model-generated executed action and resulting response. Both learners receive the same initial response, realized public action–response sequence, block label, supervised targets, and number of updates.

At each teaching turn, the acting model receives only the public information available before selecting that turn's action:

```
You are interacting with one persistent system whose response
rule is unknown.
System:  <ENTITY>
Initial response:  <INITIAL-RESPONSE>

Previous turns:
<RENDERED ACTION-RESPONSE HISTORY OR NONE>

Choose the action that will help you learn the hidden rule.
Answer <ACTION-A> or <ACTION-B> only.
```

Under thinking OFF, each learner scores the two complete candidate action strings by length-normalized log probability using its own current model state and records the higher-scoring candidate as its preferred action. The environment executes the acting model's preferred action, updates its hidden state, and emits the corresponding response on that turn.

After five turns, the rule label is revealed and attached to the completed episode record. The label header is present only after the episode and is omitted from examples in which the label is the prediction target.

Supervised targets. Response-prediction examples contain the initial response, preceding action–response history, and current executed action, with the current response hidden and later turns omitted. They append

```
Predict the system response.
Answer <RESPONSE-X> or <RESPONSE-Y> only.

Answer:  <OBSERVED RESPONSE>
```

Label-identification examples contain the complete five-turn trace and the six candidate nonce labels, but not the correct label header. They append

```
Labels:  <SIX COUNTERBALANCED NONCE LABELS>
Which concept label denotes this hidden response rule?

Answer:  <NONCE>
```

Cross-entropy is masked to answer tokens. Candidate order is independently counterbalanced and shared across matched acting–observer cells. Both learners use the same teacher-forced response and nonce-label targets with matched update counts; neither is trained with a reinforcement-learning objective.

D.3 Matched and unmatched quantities

Table 6 summarizes which quantities are matched across the two model roles and which remain learner-specific.

D.4 Online adaptation schedule

For every model, seed, and hidden rule:

1. Initialize paired acting model and observer model learners from the same checkpoint and adapter initialization.

Table 6: Standard acting–observer comparison. Matching concerns public events and supervised learning quantities, not complete inputs or internal learner states.

Property	Acting model	Observer model
Trajectory source relative to learner	Own evolving policy	Matched acting model's policy
Adaptation objective	Supervised response and label prediction	Same supervised objective
Rule-specific initial state	Same	Same
Public executed action	Own selected action	Same realized action
Public response	Rule-generated response	Same realized response
Nonce label	Same block label	Same block label
Supervised response target	Observed response	Same observed response
Supervised label target	Block nonce label	Same nonce label
Optimizer/update count	One update per episode	Same count
Retained adapter	Own adapter and optimizer	Separate adapter and optimizer
Determines which action is executed	Yes	No

2. At each episode's start, reset the environment to the rule-specific initial response state and set the consecutive-action counter to zero.
3. Present each learner with the initial response and accumulated within-episode public action–response history available before execution.
4. Each learner independently scores both candidate actions and records the higher-scoring candidate as its preferred action.
5. Execute the acting model's preferred action and apply the rule-specific transition function to the current state. The observer model's preference does not affect execution.
6. Emit the updated response, X or Y, on the same turn.
7. Append the same realized action–response pair to both learners' records.
8. Repeat Steps 3–7 for five turns.
9. Reveal the nonce rule label after the complete episode.
10. Construct the response-prediction and *label-identification* examples.
11. Perform one clipped AdamW update per learner.
12. Retain the updated adapters for the next episode and repeat from Step 2.

Each hidden rule contributes 32 five-turn teaching episodes, or 160 interaction turns. Online adaptation uses rank-16 LoRA (Hu et al., 2022), $\alpha = 32$, and dropout .05. Rule order is shared within each matched model pair, and ordinary text context is cleared at rule-block boundaries. Only the acting model's updated policy determines which actions are executed in future teaching episodes; the observer model's updated policy does not. Thinking-specific private prefixes are defined in Appendix E.

D.5 Evaluation and scoring

Evaluation takes place after learning. The learned adapter is frozen and the teaching history is removed. Every query starts in a fresh context containing only that query's instructions and test record. No correctness feedback is given, and test queries do not update the model. There are three components: identifying a learned label from a new trace, applying a learned rule to a new partner, and predicting what would happen if an earlier action had been different.

Each of the six rules contributes three *label-identification*, five novel-partner, and five counterfactual-action queries, yielding 78 evaluation items in total.

***Label identification*: name the rule shown in a new trace.** The model sees an initial response and a new five-turn action–response trace generated by one of the six rules. It selects the previously learned nonce label for that rule from six counterbalanced candidates. The correct label is not included in the trace. The trace must distinguish the target rule from all five alternatives; matching an entity name or other irrelevant wording is insufficient.

For example, with initial response Y, both Rule 2 and Rule 6 produce $A \to Y,\ A \to X$. Those two turns do not identify which rule is present. A subsequent B distinguishes them:

$$\begin{aligned} \text{Rule 2:} &\quad A \to Y,\ A \to X,\ B \to X, \\ \text{Rule 6:} &\quad A \to Y,\ A \to X,\ B \to Y. \end{aligned}$$

Rule 2 stays at X after B, whereas Rule 6 returns to Y. Diagnostic traces include this kind of distinguishing action when needed. Rules 1 and 5 similarly require a B after the first A has changed the output to X.

```
Identify the label of the hidden response rule in this new
trace.
Initial response:  <X/Y>
Turn 1:  <ACTION> -> <RESPONSE>
...
Turn 5:  <ACTION> -> <RESPONSE>
Labels:  <SIX COUNTERBALANCED NONCE LABELS>
Answer with the exact label only.
```

Novel-partner transfer: apply the rule to a new partner and history. The model is told that a new partner follows a previously learned nonce-labeled rule. It receives the initial response, a short action–response history, and one new action, then predicts X or Y. Both the partner identity and the tested action–history configuration are held out from teaching. For example, if *pimel* names Rule 2:

```
A new system follows the rule called pimel.
Initial response:  Y.
Previous turn:  A -> Y.
Now A is executed.  What response follows?
Choices:  X | Y
Answer:
```

The correct answer is X: this is the second consecutive A, and it changes the response on the current turn. The model is given the nonce label, not the Rule 2 definition. Success requires applying the learned rule to the supplied test history.

