ACTION2DELTA: DECISION-ALIGNED DYNAMICS MID-TRAINING FOR LANGUAGE AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning action consequences from environment interaction provides a route to improving language-agent decisions. Next-state prediction learns these effects by fitting successor descriptions, but aggregate accuracy can obscure confusion between candidate actions. Conventional generative objectives provide limited supervision for distinguishing consequences at a common starting state and assessing their impact on action selection. We propose Action2Delta, a framework that organizes environment experience around action–consequence discrimination and decision relevance, comprising Contrastive Delta Dynamics (CDD) and Verified Regret Alignment (VRA). CDD executes different actions from the same real environment state and represents their observable consequences as deltas. It scores the same observed delta under different action conditions to learn their correspondence. VRA replaces predicted successors with real ones under a fixed value evaluator and uses real rollout returns to verify the resulting decision gains. These gains weight dynamics training, while the returns supervise the policy. On ALFWorld and ScienceWorld, static Action2Delta reaches 86.57% success and 72.73 macro-average score, exceeding uniform training with the same real feedback by 1.74 percentage points and 1.80 score points. The deployed actor selects actions directly, without world-model queries or environment branching. Mechanism analyses show improved action–consequence discrimination and lower candidate decision regret, with additional task gains on new goal compositions and recoverable adverse starts.

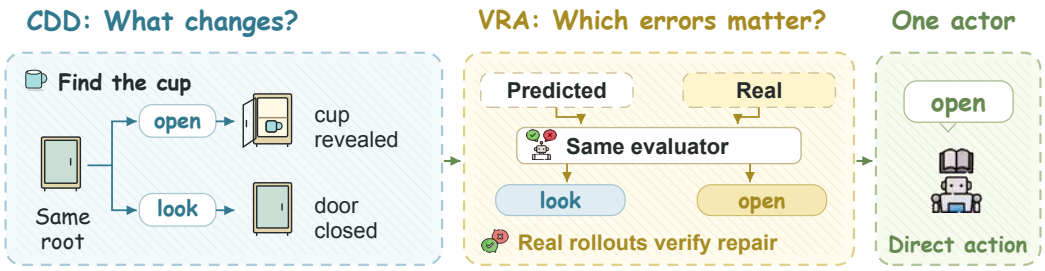

Figure 1: Action2Delta learns what actions change and which prediction errors matter for decisions. CDD distinguishes action consequences; VRA verifies successor-replacement gains with real rollouts under a fixed evaluator. These signals train a direct actor. The cabinet example is schematic.

1 INTRODUCTION

Language agents must choose actions whose consequences advance a goal across multiple environment steps. An instruction-following policy can produce a valid command while misunderstanding whether it exposes an object, changes a prerequisite, or makes recovery harder. Environment interaction supplies supervision about these effects in addition to demonstrations of what an expert did. Recent work studies text world simulation, environment-derived mid-training, and self-experience as ways to acquire this knowledge before task optimization (Wang et al., 2024; Zhang et al., 2025;

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.

Chen et al., 2025; Lei et al., 2026; Yu et al., 2026). Our focus is how to organize this experience so that learning action consequences improves the decisions of the deployed policy.

Decision usefulness requires more than an accurate average successor description. Many unchanged facts are easy to reproduce, whereas a single mistaken prerequisite can reverse the ordering of candidate actions. Even a compact delta predictor can exploit common transitions without distinguishing the effects of alternative actions at the same starting state. Conversely, improving a field-level prediction metric need not reduce the loss incurred by the chosen action. This motivates our key question: *how can dynamics mid-training distinguish action consequences, identify errors that matter to selection, and transfer the resulting supervision into a directly executable policy?*

Existing approaches address complementary parts of this question. Generative world models predict observations or state changes; early-experience and state-reaching methods reorganize environment data for agent training (Zhang et al., 2025; Li et al., 2025; Lei et al., 2026; Zuo et al., 2026). Value-aware model learning connects model approximation to downstream value (Farahmand et al., 2017). Inference-time systems instead use world models, memory, or deliberation to improve online choices (Zhang et al., 2026; Wang et al., 2026; Lin et al., 2023). These approaches motivate a training procedure that explicitly compares action conditions and verifies the decision effect of replacing a modeled successor. Such verification can prioritize model supervision while the final agent retains a direct action interface.

Three design requirements determine this procedure. Action comparisons need a common underlying environment state. Decision attribution needs a fixed evaluator so that the source of a changed ranking is isolated. Finally, model diagnostics need a specified path into the deployed actor. Action2Delta addresses these requirements with verified sibling branches, evaluator-fixed replacement, and separate actor-side dynamics supervision.

Our approach contrasts actions with the starting state and target delta held fixed, then verifies the decision effect of successor replacement under a fixed evaluator. We call this decision-oriented experience organization Action2Delta. CDD contrasts the likelihood of a fixed real delta under different actions from a verified common root. VRA then replaces predicted successors with real successors and uses real continuation returns to verify whether the selected action improves. The positive repair signal weights dynamics supervision in both the world model and the actor, while real candidate returns define the policy's soft targets. Figure 1 illustrates the core idea. Our contributions are:

- **A decision-oriented experience framework.** We connect action-conditioned dynamics learning, verified successor replacement, and direct policy training in a common branch-data pipeline.
- **Action–consequence discrimination.** CDD uses same-root, same-target action contrast and explicit treatment of equivalent consequences to strengthen action dependence.
- **Verified repair feedback.** VRA measures the positive decision repair induced by successor replacement and separates dynamics weighting from return-based policy supervision.
- **Systematic empirical analysis.** Experiments on ALFWorld and ScienceWorld connect improved action pairing to direct-actor outcomes. Shared-feedback controls, policy diagnostics, and generalization tests examine the two mechanisms and the reuse of learned consequences.

2 Related Work

Relevant research spans environment models, decision-oriented training, and world-model-assisted execution. These lines differ in how they obtain environment experience, assign a learning objective, and spend computation at deployment. We position Action2Delta at their intersection while preserving these distinctions in the controlled comparisons.

World models support both environment prediction and downstream control. Early neural world models and Dreamer illustrate the broader connection between predictive representations and learned behavior (Ha & Schmidhuber, 2018; Hafner et al., 2025). For language agents, Wang et al. (2024) compare full-state and state-change simulation. AEE collects early multi-action experience, SPA learns from exploration through state and transition prediction, and From Word to World investigates implicit text world models (Zhang et al., 2025; Chen et al., 2025; Li et al., 2025). State2State derives reachable-state objectives from environment interaction, RWML uses reinforcement learn-

ing to improve observation prediction, and Qwen-AgentWorld develops world-model training for general agents (Lei et al., 2026; Yu et al., 2026; Zuo et al., 2026). Action2Delta adds a fixed-target contrast across real sibling actions and connects that contrast to decision-verified replay.

Decision-oriented model learning aligns model training with downstream decisions. Value-aware model learning studies losses sensitive to downstream value (Farahmand et al., 2017). Contrastive predictive learning provides a broader precedent for discriminative representation objectives (van den Oord et al., 2018), while AWR and DPO organize policy updates around returns or preferences (Peng et al., 2019; Rafailov et al., 2023). Policy distillation transfers an improved action distribution to an executable policy (Rusu et al., 2016). Action2Delta instantiates this perspective through the verified change in a fixed evaluator's selected action when only its successor input is replaced. The verified repair signal weights dynamics samples, while real candidate returns define the actor's target distribution. GRPO and GiGPO provide downstream optimization settings for measuring compatibility (Shao et al., 2024; Feng et al., 2025).

Inference-time agents combine acting with reasoning, memory, or lookahead. ReAct and SwiftSage study reasoning-driven execution and fast–slow control (Yao et al., 2023; Lin et al., 2023). CoEx, MemWM, and WorldEvolver incorporate abstraction, memory, or updated rules into world-model use (Kim & Hwang, 2025; Wang et al., 2026; Zhang et al., 2026). Recent work examines belief interfaces and whether agents can effectively exploit world models as foresight tools (Kumar et al., 2026; Qian et al., 2026). ALFWorld, ScienceWorld, and AgentBoard provide complementary environments and diagnostics for these agents (Shridhar et al., 2021; Wang et al., 2022; Ma et al., 2024); our controlled study uses Qwen3 models (Yang et al., 2025). Action2Delta brings these lines together through same-root action discrimination, verified successor replacement under a fixed evaluator, and transfer to a direct actor. Action2Delta concentrates the extra branching and verification in training, while additional test-time planning is evaluated separately.

3 METHOD

Action2Delta learns from real sibling transitions and transfers their decision-relevant supervision into an actor. Figure 2 shows two learning mechanisms and the policy update they support. The workflow has three stages: (i) CDD fits observable deltas and contrasts their action conditions; (ii) VRA verifies the selection changes induced by real-successor replacement; and (iii) joint dynamics and return-based supervision updates the deployment policy. Training uses a world-model copy f_θ and an actor copy π_ϕ; evaluation uses only π_ϕ and the public-history ledger.

3.1 PUBLIC STATE AND VERIFIED ACTION BRANCHES

The model input contains only evidence available before the current action. Let $h_t = (o_0, a_0, \ldots, o_t)$ denote public history, $z_t = E(h_t)$ a deterministic evidence ledger, g the goal, and b_t the remaining action budget. The actor receives $x_t = (h_t, z_t, g, b_t)$; the dynamics model predicts $f_\theta(\delta \mid h_t, z_t, a)$ without an explicit goal. The ledger records location, held objects, visible entities, and observed attributes, preserving unparsed feedback. Unknown facts remain unknown, and environment feedback establishes action effects.

Observable deltas distinguish physical change from new information. A canonical extractor d and reconstructor R satisfy $\delta_t = d(z_t, z_{t+1})$ and $R(z_t, \delta_t) = z_{t+1}$. Deltas contain `changes`, `reveals`, `visibility`, and `feedback`. For example, opening a cabinet changes its door state and reveals a cup's existing location. The ledger summarizes public evidence, and the full history provides the model's temporal context.

Verified branching provides the common starting point required for action comparison. A root group is

$$\mathcal{G}_t = (h_t, z_t, \{(a_i, \delta_i)\}_{i=1}^{B}), \qquad \delta_i = d(z_t, z'_{t,i}). \tag{1}$$

Each branch executes its action after restoring the same simulator state or resetting the task and replaying the same prefix. Restoration checks include available internal state, time, and randomness, but internal state is not supplied to the models. Candidates combine the common reference policy's preferred action, policy samples, and commands formed from public objects. Commands are deduplicated, and all siblings remain within one data split. In time-dependent environments, deltas include observable changes that occur while the action advances the environment.

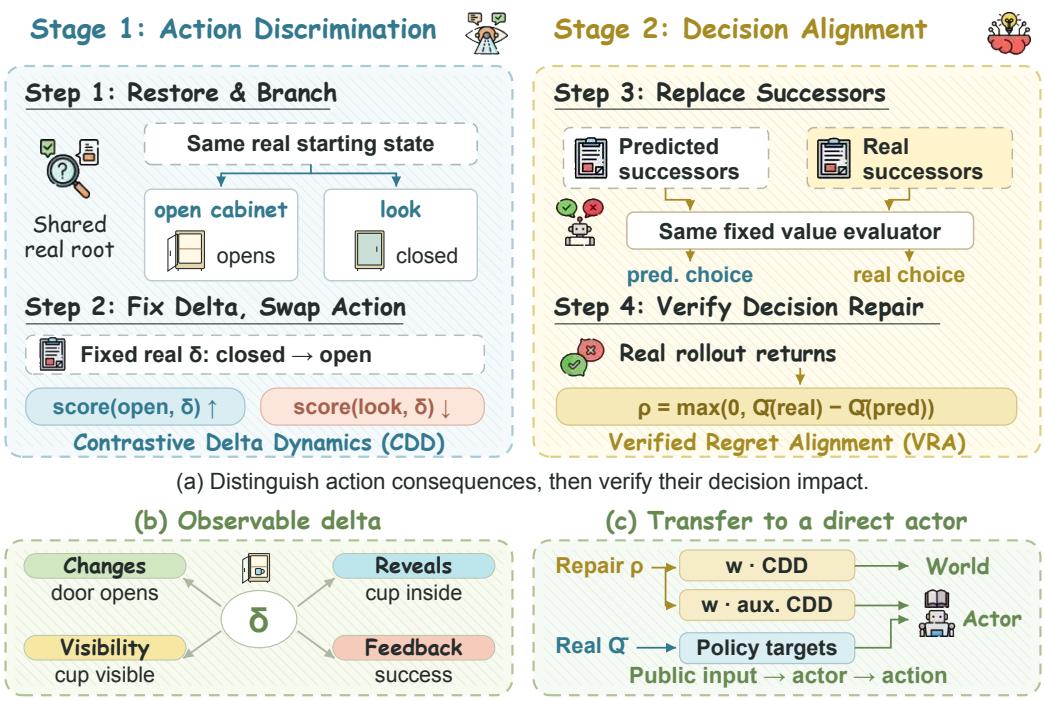

Figure 2: Action2Delta learns action-discriminative and decision-relevant dynamics. (a) CDD contrasts a fixed real delta across sibling actions; VRA replaces predicted successors under a fixed evaluator and verifies choice repair using real rollout returns. (b) Observable deltas distinguish changes, reveals, visibility, and feedback; cabinet examples are schematic. (c) Repair weights CDD for separate world-model and actor copies, while real returns define actor policy targets. Deployment uses only the actor.

