Extended Abstract

Motivation Reinforcement learning (RL) fine-tuning reliably improves LLM agent performance within a training environment, but the gains often fail to transfer when the agent is deployed elsewhere. Xi et al. (2026) study this failure systematically across five environments and find that BabyAI-trained agents show uniquely negative cross-environment transfer. They attribute this to BabyAI's interface design: a per-step list of valid actions provided to the agent at every turn, which they hypothesise induces a scaffold dependency that limits generalisation. This attribution is plausible but untested. We test it directly.

Method We hold model, algorithm, training data, and all hyperparameters fixed and vary only the per-step action list during GRPO fine-tuning on BabyAI. Two training conditions are compared: `per_step`, which provides the full valid-action list at every turn (replicating Xi et al.), and `examples`, which supplies only static format examples in the system prompt. We evaluate both conditions on BabyAI (held-in) and ALFWorld (held-out), and conduct two interpretability probes: one withholding the action list at BabyAI test time, and one injecting a BabyAI-style list into ALFWorld at test time (Condition A).

Implementation All experiments use Qwen2.5-3B-Instruct fine-tuned with GRPO via VERL v0.7.1 on 8×H100 GPUs for 300 training steps. BabyAI is accessed through AgentGym with the same 810/90 train/test split as Xi et al. (2026). Evaluation uses fixed random samples of 50/60 BabyAI and 80/134 ALFWorld games (seed 42). Condition A is replicated across four independent game subsets (seeds 42–44).

Results Both training conditions achieve comparable held-in gains on BabyAI ($+0.048$ and $+0.050$ over base), yet held-out performance on ALFWorld remains unchanged at 0.013 for both — identical to the untrained base model. The generalization gap $g = \Delta\text{Held-In} - \Delta\text{Held-Out}$ is 0.048 and 0.050 respectively, a difference consistent with noise rather than a directional effect. The interpretability probe shows that neither trained model collapses when the list is removed at test time (0.273 and 0.281, versus the base model's 0.188 in the same no-list condition). Condition A yields 0.013 across all replications — providing the familiar scaffold in ALFWorld does not help.

Discussion The three experiments converge on a negative result: the per-step action list is neither the source of the in-domain gain nor the cause of the transfer failure. Qualitative analysis of ALFWorld trajectories reveals that the trained model applies a reactive, shortcut-driven heuristic learned from BabyAI — identifying and acting on the nearest plausible object — that is structurally incompatible with ALFWorld's multi-step, state-transformation tasks. The failure is representational: single-environment GRPO fine-tuning specialises the policy to BabyAI's task structure in a way that is robust to surface-level prompt perturbations but does not transfer to environments with different sub-goal structure and object semantics.

Conclusion Xi et al.'s scaffold-dependency hypothesis does not hold under controlled experimental conditions. Interface redesign alone is unlikely to close the cross-environment transfer gap for RL fine-tuned agents. The bottleneck is the training distribution, not the training interface, pointing to multi-environment co-training and task-structure diversification as more promising directions for building broadly deployable agents.

How Does Scaffolding Affect Cross-Environment Generalization in LLM RL Fine-Tuning?

Hana Liu
Stanford University
hmyliu@stanford.edu

Alfred Sjöqvist
Stanford University
alfred.sjoqvist@stanford.edu

Abstract

Reinforcement learning fine-tuning consistently improves LLM agent performance within a training environment, but gains frequently fail to transfer when the agent is deployed elsewhere. Xi et al. (2026) attribute the unusually poor cross-environment transfer of BabyAI-trained agents to a specific interface feature: a per-step list of valid actions provided at every turn, which they hypothesise induces a scaffold dependency that limits generalisation. We test this hypothesis directly with a controlled ablation, holding model, algorithm, and training data fixed while varying only action-list availability during GRPO fine-tuning. Two conditions — `per_step`, which replicates Xi et al.'s interface, and `examples`, which provides only static format examples — achieve nearly identical held-in gains on BabyAI ($+0.048$ and $+0.050$) and identical held-out performance on ALFWorld (0.013), equal to the untrained base model. Two further probes rule out scaffold dependency and prompt-format mismatch as causes: trained models retain BabyAI performance when the list is withheld at test time, and injecting the list into ALFWorld prompts yields no improvement. Qualitative trajectory analysis identifies the actual failure mode: GRPO fine-tuning specialises the policy to BabyAI's simple, reactive navigating task in a way that is incompatible with ALFWorld's state-tracking and long-horizon planning demands. The transfer failure is not caused by the interface, pointing to training distribution diversity rather than interface redesign as the right mitigation.

1 Introduction

Reinforcement learning fine-tuning has become a standard recipe for turning a pretrained large language model (LLM) into an agent capable of multi-turn, interactive behaviour. While RL fine-tuning consistently improves in-domain success, the recurring concern is whether the gains so obtained are *portable*: an agent that excels in its training environment often collapses when deployed in another, even a closely related one.