Counterfactual-action transfer: change an earlier action and work forward. The model sees a history and is asked to replace one earlier action. Starting from the specified initial response, it must work through the changed action sequence and predict the response to a final query action. All remaining actions are kept fixed; their responses must be recomputed and are not constrained to match the original trace.

For Rule 2, consider initial response Y and original history $A \to Y,\ A \to X$. If the second action is changed to B, the revised history is $A \to Y,\ B \to Y$. The B clears the count, so a final query action A produces Y, not X:

```
The system follows the rule called pimel.  Initial response:
Y.
Original history:  A -> Y; A -> X.
Suppose the second action had been B instead of A.
After this changed history, A is executed.  What response
follows?
Choices:  X | Y
Answer:
```

Here *pimel* again denotes Rule 2. Treating the original X as fixed would give the wrong answer. These items test application of the transition rule to a changed history, rather than recall of an

observed action–response string. For longer histories, the learner must carry the change through every remaining action before answering.

What the five transfer items cover. We describe queried positions using four plain-language categories:

- **Before switch**: the response has not changed yet.
- **Switch point**: the current action causes the response to change.
- **After switch**: the response has already changed; the query tests what happens next.
- **Reset test**: another action tests whether the response changes back.

For Rule 2, the sequence $A \to Y,\ A \to X,\ B \to X$ illustrates Before switch, Switch point, and After switch. For Rule 6, the final $B \to Y$ is a Reset test. These labels describe the tested behavior; they are not additional prediction targets shown to the model.

For each rule, the five novel-partner items and five counterfactual-action items cover applicable positions plus another held-out action–history combination. The categories are not forced into a one-item-per-category allocation. Rules 1–4 include queries showing what happens after their one-way change. Rules 5–6 include B reset queries. Rules 2, 4, and 6 distinguish the first trigger action from the second. Rule 5 does not require remembering a partly completed action sequence. All evaluated configurations are held out from realized teaching records and supervised adaptation targets, as specified in Appendix A.

Table 7: Representative canonical test inventory for all six rules. A/Y means action A followed by response Y; "none" means no action has occurred since initialization. The final column is the response to the query action. These examples make the test construction explicit; they are not the complete 78-query inventory and are not additional items. Test-type labels and Rule numbers are explanatory only.

Rule	Test type	Initial	Previous history	Query	Correct
1	Before switch	Y	none	B	Y
1	Switch point	Y	none	A	X
1	After switch	Y	A/X	B	X
1	Held-out history	Y	B/Y, A/X, B/X	B	X
2	Before switch	Y	none	A	Y
2	Switch point	Y	A/Y	A	X
2	After switch	Y	A/Y, A/X	B	X
2	Held-out history	Y	A/Y, B/Y	A	Y
3	Before switch	X	none	A	X
3	Switch point	X	none	B	Y
3	After switch	X	B/Y	A	Y
3	Held-out history	X	A/X, B/Y, A/Y	A	Y
4	Before switch	X	none	B	X
4	Switch point	X	B/X	B	Y
4	After switch	X	B/X, B/Y	A	Y
4	Held-out history	X	B/X, A/X	B	X
5	Before switch	Y	none	B	Y
5	Switch point	Y	none	A	X
5	Reset test	Y	A/X	B	Y
5	Held-out history	Y	A/X, B/Y	A	X
6	Before switch	Y	none	A	Y
6	Switch point	Y	A/Y	A	X
6	Reset test	Y	A/Y, A/X	B	Y
6	Held-out history	Y	A/Y, A/X, B/Y	A	Y

Table 7 shows representative factual histories and next-response queries. Novel-partner items attach the learned nonce label and a new identifier to such a configuration. Counterfactual items instead specify an original history and one changed action; the answer is computed from the revised history, as in the Rule 2 example above. A single response need not identify a rule: Rules 2 and 6 agree at the second A and differ at the subsequent B. Rule identification requires a complete distinguishing trace, whereas transfer tests whether a specified learned rule is applied correctly.

Scoring. The model scores the complete candidate answers using length-normalized answer-token log probabilities and selects the higher-scoring response or label. Ties receive 0.5 credit. The programmed transition engine supplies the answer key. All six rules contribute equally: accuracy is averaged within rule and evaluation type, then macro-averaged across rules. Novel-partner and counterfactual-action transfer receive equal weight in the primary *structural-transfer* score. *Label identification* remains a separate secondary outcome.

Secondary distribution-sensitive scoring (models only). Accuracy remains the primary outcome. For the supplementary analysis, let $s_i(c)$ be the length-normalized answer-token log score of candidate c on item i, as used in the accuracy evaluation. We normalize over that item's candidate set $\mathcal{C}_i$:

$$q_i(c) = \frac{\exp s_i(c)}{\sum_{c' \in \mathcal{C}_i} \exp s_i(c')}, \qquad \ell_i = -\log q_i(y_i) = \mathrm{logsumexp}_{c \in \mathcal{C}_i} s_i(c) - s_i(y_i), \tag{3}$$

where y_i is the correct answer. The candidate set contains two responses for *structural-transfer* items and six nonce labels for *label-identification* items. We use natural logarithms, so NLL is measured in nats; lower values indicate more relative score assigned to the correct candidate. The normalization uses the existing candidate scores without an additional temperature or fitted calibration. It defines a distribution over the available answers, rather than the model's full vocabulary or unnormalized sequence probabilities. NLL is computed for each item before averaging; it is not the negative logarithm of an average probability.

Because the softmax is applied to model-specific candidate scores, absolute NLL values are not treated as calibrated quantities comparable across model families. The inferential target is the acting–observer contrast within each matched model configuration.

Within each model, role, and seed/block, we average item NLL within rule and evaluation type, then give the six rules equal weight. Novel-partner and counterfactual-action NLL receive equal weight in the *structural-transfer* composite, matching accuracy aggregation. *Label-identification* NLL is computed separately and is not combined with *structural-transfer* NLL. Table 8 reports the *structural-transfer* composite.