3.2 Contrastive Delta Dynamics

CDD learns both the content of a transition and which action explains it. Define the length-normalized log likelihood of a real delta as

$$\ell_\theta(\delta; a, h, z) = |\delta|^{-1} \log f_\theta(\delta \mid h, z, a), \qquad \mathcal{L}_\Delta = -\mathbb{E}[\ell_\theta(\delta; a, h, z)]. \tag{2}$$

For sibling actions with different verified consequences, keep the target δ_i fixed and swap only the action condition:

$$\mathcal{L}_{\text{pair}}(i, j) = \text{softplus}(m - \ell_\theta(\delta_i; a_i, h, z) + \ell_\theta(\delta_i; a_j, h, z)), \qquad \mathcal{L}_{\text{CDD}} = \mathcal{L}_\Delta + \lambda_C \mathcal{L}_{\text{pair}}. \tag{3}$$

We average both directions over valid unordered pairs within each root, then average roots. Hard negatives require distinct verified consequences. Equivalent observed consequences are filtered or represented as multiple positives. Roots without a valid pair contribute only $\mathcal{L}_\Delta$. Overlapping stochastic outcomes require repeated verification, soft labels, or omission from hard contrast. Explicit empty deltas remain valid prediction targets.

The contrastive term directly penalizes ignoring the action. An action-independent predictor gives the same likelihood to the fixed delta under both conditions, leaving the pair loss at $\text{softplus}(m)$. Reducing this term therefore drives action dependence. Appendix B relates reversible deltas to information preservation and distinguishes the practical normalized loss from an ideal sequence-entropy identity.

3.3 Verified Regret Alignment

VRA measures a decision change using real returns under a common continuation policy. Freeze π_{ref} for a feedback phase. After executing candidate a_i, perform M continuations with the same

remaining budget and independent rollout seeds:

$$Q_i = \mathbb{E}[Y \mid x_t, a_i, \pi_{\text{ref}}], \qquad \bar{Q}_i = M^{-1} \sum_{r=1}^{M} Y_{i,r}, \qquad Y \in [0,1]. \tag{4}$$

All expectations condition on the fixed simulator root and acquired action branch, suppressed in the notation. Averaging stochastic first-step outcomes requires resampling those outcomes as well. ALFWorld uses binary final success. ScienceWorld training clips the final score to $[0, 100]$ and divides by 100; evaluation retains the raw score. A value head $v_\psi(z', g, b_t - 1)$ is trained on these returns with two-fold cross-fitting by complete task or root group. A diagnostic root uses a head not fitted on that root.

Successor replacement holds the candidate set and evaluator fixed. For a predicted delta $\hat{\delta}_i$, form $\hat{z}'_i = R(z_t, \hat{\delta}_i)$ and the real ledger $z'_i = R(z_t, \delta_i)$. Using the same value head, goal, budget, and tie rule, choose

$$j_{\text{pred}} = \arg\max_i v_\psi(\hat{z}'_i, g, b_t - 1), \qquad j_{\text{real}} = \arg\max_i v_\psi(z'_i, g, b_t - 1). \tag{5}$$

Real rollout returns then verify whether this input replacement improved the choice. The decision-repair regret and normalized replay weight are

$$\rho_t = [\bar{Q}_{j_{\text{real}}} - \bar{Q}_{j_{\text{pred}}}]_+, \qquad w_t = 1 + \kappa\rho_t, \qquad \tilde{w}_t = w_t / \mathbb{E}_{\text{batch}} w_t. \tag{6}$$

The world model minimizes $\mathcal{L}_{\text{world}} = \mathbb{E}_t[\tilde{w}_t \mathcal{L}_{\text{CDD}}(\mathcal{G}_t)]$, with uniform replay from the original dynamics pool. The base weight retains zero-repair roots. Raw weights lie in $[1, 1 + \kappa]$, and batch normalization gives $\mathbb{E}_{\text{batch}} \tilde{w}_t = 1$.

DRR has a precise candidate-local interpretation. For true returns, let $\mathcal{R}_j = \max_i Q_i - Q_j$ and $\rho_t^{\text{true}} = [Q_{j_{\text{real}}} - Q_{j_{\text{pred}}}]_+$. Then

$$\mathcal{R}_{j_{\text{pred}}} - \mathcal{R}_{j_{\text{real}}} = Q_{j_{\text{real}}} - Q_{j_{\text{pred}}}, \qquad |\rho_t - \rho_t^{\text{true}}| \le 2\epsilon_Q \tag{7}$$

when $\max_i |\bar{Q}_i - Q_i| \le \epsilon_Q$. DRR estimates the positive regret reduction induced by successor replacement within the fixed candidate set and evaluator. Signed return differences retain both improvements and degradations for auditing. Appendix B gives the estimation analysis.

3.4 Policy transfer and training procedure

Real returns supervise the actor independently of the DRR weighting rule. Normalize the frozen reference policy on $C_t = \{a_i\}_{i=1}^B$ to obtain $p_i > 0$. Exact command-sequence probabilities, including termination, define this prior. For $\tau > 0$, the KL-regularized candidate objective has the solution

$$q^* = \arg\max_{q \in \Delta(C_t)} \left\{ \sum_i q_i \bar{Q}_i - \tau \,\mathrm{KL}(q \| p) \right\}, \qquad q_i^* = \frac{p_i e^{\bar{Q}_i/\tau}}{\sum_j p_j e^{\bar{Q}_j/\tau}}. \tag{8}$$

The actor combines these targets with shared expert supervision and its own weighted CDD auxiliary objective:

$$\mathcal{L}_{\text{actor}} = -\mathbb{E}_t \sum_i q_i^* \log \pi_\phi(a_i \mid x_t) + \lambda_{\text{BC}} \mathcal{L}_{\text{expert}} + \lambda_{\text{keep}} \mathbb{E}_t[\tilde{w}_t \mathcal{L}_{\text{CDD}}^\phi(\mathcal{G}_t)]. \tag{9}$$

All actor inputs precede the action; real successors and returns are labels. The auxiliary $\mathcal{L}_{\text{CDD}}^\phi$ uses actor parameters with the dynamics prompt. CDD initializes the world model, its weights initialize the actor, and task SFT precedes VRA. Subsequent updates use separate parameter copies; the weighted auxiliary loss provides the actor's dynamics-supervision path.

The local policy analysis separates feedback quality from distillation quality. If $\mathrm{TV}(\pi_C, q^*) \le \epsilon_d$ for the actor normalized on C_t, Appendix B proves

$$\sum_i (\pi_{C,i} - p_i) Q_i \ge \tau \,\mathrm{KL}(q^* \| p) - 2\epsilon_Q - \epsilon_d. \tag{10}$$

The bound relates candidate-return change to target improvement, feedback error, and distillation error under a fixed reference continuation. A positive right-hand side certifies improvement for this candidate comparison.

Algorithm 1 Action2Delta with static feedback and optional online updates

Require: Base model, verified dynamics pool $\mathcal{D}$, expert data $\mathcal{E}$, common static cache, online rounds K

$\theta \leftarrow \mathrm{TrainCDD}(\mathcal{D})$ using Eq. (3); $\phi \leftarrow \mathrm{TaskSFT}(\theta, \mathcal{E})$
for phase $k = 0, \ldots, K$ **do**
  Freeze the common reference if $k = 0$, otherwise the current actor
  Load the shared static cache, or collect new roots, branches, and returns if $k > 0$
  Compute $\bar{Q}$ by Eq. (4); fit value heads on disjoint root folds
  for root group $\mathcal{G}_t$ **do**
    Predict $\hat{\delta}_i$; select $j_{\mathrm{pred}}, j_{\mathrm{real}}$ with the held-out head, Eq. (5)
    Store signed repair, w_t from Eq. (6), and q^* from Eq. (8)
  end for
  Update θ with weighted CDD and shared old-data replay
  Update ϕ with Eq. (9) on the same records
end for
return Actor π_ϕ for direct execution from public history and ledger

Static and online variants differ in when they acquire new feedback. The static phase freezes the roots, candidates, reference policy, and rollout cache so that weighting schemes use identical evidence. Each optional online phase collects training roots under an updated actor and recomputes the feedback. Algorithm 1 makes this distinction explicit. Deployment invokes neither the separate world model nor the auxiliary value head and performs no counterfactual environment branching.

4 Experiments

We evaluate whether action discrimination improves consequence learning, whether verified replacement gains help allocate feedback, and whether these signals improve the deployed actor. We then examine compositional generalization, further policy optimization, and resource requirements.

4.1 Experimental settings

The main evaluation uses a fixed, directly deployed actor. We use Qwen3-4B (Yang et al., 2025) for the core study, extend to Qwen3-8B/14B/32B, and evaluate Gemma 4 12B (Google DeepMind, 2026) for cross-family generalization. Models train separately on ALFWorld (Shridhar et al., 2021) and ScienceWorld (Wang et al., 2022). ALFWorld uses the official unseen split with 134 instances per training seed and a 40-action limit. ScienceWorld uses the official test tasks, all 30 task types, an empty simplification list, and a 100-action limit. The primary metrics are ALFWorld success rate and ScienceWorld type-macro-averaged final raw score, respectively. Negative terminal scores remain in the latter. All main comparisons use the same public observation ledger, greedy action decoding, and no legal-action enumeration. The actor performs no test-time world-model inference, value evaluation, or branch rollout.

Shared data separate the training objectives. Each environment supplies 100,000 transitions, 200 successful demonstration trajectories, and a static feedback cache containing 1,000 roots, four candidate actions per root, and eight continuations per candidate. CDD preserves sibling groups and excludes equivalent consequences from hard negatives. Uniform, NLL, total-regret, and VRA feedback share roots, candidates, rollout returns, and actor soft targets; their dynamics weights differ. Appendix D provides the configuration, additional controls, and statistical reporting protocol.

4.2 Main results

Action2Delta improves direct-actor task performance. Table 1 reports 86.57% ALFWorld success and 72.73 ScienceWorld score, improving ledger SFT by 16.42 percentage points and 9.90 score points. Delta-only to CDD adds 3.73 points and 2.20 points, respectively. Shared real feedback then adds 4.48 and 2.80 through uniform training. The full pipeline therefore combines gains from action contrast, real feedback, and replay allocation.

Table 1: Static direct-actor results. All rows use the same ledger. Rows below the rule share real feedback and policy soft targets. Values are reported means. Science denotes type-macro-averaged raw score.

ID	Training method	ALF SR ↑	Science ↑
B01	Task SFT	70.15	62.83
B03	Full-state dynamics → SFT	75.12	65.53
B04	Delta-only → SFT	76.62	65.93
B05	Weighted full-state → SFT	78.11	67.73
B06	AEE-IWM adapted → SFT	78.86	68.53
B08	CDD → SFT	80.35	68.13
B10	CDD + uniform feedback	84.83	70.93
B15	CDD + NLL-priority feedback	86.07	71.83
B16	CDD + total-regret feedback	84.33	70.53
B11	Action2Delta, static	**86.57**	**72.73**

Table 2: Consequence-learning diagnostics. Change F1 averages over samples with nonempty changes. Pair accuracy tests the same observed delta under alternative action conditions. Regret uses a fixed candidate set and evaluator. F1 and accuracy are percentages; regret uses normalized returns.

	ALFWorld			ScienceWorld		
Method	F1 ↑	Pair ↑	Regret ↓	F1 ↑	Pair ↑	Regret ↓
Full-state	82.31	71.62	.139	68.42	65.17	.183
Delta-only	84.71	73.03	.135	70.22	66.37	.179
Weighted full-state	86.91	75.28	.124	73.52	68.57	.167
CDD	86.41	83.19	.115	72.82	74.87	.157
CDD + uniform	87.11	83.84	.101	73.32	75.57	.143
CDD + NLL	87.29	84.21	.098	73.68	75.92	.139
CDD + VRA	86.81	83.61	.093	73.02	75.27	.134

Controlled comparisons separate the training contributions. B04 and B08 share delta targets, the transition pool, and the task interface; CDD additionally scores alternative action conditions. B10 and B11 share roots, returns, and policy soft targets and differ in dynamics weights. These comparisons distinguish the contributions of action contrast and feedback allocation.

Verified weighting provides an incremental gain under shared feedback. VRA improves on uniform weights by 1.74 ALFWorld percentage points and 1.80 ScienceWorld score points. The margins are 0.50/0.90 over NLL weights and 2.24/2.20 over total-regret weights. The shared-feedback comparison supports allocating dynamics supervision according to verified decision repair. Adapted external objectives, including AEE (Zhang et al., 2025), are evaluated under the common environment interface and training protocol. Complete B-series results appear in Table 8.

4.3 Mechanism and ablation analysis

Action contrast improves discrimination beyond delta generation. CDD raises pair accuracy from 73.03% to 83.19% on ALFWorld and from 66.37% to 74.87% on ScienceWorld (Table 2). Shuffling action–consequence assignments reduces accuracy to 50.62%/50.37%. Removing the contrastive term reduces both task means (Table 3), linking action discrimination to the deployed actor's performance.

The ablations separate supervision and representation choices. Retaining equivalent actions as negatives yields 84.08%/70.63, and merging reveals with changes yields 85.57%/71.41. At equal FLOPs, weighted full-state training reaches 87.06%/71.93, compared with 86.57%/72.73 for Action2Delta. Appendix D.3 provides the complete comparisons.

Table 3: Ablations of static Action2Delta. VRA is retained unless the row changes the relevant auxiliary supervision. A03 matches FLOPs using weighted full-state prediction.