This portability question has far-reaching impact. RL fine-tuned agents are being actively deployed as code assistants, browser agents, terminal helpers, and computer-use systems, yet cross-environment generalization failure is an acknowledged open problem. Each deployment interface differs in observation format, action vocabulary, and task structure. We frame cross-environment differences along two axes: *formal* differences in how observations and actions are represented at the interface level, and *substantial* differences in what the environment intrinsically demands of the agent (navigation, tool use, long-horizon planning). The two are typically confounded, because environments that differ in what they ask of the agent also differ in how they present that ask. Disentangling them and understanding whether interface design drives transferability failures is, therefore, a precondition for deploying such agents beyond their training environment.

In this work we test one specific instance of that question. Xi et al. (2026) attribute the cross-environment transfer failure of agents fine-tuned in the environment of BabyAI to a single feature of its interface: a per-step list of valid actions shown to the agent at every turn. They argue that this scaffold induces overfitting and over-reliance that limit generalization. This attribution makes an important suggestion about the impact of interface design on RL fine-tuned agents' generalization, but it remains qualitative and untested. We test it directly. Specifically, we ask: **does the BabyAI interface design (providing the LLM a per-step valid action list) during RL fine-tuning cause cross-environment transfer failure, and can it be mitigated by reducing scaffolding?**

Baseline. We evaluate the unmodified Qwen2.5-3B-Instruct model on both the training and transfer environments before any RL fine-tuning to establish reference scores. The base model achieves a success rate of 0.273 on BabyAI and 0.013 on ALFWorld.

Extension – Interface Scaffolding Design. We treat action list availability during GRPO training as the sole independent variable, and compare two conditions: `per_step`, which provides the full valid action list at every turn, and `examples`, which supplies only static format examples in the system prompt. Both conditions achieve comparable held-in gains on BabyAI ($+0.048$ and $+0.050$ over base, respectively), yet their held-out performance on ALFWorld remains identical to the untrained base model (0.013). Removing the per-step action list during training does not help generalization.

Extension – Mechanistic Interpretability. We probe *why* the transfer failure persists with two targeted evaluations. First, we evaluate both trained models on BabyAI with the action list withheld at test time: both maintain strong performance (0.273 and 0.281, versus 0.188 for the untrained base model), showing that neither policy has formed a hard dependency on the list. Second, we inject a BabyAI-style action list into the ALFWorld prompt at each turn (Condition A): held-out performance remains at 0.013, ruling out surface-level format mismatch as the cause of failure. Together, these results suggest the trained policy encodes a BabyAI-specific task strategy that does not generalize, regardless of how either environment is prompted.

Contributions.

1. Controlled ablation of BabyAI's per-step action list during GRPO fine-tuning, demonstrating the model effectively overcomes the absence of valid action list during training in BabyAI.
2. An interpretability experiment showing that models trained with and without per-step action list retain in-domain performance when the action list is removed from BabyAI at test time, directly contradicting the scaffold-dependency hypothesis of Xi et al. (2026).
3. An Adapted-ALFWorld diagnostic experiment that rules out prompt-format mismatch as the cause of transfer failure.
4. By elimination, identification of *weight-level environment specialisation* as the residual failure mode, and a discussion of what this implies for future work on RL fine-tuned agents.

Section 2 reviews prior work; Section 3 and 4 describes the three experiments; Section 5 presents results; Section 6 discusses implications; Sections 7 and 8 cover limitations and future work.

2 Related Work

Our work intersects three lines of prior research: the empirical literature on cross-environment generalisation in RL fine-tuned LLM agents, the environments and training infrastructure we rely on, and the RL algorithm we hold fixed across all conditions.

Cross-environment transfer in RL fine-tuned LLM agents. The closest prior work is Xi et al. (2026) [1], which characterises the generalisation profile of reinforcement-fine-tuned LLM agents across five environments (BabyAI, WebShop, ALFWorld, TextCraft, SearchQA). Their systematic study reveals a striking failure pattern: agents fine-tuned on BabyAI transfer worse than agents fine-tuned on any other source environment they consider. In one of their reported cases, a 7B agent's WebShop success rate drops from 28.6 prior to BabyAI training to 10.3 afterward, a 64% relative drop. The descriptive finding is solidly supported across model scales and metrics. Their *mechanistic* attribution, however, is qualitative: they hypothesise that the failure arises because BabyAI shows the agent a list of valid actions at every turn, and that "the policy gradually becomes dependent on this information." This explanation appears as a single paragraph in their paper and is not tested in a controlled experiment. Our project closes that gap: we hold model, task, optimiser and rollout budget fixed and vary only the per-step action list during fine-tuning.