Acting–observer contrasts use the same matched seed/block units as the primary accuracy comparison. We compute acting model-minus-observer model mean NLL within each unit, then average these paired differences. Negative gaps favor acting models. Confidence intervals are obtained by resampling complete matched seed/block units jointly across roles, retaining their rules and evaluation items; individual items are not treated as independent learners. The NLL analysis is a secondary sensitivity check and does not replace accuracy in the human–model comparison.

E Native Thinking and Action–Experience Alignment

This appendix specifies the thinking and alignment experiments. Unless stated below, checkpoints, public records, supervised targets, update schedules, and evaluation follow Appendix D.

E.1 Checkpoints and training schedule

The experiments use Qwen3-0.6B, 1.7B, 8B, and 14B. Native thinking is switched within a checkpoint. Each condition uses 10 seeds. Each rule contributes 32 five-turn teaching episodes, with one update per episode. The alignment experiment crosses thinking OFF/ON with assigned match probability $m \in \{0, 0.5, 1\}$ and uses the same evaluation tasks and fixed test-stage mode as the role comparison. At full match ($m = 1$), the alignment condition coincides with the corresponding acting model condition and uses the same runs.

Table 8: Secondary distribution-sensitive evaluation under standard online supervised adaptation. Entries are candidate-normalized *structural-transfer* NLL, with the same rule and evaluation-type weights as accuracy. Gap is acting model minus observer model; negative values favor acting models. Absolute NLL values are interpreted only within model configuration rather than compared across model families. Confidence intervals are based on matched seed/block resampling.

Model	Params	Acting model NLL	Observer model NLL	Gap	95% CI
SmolLM2	0.36B	0.600	0.597	$+0.003$	$[-0.019, 0.023]$
Qwen3	0.6B	0.553	0.557	-0.004	$[-0.018, 0.011]$
Llama 3.2	1B	0.522	0.525	-0.003	$[-0.021, 0.015]$
Gemma 3	1B	0.493	0.490	$+0.003$	$[-0.013, 0.018]$
Qwen3	1.7B	0.463	0.469	-0.006	$[-0.019, 0.007]$
Llama 3.2	3B	0.450	0.461	-0.011	$[-0.023, -0.001]$
Gemma 3	4B	0.412	0.418	-0.006	$[-0.022, 0.009]$
Qwen3	8B	0.381	0.389	-0.008	$[-0.021, 0.004]$
Llama 3.1	8B	0.389	0.387	$+0.002$	$[-0.014, 0.017]$
Gemma 3	12B	0.340	0.346	-0.006	$[-0.019, 0.006]$
Qwen3	14B	0.315	0.324	-0.009	$[-0.019, -0.001]$

E.2 Pre-action computation and preferred-action extraction

On each teaching turn, the learner receives the rule-specific initial response and the current public interaction history h_t. It performs pre-action computation to form an action preference before the executed action and response are revealed. With thinking ON, it generates an explicit learner-specific reasoning prefix R_t^-; with thinking OFF, it directly scores the candidate actions without generating a reasoning trace. Candidate action strings are scored conditional on the current history and the reasoning prefix when present. In the expression below, R_t^- is empty under thinking OFF. The higher-scoring candidate is saved as the preferred action:

$$a^* = \arg\max_{a \in A} |a|^{-1} \log p_\theta(a \mid h_t, R_t^-). \tag{4}$$

This argmax is the behavioral output of pre-action computation and is saved before the match draw, action execution, or response reveal. In the acting–observer comparison, both roles independently perform pre-action computation and form their own preferred action from the current public history. With thinking ON, both generate their own learner-specific pre-action reasoning trace before action scoring. Only the acting model's preferred action determines the executed action; the observer model's preferred action is recorded but has no causal effect on the trajectory. The observer model receives the matched acting model-generated executed action and resulting response. In the alignment experiment, the match intervention separately determines the executed action. Preferred and executed actions are recorded separately.

Under thinking ON, either role in the acting–observer comparison and learners in each alignment condition may also generate pre-response reasoning after the executed action is revealed but before its response. An episode-level reasoning call may summarize the completed public record before the nonce label is revealed. These later traces are additional reasoning stages: they do not replace either role's pre-action reasoning or change the saved preferred action for that turn. Thinking OFF generates none of these explicit reasoning traces.

E.3 Action–experience alignment intervention

For every teaching turn, the procedure is:

1. compute and save a^* using Equation 4;
2. read the run's assigned $m \in \{0, 0.5, 1\}$;
3. draw a match indicator from a deterministic seed-specific RNG with probability m;
4. execute a^* when matched and the alternative binary action when mismatched;
5. generate the partner response from the executed action, state, history, and hidden rule;

6. reveal the executed action and response to the learner; and
7. continue the standard response and label adaptation pipeline after the relevant targets are revealed.

Randomization occurs independently on each teaching turn. Assigned m, rather than the realized match fraction, is the experimental predictor. Realized fractions are logged only as manipulation checks. The preferred action must be timestamped before the match draw and before the executed action is exposed to the learner.

The intervention jointly changes whether subsequent experience realizes the learner's current preference and how on-policy that experience is with respect to that preference. These changes define the action–experience alignment manipulation.

E.4 Teacher-forced updates with private prefixes

For assigned match m and thinking mode c, the adaptation loss is

$$\mathcal{L}_{\text{adapt}}^{m,c} = -\sum_{i} \sum_{j \in \mathcal{M}_i} \log p_{\theta_{m,c}}(y_{i,j} \mid x_i^{m,c}, y_{i,<j}) \,. \tag{5}$$

The prefix $x_i^{m,c}$ contains only information available before the target, and $\mathcal{M}_i$ masks the observed-response or label answer tokens. In thinking-ON conditions, generated reasoning tokens are fixed conditioning data, not supervised answers. At update time, the saved prefix is rerun under the unchanged adapter, allowing gradients on observed targets to depend on that learner's reasoning without a score-function gradient through reasoning generation.

A response example includes the executed action and, under thinking ON, eligible pre-response reasoning, never a post-response explanation. A label example includes the completed public episode and, under thinking ON, eligible episode reasoning, never reasoning generated after the label was first supplied. Thinking-OFF examples contain no generated reasoning prefix. One clipped AdamW update follows each episode. At evaluation, the adapter is loaded and frozen, the teaching history is removed, and each query begins in a fresh context.