ID	Modification	ALF SR	Science
B11	Complete static framework	86.57	72.73
A01	Remove action contrast	85.32	71.53
A02	Shuffle action–consequence assignments	77.86	66.43
A03	Weighted full-state, equal FLOPs	87.06	71.93
A06	Remove actor dynamics retention	85.82	71.03
A07	Keep equivalent actions as negatives	84.08	70.63
A08	Merge reveals with physical changes	85.57	71.41

Table 4: Generalization under shared feedback. ALF columns report success rate, using recovery success for G05; Science columns report macro score. Each row has its own evaluation construction. Closure holdout applies only to ScienceWorld.

	ALFWorld		ScienceWorld	
Evaluation	Uniform	VRA	Uniform	VRA
G02, unseen goal compositions	68.47	71.27	59.41	61.51
G03, action–object holdout	55.67	56.17	50.63	50.93
G05, recoverable adverse starts	63.17	66.27	58.27	61.17
G06, consistent entity aliases	78.11	77.51	66.39	66.09
40% observation masking	58.73	57.97	51.17	50.83
Related transition closure holdout	—	—	40.17	40.03

Successor replacement reduces candidate decision regret. With the same value evaluator, replacing predicted successors by real successors lowers candidate regret from .115 to .091 on ALFWorld and from .157 to .126 on ScienceWorld. Mean net repairs are .024/.031; the positive-part means used for weighting are .0297/.0413. Replacement also harms 6.30%/8.70% of audited choices. DRR uses the positive verified replacement gain to allocate dynamics supervision. VRA also has lower Change F1 than uniform replay but lower regret (Table 2), supporting evaluation by decision relevance.

The actor retains verified candidate-return gains. An independent candidate-return audit reports soft-target gains of .041/.052 and actor gains of .028/.033. Removing the actor's dynamics retention loss yields 85.82%/71.03. These diagnostics show how verified local gains enter policy behavior on the audited candidate sets. Table 23 reports the audit quantities.

4.4 Robustness, Generalization, and Compatibility

Learned consequences support new goal combinations and recovery. VRA improves on uniform feedback by 2.80/2.10 on new goal compositions and 3.10/2.90 on recoverable adverse starts (Table 4). Strict action–object holdout gives gains of .50/.30. The ordering reverses under entity aliases, 40% observation masking, and ScienceWorld transition-closure holdout. Source-environment CDD also improves both transfer directions in Appendix D.5.

Within-backbone gains extend across scales and model families. Table 5 extends the comparison to Qwen3-14B/32B and Gemma 4 12B. Relative to uniform feedback, VRA adds .50/.25/.50 ALFWorld percentage points, respectively. ScienceWorld scores increase by .3/.2 on Qwen3-14B/Gemma, while both Qwen3-32B variants score 76.7 at the reported precision. The DRR increment narrows as the Qwen3 uniform baseline improves. Appendix D.7 gives the six training paths and their common evaluation protocol.

Action2Delta integrates with further policy optimization. At 4B, GiGPO (Feng et al., 2025) yields 94.78%/77.23 after Action2Delta and 92.79%/75.73 after SFT. GRPO yields 93.03%/76.81 for Action2Delta and 93.28%/75.61 for State2State (Lei et al., 2026). Two online rounds reach

Table 5: Static direct actors across model scales and families. Uniform (B10) and VRA (B11) share feedback within each model. ALF is success rate (%); Science is type-macro score. Full training-path comparisons appear in Table 20.

	ALFWorld		ScienceWorld	
Backbone	Uniform	VRA	Uniform	VRA
Qwen3-4B	84.83	86.57	70.93	72.73
Qwen3-8B	93.03	93.78	74.17	74.67
Qwen3-14B	95.02	95.52	75.2	75.5
Qwen3-32B	97.01	97.26	76.7	76.7
Gemma 4 12B	93.78	94.28	73.0	73.2

88.56%/74.83, compared with 88.06%/74.13 for matched AWR/RFT. Table 19 gives the 8B extension.

4.5 Resource requirements

Direct deployment shifts the extra work to training. Static training uses 32,000 real continuations per environment. Token-based accounting gives 16.394/33.520 exaFLOPs for gradient training and 6.717/32.096 exaFLOPs for rollout inference for ALFWorld/ScienceWorld, respectively. CDD costs more gradient computation than delta-only because wrong-action branches also backpropagate. B10 and B11 share the feedback cache and its acquisition cost. B11 training-chain time is 88.91/279.27 minutes. Appendix D.8 details computation, collection, time, and memory use. Deployment uses a single actor, concentrating branching and value evaluation in training.

5 Conclusion

Action2Delta studies how environment dynamics supervision can improve the choices of a language agent. Its central idea is to organize experience by action discrimination and verified decision relevance. CDD contrasts real sibling consequences with a fixed target, and VRA uses real-successor replacement to measure a candidate-local repair signal. Return-based policy targets and weighted dynamics supervision transfer both sources of information to a direct actor. Experiments across two text environments show improved action pairing, lower candidate regret, and stronger direct-actor results under shared feedback.

AI use statement

AI tools assisted with manuscript refinement and data organization.

Reproducibility statement

Section 3 defines the learning objectives and training algorithm. Appendix A specifies the public representation, branch construction, and value evaluator. Appendix B gives the mathematical assumptions and proofs. Appendices D and E provide configurations, extended comparisons, and resource reporting; Appendix D.10 defines the statistical quantities and comparison scope.

REFERENCES

Shiqi Chen, Tongyao Zhu, Zian Wang, Jinghan Zhang, Kangrui Wang, Siyang Gao, Teng Xiao, Yee Whye Teh, Junxian He, and Manling Li. Internalizing World Models via Self-Play Finetuning for Agentic RL. arXiv preprint arXiv:2510.15047v1, 2025. URL https://arxiv.org/abs/2510.15047v1. Version 1, October 2025.

Amir-Massoud Farahmand, Andre Barreto, and Daniel Nikovski. Value-Aware Loss Function for Model-based Reinforcement Learning. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pp. 1486–1494. PMLR, 2017. URL https://proceedings.mlr.press/v54/farahmand17a.html.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-Group Policy Optimization for LLM Agent Training. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2505.10978.

Google DeepMind. Gemma 4 12B IT Model Card. Official model documentation, 2026. URL https://huggingface.co/google/gemma-4-12B-it. Accessed September 24, 2026.

David Ha and Jürgen Schmidhuber. World Models. arXiv preprint arXiv:1803.10122, 2018. URL https://arxiv.org/abs/1803.10122.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640(8059):647–653, 2025. doi: 10.1038/s41586-025-08744-2. URL https://www.nature.com/articles/s41586-025-08744-2.

Minsoo Kim and Seung-won Hwang. CoEx – co-evolving world-model and exploration. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 21629–21651, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.1179. URL https://aclanthology.org/2025.findings-emnlp.1179/.

Shubham Kumar, Harshit Kumar, Narendra Ahuja, and Saurabh Jha. Towards a Belief-Based World Model for LLM Agents. arXiv preprint arXiv:2609.00455, 2026. URL https://arxiv.org/abs/2609.00455.

Xuanyu Lei, Yiqi Zhu, Chenliang Li, Kaiming Liu, Peng Li, Ming Yan, Jieping Ye, Ya-Qin Zhang, and Yang Liu. State2State: Environment-Derived Mid-Training for LLM Agents. arXiv preprint arXiv:2608.04934, 2026. URL https://arxiv.org/abs/2608.04934.

Yixia Li, Hongru Wang, Jiahao Qiu, Zhenfei Yin, Dongdong Zhang, Cheng Qian, Zeping Li, Pony Ma, Guanhua Chen, and Heng Ji. From Word to World: Can Large Language Models be Implicit Text-based World Models? arXiv preprint arXiv:2512.18832, 2025. URL https://arxiv.org/abs/2512.18832.

Bill Yuchen Lin, Yicheng Fu, Karina Yang, Faeze Brahman, Shiyu Huang, Chandra Bhagavatula, Prithviraj Ammanabrolu, Yejin Choi, and Xiang Ren. SwiftSage: A Generative Agent with Fast and Slow Thinking for Complex Interactive Tasks. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2305.17390.

Chang Ma, Junlei Zhang, Zhihao Zhu, Cheng Yang, Yujiu Yang, Yaohui Jin, Zhenzhong Lan, Lingpeng Kong, and Junxian He. AgentBoard: An Analytical Evaluation Board of Multi-turn LLM Agents. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2401.13178.

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-Weighted Regression: Simple and Scalable Off-Policy Reinforcement Learning. arXiv preprint arXiv:1910.00177, 2019. URL https://arxiv.org/abs/1910.00177.

Cheng Qian, Emre Can Acikgoz, Bingxuan Li, Xiusi Chen, Yuji Zhang, Bingxiang He, Qinyu Luo, Dilek Hakkani-Tür, et al. Current Agents Fail to Leverage World Model as Tool for Foresight. arXiv preprint arXiv:2601.03905, 2026. URL https://arxiv.org/abs/2601.03905.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. In *Advances in Neural Information Processing Systems*, volume 36, pp. 53728–53741, 2023. doi: 10.52202/075280-2338. URL https://arxiv.org/abs/2305.18290.

Andrei A. Rusu, Sergio Gomez Colmenarejo, Caglar Gulcehre, Guillaume Desjardins, James Kirkpatrick, Razvan Pascanu, Volodymyr Mnih, Koray Kavukcuoglu, and Raia Hadsell. Policy Distillation. In *International Conference on Learning Representations*, 2016. URL https://arxiv.org/abs/1511.06295.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, et al. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. arXiv preprint arXiv:2402.03300, 2024. URL https://arxiv.org/abs/2402.03300.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. ALFWorld: Aligning Text and Embodied Environments for Interactive Learning. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2010.03768.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation Learning with Contrastive Predictive Coding. arXiv preprint arXiv:1807.03748, 2018. URL https://arxiv.org/abs/1807.03748.

Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. ScienceWorld: Is your Agent Smarter than a 5th Grader? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022. URL https://arxiv.org/abs/2203.07540.

Ruoyao Wang, Graham Todd, Ziang Xiao, Xingdi Yuan, Marc-Alexandre Côté, Peter Clark, and Peter Jansen. Can Language Models Serve as Text-Based World Simulators? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 2024. URL https://arxiv.org/abs/2406.06485.

Yujun Wang, Tao Zhang, Jinhe Bi, Aniri, Wenxuan Ye, Boliang Liu, Sikuan Yan, Shuning Wang, et al. MemWM: Memory-Augmented Text-Based World Model. arXiv preprint arXiv:2608.07107, 2026. URL https://arxiv.org/abs/2608.07107.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, 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. URL https://arxiv.org/abs/2210.03629.

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

Kai Zhang, Xiangchao Chen, Bo Liu, Tianci Xue, Zeyi Liao, Zhihan Liu, Xiyao Wang, Yuting Ning, et al. Agent Learning via Early Experience. arXiv preprint arXiv:2510.08558, 2025. URL https://arxiv.org/abs/2510.08558.

Xuan Zhang, Wenxuan Zhang, See-Kiong Ng, and Yang Deng. Self-Evolving World Models for LLM Agent Planning. arXiv preprint arXiv:2606.30639, 2026. URL https://arxiv.org/abs/2606.30639.

Yuxin Zuo, Zikai Xiao, Li Sheng, Fei Huang, Jianhong Tu, Yuxuan Liu, Tianyi Tang, Xiaomeng Hu, et al. Qwen-AgentWorld: Language World Models for General Agents. arXiv preprint arXiv:2606.24597, 2026. URL https://arxiv.org/abs/2606.24597.

A Representation and Branch Construction

This appendix specifies the public representation and the operations needed to form comparable training branches. The specification distinguishes the agent's evidence from the simulator state used to validate restoration. The representation, branch construction, and scoring rules define the common interface used throughout training and evaluation.

A.1 Notation and information boundaries

Table 6 collects the symbols shared by the objectives and algorithm. The dynamics model, value head, and actor have different information requirements. A task goal determines which consequences are useful, while an environment transition can be described independently of that preference.

Table 6: Notation used in the framework. Simulator state is restricted to data collection and restoration checks.

Symbol	Meaning
s_t, o_t, a_t	Internal simulator state, public observation, and executed action
$h_t, z_t = E(h_t)$	Public interaction history and deterministic evidence ledger
g, b_t, x_t	Goal, remaining budget, and actor input (h_t, z_t, g, b_t)
d, R, δ_i	Canonical delta extractor, reconstructor, and real action consequence
$\mathcal{G}_t, C_t, B$	Verified root group, candidate action set, and candidate count
f_θ, π_ϕ	Separate dynamics and actor parameter copies
v_ψ	Held-out value head, frozen during successor replacement
$\pi_{\text{ref}}, Y, Q_i, \bar{Q}_i$	Reference continuation policy, normalized return, candidate value, and its estimate
$j_{\text{pred}}, j_{\text{real}}$	Choices obtained from predicted and real successor inputs
$\rho_t, w_t, \tilde{w}_t$	Positive repair estimate, raw weight, and batch-normalized weight
p, q^*, π_C	Reference prior, soft target, and actor distribution normalized on candidates
$m, \lambda_C, \kappa, \tau$	Contrast margin, contrast coefficient, repair coefficient, and policy temperature
$\epsilon_T, \epsilon_V, \epsilon_Q, \epsilon_d$	Transition, value, return-estimation, and distillation errors

Ledger updates require public evidence of an outcome. Position and possession fields are replaced only after corresponding environment feedback. Previously unknown object locations or attributes enter `reveals`; current visibility is tracked separately from the last known fact. If a new observation contradicts a stored single-valued fact, the latest explicit evidence updates that field. Evidence timestamps and internal tracking identifiers are audit metadata, not changing target fields. Unparsed text is retained in the history and raw-evidence field, avoiding the assumption that a parser captures all relevant observations.