Environments. We use BabyAI [6] as the training environment. BabyAI is a curriculum of grounded language tasks of increasing complexity, originally introduced to study sample efficiency in grounded language learning. Following Xi et al. (2026), we train on a mixture of three difficulty levels (GoToRedBall, Pickup and Open) so that our setup is directly comparable to theirs. As our held-out environment we use ALFWorld [3], which unifies a text-game interface over embodied household tasks. ALFWorld is deliberately chosen as the transfer target because its action-set abstraction is structurally similar to BabyAI's: if interface mismatch were the dominant bottleneck for transfer, ALFWorld should be a permissive target, not an adversarial one. A negative transfer result here is therefore informative.

Training infrastructure. Distributed GRPO training with multi-turn rollouts is orchestrated by VERL [5], a flexible RLHF framework that supports the hybrid actor–rollout setup we use. The environment side is wrapped through AgentGym [2], which provides standardised multi-turn interfaces for RL training of LLM agents and is the same framework used by Xi et al. (2026), ensuring our setup remains directly comparable to theirs.

Reinforcement Learning with GRPO We fine-tune the base model using Group Relative Policy Optimization (GRPO; Shao et al., 2024), a critic-free variant of PPO that eliminates the need for a separate value network. For each training prompt q, GRPO samples a group of G trajectories $\{o_1, \ldots, o_G\}$ from the current policy and computes advantages by normalizing rewards within the group:

$$A_i = \frac{r_i - \text{mean}(\{r_j\}_{j=1}^{G})}{\text{std}(\{r_j\}_{j=1}^{G})} \tag{1}$$

This group-relative formulation provides a low-variance baseline without requiring critic training, making it well-suited to the multi-turn agentic setting where value estimation is expensive. A KL penalty between the current and reference policy is added directly to the loss to prevent policy collapse during training.We hold GRPO fixed across all conditions so that any divergence in learning dynamics or transfer outcome is attributable to the interface variable rather than to the optimiser.

The gap. To our knowledge, no prior work isolates the interface-format variable while holding model, task and algorithm fixed. Xi et al. (2026)'s attribution of BabyAI's transfer drop to per-step action-list scaffolding therefore remains an open conjecture rather than a tested claim. Our experimental design addresses this directly, and the result, as we show in Sections 5 and 6, contradicts the conjecture.

3 Methods

3.1 Interface Conditions

We compare two training interface conditions (see Figure 1) that vary only in how action information is presented to the agent at each interaction turn, holding the model, algorithm, data, and all other hyperparameters constant.

`per_step`: At every turn, the agent receives an enumerated list of all valid actions for the current state (e.g., `"turn left"`, `"go to red ball 1"`). This replicates the interface used by Xi et al. (2026) and represents the maximally scaffolded condition.

`examples`: The system prompt contains a static block of generic action format examples, but no per-instance list is provided at any turn. The agent must infer admissible actions from the observation alone. This is the minimally scaffolded condition.

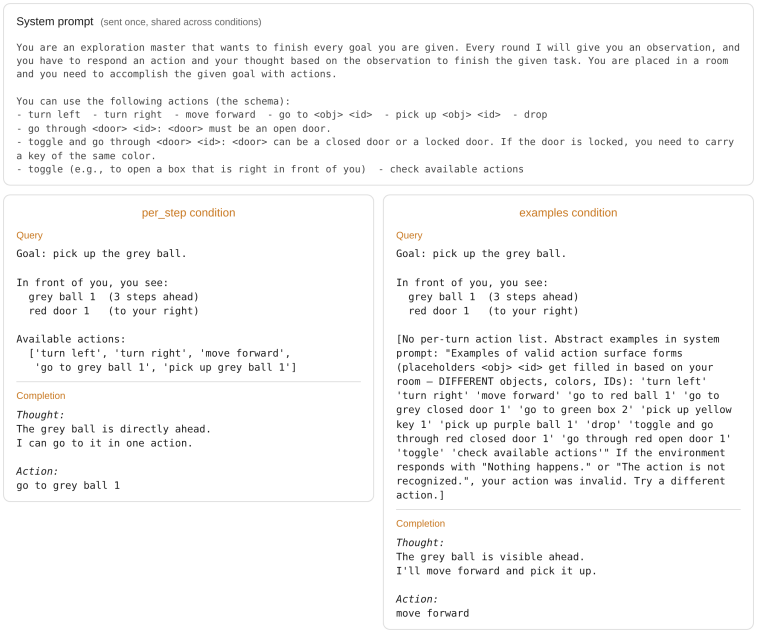

Figure 1: BabyAI interactions under `per_step` and `examples` conditions

3.2 Preliminary Experiment

Before committing to the full training budget, we conducted a preliminary experiment to verify that all interface conditions produce a learning signal and to identify any conditions that could be collapsed for the main study. We compared three conditions — above stated `per_step`, `examples`, and `shuffled` (identical to `per_step` but with the action list randomly permuted each turn) — across 3 random seeds for 50 training steps each, tracking training reward, policy entropy, KL loss, gradient norm, mean number of turns, and response length.