E.5 Budgets, scoring, and failure policy

In thinking-ON conditions, reasoning generation uses temperature 0.6, top-p 0.95, top-k 20, and min-p 0 (Qwen Team, 2025). Each reasoning call has a 1,536-token output allowance; thinking-extended examples have an 8,192-token total input allowance. The allowances, target masks, number of revealed targets, and optimizer steps are matched across assigned match conditions. The primary transfer readout uses length-normalized candidate scoring with test thinking OFF.

Logs retain the candidate scores, preferred action, match RNG draw, assigned m, executed action, reasoning tokens when present, target masks, reveal timestamps, and checksums. Failures include unfinished thinking-ON reasoning, invalid action scores, missing match draws, input overflow, missing target masks, or response leakage. A seeded fallback keeps the binary action defined after an invalid action call and marks the complete seed block for sensitivity analysis. No condition receives outcome-based retries.

E.6 Control comparisons

Learning versus test thinking. The Qwen3-8B role comparison crosses learning thinking OFF/ON with test thinking OFF/ON while holding each learned adapter fixed within test-mode comparisons.

Reasoning-conditioned updates. The original role control compares full reasoning-conditioned updates with public-record-only updates at learning thinking ON and test thinking OFF. The alignment-specific control makes the same update-context comparison across assigned match conditions. Its direct target is the reduction in the Thinking $\times$ Match coefficient.

Trajectory source. At fixed observer model thinking OFF, replay from thinking-OFF acting models is compared with replay from thinking-ON acting models. This isolates trajectory source at a fixed observer model computation setting.

Table 9: Qwen3-8B learning versus test thinking. Accuracies are percentages; gaps are acting model minus observer model in percentage points.

Learn thinking	Test thinking	Acting model	Observer model	Gap
OFF	OFF	86.2	85.9	+0.3
OFF	ON	88.1	87.1	+1.0
ON	OFF	90.5	86.0	+4.5
ON	ON	91.5	88.5	+3.0

E.7 Statistical targets

Assigned match is coded continuously as 0, 0.5, or 1 in the primary analysis:

$$Y_{s,c,m} = \beta_0 + \beta_T c + \beta_M m + \beta_{TM}(c \times m) + u_s + \varepsilon, \tag{6}$$

where Y is *structural-transfer* accuracy, c is thinking OFF/ON, m is assigned match, and u_s represents the seed block. The primary coefficient β_{TM} is the increase in the 0-to-100% match slope under thinking ON relative to thinking OFF. The Qwen3-8B endpoint changes are reported in Section 5.2; the primary inferential target remains the ordered Thinking $\times$ Match coefficient.

Scale-specific β_{TM} estimates are primary for replication. A pooled Thinking $\times$ Match $\times$ Scale model is a heterogeneity analysis, not the central claim. Resampling preserves complete seed-level condition blocks, including thinking modes, assigned match levels, rules, and repeated evaluations. Shared endpoints are retained jointly within each resampled seed block. Turns, test items, and repeated generations are not independent learners. Realized match rate is excluded from the primary predictor.

E.8 Quantitative bridge from randomized alignment to the role gap

For each thinking-ON model, the observer model's natural preferred–executed match rate is computed from the saved role-comparison logs:

$$\widehat{m}_{\text{obs}} = \text{mean}\, \mathbf{1}\{a^*_{\text{obs}} = a_{\text{exec}}\}.$$

The acting model's preferred action determines execution, giving action-level match 1. Let s_{ON} be the model-specific 0-to-100% assigned-match slope from the randomized thinking-ON alignment intervention, and let Δ_{role} be the observed thinking-ON acting–observer difference. The bridge quantities are

$$\text{Predicted alignment contribution} = s_{\text{ON}}(1 - \widehat{m}_{\text{obs}}),$$
$$\text{Residual role difference} = \Delta_{\text{role}} - s_{\text{ON}}(1 - \widehat{m}_{\text{obs}}).$$

The thinking-ON slopes are 1.0, 2.1, 3.5, and 2.2 percentage points at 0.6B, 1.7B, 8B, and 14B, respectively. Seed/block-level resampling propagates uncertainty through the full calculation, including the slope, natural observer model match rate, and observed role gap, to obtain the residual 95% CI. The residual and its interval are the primary inferential quantities; individual turns are not treated as independent learners.

Natural observer model match is an endogenous descriptive quantity, whereas assigned match is randomized and remains the causal predictor in the alignment experiment. The bridge transports the randomized alignment slope to the observed action-level agreement difference and is used as a consistency analysis rather than a mediation estimate.

Table 2 reports the full bridge quantities.

E.9 Frozen in-context learning protocol

The frozen in-context comparison (Section 5.4, Table 3) evaluated Qwen3-1.7B and Qwen3-8B with the teaching history retained in context and no parameter updates. Checkpoints, hidden rules, public records, role assignment, and evaluation tasks follow Appendix D, except that no adapter is trained and the evaluation context contains the learner's retained teaching history rather than a fresh context.

Under thinking ON, each learner generated a bounded pre-action reasoning trace, with a maximum of 24 tokens per turn, before forming its preferred action (Equation 4). This trace and the preferred action were retained verbatim in that learner's own subsequent context, together with the executed action, the observed response, and the nonce rule label once revealed. Private reasoning was never shared across roles. Both learners received the same executed action and response, but only the acting model's preferred action controlled execution; the observer model's preferred action was recorded in its own context but had no causal effect on the public trajectory. Pre-response and episode-level reasoning were not retained in context. No summarization or compression was applied to retained traces.

Thinking-OFF runs retained the same public history without private reasoning. Because the two roles then share identical context and identical frozen weights, their evaluations coincide and the acting–observer difference is zero by construction; the thinking-OFF rows in Table 3 are reported as reference levels rather than as an inferential comparison.

E.10 Supplementary results

Table 10 reports structural-transfer accuracy at each assigned match level and thinking condition.

Table 10: *Structural-transfer* accuracy (%) at each assigned match level. The final column gives the change from 0% to 100% match in percentage points. The $m = 1$ endpoint corresponds to the acting model condition.