The delta format preserves the distinction between changing and discovering a fact. The following schema example illustrates opening a cabinet whose contents were previously unknown.

```
action: open cabinet_1
changes:    cabinet_1.open = false -> true
reveals:    cup_1.location = cabinet_1
visibility: cup_1.visible = true
feedback:   status = succeeded
```

The cup's location is newly observed rather than asserted to have changed physically. A failed action retains the explicit failure feedback and any observed changes. Missing failure reasons remain unknown. An empty `changes` list is serialized explicitly, since an action can leave known physical fields unchanged while revealing information or advancing time.

A.2 Restoration, stochasticity, and split integrity

Restoration establishes equivalence of roots before actions are compared. A snapshot interface is preferred when its state, clock, and random-state restoration can be checked. Otherwise, collection resets the same task instance and replays the same action prefix. Public observations and available internal diagnostics are compared before each branch; inconsistent roots are logged rather than

treated as valid same-root contrasts. ScienceWorld time evolution makes this check relevant even when the public descriptions look identical.

Stochastic transitions require an explicit conditioning convention. If the initial action outcome is restored deterministically, repeated continuations estimate the value of that acquired successor under the shared reference policy. To estimate an action value that averages over stochastic first-step outcomes, collection must also resample those outcomes. A single difference between sampled outcomes is insufficient to establish distinct consequence distributions for hard contrastive negatives. Repeated verification, soft outcome labels, and exclusion from hard contrast provide alternative treatments of overlapping outcomes.

Data splits keep related evidence together. Entire sibling groups share a partition, and partitions follow source task instances or scenes. Goal-composition holdout retains primitive transition knowledge while excluding the target compositions from actor training and feedback. Action–object holdout removes the designated combinations from dynamics, expert, and feedback data. Transition-closure holdout additionally excludes the related primitive transitions. These protocols evaluate distinct forms of generalization.

A.3 Scoring, parsing, and model copies

Candidate scoring uses a value head trained on disjoint root groups. The value evaluator uses a frozen backbone's final-layer masked mean representation, encodes the goal, concatenates normalized remaining budget, and fits a two-layer MLP of width 256 with a sigmoid output and squared-error return supervision. Two-fold cross-fitting assigns each root a head not trained on it. The same head and input template are used for predicted and real successors, including the tie-breaking rule. This prevents changing evaluator parameters during the replacement comparison, although representation-dependent value errors can still affect the diagnostic.

Malformed predictions follow a fixed fallback rule. A successor that cannot be parsed is marked invalid. If every candidate successor is invalid, selection follows a predetermined reference-policy order. Full-state prediction controls are mapped to the same ledger representation before value evaluation. Format errors, semantic errors, and signed repair describe distinct aspects of the successor evaluation.

The actor has an explicit path for weighted dynamics supervision. CDD pretraining initializes f_θ; its weights initialize π_ϕ, which then receives task SFT. During a static VRA update, the world-model and actor copies are optimized separately. Their shared records contain real deltas, repair weights, and policy targets, but an updated world-model checkpoint is not automatically copied into the actor. The auxiliary CDD loss in Eq. (9) applies repair-weighted dynamics supervision to the actor parameters. Appendix D.11 specifies the B10, B11, A06, and A14 update paths.

B Candidate-Local Analysis

The analysis describes what the representation, repair signal, and candidate update establish under explicit assumptions. Throughout, continuation values refer to one fixed reference policy and bounded normalized returns. The results characterize a fixed candidate problem; they do not imply that the full learning procedure monotonically improves whole-task performance.

B.1 Reversible deltas preserve conditional information

Proposition 1 (Information preservation)**.** *Let H be public history, $Z = E(H)$, A an action, Z' a successor ledger, and $\Delta = d(Z, Z')$ a canonical delta. For every fixed h, assume $d(E(h), \cdot)$ and $R(E(h), \cdot)$ are inverse maps on the admissible successor set. If the relevant conditional entropies are finite, then*

$$H(Z' \mid H, A) = H(\Delta \mid H, A). \tag{11}$$

Proof. Fix $H = h$ and $A = a$. The current ledger $E(h)$ is determined, so the bijection $z' \leftrightarrow d(E(h), z')$ preserves every conditional probability. Entropy sums the same probabilities under a relabeling of outcomes. Averaging the resulting equality over (H, A) proves the claim. □

Information preservation concerns the observable ledger, not the full simulator state. For F fixed fields with at most V values each, a simple full encoding has length $O(F \log V)$. If k fields change, recording their indices and new values has length $O(k(\log F + \log V))$. Sparse changes can shorten explicit targets without implying that total training FLOPs decrease. The contrastive loss introduces additional forward and backward computation.

B.2 Action conditions contain learnable information

Proposition 2 (Ideal action-conditional information gain)**.** *For unrestricted discrete conditional models on a fixed data distribution,*

$$\inf_q \mathbb{E}[-\log q(\Delta \mid H)] - \inf_q \mathbb{E}[-\log q(\Delta \mid H, A)] = I(A; \Delta \mid H). \tag{12}$$

Proof. For any conditioning variable X, the cross-entropy decomposes as

$$\mathbb{E}[-\log q(\Delta \mid X)] = H(\Delta \mid X) + \mathbb{E}_X \, \mathrm{KL}(P(\Delta \mid X) \| q(\Delta \mid X)). \tag{13}$$

The infimum is attained by the true conditional distribution. Subtract the conditional entropies for $X = H$ and $X = (H, A)$. □

The identity motivates same-root action variation without establishing a finite-model guarantee. Since $I(A; \Delta \mid H) \leq H(A \mid H)$, near-deterministic action choice in demonstrations limits the available within-history contrast. Real sibling actions create such variation when their consequences are distinguishable. This result uses unnormalized sequence cross-entropy. CDD instead optimizes length-normalized likelihood with filtered pairs, so its practical loss is not asserted to equal or bound this mutual information.

B.3 Dynamics and value errors both affect selection

Proposition 3 (Candidate regret under approximation)**.** *Let u include all history, goal, and budget information required to define a continuation value. Let true and approximate successor distributions T and $\hat{T}$ share a state space. Assume $V \in [0, 1]$, and over a fixed candidate set C,*

$$\max_{a \in C} \mathrm{TV}(T(\cdot \mid u, a), \hat{T}(\cdot \mid u, a)) \leq \epsilon_T, \qquad \|V - \hat{V}\|_\infty \leq \epsilon_V. \tag{14}$$

The value bound must cover the support of both distributions. Define $Q(a) = \mathbb{E}_T V(U')$ and $\hat{Q}(a) = \mathbb{E}_{\hat{T}} \hat{V}(U')$. If $a^* = \arg\max_C Q$ and $\hat{a} = \arg\max_C \hat{Q}$, then

$$Q(a^*) - Q(\hat{a}) \leq 2(\epsilon_T + \epsilon_V). \tag{15}$$

Proof. Using $\mathrm{TV}(P, Q) = \sup_S |P(S) - Q(S)|$ and $0 \leq V \leq 1$,

$$|Q(a) - \hat{Q}(a)| \leq |\mathbb{E}_T V - \mathbb{E}_{\hat{T}} V| + |\mathbb{E}_{\hat{T}}(V - \hat{V})| \tag{16}$$

$$\leq \epsilon_T + \epsilon_V. \tag{17}$$

Insert $\hat{Q}(a^*)$ and $\hat{Q}(\hat{a})$ into $Q(a^*) - Q(\hat{a})$. The middle difference is nonpositive by the definition of $\hat{a}$, and the two remaining terms are bounded by the preceding inequality. □

Approximate scoring adds another error term. If the implemented score $\tilde{Q}$ satisfies $\max_C |\tilde{Q} - \hat{Q}| \leq \epsilon_{\mathrm{app}}$, the same proof gives $2(\epsilon_T + \epsilon_V + \epsilon_{\mathrm{app}})$. Single predicted successors or finite successor sampling are examples where this term matters. The proposition does not assume that the public ledger itself satisfies the required sufficient-state condition; this is why it is stated for the information-rich variable u.

B.4 What successor replacement measures

Proposition 4 (Repair identity and estimation error)**.** *Hold the candidate set, value evaluator, remaining inputs, and tie rule fixed. Define* j_{pred} *and* j_{real} *as in Eq. (5), and true candidate regret* $\mathcal{R}_j = \max_i Q_i - Q_j$*. Then*

$$\rho^{\mathrm{true}} = [\mathcal{R}_{j_{\mathrm{pred}}} - \mathcal{R}_{j_{\mathrm{real}}}]_+ = [Q_{j_{\mathrm{real}}} - Q_{j_{\mathrm{pred}}}]_+. \tag{18}$$

If $\max_i |\bar{Q}_i - Q_i| \le \epsilon_Q$*, then* $|\rho - \rho^{\mathrm{true}}| \le 2\epsilon_Q$*.*

Proof. The shared $\max_i Q_i$ term cancels when the regrets are subtracted. For the estimate, the positive-part map is 1-Lipschitz, so

$$|\rho - \rho^{\mathrm{true}}| \le |(\bar{Q}_{j_{\mathrm{real}}} - \bar{Q}_{j_{\mathrm{pred}}}) - (Q_{j_{\mathrm{real}}} - Q_{j_{\mathrm{pred}}})| \tag{19}$$

$$\le |\bar{Q}_{j_{\mathrm{real}}} - Q_{j_{\mathrm{real}}}| + |\bar{Q}_{j_{\mathrm{pred}}} - Q_{j_{\mathrm{pred}}}| \le 2\epsilon_Q. \tag{20}$$

□

The result holds for any fixed evaluator, including an inaccurate one. If every predicted successor equals its acquired real counterpart, both selections coincide and DRR is zero. The converse fails when errors preserve the ranking or the changed selection has no positive verified return gain. Moreover, taking a positive part of a noisy difference need not be unbiased. Signed repair, negative repair frequency, and independent-repeat stability therefore carry information that the nonnegative training weight alone discards.

B.5 Regularized targets and actor distillation

Proposition 5 (Local soft-target and actor improvement)**.** *Let* $p_i > 0$ *be a distribution on fixed candidates and* $\tau > 0$*. The optimizer of Eq. (8) is unique and obeys*

$$\sum_i (q_i^* - p_i)\bar{Q}_i \ge \tau\, \mathrm{KL}(q^* \| p). \tag{21}$$

If $\max_i |\bar{Q}_i - Q_i| \le \epsilon_Q$ *and* $\mathrm{TV}(\pi_C, q^*) \le \epsilon_d$*, then*

$$\sum_i (\pi_{C,i} - p_i) Q_i \ge \tau\, \mathrm{KL}(q^* \| p) - 2\epsilon_Q - \epsilon_d. \tag{22}$$

Proof. The objective is strictly concave on the candidate simplex. Adding a normalization multiplier and differentiating gives

$$\bar{Q}_i - \tau(\log(q_i/p_i) + 1) + \lambda = 0, \tag{23}$$

and hence $q_i^* \propto p_i \exp(\bar{Q}_i/\tau)$. Comparing the optimal objective with its value at feasible p yields the first inequality. Replacing $\bar{Q}$ by Q changes the difference by at most $\epsilon_Q \|q^* - p\|_1 \le 2\epsilon_Q$. Since $Q_i \in [0,1]$, replacing q^* by π_C changes expected value by at most $\mathrm{TV}(\pi_C, q^*) \le \epsilon_d$. Combining these statements proves the actor bound. □

Mass outside the candidate set and task distribution shift remain separate limitations. If $\beta = 1 - \pi(C)$, then the full policy's expected candidate-comparator gain is bounded below by the same expression minus β, using bounded returns. This compares against the candidate-restricted prior p, not necessarily the unrestricted reference policy. Pinsker's inequality can upper-bound ϵ_d by $\sqrt{\mathrm{KL}(\pi_C \| q^*)/2}$ when KL uses natural logarithms. An average empirical KL cannot substitute for a per-root uniform assumption.

B.6 Finite rollout counts

Proposition 6 (Fixed-candidate concentration)**.** *Condition on a predetermined simulator root, candidate set, and acquired action branches. Let* $Y_{i,r} \in [0,1]$ *be independent across repeats* r *for each*

candidate under a fixed continuation policy, with conditional mean Q_i. With probability at least $1-\alpha$,

$$\max_{i\le B} |\bar{Q}_i - Q_i| \le r_{B,M,\alpha} = \sqrt{\frac{\log(2B/\alpha)}{2M}}. \tag{24}$$

Consequently, the repair error is at most $2r_{B,M,\alpha}$ and the actor gain is at least $\tau\,\mathrm{KL}(q^*\|p) - 2r_{B,M,\alpha} - \epsilon_d$ on the same event.

Proof. Hoeffding's inequality bounds one candidate's estimation error probability by $2\exp(-2M\epsilon^2)$. A union bound over B candidates gives $2B\exp(-2M\epsilon^2)$; solve for ϵ at probability α and substitute into the preceding propositions. Independence between different candidates is not required. □

The general bound is loose at the default rollout budget. For $B = 4$, $M = 8$, and $\alpha = 0.05$, the radius is approximately 0.563, so $2r \approx 1.126$. Thus this distribution-free result does not certify the small empirical gains. For N predetermined roots, replacing $2B$ by $2BN$ provides simultaneous coverage, with an even looser bound. Independent-repeat diagnostics assess estimation stability at the finite rollout budget.