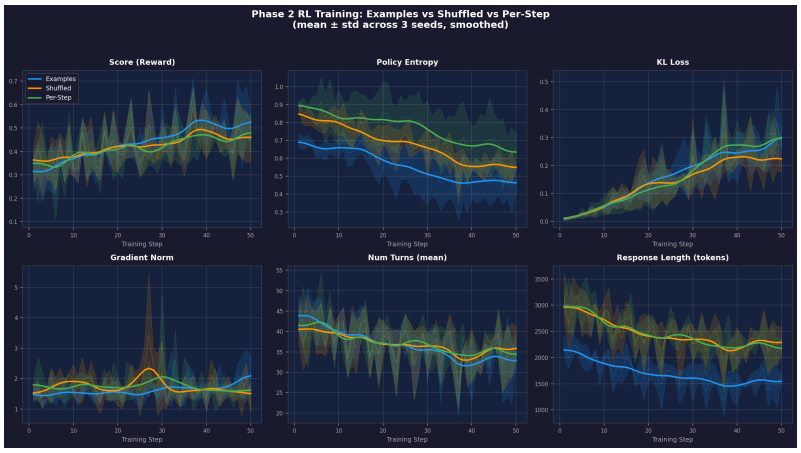

Figure 2: Examples vs Shuffled vs Per-Step (mean + std across 3 seeds, smoothed (Graphics generated by Claude, data from experiment)

All three conditions learned successfully, with reward rising from approximately 0.30–0.35 to 0.45–0.50 by the end of training. `examples` achieved the highest final reward (0.50) and the largest gradient norm and KL loss, indicating more aggressive policy updates relative to the reference model. `per_step` exhibited the highest entropy (0.655 vs. 0.477 for `examples` and 0.546 for `shuffled`), suggesting it maintains a more exploratory policy under the given training budget. `shuffled` produced training dynamics nearly indistinguishable from `per_step`, indicating that list order is not a meaningful variable; we therefore dropped it from subsequent experiments. These results confirmed that both `per_step` and `examples` are viable training conditions and motivated the full 300-step comparison reported in Section 5.

3.3 Generalization Evaluation

To measure the effect of interface design on cross-environment transfer, we evaluate each trained checkpoint on both the held-in and held-out environment. Held-in performance (ΔHeld-In) is the improvement in BabyAI success rate over the base model; held-out performance (ΔHeld-Out) is the corresponding improvement on ALFWorld. We define the generalization gap as:

$$g = \Delta\text{Held-In} - \Delta\text{Held-Out} \tag{2}$$

A positive g indicates that in-domain gains are not matched by held-out gains. Our central hypothesis, following Xi et al. (2026), is that interface scaffolding during training drives g upward — and that reducing scaffolding should narrow it by producing a less environment-specific policy.

3.4 Interpretability Probes

To characterize the mechanism behind our main results, we conduct two targeted post-hoc evaluations on the trained checkpoints.

Interpretability Experiment – List Dependency. We evaluate both trained models on the BabyAI held-in test set with the per-step action list withheld at inference time. If `per_step` training induces a hard reliance on the list as a lookup mechanism, performance should drop sharply when the scaffold is removed. Comparing the two conditions under this perturbation reveals whether the trained policies differ in their structural dependence on the action list.

Diagnostic Experiment (Condition A) – Format Mismatch. To diagnose the failure mode in transferring to ALFWorld, we evaluate both trained models on ALFWorld with a BabyAI-style per-step action list artificially injected into the prompt at each turn. In the standard ALFWorld setup (matching Xi et al.), the environment's admissible-action list appears only at episode reset in the initial observation; subsequent turn prompts contain the raw text observation alone.

Under Condition A, we modify the per-turn prompt to append a randomly sampled subset of up to 15 actions drawn from the environment's `admissible_commands` at each step, formatted as

"AVAILABLE ACTIONS: <action1>,<action2>,...", mirroring the per-step scaffold the model receives during BabyAI training. The 15-action cap was imposed due to compute budget constraints, as longer prompts increase inference cost per rollout; in a follow-up replication we removed the cap and exposed the full admissible list, confirming that the results are robust to this choice. All other aspects of the ALFWorld evaluation are held constant.

If the cross-environment transfer failure is driven by a surface-level format mismatch, then supplying this familiar scaffold at test time should partially recover held-out performance. A null result under Condition A implicates a deeper representational failure rather than a prompt formatting issue.

4 Experimental Setup

4.1 Model and Infrastructure

All experiments use `Qwen/Qwen2.5-3B-Instruct` as the base model, fine-tuned with the verl v0.7.1 framework using SGLang for inference. Training runs on 8×H100 GPUs via Modal cloud compute, with model weights stored in `bfloat16` precision and gradient checkpointing enabled throughout.