Model	Thinking	$m = 0$	$m = .5$	$m = 1$	$0 \rightarrow 100$
0.6B	OFF	73.8	74.3	74.1	+0.3
0.6B	ON	75.4	75.6	76.4	+1.0
1.7B	OFF	81.8	82.3	82.1	+0.3
1.7B	ON	82.9	84.2	85.0	+2.1
8B	OFF	85.9	86.5	86.2	+0.3
8B	ON	87.0	88.3	90.5	+3.5
14B	OFF	89.9	90.4	89.2	-0.7
14B	ON	90.5	92.0	92.7	+2.2

Table 11 reports absolute acting and observer accuracies, role gaps, and their confidence intervals for both thinking conditions, together with the direct gap changes shown in Figure 3A.

Table 11: Qwen3 role-comparison results with learning-stage thinking OFF and ON. Acting and observer columns report structural-transfer accuracy (%); gap is acting minus observer in percentage points. The lower panel reports the direct change $\Delta\text{gap} = \text{Gap}_{\text{ON}} - \text{Gap}_{\text{OFF}}$ and its 95% CI, as plotted in Figure 3A. Test thinking is OFF throughout.

Model	Thinking	Acting	Observer	Gap	95% CI of gap
0.6B	OFF	74.1	73.6	+0.5	$[-0.8, 1.8]$
0.6B	ON	76.4	75.5	+0.9	$[-0.5, 2.4]$
1.7B	OFF	82.1	81.8	+0.3	$[-0.9, 1.5]$
1.7B	ON	85.0	82.9	+2.1	$[0.4, 3.8]$
8B	OFF	86.2	85.9	+0.3	$[-0.9, 1.6]$
8B	ON	90.5	86.0	+4.5	$[2.3, 6.4]$
14B	OFF	89.2	88.6	+0.6	$[-0.6, 1.9]$
14B	ON	92.7	88.9	+3.8	$[1.7, 6.0]$

Model	Δgap (pp)	95% CI of Δgap
0.6B	+0.4	$[-1.8, 2.0]$
1.7B	+1.8	$[0.6, 3.4]$
8B	+4.2	$[2.6, 6.4]$
14B	+3.2	$[1.7, 4.5]$

Table 12 reports the endpoint effects and direct interaction contrasts summarizing Figure 3B, with 95% CIs for the interactions.

Table 12: Action–experience alignment across Qwen3 scales. Effects and interactions are percentage-point changes from 0% to 100% assigned match. Match is treated as an ordered predictor. Intervals are 95% CIs of the interaction.

Model	OFF $0 \rightarrow 100$	ON $0 \rightarrow 100$	Interaction	95% CI
0.6B	+0.3	+1.0	+0.7	[-0.7, 2.3]
1.7B	+0.3	+2.1	+1.8	[0.2, 3.2]
8B	+0.3	+3.5	+3.2	[1.8, 5.6]
14B	-0.7	+2.2	+2.9	[1.3, 4.4]

F Extended Related Work

Active, yoked, and interventional learning. A long line of work asks how learning changes when learners select or generate evidence rather than merely receive it (Castro et al., 2008); Coenen et al. (2019) review broader challenges in understanding human inquiry. In causal learning, interventions can support structure learning beyond passive observation (Steyvers et al., 2003; Lagnado & Sloman, 2004; Bramley et al., 2017), learners adaptively select which interventions to make (Coenen et al., 2015), and the process of deciding which intervention to make can itself affect what is learned (Sobel & Kushnir, 2006). Lampinen et al. (2023) showed that agents can learn generalizable causal experimentation strategies by passively imitating expert intervention trajectories, and that pretrained language models can exhibit related generalization from few-shot examples. Their setting asks whether passive training can teach a learner how to intervene later. Our question is complementary: we hold the realized intervention record fixed and ask whether that same record is learned differently when the learner's own choice generated it rather than when it was received as replay. Section 5.5 applies this matched-evidence contrast in a DAG-based causal-intervention environment while preserving the same acting–observer role manipulation. More directly related to our design, active–yoked studies, dating back to classic yoked-control designs (Held & Hein, 1963), compare learners who select evidence with learners who receive matched evidence selected by another person (Sobel & Kushnir, 2006; Markant & Gureckis, 2014). Markant & Gureckis (2014) found learning differences even when yoked learners viewed the same sequence of observations, and recent work has similarly found benefits of generating rather than observing matched predictions (Brod et al., 2025), consistent with the broader generation effect in memory (Slamecka & Graf, 1978). Related work on contingent social interaction also reports learning benefits for live interaction relative to recorded exposure (Kuhl et al., 2003; De Felice et al., 2021). **INTERGROUND** builds on this active–yoked tradition, but studies hidden transition-rule learning in both humans and adaptive language models and separates memory for rule labels from transfer of the learned structure.

On-policy, replayed, and self-generated experience. In sequential learning, a learner's actions can determine the distribution of states and observations from which it subsequently learns (Sutton & Barto, 2018; Ross et al., 2011). Here we use *on-policy* to describe the source of experience relative to the learner's current policy, not the learning objective: our language models are updated by supervised adaptation rather than reinforcement learning (cf. Ouyang et al., 2022; Shao et al., 2024). Offline reinforcement learning (Levine et al., 2020) and sequence modeling (Chen et al., 2021; Janner et al., 2021) can learn from previously collected trajectories, while agent tuning can focus adaptation on selected critical steps (Chen et al., 2025). Recent language-model work has explicitly trained on self-generated sequences. On-policy distillation trains a student on outputs sampled from its own policy to reduce the mismatch between training and student-generated sequences (Agarwal et al., 2024; Gu et al., 2024), and on-policy data has been argued to be preferable for preference fine-tuning (Tajwar et al., 2024). Student-guided instruction tuning similarly selects or refines examples for compatibility with the model being adapted (Li et al., 2024). Recent work has also introduced on-policy replay into ordinary supervised fine-tuning by generating replay data from the current model rather than changing the optimization objective (Chen et al., 2026). These approaches compare training distributions generated by different policies. **INTERGROUND** instead holds the realized public interaction record fixed across an acting learner and a yoked replay

learner, and separately manipulates whether executed actions align with the learner's prior action preference.