C Scope, Reproducibility, and Responsible Use

The current study concerns text simulators with recoverable training states. Restoration and repeated counterfactual execution should not be assumed available in websites, physical systems, or user accounts where actions are irreversible. A deployment extension would need a separate acquisition and verification protocol rather than simply reusing the simulator branching assumption.

Interpretation of the model and decision diagnostics. The ledger summarizes observable evidence and is not assumed to be a sufficient Markov state. CDD encourages action dependence, while generalization is evaluated empirically. DRR measures positive return changes caused by successor replacement under the chosen evaluator, candidate set, and reference continuation. It is not an evaluator-independent decomposition of dynamics error. Zero DRR can coexist with inaccurate successors, and imperfect evaluation can produce negative repair. The policy bound is local to the candidate set; its finite-feedback lower bound can be nonpositive. Actor gains on that set establish a behavioral diagnostic rather than evidence of an internal planning algorithm.

Empirical scope and resource interpretation. The experiments compare descriptive task means in two text environments. Pairwise significance requires matched task outcomes and the corresponding statistical analysis. The weighted full-state control is stronger on ALFWorld at equal FLOPs, and strict transfer and observation perturbations include reversals against uniform feedback. Resource reporting includes token-based FLOP accounting, training-chain time, and memory use for the specified configurations.

The unit of reproducibility is a complete root group with its continuation records. Environment version, restoration procedure, parser version, split membership, candidate settings, cross-fitting assignment, rollout seeds, and termination rules determine the experiment configuration. Appendix D.10 defines aggregation and the scope of the auxiliary comparisons.

D Additional experimental details and results

This appendix describes the environment interface, training configuration, extended baselines, and mechanism diagnostics. It then examines generalization, feedback allocation, model scale, and resource use. The concluding subsections define statistical aggregation, comparison conditions, and world-model and actor update paths.

D.1 Environment, data, and optimization configuration

Only public interaction evidence enters the model input. The ledger records location, inventory, visible entities, containment, device state, and observed treatment properties. ALFWorld properties

include cleaning, heating, and cooling. ScienceWorld properties include publicly observed temperature, phase, and device feedback. Unknown fields remain unknown, newly observed facts are recorded as reveals, and updates to known facts are recorded as changes. Unstructured feedback retains its original evidence text. Issuing an action alone does not establish that a physical change occurred. The dynamics input excludes explicit task goals and goal-reward text, whereas the policy and value evaluator retain the goal.

Sibling transitions are collected from a common environment root. Sibling collection uses state restoration or a task reset followed by replay of a common prefix. The dynamics pool contains 20,000 roots with four branches each and 20,000 additional single-branch transitions. Splits preserve original task instances and complete sibling groups across dynamics, demonstration, feedback, and online data. Equivalent-consequence actions are excluded from hard-negative pairs. Static VRA uses 1,000 training roots and deduplicated candidate actions. Eight independently seeded continuations use the same reference policy and remaining action budget for each candidate. Candidate priors normalize the sequence likelihood of the complete command, including its end token.

Table 7: Default configuration. One setting applies to both environments unless separate values are shown. Statistical definitions appear in Appendix D.10.

Configuration	Value
Base models	Qwen3-4B; Qwen3-8B/14B/32B and Gemma 4 12B extensions
Training seeds / data seed	17, 29, 43 / 1566
Training precision / global batch	bf16 / 128 branches, retaining sibling groups
Dynamics pool / demonstrations	100,000 transitions / 200 successful trajectories per environment
Static feedback	1,000 roots, $B = 4$, $M = 8$, 32,000 continuations
Learning rates	Dynamics 10^{-5}; policy 5×10^{-6}; GRPO 10^{-6}
Training passes	Dynamics 2; task SFT at most 3; static VRA 2
Checkpoint selection	Common development-set rule
Maximum input tokens	ALFWorld 4,096; ScienceWorld 8,192
Maximum output tokens	Dynamics 512; actor 128
Contrastive objective	$\lambda_C = 0.1$, margin $m = 0.2$
Feedback and policy coefficients	$\kappa = 2$, $\tau = 0.2$, $\lambda_{\mathrm{BC}} = 0.1$, $\lambda_{\mathrm{keep}} = 0.1$
Replay	At least 30% of dynamics samples from the fixed old pool
Reference rollout decoding	Temperature 0.7, top-$p = 0.95$
GRPO rollout / reference KL	8 trajectories per task group / 0.01
Downstream RL environment steps	ALFWorld 2 million; ScienceWorld 6 million
Evaluation action limits	ALFWorld 40; ScienceWorld 100
Evaluation decoding and tools	Greedy actor; no legal-action enumeration or world-model calls

Training labels and evaluation scores use different ScienceWorld scales. The value label clips the final ScienceWorld score to $[0, 100]$ and divides by 100; the main evaluation preserves final raw scores, including negative termination. ALFWorld labels are binary final success. ScienceWorld macro score first averages within each task type and then averages the 30 types equally. Success@100 counts episodes ending at 100. ALFWorld's main unseen metric uses the specified 134 instances per seed. Seen-split and family-macro results are reported as separate ALFWorld metrics.

Static feedback is separated from online interaction. B10, B11, B15, and B16 share fixed roots, candidate actions, the reference policy, cached returns, and policy soft targets. Their dynamics weights are uniform, DRR, NLL, and total candidate regret, respectively. B12 adds two rounds of training-environment sampling from updated policies. B13 uses the same additional online interaction budget with verified AWR/RFT. Static passes over an existing cache do not count as new online rounds. Test parameters remain fixed and memory is cleared between tasks.

D.2 Complete direct-actor and downstream results

Complete direct-actor comparisons. Table 8 includes the raw-observation baseline, trajectory continuation training, adapted world-model objectives, and both static and online feedback. Raw-trajectory CPT does not improve ledger SFT in these reported means. The weighted-full-state baseline is competitive with CDD on ScienceWorld before feedback.

Table 8: Complete B-series outcomes. ALF is unseen SR in percent; Science is type-macro score; S@100 is ScienceWorld success@100 in percent. Online rows use additional interaction and are separated from static rows. Adapted labels denote objectives implemented under the common protocol, not published benchmark numbers.

ID	Method	ALF	Science	S@100
B00	Task SFT, raw observations	67.91	60.73	42.83
B01	Task SFT, shared ledger	70.15	62.83	43.83
B02	Raw-trajectory CPT → SFT	69.65	62.33	43.33
B03	Full-state dynamics → SFT	75.12	65.53	44.43
B04	Delta-only → SFT	76.62	65.93	44.83
B05	Weighted full-state → SFT	78.11	67.73	45.93
B06	AEE-IWM adapted → SFT	78.86	68.53	46.33
B07	AEE-SR adapted → SFT	76.87	64.63	42.03
B08	CDD → SFT	80.35	68.13	46.73
B09	Full-state dynamics + VRA	81.09	69.33	47.83
B10	CDD + uniform feedback	84.83	70.93	49.23
B11	Action2Delta, static	86.57	72.73	50.73
B14	Next-observation modeling → SFT	75.87	66.53	45.33
B15	CDD + NLL-priority feedback	86.07	71.83	49.83
B16	CDD + total-regret feedback	84.33	70.53	48.83
B17	QAW-style observation adaptation	81.34	68.83	46.53
B12	Action2Delta, two online rounds	88.56	74.83	51.93
B13	Matched online verified AWR/RFT	88.06	74.13	51.53

Downstream improvements depend on the optimizer and environment. Table 9 compares GRPO initializations and GiGPO integration. R00–R06 share downstream interaction budgets; R07 instead extends task-SFT RL to account for the additional upstream budget. Action2Delta improves the shared-budget SFT initialization and remains above R07 on the primary means, but State2State is higher on ALFWorld under GRPO. The GiGPO comparison supports compatibility with step-level credit assignment.

Table 9: Downstream policy optimization. External objectives are adapted to the common interface. R00–R06 share a downstream environment-step budget. R07 extends SFT-based RL and is interpreted using the analytical cumulative-cost accounting.

ID	Route	ALF SR	Science	S@100
R00	Task SFT → GRPO	85.82	73.41	52.17
R01	Full-state → SFT → GRPO	88.31	74.91	53.17
R02	AEE-IWM → SFT → GRPO	90.05	75.81	53.57
R03	State2State → GRPO	93.28	75.61	54.77
R04	RWML adapted → GRPO	91.54	75.41	53.37
R05	SPA adapted → GRPO	89.05	74.21	52.57
R06	Action2Delta → GRPO	93.03	76.81	54.27
R07	Task SFT → longer GRPO	92.04	76.21	54.07
R08a	Task SFT → GiGPO	92.79	75.73	54.43
R08b	Action2Delta → GiGPO	94.78	77.23	55.17

D.3 Dynamics diagnostics and ablations

Generation quality and action discrimination measure different properties. All complete-state outputs are converted to the common delta schema before evaluation. Change F1 is a sample average over nonempty changes, while Reveal F1 measures newly observed facts. Exact match compares the complete normalized delta. No-change accuracy and unchanged-field preservation address empty-change cases and persistent facts. The likelihood gap is length-normalized in nats per token. Table 10 supplements the main pair-accuracy and regret results with these generation diagnostics. VRA's lower regret does not require the best score on each generation metric.

Table 10: Additional consequence-model diagnostics. All values except the likelihood gap are percentages. Preserve measures unchanged-field preservation. F1 follows the nonempty-sample convention, not global micro averaging.

Env.	Method	Reveal F1	EM	No-change	Preserve	Gap	Format
ALF	Full-state	79.24	64.18	88.73	96.84	.071	99.12
	Delta-only	80.05	66.93	89.31	97.10	.083	99.38
	Weighted full-state	81.38	70.16	90.06	97.63	.098	99.25
	CDD	80.92	69.83	91.74	97.42	.168	99.41
	CDD + uniform	81.47	70.54	92.10	97.89	.176	99.48
	CDD + NLL	81.63	71.01	92.24	97.96	.181	99.43
	CDD + total-regret	81.08	70.02	91.86	97.52	.170	99.39
	CDD + VRA	81.19	70.28	92.03	97.71	.174	99.46
Science	Full-state	64.31	43.87	80.46	92.73	.052	98.63
	Delta-only	65.08	45.32	81.15	93.18	.061	98.87
	Weighted full-state	66.47	48.59	82.34	94.05	.079	98.81
	CDD	65.94	48.02	85.13	93.62	.127	99.04
	CDD + uniform	66.58	48.81	85.77	94.18	.134	99.11
	CDD + NLL	66.73	49.19	85.96	94.22	.138	99.08
	CDD + total-regret	65.83	48.36	85.18	93.73	.130	98.99
	CDD + VRA	66.21	48.43	85.41	93.96	.132	99.07

Ablations of supervision and training design. Table 11 compares supervision, representation, training order, and online updating. Appendix D.11 defines world-only weighting, and Appendix D.10 discusses weight-distribution controls. Removing action contrast (A01), shuffling action–consequence assignments (A02), and replacing the ledger by the last observation (A09) reduce both primary means. A03 shows that a matched-compute weighted full-state objective can outperform delta training on ALFWorld. Adding an inverse-dynamics task (A12) and supplying the explicit goal (A10) yield slightly higher primary means than the default configuration.

Table 11: Reported ablations. Static comparisons preserve the corresponding data and feedback unless the row explicitly changes them. A11 conditions have no VRA. Online conditions have two extra optimization or data rounds. Weight-distribution controls appear in Table 24; Appendix D.11 defines the static update-path control.

ID	Condition	ALF SR	Science	S@100
B11	Complete static method	86.57	72.73	50.73
A01	Remove action contrast, retain VRA	85.32	71.53	50.03
A02	Shuffle action–consequence assignments	77.86	66.43	44.43
A03	Weighted full-state + VRA, matched FLOPs	87.06	71.93	49.93
A04/B10	Uniform feedback weights	84.83	70.93	49.23
A05/B15	NLL feedback weights	86.07	71.83	49.83
A05/B16	Total-regret feedback weights	84.33	70.53	48.83
A06	Remove actor dynamics retention	85.82	71.03	49.53
A07	Keep equivalent outcomes as negatives	84.08	70.63	48.93
A08	Merge reveals and changes	85.57	71.41	49.97
A09	Last observation instead of ledger	79.85	66.07	44.57
A10	Include explicit dynamics goal	87.06	73.12	51.07
A12	Add multi-positive inverse dynamics	86.82	73.06	51.03
A13	Train on successful experience only	86.82	70.03	48.23
A16	DPO instead of return soft-BC	86.07	72.39	50.17
A11/B08	CDD $\rightarrow$ SFT	80.35	68.13	46.73
A11 reverse	SFT $\rightarrow$ CDD	75.87	64.81	42.93
A11 joint	Joint CDD and SFT	79.35	67.59	46.23
A15 static	Two additional fixed-data rounds	87.31	73.17	51.03
A15/B12	Two new-data online rounds	88.56	74.83	51.93
A17 frozen	Online with first-round weights frozen	88.31	74.19	51.47

Semantic and recovery diagnostics. Merging reveals and changes raises the rate of newly discovered facts mislabeled as physical changes from 2.17%/5.39% to 7.23%/12.71% on ALFWorld/ScienceWorld. Explicit-goal dynamics raises goal-property hallucination from 2.61%/5.17% to 5.43%/9.21% on the counterfactual-goal audit, despite its higher task means. Successful-only training reaches 60.21%/55.39 on recovery, below mixed-success/failure training at 66.27%/61.17.