4.2 Training Data and Environments

We use the BabyAI task suite via the AgentGym interface as the held-in training environment, with the same 810/90 train/test split as Xi et al. (2026). ALFWorld serves as the held-out transfer environment and is never seen during training. For evaluation, due to compute constraints, we evaluate on a random sample of 50/60 BabyAI and 80/134 ALFWorld games (fixed seed, 42). Held-in evaluation always provides the full per-step action list regardless of training condition, keeping our BabyAI numbers directly comparable to Xi et al.'s Table 3. Held-out evaluation uses ALFWorld's standard interface, which provides no such list.

4.3 Hyperparameters

Both interface conditions are trained for 300 GRPO steps with a batch size of 16 and $G = 8$ rollouts per prompt. The actor learning rate is 10^{-6}, with a KL loss coefficient of 0.001 and an entropy coefficient of 0.001. Maximum prompt and response lengths are 2048 and 8192 tokens respectively. All reported results are from single training runs per condition; the preliminary experiment in Section 3.2 used 3 random seeds.

4.4 Reward Function

The two environments use structurally different reward signals that reflect their different task horizons.

BabyAI (sparse, but step-efficiency). BabyAI tasks are short-horizon navigation problems solvable in tens of steps, so we use the step-efficiency reward from Chevalier-Boisvert et al. [6]:

$$r = \begin{cases} 1 - 0.9\,\dfrac{t}{T} & \text{if the agent completes the goal at step } t, \\ 0 & \text{otherwise (timeout or illegal sequence),} \end{cases}$$

where $T = 30$ is the maximum number of turns per episode. The reward thus ranges in $(0.1,\ 1]$ on success, decaying linearly with the number of steps taken, and is exactly zero on failure. VERL collects per-turn reward deltas in `extra_info["turn_scores"]`; the final scalar passed to GRPO is their sum, which equals r above because only the completion step is non-zero.

ALFWorld (sparse, binary). ALFWorld household tasks require many interdependent sub-goals, making intermediate shaping signals unreliable. We therefore use a binary reward:

$$r = \mathbf{1}[\text{task completed within } T \text{ turns}], \quad T = 30.$$

The ALFWorld environment itself emits 1.0 only upon full task completion and 0.0 at every other step.

5 Results

5.1 Main Experiment

Table 1 summarises the held-in and held-out results for both interface conditions at 300 GRPO training steps, alongside the untrained base model.

Method	Held-In (BabyAI) ↑	Held-Out (ALFWorld) ↑	$g = \Delta$Held-In $-$ ΔHeld-Out
Base model	0.273	0.013	—
`per_step`@300	0.321	0.013	0.048
`examples`@300	0.323	0.013	0.050

Table 1: Main experiment results. Held-In: success rate on BabyAI (training environment), always evaluated with the full per-step action list. Held-Out: zero-shot success rate on ALFWorld. g is the generalization gap defined in Section 3.

Both conditions achieve nearly identical held-in gains: `per_step` improves BabyAI success rate by $+0.048$ over the base model, and `examples` by $+0.050$. The training curves in Figure 3 show that `per_step` benefits from an early advantage, which suggests that the per-step list provides a meaningful scaffold before the policy has learned the task structure. But `examples` closes this gap by step 200 and matches it at convergence. At the same time, held-out performance on ALFWorld remains at 0.013 for both trained conditions, equal to the untrained base model. Removing the per-step action list during training does not close the generalization gap.

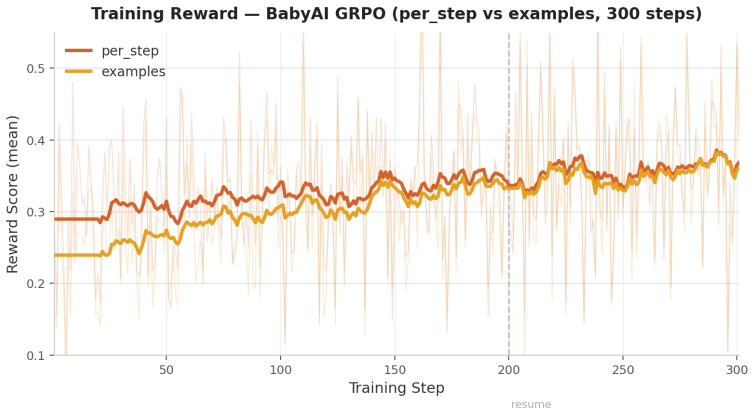

Figure 3: Training reward curves for `per_step` and `examples` over 300 GRPO steps (mean across episodes, smoothed). `examples` converges to `per_step` performance by approximately step 200.