Reasoning, hypothesis testing, and learner-specific computation. Intermediate reasoning can alter language-model decisions and training dynamics (Nye et al., 2021; Wei et al., 2022; Kojima et al., 2022; Li et al., 2025), explanations in context can change what is learned from examples (Lampinen et al., 2022), and self-generated rationales can be used as training data for subsequent model improvement (Zelikman et al., 2022; 2024; DeepSeek-AI et al., 2025). In human active learning, evidence selection is likewise often understood as hypothesis-dependent: learners choose observations or interventions in light of their current beliefs (Schulz & Bonawitz, 2007; Gureckis & Markant, 2012; Markant & Gureckis, 2014). Recent work finds that in-context learning can support more flexible generalization than finetuning, and that augmenting finetuning data with in-context inferences can improve generalization (Lampinen et al., 2025). ReAct interleaves reasoning with actions, while Reflexion uses feedback-driven verbal reflection to support subsequent attempts (Yao et al., 2023; Shinn et al., 2023). Complementing this work, **INTERGROUND** compares acting and observing under matched public interaction records, and separately randomizes whether executed actions match the learner's pre-action preferences. **INTERGROUND** connects these ideas by asking whether computation performed *before* an action changes how the resulting experience is learned. Rather than treating reasoning only as a test-time inference procedure, we test whether learner-specific pre-action computation changes the effect of alignment between a learner's preferred action and the experience that follows.

G CAUSAL-INTERVENTION REPLICATION

The primary **INTERGROUND** environment uses six deterministic finite-state transition rules over two actions and two responses. We use a second, structurally distinct environment to test whether the thinking-dependent acting–observer difference extends to intervention learning over a multivariable causal system. The replication preserves the central role manipulation—whether the learner's own preference determines the executed experiment—while changing the latent structure, action space, and form of environmental feedback. We evaluate Qwen3-8B with five matched seeds under acting model and observer model roles with learning-stage thinking OFF or ON.

G.1 ENVIRONMENT AND CAUSAL MECHANISMS

Each hidden causal system contains five binary variables

$$X_1, \ldots, X_5 \in \{-1, +1\}$$

connected by a directed acyclic graph (DAG). We use graphs with five directed edges, exactly two root variables, maximum in-degree two, no isolated variables, and at least one directed path of length two or greater. These constraints produce systems containing both local and multistep causal effects while keeping the complete state small enough to present directly to the model.

For each system, we first sample a topological ordering of the five variables and then sample five directed edges consistent with that ordering. Samples that violate the root-count, in-degree, connectivity, or path-length constraints are rejected. For every non-root variable X_j, the structural equation is

$$X_j = \operatorname{sign}\left(b_j + \sum_{i \in \mathrm{Pa}(j)} w_{ij} X_i\right),$$

where

$$w_{ij} \in \{-1, +1\}, \qquad b_j \in \{-0.5, +0.5\}.$$

The nonzero bias prevents ties, so the argument of $\operatorname{sign}(\cdot)$ is never zero. Root values are externally specified by the root configuration, and all remaining variables are evaluated in topological order.

After generating a valid mechanism, variable identities are randomly permuted so that their surface names do not reveal topological position. The graph, edge signs, biases, root identities, and topological order are never shown to the learner.

Each matched seed contains six independently generated causal systems. As in the primary environment, the six systems receive arbitrary nonce labels from the label pool defined in Appendix D. The causal systems, variable mappings, and nonce-label assignments are fixed within a seed and shared across acting model and observer model roles and across thinking conditions.

G.2 Root Configurations and Support Split

Each causal system has exactly two roots and therefore four possible assignments of root values in $\{-1, +1\}^2$. For each system and seed, a seeded permutation assigns three root configurations to teaching and reserves the fourth root configuration for evaluation. This split is created before model interaction and is identical for acting model and observer model roles and for thinking OFF and ON.

For any root configuration, the corresponding baseline state is obtained by assigning the root values and evaluating every non-root structural equation in topological order. The three teaching configurations are used only during learning. The held-out fourth root configuration is never used to construct the baseline state of a teaching episode or adaptation example. Its full state may nevertheless appear as an intervention outcome during teaching. This split therefore holds out the root configuration used to construct the evaluation baseline state, not necessarily every occurrence of that state or every equivalent consequence.

This support split requires evaluation to transfer the learned causal mechanism to a baseline state derived from a root configuration not used for teaching while keeping the underlying mechanism fixed.

G.3 Intervention semantics

Each teaching episode is associated with one baseline state and contains five intervention turns. On a turn, the learner selects one of the five variables X_k as the intervention target. Let $x_k^{(0)}$ denote that variable's value in the episode baseline. The executed intervention is

$$\mathrm{do}(X_k = -x_k^{(0)}),$$

so the selected variable is forced to the value opposite its baseline value.

The intervention replaces the structural equation for X_k. Its descendants are then recomputed in topological order using the intervened value, while variables that are neither the intervention target nor descendants of it retain their baseline values. The environment reveals the complete resulting five-variable state.

Interventions do not persist across turns. Before every new intervention, the physical system is reset to the same episode baseline and the new intervention is applied independently. Thus, the five turns constitute separate experiments on a common hidden causal system rather than successive modifications of a persistent state.

The public interaction history nevertheless accumulates across turns. Before choosing the next intervention, a learner can therefore condition on the baseline state and on all previously executed interventions and their complete resulting states.

G.4 Matched acting–observer roles

The role manipulation follows the same principle as in the primary **INTERGROUND** experiment. Acting model and observer model learners begin from the same Qwen3-8B checkpoint and matched adapter initialization and are assigned the same six causal systems, baseline schedule, nonce-label mapping, and evaluation inventory.

On every teaching turn, both learners receive the same public information available before execution: the current baseline state and the accumulated history of previously executed interventions and resulting states. Each learner then independently forms a preferred intervention over the five variables.

Only the acting model's preferred intervention determines which intervention is executed. The observer model's preferred intervention is recorded but has no causal effect on the environment. After

execution, acting model and observer model both receive the same executed intervention and the same complete post-intervention state.