Training order affects the executable action policy. CDD followed by SFT yields 80.35%/68.13, compared with 75.87%/64.81 for the reverse order and 79.35%/67.59 for joint training. The corresponding actor-format validity is 99.62%/99.07%, 94.71%/91.83%, and 99.14%/98.51%. The format-validity comparison supports using task SFT to restore the action interface after dynamics learning.

D.4 Successor replacement and verification stability

Successor replacement fixes the candidate set and evaluator. The diagnostic compares actions selected from predicted successors and real successors using the same value evaluator, goal, and remaining budget. Real continuation returns then evaluate those selections. Candidate regret is the gap from the best measured candidate return. The candidate-return oracle has zero regret by definition and is not an executable evaluation policy. A degraded evaluator increases regret to .264/.311, indicating sensitivity to evaluator quality.

Table 12: Replacement outcomes and independent verification. The signed repair D is the real-return difference between the real-successor-selected and predicted-successor-selected actions. Its positive part, rather than its signed mean, is used by DRR. The stability panel is ScienceWorld only.

Replacement diagnostic	ALFWorld	ScienceWorld
Predicted successor regret	.115	.157
Real successor regret, same evaluator	.091	.126
Agreement with best candidate, predicted (%)	71.83	64.71
Agreement with best candidate, real (%)	77.46	71.38
Positive / negative / zero repair (%)	18.40 / 6.30 / 75.30	23.10 / 8.70 / 68.20
Mean D among positive repairs	.1616	.1786
Mean D among negative repairs	$-.0910$	$-.1180$
Mean signed repair $\mathbb{E}[D]$	.0240	.0310
Mean DRR $\mathbb{E}[\max(D, 0)]$	.0297	.0413

M	Q MAE	DRR rank Spearman	Positive-sign repeat (%)	Science score
4	.094	.42	68.73	71.64
8	.067	.58	76.41	72.73
16	.049	.69	82.17	73.16

Additional continuations improve the reliability of the signal. Table 12 reports decreasing Q error and increasing independent rank and sign agreement as M grows. At the default $M = 8$, positive-sign agreement is 76.41%, so verified gains still have sampling uncertainty. A zero DRR may result from an unchanged selected action, an evaluator failure, a nonpositive replacement gain, or an inaccurate consequence irrelevant to the current decision. It does not certify a correct dynamics model. Independent continuation repeats assess signal stability at each rollout budget.

D.5 Generalization, transfer, and adaptation

The generalization protocols vary what is held out. G02 removes goal compositions while retaining atomic transition experience. G03 removes action–object combinations and retrains on the isolated training set. G04 buckets tasks by a fixed teacher or official reference-path length within each task family. G05 starts each method at the same adverse state, reached by a shared policy and an action verified to be recoverable; the remaining budget is identical, and no rollback tool is added at test time. G06 consistently substitutes entity aliases in observations, goals, and commands while retaining category information. G07 appends true but irrelevant public facts. G08 masks noncritical

observation fragments under a common rule while checking solvability. Table 13 consolidates these outcomes.

Table 13: Full principal generalization comparisons. Each cell gives ALF success / Science macro score; ALF success is recovery success in G05. Evaluation sets differ between rows and do not establish one common difficulty scale. B00 uses raw observations, while subsequent methods use the ledger.

Protocol	B00	Delta-only	CDD	Uniform	VRA
G02, goal composition	54.19/48.37	60.73/52.94	65.82/56.73	68.47/59.41	71.27/61.51
G03, action–object	42.73/43.91	49.81/47.12	53.39/49.47	55.67/50.63	56.17/50.93
G04, long tasks	48.31/46.18	54.29/50.93	58.17/54.73	62.43/57.21	64.73/59.01
G05, recovery	45.71/44.63	52.93/49.81	58.41/54.39	63.17/58.27	66.27/61.17
G06, entity aliases	63.83/57.47	70.17/61.81	72.93/63.27	78.11/66.39	77.51/66.09
G07, irrelevant facts	62.47/55.73	70.33/61.93	73.67/64.53	79.27/67.47	80.17/68.07
G08, 20% masking	54.13/49.37	61.67/55.13	65.31/57.27	72.47/61.43	73.07/61.91

Transition closure and observation quality delimit the gains. The ScienceWorld type holdout uses five separately retrained folds over phase changes, circuits, growth, mixing, and genetics. One variant retains relevant atomic transitions; the other also removes their related transition closure. Table 14 shows that VRA falls slightly below uniform feedback in the latter. Its margin over uniform feedback also shrinks with masking and becomes negative at 40%. At 100% additional irrelevant text, VRA's Science score is 61.97 versus 62.13 for uniform feedback.

Table 14: ScienceWorld type holdout and observation-masking boundaries. Holdout cells give macro score / success@100; five folds are averaged equally. Masking rates apply to the shared observation perturbation rule.

Holdout	Delta-only	CDD	Uniform	VRA
Goal type only	47.83/23.17	51.07/25.73	54.39/28.13	55.87/29.07
Related transition closure	37.91/15.63	38.73/16.03	40.17/17.33	40.03/17.07

Masking	Uniform ALF	VRA ALF	Uniform Science	VRA Science
0%	84.83	86.57	70.93	72.73
10%	78.37	79.63	66.81	67.63
20%	72.47	73.07	61.43	61.91
40%	58.73	57.97	51.17	50.83

Cross-environment transfer tests CDD initialization. Source-environment CPT, task SFT, and CDD are each followed by independent target adaptation using the same 0, 50, or 200 target trajectories and update budget. Zero-target conditions receive the public command specification. Source CDD improves the reported primary means in both directions and all three data settings (Table 15). Because this experiment varies source CDD without a separate source-VRA factor, it supports CDD transfer and cannot identify a VRA-specific transfer effect.

Table 15: Bidirectional environment transfer. ALFWorld to ScienceWorld reports target macro score; ScienceWorld to ALFWorld reports target SR. Columns are numbers of target demonstration trajectories.

	ALFWorld → ScienceWorld			ScienceWorld → ALFWorld		
Source training	0	50	200	0	50	200
None	28.37	48.17	60.73	31.34	54.23	67.91
Trajectory CPT	28.03	48.39	61.09	32.09	54.61	68.16
Task SFT	27.41	47.73	60.13	30.85	53.77	67.41
CDD	30.13	50.63	62.47	35.07	57.83	70.15

Adaptation to modified simulator rules. G10 changes action preconditions in ALFWorld and equipment or material conditions in ScienceWorld, with corresponding interface descriptions supplied to all methods. Training interactions come from the modified simulator. Table 16 shows no zero-feedback advantage for VRA but an advantage after collecting new-rule feedback. The comparison measures adaptation using feedback from the modified transition rules.

Table 16: Adaptation to changed simulator rules using real new-rule feedback. Root counts refer to training roots under the modified rules, not test-time branches.

New-rule roots	Uniform ALF	VRA ALF	Uniform Science	VRA Science
0	47.31	46.87	43.77	43.41
100	56.13	56.87	48.93	49.37
500	68.47	70.63	57.19	59.07
2,000	77.83	80.17	65.73	67.61

D.6 Sensitivity, feedback allocation, and model scale

The default parameters balance improvement and feedback coverage. The one-factor sweeps in Table 17 preserve the other default settings. More dynamics data and additional continuations improve the reported means, with smaller increments at the upper end. Increasing the contrastive coefficient or DRR coefficient beyond the default does not continue to improve performance. Sampling three successors for value averaging changes training and audit inference only; it does not add test-time world-model calls to the main actor.

Table 17: Reported one-factor sensitivity analysis. The B sweep fixes 1,000 feedback roots and therefore varies total continuations. The M sweep likewise changes feedback cost. Default settings are 100k transitions, $B = 4$, $M = 8$, $\lambda_C = .1$, $\kappa = 2$, $\tau = .2$, and zero new online rounds.

Factor	Value	ALF SR	Science	S@100
Dynamics transitions	10,000	81.59	69.17	47.93
	30,000	84.33	71.49	49.63
	100,000	86.57	72.73	50.73
Candidate count B	2	85.07	71.62	50.03
	4	86.57	72.73	50.73
	8	87.06	73.24	51.17
Continuations M	4	85.32	71.64	49.77
	8	86.57	72.73	50.73
	16	86.82	73.16	51.03
Contrastive coefficient λ_C	0	85.32	71.53	50.03
	.05	86.32	72.21	50.43
	.10	86.57	72.73	50.73
	.20	85.82	72.08	50.07
DRR coefficient κ	0	84.83	70.93	49.23
	1	86.07	72.26	50.43
	2	86.57	72.73	50.73
	4	85.82	71.91	49.93
Policy temperature τ	.1	86.07	72.19	50.23
	.2	86.57	72.73	50.73
	.5	85.57	71.83	49.83
New online rounds	0	86.57	72.73	50.73
	1	87.81	74.09	51.63
	2	88.56	74.83	51.93
Successor decoding	Greedy	86.57	72.73	50.73
	Three samples, value average	86.82	73.21	51.07

Fixed continuation counts expose a root-coverage tradeoff. Table 18 holds roots $\times B \times M = 32,000$. Raising B at fixed roots is beneficial in the previous table, but raising it while reducing the number of roots is not. The default has the highest reported means among the five tested allocations.

Equal continuation counts do not imply equal simulator steps, prefix replay, tokens, or wall-clock time, because branch lengths vary.

Table 18: Feedback allocation under 32,000 continuations. This controls the number of continuations, not total compute or real environment steps.

B	M	Roots	ALF SR	Science	S@100
2	8	2,000	86.32	72.31	50.37
4	8	1,000	86.57	72.73	50.73
8	8	500	85.57	71.79	49.83
4	4	2,000	86.32	72.07	50.13
4	16	500	86.07	72.39	50.47

The 8B extension preserves smaller primary-metric margins. Table 19 reports the complete scale extension. Static VRA exceeds uniform feedback by .75 ALFWorld percentage points and .50 ScienceWorld score points, compared with 1.74/1.80 at 4B. ScienceWorld success@100 is lower by .20 points, and the online matched-budget success@100 comparison is also slightly lower. Under GRPO, the State2State route has higher ALFWorld success and ScienceWorld success@100, while Action2Delta has higher ScienceWorld macro score.

Table 19: Complete Qwen3-8B extension under the reported common environment protocol. Static, online, and downstream RL conditions use different training budgets and are separated.

8B training condition	ALF SR	Science	S@100
Task SFT, raw observations	88.06	64.47	47.13
Task SFT, ledger	88.81	66.13	47.83
Full-state → SFT	89.30	68.41	48.53
Delta-only → SFT	90.05	68.87	48.83
Weighted full-state → SFT	91.29	70.63	49.63
AEE-IWM → SFT	90.80	70.11	49.07
CDD → SFT	91.54	70.39	49.97
CDD + uniform feedback	93.03	74.17	53.03
Action2Delta, static	93.78	74.67	52.83
Action2Delta, online	94.78	75.81	53.67
Matched online AWR	94.53	75.57	53.73
Task SFT → GRPO	93.03	75.73	54.13
State2State → GRPO	97.51	78.17	56.13
Action2Delta → GRPO	96.77	78.81	55.83

D.7 Generalization across model scales and families

The cross-backbone study separates two questions: how the training effects change with model size within Qwen3, and how they transfer to another model family. Table 20 extends the Qwen3-4B/8B comparison to Qwen3-14B and Qwen3-32B, with Gemma 4 12B (Google DeepMind, 2026) as the cross-family condition. The instruction-tuned checkpoints are `Qwen/Qwen3-14B`, `Qwen/Qwen3-32B`, and `google/gemma-4-12B-it`. All inputs are textual, and each model uses its native chat template with thinking disabled.

Action contrast and feedback improve the larger backbones. CDD improves over delta-only by 1.00/.50/.75 ALFWorld percentage points and 1.4/1.1/1.1 ScienceWorld points on Qwen3-14B, Qwen3-32B, and Gemma 4 12B, respectively. Adding uniform feedback yields a further 1.00/.75/1.00 ALFWorld points and 3.8/4.0/2.9 ScienceWorld points. Relative to uniform replay, DRR adds two, one, and two ALFWorld successes out of 402; its ScienceWorld increments are .3, .0, and .2. NLL and DRR coincide at the displayed ScienceWorld precision on Qwen3-32B. These within-backbone comparisons distinguish the gains from consequence discrimination, real feedback, and replay allocation.

Table 20: Complete cross-backbone comparisons. B01: ledger SFT; B04: delta-only; B08: CDD; B10: uniform feedback; B15: NLL-priority feedback; B11: static Action2Delta. ALF success counts total three seeds, with 134 instances per seed; SR is $100K/402$. Science denotes type-macro score.