5.2 Interpretability Experiment

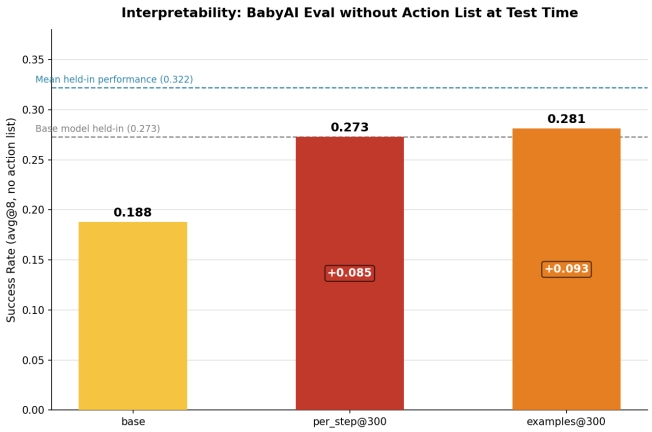

Figure 4: BabyAI success rate evaluated *without* the per-step action list at test time. The gray dashed line marks the base model held-in performance *with* the action list (0.273); the blue dashed line marks the mean held-in performance of the two trained models *with* the action list (0.322).

Figure 4 shows BabyAI performance when the action list is withheld at inference time. The untrained base model scores 0.188 without the list, a drop of 0.085 from its standard held-in score of 0.273, reflecting its reliance on the per-step scaffold as a navigational aid prior to any fine-tuning. Both trained models substantially exceed this unscaffolded base: `per_step@300` scores 0.273 and `examples@300` scores 0.281, closely approaching their respective with-list scores of 0.321 and 0.323 (Table 1). Neither model collapses when the scaffold is removed. Notably, `per_step@300` performs closely in relation to `examples@300`, despite never having been trained without the list. This indicates that the trained policy does not depend on the per-step list at inference time, directly contradicting the scaffold-dependency hypothesis of Xi et al. (2026).

5.3 Diagnostic Experiment — Condition A

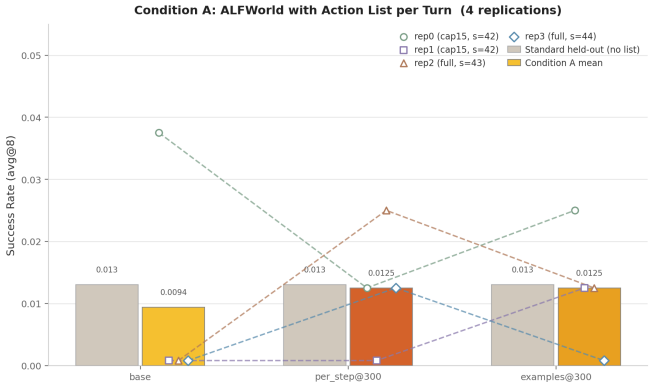

Figure 5: ALFWorld success rate under Condition A (BabyAI-style per-step action list injected into the ALFWorld prompt), averaged over 4 independent replications (seeds 42–44) with different game subsets, alongside the standard held-out evaluation.

Figure 5 shows held-out performance under Condition A. Across four independent replications with different game subsets (seeds 42–44), both trained models score 0.013 under Condition A, identical to the standard ALFWorld evaluation. Providing the familiar BabyAI-style action list at ALFWorld test time yields no improvement whatsoever. This rules out prompt-format mismatch as the cause of

the transfer failure: the models are not simply failing because they expect a list that ALFWorld does not supply. The failure is representational.

6 Discussion

6.1 Summary of Evidence

Three experiments converge on a single conclusion. The main ablation shows that removing the per-step action list during training has no effect on cross-environment transfer: both conditions improve BabyAI performance by ≈0.05 and leave ALFWorld performance unchanged at 0.013. The interpretability probe shows that trained models do not depend on the list at inference time. The Condition A diagnostic shows that providing the list in ALFWorld costs nothing either. The interface is not the bottleneck. Xi et al.'s (2026) attribution of BabyAI's transfer failure to its per-step action list is not supported.

6.2 What the Failure Actually Is

Figure 6: Case study in BabyAI. The base model wanders inefficiently, issuing redundant `check available actions` calls; the `per_step@300` model navigates directly to the target in 7 steps. The efficiency gain illustrates what GRPO fine-tuning actually learns: a BabyAI-specific navigation strategy, not a general reasoning capability.

Figure 6 illustrates what the held-in gain looks like in practice. The trained model identifies the target object from the observation and executes a near-optimal path, which is the right behaviour for an agent navigating BabyAI.

To understand why the transfer fails, we consider what BabyAI and ALFWorld structurally demand of an agent. BabyAI tasks are single-goal, short-horizon problems of navigating a 2-D grid space: *go to the red ball*, *pick up the key*. The task is complete in one action primitive, and the observation prompt makes the presence/absence of the target object explicit at every turn. The optimal strategy is therefore almost purely reactive: identify the relevant object from the list, navigate directly to it, act. If the objective is not directly present, move to increase the scanned area. GRPO concentrates the policy on this strategy because it reliably earns the step-efficiency reward.