Thus, within each matched pair, the two roles receive the same realized public intervention record and the same supervised learning targets. They differ in whether their own pre-action preference determines the experiment whose result they subsequently observe. As in the primary experiments, matching concerns the public record and supervised quantities, not complete private computation, learner state, adapters, or optimizer state. Acting model and observer model retain separate adapters and optimizer states throughout online adaptation.

G.5 Pre-action computation and thinking

We use the same learning-stage thinking manipulation as in Section 5 and Appendix E. On each teaching turn, the learner forms its intervention preference before the executed intervention and post-intervention state are revealed.

Under thinking OFF, the learner directly scores the five complete candidate intervention strings using its current model state and records the highest-scoring candidate as its preferred intervention. Under thinking ON, the learner first generates its own learner-specific pre-action reasoning trace and then scores the same five candidate interventions conditional on that trace.

Using the notation of Appendix E, the preferred intervention is

$$a_t^* = \arg\max_{a \in \mathcal{A}_{\mathrm{DAG}}} |a|^{-1} \log p_\theta(a \mid h_t, R_t^-),$$

where

$$\mathcal{A}_{\mathrm{DAG}} = \{\mathrm{do}(X_1), \ldots, \mathrm{do}(X_5)\}$$

and R_t^- is empty under thinking OFF.

Under thinking ON, acting model and observer model generate separate private reasoning traces. The acting model's resulting preference controls execution; the observer model's does not. Preferred interventions are saved before execution and before the post-intervention state is exposed.

Reasoning generation, candidate scoring, update-prefix handling, failure policy, and generation settings otherwise follow Appendix E. Test-stage thinking is fixed OFF, matching the primary transfer readout.

G.6 Teaching schedule

Each seed contains six hidden causal systems. Each system contributes 12 five-intervention teaching episodes, for 60 intervention turns per system and 360 intervention turns per seed.

The three teaching root configurations each occur in four episodes. Their order is determined by a seed-specific schedule that is fixed before interaction and shared across matched acting–observer conditions. Each episode resets to its scheduled baseline before the first intervention and returns to that same baseline before every subsequent intervention.

At the end of each five-intervention episode, the nonce label associated with the causal system is revealed. Ordinary text context is cleared according to the same block structure used in the primary online-adaptation experiments; learned adapter and optimizer state are retained across episodes.

G.7 Supervised online adaptation

After each executed intervention, the complete post-intervention state is part of the public record for both learners. For supervised consequence prediction, one non-intervened variable is selected by a deterministic seed-specific schedule. The adaptation example contains the baseline state, the public intervention history available before the current consequence, the current executed intervention, and the identity of the queried variable. The queried variable's post-intervention value is hidden and used as the binary answer target.

A consequence-prediction example has the form:

> Baseline state:
> A = <VALUE>, B = <VALUE>, C = <VALUE>, D = <VALUE>, E = <VALUE>.
> Previous experiments: <HISTORY>.
> Executed intervention: <INTERVENTION>.
> Predict the post-intervention value of <QUERY-VARIABLE>.
> Answer -1 or $+1$ only.
> Answer: <OBSERVED-VALUE>

The queried variable is never the intervened variable. Acting model and observer model receive the same queried variable and the same observed target.

After the fifth intervention, a *label-identification* example is constructed from the completed episode and the six nonce-label candidates, following the same teacher-forced labeling procedure as Appendix D. Cross-entropy is masked to the answer tokens; generated reasoning tokens in thinking-ON conditions are conditioning context rather than supervised targets.

All consequence-prediction and *label-identification* examples from an episode are used in one clipped AdamW update. Online adaptation otherwise follows the primary Qwen3 protocol: rank-16 LoRA with $\alpha = 32$ and dropout 0.05. Acting model and observer model receive the same supervised targets and the same number of updates.

G.8 Intervention-selection prompt

Before each teaching intervention, the learner receives the current public information and is asked to select the experiment that would help it learn the hidden causal system. The model-facing template is:

> You are studying one persistent system with an unknown causal mechanism.
> Baseline state:
> A = <VALUE>, B = <VALUE>, C = <VALUE>, D = <VALUE>, E = <VALUE>.
> Previous experiments:
> <RENDERED INTERVENTION-OUTCOME HISTORY OR NONE>.
> Choose one variable to intervene on to help learn the hidden causal mechanism.
> The selected variable will be set to the value opposite its baseline value.
> Answer A, B, C, D, or E only.

Candidate strings corresponding to the five variable choices are scored using the same length-normalized procedure used for action selection in the primary environment. The learner is not shown the hidden graph, edge signs, structural equations, or evaluation queries.

G.9 Held-out intervention transfer

The primary outcome is held-out intervention-transfer accuracy. Evaluation is performed after learning with the adapter frozen, the teaching history removed, and test-stage thinking OFF. No correctness feedback is provided and evaluation does not update the model.

Each canonical test query identifies a previously learned causal system by its nonce label, provides the baseline state associated with the held-out root configuration, specifies an intervention, and asks for the resulting value of one non-intervened variable:

> System label: <NONCE>.
> Baseline state:
> A = <VALUE>, B = <VALUE>, C = <VALUE>, D = <VALUE>, E = <VALUE>.
> Intervene on <VARIABLE> and set it to the value opposite its baseline value.
> After this intervention, what is the value of <QUERY-VARIABLE>?
> Choices: $-1 \mid +1$
> Answer:

The queried variable is never the intervened variable. The programmed causal engine supplies the answer by applying the specified intervention to the baseline and recomputing all descendants in topological order.

To reduce dependence on any single serialization, each canonical query is evaluated under 10 independently generated, semantically equivalent surface realizations. These realizations preserve the causal system, baseline, intervention target, queried variable, and correct answer while independently varying only nuisance aspects of presentation such as variable ordering in the rendered state and answer-choice ordering. The same realization set is used across acting model and observer model roles and across thinking OFF and ON.

G.10 Evaluation inventory and support

Each of the six causal systems contributes eight canonical held-out intervention-transfer queries, for 48 canonical queries per seed. Each canonical query is evaluated under 10 semantically equivalent surface realizations, yielding 480 scored evaluations per role–thinking condition and seed.