Backbone	ID	ALF successes	ALF SR	Science
Qwen3-14B	B01	371	92.29	67.4
	B04	374	93.03	70.0
	B08	378	94.03	71.4
	B10	382	95.02	75.2
	B15	383	95.27	75.4
	B11	384	95.52	75.5
Qwen3-32B	B01	384	95.52	69.1
	B04	385	95.77	71.6
	B08	387	96.27	72.7
	B10	390	97.01	76.7
	B15	390	97.01	76.7
	B11	391	97.26	76.7
Gemma 4 12B	B01	368	91.54	67.0
	B04	370	92.04	69.0
	B08	373	92.79	70.1
	B10	377	93.78	73.0
	B15	378	94.03	73.1
	B11	379	94.28	73.2

Matched training routes compare the training objectives. For each backbone, the protocol compares ledger SFT (B01), delta-only mid-training (B04), CDD (B08), CDD with uniform feedback (B10), NLL-priority feedback (B15), and DRR feedback (B11). B08 versus B04 tests action contrast; B10 versus B08 tests access to real feedback; B11 versus B10 and B15 tests the feedback-allocation criterion. Weighting applies to both the world-model and actor dynamics objectives; policy soft targets and behavior-cloning terms remain shared. For this extension, B15 uses raw weights $1 + \kappa L_t/(1 + L_t)$, where L_t averages length-normalized delta NLL over the root's real branches under the fixed pre-feedback world model. Each active parameter path normalizes weights within its minibatch. The primary effects are within-backbone differences in ALFWorld success rate and ScienceWorld type-macro score.

Shared evidence and evaluation define each comparison. The protocol fixes the task splits, public ledger, 100,000 dynamics transitions, 200 demonstrations, and training seeds $\{17, 29, 43\}$. It uses full-parameter bf16 training, the coefficients in Table 7, and common checkpoint selection on development data. Within each backbone, environment, and seed, B10, B15, and B11 share the CDD/SFT initialization, 1,000 roots, four candidates per root, eight continuations per candidate, reference policy, value-head folds, and policy soft targets. Each backbone receives its own reference-policy return cache. Native tokenization is retained, with a shared public-history selection rule and recorded input truncation.

Evaluation uses a 40-action limit on the 134 ALFWorld unseen instances and a 100-action limit on the ScienceWorld test list covering all 30 task types, with one evaluation per instance and training seed. The direct actor uses greedy decoding without auxiliary model calls. Model identity forms part of the evaluation-group identifier. The table reports aggregate means and total ALFWorld successes using the definitions in Appendix D.10. The paired evaluation protocol retains shared task identities and records model and tokenizer revisions, cache hashes, sequence lengths, and resource use.

D.8 Cost accounting and data quality

Training and collection costs include contrastive learning and rollouts. The accounting uses $6PT$ for gradient training and $2PT$ for inference, where P is parameter count and T is the number of nonpadding-equivalent tokens. CDD includes two wrong-action gradient branches on 80% of sibling transitions. Table 21 reports the corresponding costs, including dynamics acquisition, prefix

replay, and real continuations. Shared expert-data generation, development evaluation, and final testing are outside these totals. The token-based FLOP accounting omits additional attention, I/O, and scheduling overhead.

Table 21: Analytical training and collection costs. Tokens and environment steps are in millions; FLOPs are in 10^{18}. Feedback inference is separated from gradient training. Zero collection in B01 excludes shared demonstration generation.

Env.	Method	Train tokens	Train FLOPs	Env. steps	Rollout tokens	Rollout FLOPs
ALF	B01	13.603	.326	.000	.000	.000
	B03	322.595	7.742	1.100	.000	.000
	B04	256.195	6.149	1.100	.000	.000
	B05	322.595	7.742	1.100	.000	.000
	B08	645.955	15.503	1.100	.000	.000
	B10	681.659	16.360	2.156	839.680	6.717
	B11	683.087	16.394	2.156	839.680	6.717
Science	B01	76.166	1.828	.000	.000	.000
	B03	626.524	15.037	2.300	.000	.000
	B04	552.924	13.270	2.300	.000	.000
	B05	626.524	15.037	2.300	.000	.000
	B08	1324.764	31.794	2.300	.000	.000
	B10	1393.900	33.454	4.764	4012.032	32.096
	B11	1396.665	33.520	4.764	4012.032	32.096

Sequence lengths and replay define the accounting model. Mean demonstration lengths are 16/45 steps on ALFWorld/ScienceWorld, with SFT inputs of 1,300/2,600 tokens. Dynamics inputs are 1,150/2,300 tokens, full successors 400/480 tokens, and deltas 68/112 tokens. Dynamics prefixes average 10/22 steps, VRA prefixes 12/28, and continuations 20/48. The VRA accounting specifies 8,000 branch executions across its training passes, old-data replay equivalent to 10% of warm-up tokens, and an additional 4% compute allowance for B11. The static cache contains 4,000 unique first-step branches; training-pass counts and unique environment executions are distinct accounting quantities. The tabulated collection costs follow the stated analytical branch-count convention.

Training time and memory use. B11 training-chain time is 88.91 minutes on ALFWorld and 279.27 minutes on ScienceWorld. The totals cover gradient training, rollout inference, environment execution, and orchestration. Per-device training memory is 21.45/34.90 GiB for 4B/8B with eight-way state sharding, 16 bytes per parameter for parameters, gradients, and optimizer states, 12/18 GiB for activations, and a 2 GiB buffer. A single bf16 inference copy uses 17.45/24.90 GiB for 4B/8B, including an 8 GiB KV cache and a 2 GiB buffer.

Interaction budgets must include upstream training. The reported cumulative accounting starts Action2Delta RL at 2.156/4.764 million environment steps and 23.112/65.616 exaFLOPs on ALFWorld/ScienceWorld. Adding the full downstream budget reaches 4.156/10.764 million steps and 63.112/185.616 exaFLOPs. Therefore a downstream learning curve alone does not establish total sample efficiency. The cumulative and stage tables use different SFT offsets, .302/1.693 and .326/1.828 exaFLOPs respectively. Comparisons within either table use that table's stated accounting baseline.

Data audits expose the limits of the public representation. Table 22 reports replay consistency, parser quality, and useful contrast density. Only 71.39%/63.17% of sibling roots have at least two distinguishable consequences, and equivalent-consequence pairs occur in 18.27%/26.43% of candidate pairs. Parser errors and incomplete public observations can still merge distinct underlying states. The accompanying annotation sets and denominators define the sampling units for these diagnostics.

Table 22: Data-quality and coverage diagnostics in percent. The row descriptions identify the population used for each ratio.

Audit quantity	ALFWorld	ScienceWorld
Root–action samples with explicit physical changes	62.31	48.27
Samples with both changes and reveals empty	21.43	28.63
Parser coverage of expressible public fields	94.17	84.39
Incorrect fields among extracted fields	1.37	3.81
Repeated replay consistency	99.73	98.91
Sibling roots with at least two distinct consequences	71.39	63.17
Equivalent-consequence candidate pairs	18.27	26.43
Targets exceeding 512 tokens	.17	2.31
Initial environment failure or crash	.13	.71
Episodes still missing after the fixed retry policy	.03	.19

D.9 Actor audit and deployment interpretation

Independent candidate returns test behavioral transfer to the actor. Table 23 separates value estimation error, target movement, distillation error, and the realized gain on an independently audited candidate set. The actor achieves candidate-return gains of .028/.033, compared with .041/.052 for the soft target. Mean errors and mean KL values are descriptive diagnostics; they cannot replace the uniform per-root errors required by a local policy-improvement bound. They also do not establish the actor's internal reasoning mechanism.

Table 23: Reported local policy diagnostics. Returns are normalized to $[0, 1]$; KL uses natural logarithms. Gains are relative to the same candidate prior on an independent continuation audit. These are local candidate-set quantities, not full-episode performance guarantees.

Diagnostic	ALFWorld	ScienceWorld
Candidate Q MAE, $M = 8$	.043	.067
95th percentile of absolute Q error	.112	.173
Mean KL of soft target relative to candidate prior	.086	.113
Mean KL of actor relative to soft target	.019	.031
Independent soft-target candidate-return gain	.041	.052
Independent actor candidate-return gain	.028	.033

Direct execution and world-model lookahead. The main actor makes zero auxiliary world-model and slow-policy calls by design. Adding depth-two lookahead using the trained world model raises B11 to 88.31% ALFWorld success and 74.27 ScienceWorld score, at 46.13/54.71 additional world-model calls per episode. The lookahead extension uses an additional inference budget. The full comparison in Table 40 uses ScienceWorld type-macro score and success@100. The ALFWorld direct-actor and own-model comparison uses the same action budget.

D.10 Statistical aggregation and comparison scope

Each evaluation condition has a fixed denominator. ALFWorld uses 134 unseen task instances per training seed and the seeds $\{17, 29, 43\}$. For one evaluation per instance, $N_s = 134$ and the seed rate is $r_s = 100K_s/134$. The primary mean is $\bar{r} = \frac{1}{3}\sum_s r_s = 100\sum_s K_s/402$. Cross-seed variation is the sample standard deviation $\sqrt{\sum_s (r_s - \bar{r})^2/(3-1)}$. Each action budget or planning condition defines its own evaluation group. Repeated evaluation, when used, is averaged within an instance before seed aggregation; repeats do not create additional independent tasks.

ScienceWorld averages task types within each seed. For training seed s, let $y_{s,c,j,r}$ be the raw final score on task type c, variation j, and evaluation repeat r. Average repeats within each variation, variations within each type, and the 30 type means equally. The final score averages these seed-level macro scores. Success@100 averages $\mathbf{1}[y = 100]$ within each variation across repeats, then weights instances and seeds equally. The separately labeled micro score also weights instances equally.

Negative terminal scores remain in the raw-score metric. Task lists, repeats, action budgets, and training seeds define each evaluation group independently of the reported outcome.

Paired uncertainty preserves shared tasks. A comparison pairs results by task, training seed, environment seed, action budget, and repetition. Training seeds and task instances are separate sources of variation. ScienceWorld analysis preserves task-type grouping. Repeated seeds quantify training variation; they do not multiply the number of independent task instances.

Weight-distribution controls keep the multiset fixed. For binary ALFWorld returns with $M = 8$ and $\kappa = 2$, every raw DRR weight belongs to $\{1, 1.25, \ldots, 3\}$. Weight-distribution matching assigns one fixed multiset of raw weights to roots according to each priority score. Table 24 compares the NLL and total-regret orderings.

Table 24: Weight-distribution controls. Both comparisons retain the DRR weight multiset and vary the root priority ordering.

Condition	ALF SR	Science	Comparison scope
NLL quantile-matched weights	86.32	72.07	Weights follow NLL ordering with a fixed weight distribution.
Total-regret quantile matching	84.83	71.07	Weights follow total-regret ordering with the same distribution.

Factorial comparisons distinguish module effects from interactions. The B03/B08/B09/B11 factorial changes both the representation and the action-contrast objective. Its .25/ .80 interaction is therefore not an action-contrast-specific interaction. Holding delta representation fixed and using B04/B08/A01/B11 gives a displayed-value interaction of -2.48 ALFWorld percentage points and -1.00 ScienceWorld score points. These interaction estimates are descriptive; paired uncertainty estimates are needed to assess their statistical significance.

Configuration identity defines a repeatable comparison. Optimizer settings, schedules, checkpoint selection, task and root identities, environment version, restoration rules, parser behavior, value-head folds, and baseline adaptations jointly identify a run. Candidate sets and rollout returns identify a shared feedback comparison. Appendix A defines the value-head architecture and cross-fitting scheme.

D.11 World and actor update paths

The static controls separate dynamics weighting in the world-model and actor parameter copies. Write the raw root weights as w_t^{W} and w_t^{A} and normalize each active path within its minibatch. All conditions share the same return-based policy targets, expert supervision, actor initialization, and cached feedback. Table 25 defines the two paths.

Table 25: Static weighting paths. DRR denotes $1 + \kappa\rho_t$. Uniform weights equal one. All actor paths use the same policy soft targets.

Condition	w_t^{W}	w_t^{A}	λ_{keep}
B10, uniform feedback	Uniform	Uniform	.1
B11, full method	DRR	DRR	.1
A06, no actor dynamics	DRR	Disabled	0
A14, world-only weighting	DRR	Uniform	.1

A14 uses uniform actor dynamics weights; A06 omits actor dynamics supervision. Static updates keep policy labels fixed and do not copy world parameters into the actor. Thus, A14 and B10 have the same actor objective and identical deterministic updates when minibatches, initialization, optimizer state, and random-number streams match. A14 controls for unintended transfer between copies. B11 versus B10 tests actor-side DRR weighting; B11 versus A06 tests actor dynamics supervision.

E EXTENDED EXPERIMENTAL RESULTS

This section extends the main comparisons with module attribution, task-level breakdowns, robustness tests, learning curves, execution diagnostics, and test-time planning. Resource tables use analytical cost accounting.

Metric conventions. ALFWorld task performance is reported as a success rate; ScienceWorld uses the type-macro score unless otherwise specified. Cost tables use analytical accounting. Statistical definitions and the scope of auxiliary comparisons appear in Appendix D.10.

E.1 MODULE ATTRIBUTION AND SUPPLEMENTARY TASK METRICS

Table 26: Four-cell module comparison and stage-wise differences. The four-cell rows compare full-state/CDD and no-VRA/VRA conditions; stage differences are calculated from the displayed means.