ALFWorld tasks are structurally incompatible with this strategy. They require multi-step sub-goal chains: find an object across multiple rooms, transform its state (heat, cool, clean), then place it at a target location. The target object is not named in a list; it must be found through exhaustive spatial search. Object state must be tracked across turns. Sub-goals must be sequenced correctly. The BabyAI-trained model's reactive, simple heuristic produces systematic failures in this setting.

This failure pattern is directly visible in the ALFWorld trajectories we collected. In a representative *heat-and-place* task (*heat some apple and put it in garbage can*), the trained model navigates immediately to the fridge (a plausible action in the BabyAI environment) but the fridge contains no apple (it holds bread, lettuce, and a potato). Rather than continuing the search, the model picks up the potato, the first graspable object it encounters, and attempts to act on it. By turn 26 it is stuck in a loop: `put potato 1 in microwave 1` → *Nothing happens* → `cool potato 1 with refrigerator 1` → *Nothing happens*. At this point, it also hallucinates `refrigerator 1`, an object name that does not exist (`fridge 1` is the correct referent in ALFWorld). This suggests that the model trained in BabyAI which contains a more limited series of objects fail to identify the actionable objects in ALFWorld due to its larger vocabulary. The episode terminates at the turn limit with cumulative reward 0. This is not a scaffold problem. It is a task-structure problem: the model has learned to exploit actions on visible objects in the horizon but has not learned the exhaustive search and state-tracking that ALFWorld requires.

6.3 Implications for RL Fine-Tuning of LLM Agents

This finding has a practical implication for how RL fine-tuning is applied in deployment settings. Interface redesign alone (removing scaffolds, randomising action lists, varying prompt formats -) is unlikely to produce agents that transfer across environments, because the bottleneck is not the interface but the training distribution. Mitigations that operate at the training distribution level, such as multi-environment co-training or domain randomisation over task structure, are better-motivated targets for future work. We discuss these directions in Section 8.

7 Limitations

Our findings rest on a deliberately narrow experimental slice, and several limitations deserve to be stated explicitly so that the scope of our claim is clear.

Single base model. All conditions use Qwen2.5-3B-Instruct. Scaffold internalisation and the form of environment specialisation we observe may not be invariant across scale: stronger pretrained priors at 7B or 14B parameters could either dissolve the specialisation (if larger models form more abstract, transferable representations) or harden it (if larger capacity permits sharper environment-specific overfitting). Our 3B result cannot adjudicate between these alternatives.

Single transfer pair. We test one transfer direction (BabyAI → ALFWorld) across two environments that share a text-game modality. The weight-level specialisation story is therefore a single-target claim. Whether the same mechanism applies across genuinely heterogeneous modalities (web, code, multi-modal interfaces) is an open question that our setup is not equipped to answer.

Near-floor Held-Out signal. ALFWorld zero-shot success sits at approximately 0.013 for every condition we tested, including the base model. With the signal so close to the floor, small positive transfer effects would not be statistically distinguishable from noise. Our claim is therefore that no *detectable* transfer occurs in our setup, not that transfer is impossible in principle.

Single algorithm. GRPO is the only RL algorithm we test. Whether the specialisation story is specific to group-relative methods or generalises to PPO, DPO, or other RL fine-tuning approaches remains untested.

Single training seed and constrained budget. We fine-tune for 300 GRPO steps per condition and use a single random seed for each. Long-horizon dynamics, including whether environment specialisation eventually relaxes under more training or hardens further, are not characterised by our data. The single-seed budget also means that run-to-run optimisation noise is folded into our point estimates: the in-domain gap of 0.002 between `per_step` and `examples` is consistent with sampling variance, and we report it as a tie rather than as a directional finding.

No mechanistic characterisation. We rule out scaffold dependency and prompt-format mismatch as causes of transfer failure, but the residual explanation, weight-level environment specialisation, is named, not characterised. We do not yet identify which layers, attention patterns, or feature dimensions encode the BabyAI-specific structure, nor whether `per_step` and `examples`-trained models share that structure or diverge on it. Mechanistic interpretability work is needed to convert our by-elimination conclusion into a positive account.

8 Conclusion and Future Work

We tested the claim, advanced by Xi et al. (2026), that BabyAI's per-step action list during RL fine-tuning induces a scaffold dependency that limits cross-environment generalisation. Across three controlled experiments (a main ablation, an interpretability probe, and an Adapted-ALFWorld diagnostic), the claim does not hold. Both `per_step` and `examples`-fine-tuned models converge to nearly identical in-domain performance; both retain BabyAI performance when the action list is stripped at test time; and neither benefits when ALFWorld prompts are reformatted to match BabyAI's style. The transfer failure is real, but it lives in the weights, not in the interface.

This finding points to three concrete next directions.

Multi-environment co-training. If single-environment RL fine-tuning produces representations that do not transfer, the most natural mitigation is to train on a mixture of environments where no single one dominates the gradient signal. Joint BabyAI + ALFWorld fine-tuning would test whether transferable representations form when the optimiser is forced to balance multiple interface formats and task distributions simultaneously.