All eight canonical queries for a system use its held-out root configuration. Four are chosen so that the intervention changes the queried variable relative to its baseline value, and four are chosen so that the queried variable remains unchanged. Correct answer values are balanced as closely as possible between -1 and $+1$. At least two canonical queries per system require the intervention effect to propagate along a directed path of length two or greater.

This balancing prevents high accuracy from being obtained by always predicting that the queried variable remains at its baseline value. At the canonical-query level, exactly half of the queries change the queried variable and half do not.

The complete set of canonical queries and all associated surface realizations are generated before any model run and are shared across acting model and observer model roles and across thinking OFF and ON. No evaluation query or realization is added, removed, or selected according to a condition's realized acting model trajectory, preferred interventions, or model accuracy.

G.11 Identifiability of held-out queries

We require each canonical held-out query to have a determined answer under the predefined causal-mechanism class given the designed teaching support.

For a target causal system, we enumerate mechanisms satisfying the graph and structural-equation constraints in Appendix G.1 that agree with the target system on the complete designed teaching support: all five possible interventions under each of the three teaching root configurations. A candidate canonical query is retained only if every mechanism consistent with this designed support predicts the same value for the queried variable.

This identifiability screen is performed when the environment and canonical evaluation inventory are generated, before any model run. It therefore does not depend on which interventions an acting model actually chooses or differ across experimental conditions. Surface realizations inherit the answer of their underlying canonical query and do not alter the causal content of the item. Realized teaching trajectories need not themselves exhaust the designed teaching support; variation in which experiments an acting learner selects remains part of the interactive learning problem. This screen establishes identifiability from the complete designed support, not necessarily from each realized trajectory.

G.12 Scoring

For each surface realization, the model scores the complete candidate answers -1 and $+1$ using length-normalized answer-token log probability and selects the higher-scoring candidate. A correct selection receives 1, an incorrect selection receives 0, and an exact score tie receives 0.5.

Scores are first averaged across the 10 surface realizations of each canonical query. The resulting canonical-query scores are then averaged across the eight queries within each causal system and macro-averaged across the six systems. Thus, every causal system and every canonical query within a system receive equal weight.

Table 13: Seed-level held-out intervention-transfer accuracy (%) in the causal-intervention replication. Each entry averages 48 canonical queries, each evaluated under 10 surface realizations. Values are rounded to one decimal place; inferential contrasts are computed from unrounded scores.

Seed index	OFF P	OFF O	ON P	ON O
1	69.9	70.5	74.6	73.4
2	72.4	71.2	76.7	74.3
3	70.6	70.8	76.4	73.6
4	72.6	71.6	77.9	74.2
5	71.5	70.9	78.4	74.0
Mean	71.4	71.0	76.8	73.9

Equivalently, because every canonical query has the same number of realizations, each seed-level role–thinking accuracy summarizes 480 scored evaluations.

All acting–observer differences, thinking interactions, and confidence intervals are computed from the unrounded scores prior to presentation rounding. Displayed one-decimal accuracies are not subtracted to construct inferential quantities.

Held-out intervention-transfer accuracy is the sole primary transfer outcome in this replication; we do not introduce separate novel-intervention and counterfactual-intervention composites.

G.13 Statistical analysis

The replication crosses role $r \in \{P, O\}$ with learning-stage thinking $c \in \{\mathrm{OFF}, \mathrm{ON}\}$ for Qwen3-8B. The four conditions use the same five matched seed blocks, causal systems, canonical evaluation queries, and surface realizations.

For each seed and condition, held-out intervention-transfer accuracy is computed from the unrounded realization-level scores using the aggregation procedure in Appendix G.12. For each thinking mode, the acting–observer difference is

$$\Delta^{c}_{\mathrm{role}} = \bar{Y}_{P,c} - \bar{Y}_{O,c}.$$

The primary inferential target is the change in the acting–observer difference when thinking is enabled:

$$\Delta_{\mathrm{gap}} = \Delta^{\mathrm{ON}}_{\mathrm{role}} - \Delta^{\mathrm{OFF}}_{\mathrm{role}} = (\bar{Y}_{P,\mathrm{ON}} - \bar{Y}_{O,\mathrm{ON}}) - (\bar{Y}_{P,\mathrm{OFF}} - \bar{Y}_{O,\mathrm{OFF}}).$$

Inference treats the complete matched seed block as the independent resampling unit. Each resampled block retains all four role–thinking cells, all six causal systems, all 48 canonical queries, and all 10 realizations of each query. Individual interventions, canonical queries, and surface realizations are not treated as independent learners.

Because the replication contains five matched seed blocks, percentile confidence intervals are computed over the complete set of $5^5 = 3125$ bootstrap resamples obtained by drawing five seed blocks with replacement. Each bootstrap replicate recomputes the thinking-OFF gap, the thinking-ON gap, and their difference from the underlying unrounded seed-level scores. Numerical values are rounded only for presentation, using round-to-nearest with ties to even. For example, 71.25% is displayed as 71.2%. With only five independent seed blocks, the intervals should be interpreted with caution; enumerating all bootstrap resamples eliminates Monte Carlo error but does not increase the number of independent observations.

G.14 Results

Table 13 reports the seed-level accuracies. With thinking OFF, acting model and observer model transfer accuracy was 71.4% and 71.0%, respectively, yielding an acting–observer difference of $+0.4$ percentage points (95% CI $[-0.2, 1.0]$). With thinking ON, transfer accuracy was 76.8% for acting models and 73.9% for observer models, yielding an acting–observer difference of $+2.9$ points (95% CI $[1.9, 3.9]$).

The direct thinking-related increase in the acting–observer difference was $+2.5$ percentage points (95% CI $[1.7, 3.3]$). All reported contrasts are computed from unrounded seed-level accuracies before presentation rounding. The evaluation counts contain 11 exact score ties across 9,600 scored evaluations; each tie receives half credit.

For comparison, the corresponding Qwen3-8B increase in the primary **INTERGROUND** environment was $+4.2$ points: the acting–observer difference increased from $+0.3$ points with thinking OFF to $+4.5$ points with thinking ON. The causal-intervention replication therefore reproduces the direction, with a smaller point estimate, of the thinking-dependent acting–observer difference under a distinct latent structure, action space, and feedback representation.