Condition	ID	ALF SR	Science	Science S@100
Full-state + SFT	B03	75.12	65.53	44.43
CDD + SFT	B08	80.35	68.13	46.73
Full-state + VRA	B09	81.09	69.33	47.83
CDD + VRA	B11	86.57	72.73	50.73
Ledger input, B01−B00	–	+2.24	+2.10	+1.00
Full-state warm-up, B03−B01	–	+4.97	+2.70	+0.60
Delta representation, B04−B03	–	+1.50	+0.40	+0.40
Action contrast, B08−B04	–	+3.73	+2.20	+1.90
Shared real feedback, B10−B08	–	+4.48	+2.80	+2.50
DRR weighting, B11−B10	–	+1.74	+1.80	+1.50
DRR vs. NLL, B11−B15	–	+0.50	+0.90	+0.90
DRR vs. total regret, B11−B16	–	+2.24	+2.20	+1.90

Table 27: Supplementary task metrics. ALF seen and family scores use independent evaluation lists; Science macro is the primary metric; the table adds micro score, the across-seed standard deviation of success@100 (in percentage points), and negative-terminal rate.

ID	ALF seen	ALF family macro	Science micro	Science S@100 SD	Negative terminal	Primary pair
B00	64.66	66.74	57.40	2.33	4.73	67.91/60.73
B01	67.10	68.98	59.61	2.19	4.50	70.15/62.83
B03	72.52	73.95	62.44	2.08	4.20	75.12/65.53
B04	74.15	75.45	62.86	2.16	4.16	76.62/65.93
B05	75.78	76.94	64.75	2.02	3.96	78.11/67.73
B06	76.59	77.69	65.59	2.30	3.87	78.86/68.53
B08	78.22	79.18	65.17	2.05	3.92	80.35/68.13
B09	79.03	79.92	66.43	2.18	3.78	81.09/69.33
B10	83.10	83.66	68.11	1.94	3.61	84.83/70.93
B11	85.00	85.40	70.00	2.09	3.41	86.57/72.73
B12	87.17	87.39	72.21	2.21	3.18	88.56/74.83
B15	84.45	84.90	69.06	2.04	3.51	86.07/71.83
B16	82.56	83.16	67.69	2.42	3.65	84.33/70.53
R03	92.32	92.11	73.03	2.17	3.09	93.28/75.61
R06	92.05	91.86	74.29	2.12	2.96	93.03/76.81
R08b	93.94	93.61	74.73	2.24	2.91	94.78/77.23

Table 28: Full-sample micro Change precision, recall, and F1. The main mechanism table uses sample-averaged F1; this table provides the global micro aggregation.

Method	ALF P	ALF R	ALF F1	Science P	Science R	Science F1
Full-state	85.13	78.46	81.66	73.18	62.37	67.34
Delta-only	87.46	81.39	84.32	75.02	64.86	69.57
CDD	89.24	84.53	86.82	78.16	69.75	73.72
CDD + VRA	89.43	84.97	87.14	78.03	69.97	73.78

E.2 WEIGHT CONCENTRATION AND ROBUSTNESS CONTROLS

Table 29: Weight concentration controls. ESS is computed over 1,000 feedback roots. Quantile-matched controls hold the reported concentration fixed while changing which roots receive high weight.

Environment / scheme	ESS	ESS/N	Primary result
ALF uniform	1000.0	1.000	B10
ALF DRR	914.1	.914	B11
ALF NLL quantile-matched	914.1	.914	86.32/72.07
ALF total-regret quantile-matched	914.1	.914	84.83/71.07
Science uniform	1000.0	1.000	B10
Science DRR	854.1	.854	B11
Science NLL quantile-matched	854.1	.854	86.32/72.07
Science total-regret quantile-matched	854.1	.854	84.83/71.07

Table 30: Task-length, recovery, observation-masking, and irrelevant-information controls.

Protocol	B04 ALF	B10 ALF	B11 ALF	B04 Science	B10 Science	B11 Science
Short tasks	87.41	92.17	92.63	76.23	81.47	82.03
Medium tasks	74.83	82.67	84.29	64.13	69.83	71.07
Long tasks / G04	54.29	62.43	64.73	50.93	57.21	59.01
G05 recovery SR	52.93	63.17	66.27	49.81	58.27	61.17
20% masking / G08	–	72.47	73.07	–	61.43	61.91
40% masking	–	58.73	57.97	–	51.17	50.83
100% irrelevant text	–	72.53	72.87	–	62.13	61.97

E.3 SCALE, DATA EFFICIENCY, AND ACTION BUDGETS

Table 31: Qwen3-8B extension and matched net gains.

Condition	ALF SR	Science	Science S@100
8B ledger SFT	88.81	66.13	47.83
8B Delta-only	90.05	68.87	48.83
8B CDD	91.54	70.39	49.97
8B CDD + uniform	93.03	74.17	53.03
8B Action2Delta static	93.78	74.67	52.83
8B Action2Delta online	94.78	75.81	53.67
8B State2State + GRPO	97.51	78.17	56.13
8B Action2Delta + GRPO	96.77	78.81	55.83
4B DRR−uniform	+1.74	+1.80	+1.50
8B DRR−uniform	+0.75	+0.50	−0.20
4B online VRA−AWR	+0.50	+0.70	+0.40
8B online VRA−AWR	+0.25	+0.24	−0.06

Table 32: Development-set data efficiency and downstream learning curves.

Dynamics data	Full-state	Delta-only	CDD	B10	B11	Metric
10k	69.73/61.43	71.37/62.29	73.81/64.17	80.07/68.37	81.63/69.71	ALF/Science
30k	73.93/64.01	75.41/64.83	78.29/67.31	83.17/70.41	85.19/71.93	ALF/Science
100k	76.17/66.09	77.43/66.53	81.29/68.71	85.73/71.39	87.61/73.23	ALF/Science
RL budget 0%	70.35/62.41	48.19/35.73	87.61/73.23	–	–	R00/R03/R06
RL budget 25%	79.17/68.17	86.57/68.83	91.43/75.87	–	–	R00/R03/R06
RL budget 50%	83.21/71.09	91.07/73.67	93.17/77.01	–	–	R00/R03/R06
RL budget 100%	88.97/74.13	94.97/76.37	94.47/77.53	–	–	R00/R03/R06

Table 33: ScienceWorld action-budget comparison. Entries report type-macro score and success@100 at each action limit.

Action limit	B00	B10	B11	R03	R06
30	52.39/33.17	62.71/42.77	64.49/44.13	71.13/49.33	72.21/49.03
100	60.73/42.83	70.93/49.23	72.73/50.73	75.61/54.77	76.81/54.27
200	64.17/45.73	73.73/52.17	75.91/54.13	78.31/58.03	79.27/57.73

E.4 Execution behavior, first errors, and task-type breakdown

Table 34: Execution accounting for the key direct actors. Attempts are averaged over successful episodes; actor calls and environment steps are averaged over all episodes.

Method	ALF attempts	ALF calls	ALF steps	Science attempts	Science calls	Science steps
B00	17.43	24.19	23.98	48.93	74.12	72.84
B01	17.16	23.53	23.34	48.09	73.32	72.11
B04	16.39	21.56	21.42	46.85	72.31	71.22
B08	15.94	20.37	20.29	45.97	71.02	70.36
B10	15.40	18.91	18.83	44.85	69.30	68.41
B11	15.19	18.32	18.26	44.13	68.21	67.39
R06	14.37	16.05	16.03	42.70	65.70	65.03
R08b	14.11	15.38	15.36	43.33	65.60	64.94

Table 35: First verified error diagnostics. RMST is restricted to the first 20 ALFWorld or 50 ScienceWorld decisions.

Method	ALF RMST@20	ALF error-free at cutoff	Science RMST@50	Science error-free at cutoff
B00	11.21	39.73	22.59	25.43
B04	13.17	47.13	26.41	31.27
B08	14.03	52.47	28.23	35.17
B10	15.19	58.73	30.19	39.43
B11	15.84	62.13	31.27	42.17

Table 36: ALFWorld family breakdown.

Family	B00	B04	B08	B10	B11	R03	R06
pick/place simple	72.74	81.45	85.18	89.66	91.40	98.11	97.86
pick two/place	67.74	76.45	80.18	84.66	86.40	93.11	92.86
clean then place	63.74	72.45	76.18	80.66	82.40	89.11	88.86
heat then place	61.74	70.45	74.18	78.66	80.40	87.11	86.86
look in light	73.74	82.45	86.18	90.66	92.40	99.11	98.86
cool then place	60.74	69.45	73.18	77.66	79.40	86.11	85.86
Family macro	66.74	75.45	79.18	83.66	85.40	92.11	91.86

Table 37: ScienceWorld scores for all 30 task types. The final row averages task types equally.

Type	B00	B04	B08	B10	B11	R03	R06
boil	66.56	71.31	73.32	75.87	77.51	80.14	81.24
melt	62.56	67.62	69.76	72.48	74.23	77.03	78.20
freeze	58.56	63.93	66.20	69.09	70.95	73.93	75.17
change state	63.56	68.54	70.65	73.33	75.05	77.81	78.96
thermometer	76.56	80.53	82.21	84.34	85.71	87.91	88.82
melting point known	51.56	57.48	59.98	63.17	65.21	68.49	69.85
melting point unknown	47.56	53.79	56.42	59.78	61.93	65.38	66.82
power component	67.56	72.23	74.20	76.72	78.33	80.92	82.00
renewable energy	57.56	63.01	65.31	68.25	70.13	73.15	74.41
conductivity	52.56	58.40	60.87	64.01	66.03	69.27	70.61
unknown conductivity	46.56	52.87	55.54	58.93	61.11	64.61	66.06
find living	87.56	90.67	91.98	93.66	94.73	96.45	97.17
find non-living	89.56	92.51	93.76	95.35	96.37	98.01	98.69
find plant	85.56	88.83	90.21	91.96	93.09	94.90	95.65
find animal	81.56	85.14	86.65	88.58	89.81	91.79	92.62
grow plant	35.56	42.73	45.76	49.61	52.09	56.06	57.71
grow fruit	29.56	37.19	40.42	44.53	47.17	51.40	53.16
chemistry mix	64.56	69.46	71.54	74.18	75.87	78.59	79.72
secondary paint	70.56	75.00	76.87	79.26	80.79	83.25	84.27
tertiary paint	62.56	67.62	69.76	72.48	74.23	77.03	78.20
longest lifespan	79.56	83.29	84.87	86.88	88.17	90.24	91.10
shortest lifespan	76.56	80.53	82.21	84.34	85.71	87.91	88.82
longest then shortest	59.56	64.85	67.09	69.94	71.77	74.70	75.92
life stages 1	71.56	75.92	77.76	80.11	81.61	84.03	85.03
life stages 2	66.56	71.31	73.32	75.87	77.51	80.14	81.24
inclined angle	47.56	53.79	56.42	59.78	61.93	65.38	66.82
friction named	49.56	55.63	58.20	61.47	63.57	66.94	68.34
friction unnamed	42.56	49.18	51.98	55.54	57.83	61.50	63.03
genetics known	38.56	45.49	48.42	52.15	54.55	58.39	59.99
genetics unknown	31.56	39.04	42.20	46.23	48.81	52.95	54.68
Macro average	60.73	65.93	68.13	70.93	72.73	75.61	76.81

E.5 PAIRED EFFECTS AND CUMULATIVE COST

Table 38: Paired effects and statistical power. ALF differences are in percentage points; Science differences are in score points. Brackets give the lower and upper values from effect sensitivity analysis. Required N is the number of independent ALF tasks for two-sided $\alpha = .05$ and power .80, using unrounded paired differences. The four power calculations use discordance rates of .12, .08, .05, and .04, respectively.

Comparison	ALF difference	Science difference	Required N
B08−B04	+3.73 [−2,+7]	+2.20 [−3,+6]	669
B11−B10	+1.74 [−3,+5]	+1.80 [−3,+6]	2064
B11−B15	+0.50 [−3,+4]	+0.90 [−3,+5]	15852
B11−B16	+2.24 [−3,+6]	+2.20 [−3,+7]	–
B12−B13	+0.50 [−2,+3]	+0.70 [−2,+4]	–
R06−R03	−0.25 [−4,+4]	+1.20 [−4,+5]	50744

Table 39: Analytical cumulative downstream cost curves. FLOPs are in 10^{18} and environment interaction is in millions of steps.

Budget	ALFWorld			ScienceWorld		
	R00 steps/FLOPs	R03 steps/FLOPs	R06 steps/FLOPs	R00 steps/FLOPs	R03 steps/FLOPs	R06 steps/FLOPs
0%	0.000/.302	1.900/21.000	2.156/23.112	0.000/1.693	4.700/51.000	4.764/65.616
25%	.500/10.302	2.400/31.000	2.656/33.112	1.500/31.693	6.200/81.000	6.264/95.616
50%	1.000/20.302	2.900/41.000	3.156/43.112	3.000/61.693	7.700/111.000	7.764/125.616
75%	1.500/30.302	3.400/51.000	3.656/53.112	4.500/91.693	9.200/141.000	9.264/155.616
100%	2.000/40.302	3.900/61.000	4.156/63.112	6.000/121.693	10.700/171.000	10.764/185.616

E.6 TEST-TIME PLANNING

Table 40: ScienceWorld test-time planning and slow-thinking extensions. Calls are episode averages. Direct actors use no auxiliary calls.

Setting	Macro score	S@100	WM calls	Slow calls
T00 direct B01	62.83	43.83	0	0
T01 direct B11	72.73	50.73	0	0
T02 full-state lookahead	69.43	47.73	59.13	0
T03 QAW-style lookahead	70.27	48.43	58.47	0
T04 MemWM-style memory	70.13	48.17	56.93	0
T05 WorldEvolver-style rules	71.53	49.03	61.23	0
T06 B11 + own WM lookahead	74.27	51.93	54.71	0
T07 SwiftSage-style	75.33	54.37	0	3.41