Scale. Repeating the ablation at 7B and 14B would clarify whether weight-level environment specialisation is a small-model artefact (washed out by stronger pretrained representations) or a structural property that persists with scale. Given the rate at which LLM-agent work is moving to larger backbones, this is the most informative single follow-up.

Mechanistic probe. Activation patching and representation-similarity analysis between `per_step` and `examples` checkpoints would convert our by-elimination conclusion into a positive account. The goal is to identify *what* gets specialised in the weights (which layers, which feature dimensions) and whether the two scaffold conditions converge on the same internal structure or diverge.

A broader version of the same study would also extend the held-out evaluation beyond ALFWorld to WebShop, TextCraft and SearchQA (the remaining environments in Xi et al. (2026)'s benchmark) to test whether the no-transfer pattern is universal or breaks for the right kind of target.

References

[1] Xi, Z. et al. *Can RL Improve Generalization of LLM Agents? An Empirical Study.* arXiv:2603.12011, 2026.

[2] Xi, Y. et al. *AgentGym-RL: Training LLM Agents for Long-Horizon Decision Making through Multi-Turn Reinforcement Learning.* arXiv:2509.08755, 2025.

[3] Shridhar, M. et al. *ALFWorld: Aligning Text and Embodied Environments for Interactive Learning.* ICLR, 2021.

[4] Shao, Z. et al. *DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models.* arXiv:2402.03300, 2024. (Introduces GRPO.)

[5] Sheng, G. et al. *HybridFlow: A Flexible and Efficient RLHF Framework.* arXiv:2409.19256, 2024. (VERL.)

[6] Chevalier-Boisvert, M. et al. *BabyAI: A Platform to Study the Sample Efficiency of Grounded Language Learning.* ICLR, 2019.

Author Contributions

- **Hana Liu.**
  - *Idea Development.* Zoomed in the research question to the BabyAI transfer failure and scaffolding dependency.
  - *Pipeline and Training Algorithm.* Hand-coded the core agent–environment interaction loop for VERL's multi-turn GRPO rollouts (`babyai_interaction.py`) and designed the step-efficiency reward function used to train the BabyAI policy (`reward.py`).
  - *Experiments.*
    - designed together and executed the main ablation experiment;
    - designed and executed the interpretability experiments, ran no-list) evaluation pass on both fine-tuned checkpoints;
    - designed and executed the Adapted-ALFWorld diagnostic across replications;
  - *Analysis.*
    - analyzed the preliminary experiment data, diagnosed and corrected AI errors in edge cases that cause polluting data (`babyai_interaction.py`), generated tensorboard curves and conducted quantitative and qualitative discussions.
    - analyzed the data from the main ablation, interpretability and diagnostic experiments, generated the main training-curve, tables, and evaluation charts.
    - *Writing.*
      - **Project Proposal.**
      - **Milestone Report.**
      - **Poster:** wrote Experiment Design, Setup, Results and Conclusion; created the template and conducted the visual design.
      - **Final Report** wrote Extended Abstract, Abstract, Methods, Results and Discussion.
- **Alfred Sjöqvist.**
  - *Idea Development.* Proposed the original paper and the interface-agnostic generalization question as the starting point.
  - *Infrastructure and pipeline.*
    - Built the end-to-end Modal training pipeline (data preparation, GRPO orchestration via VERL, checkpointing, trajectory logging)
    - the BabyAI and ALFWorld evaluation harnesses, and the run-management scripts (`modal_train_babyai.py`, `modal_train_alfworld.py`, `prepare_*_data.py`, smoke tests, diagnostic runners) used across all experiments.
    - built a human interface where we played and probed how BabyAI works (play_babyai.py).
  - *Experiments.* Designed and executed the preliminary experiment; GRPO training on Modal A100s; produced the per-run trajectory dumps used for downstream analysis. Designed together the main ablation experiment.
  - *Analysis.*
    - Performed the first-pass quantitative analysis of training curves, reward dynamics, and per-turn behaviour, particularly, the "check available actions" meta-action audit on the milestone-era traces to characterize how much the base model relies on actively querying the action list as a navigational crutch.
    - Analyzed the training data and the eval results from the main ablation experiments.
  - *Writing.*

* **Milestone Report:** Provided the first draft with experiment setup; curated, analyzed and compiled data for the report writing.
* **Poster:** Drafted the Problem Statement, Motivation, Prior Work and Limits & Further Work, References sections of the poster;
* **Final Report:** drafted the Introduction, Related Work, Limitations and Conclusion & Future Work, References sections of this final report; curated the prior-work citations.

AI Tools Disclosure

We used Claude code to assist the building of the infrastructure and the pipeline. But we did ourselves the core RL training algorithm, the model-envrionment interaction loop via VERL and the reward function. We used Claude to assist data analysis and graphics generation, but all the data were authentic from our experiments runs.