ADAPTWORLD: LEARNING TO ACT AND ADAPT OVER LONG-HORIZON TASKS IN DYNAMIC ENVIRONMENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-horizon tasks in dynamic environments require agents to revise decisions as evidence arrives while managing earlier commitments. AdaptWorld constructs executable training environments that make this adaptation consequential. Public-information reference comparisons guide bounded recipe revision; independent assessment admits task families, and execution-and-outcome filtering selects model demonstrations. Across finance, grid storage, inventory, and network recovery, revision raises recipe acceptance from 34.4% to 72.7%. We fine-tune two Qwen backbones on 5,120 episodes from 4,226 tasks. Four-domain training improves both backbones on CEO, Merchant, and local Vending. For 27B, it exceeds the strongest single-domain source by 11.3%, 18.2%, and 69.4%, respectively, at a matched supervision target; 35B retains stronger task-specific specialization. Broader task coverage helps, while the best difficulty mixture depends on the target. At four times the reference horizon, SFT increases completion by 23–31 percentage points across all six model–benchmark pairs. These results link decision-oriented environment construction to transfer and sustained execution. Code is available at https://anonymous.4open.science/r/anonymous_code3-1025.

1 INTRODUCTION

Long-horizon tasks require agents to act on incomplete information while managing earlier commitments (Chen et al., 2026; Shi et al., 2026; Zhang et al., 2026; Pan et al., 2026). Orders precede demand, investments precede reports, and commitments constrain later responses. Business benchmarks make these demands explicit (Chen et al., 2026; Shi et al., 2026; Backlund & Petersson, 2025; Zhang et al., 2026; Hu et al., 2026; Sugiura et al., 2026; Pan et al., 2026; Fan et al., 2026; Wang et al., 2026a). Training environments must reward decisions that use new evidence (Ross et al., 2007; Klenske & Hennig, 2016; Duan et al., 2016) (Figure 1).

The central construction problem is to make information consequential (Ross et al., 2007; Klenske & Hennig, 2016). A valid simulator may reward fixed rules if uncertainty is irrelevant, reports arrive too late, or resources erase decision costs. Likewise, a long transcript need not contain sustained adaptation (Chen et al., 2026; Shi et al., 2026; Backlund & Petersson, 2025). We target parameter inference, state tracking, intertemporal planning, and active sensing (Doshi-Velez & Konidaris, 2013; Ross et al., 2007; Klenske & Hennig, 2016; Duan et al., 2016). Adaptation revises decisions using public history within an episode; post-training updates model parameters across episodes (Duan et al., 2016).

AdaptWorld builds training tasks around these requirements (Hu et al., 2025; Sullivan et al., 2025; Shi et al., 2025; Song et al., 2026; Cai et al., 2025; Xu et al., 2026). Executable domain mechanisms define hidden quantities, observations, actions, and outcomes (Song et al., 2026; Wu et al., 2026; Shen et al., 2026; Xu et al., 2026). We compare public-information reference controllers against specified restrictions and simple baselines. These comparisons guide bounded recipe revision (Dennis et al., 2020; Parker-Holder et al., 2022; Guo et al., 2026; Wu et al., 2026), after which fresh worlds assess the frozen recipe. Accepted families supply rollouts; execution-and-outcome filtering selects supervised fine-tuning (SFT) demonstrations (Shi et al., 2025; Wang et al., 2026b; Xu et al., 2026). The pipeline separates executable correctness, decision value, and demonstration quality.

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

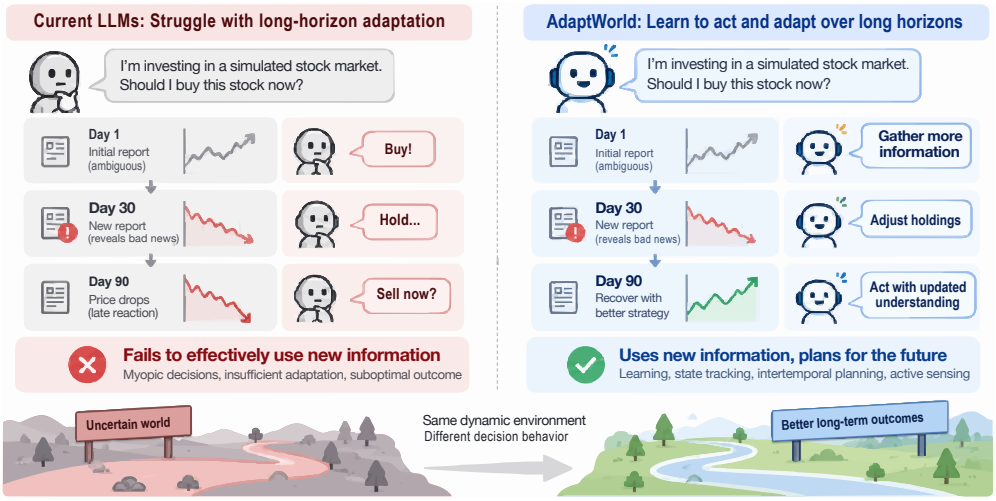


Figure 1: **Acting and adapting over time.** New evidence motivates strategy changes; earlier commitments constrain actions. Investment illustrates evidence timing and persistent consequences.

Across finance, grid storage, inventory, and network recovery, revision raises acceptance from 44 to 93 of 128 candidates. Training on 5,120 episodes improves both Qwen backbones on CEO, Merchant, and local Vending (Chen et al., 2026; Shi et al., 2026; Backlund & Petersson, 2025). Sources complement each other for 27B; 35B shows stronger specialization. At four times the reference horizon, SFT improves completion by 23–31 percentage points across all six model–benchmark pairs.

2 Related Work

Long-horizon decisions and agent training. CEO-Bench, MerchantBench, and Vending-Bench evaluate decisions with persistent business consequences (Chen et al., 2026; Shi et al., 2026; Backlund & Petersson, 2025). FM-Bench, STOCKTAKE, and WildGrid emphasize delayed returns or incomplete information (Wang et al., 2026a; Deb & Krishnan, 2026; Pezeshkpour & Hruschka, 2026); AppWorld and Tool-Decathlon evaluate application and tool workflows (Trivedi et al., 2024; Li et al., 2026). Reflexion uses feedback through the agent harness (Shinn et al., 2023), while AgentGen, PAPRIKA, AgentGym-RL, MEM1, and TurnOPD study generated tasks, interactive learning, memory, and supervision allocation (Hu et al., 2025; Tajwar et al., 2025; Xi et al., 2026; Zhou et al., 2026b;a). AdaptWorld uses standard SFT to study the training environments and demonstrations supplied to the learner.

Executable environment synthesis. AWM, EnvCraft, and Echoverse construct executable tool or computer-use environments (Wang et al., 2026c; Zeng et al., 2026; Pandya et al., 2026). ScaleEnv, AutoForge, and EnvScaler combine synthesis with task or trajectory verification (Tu et al., 2026; Cai et al., 2025; Song et al., 2026); AgentMercury builds persistent business worlds (Jeong & Yoon, 2026). AdaptWorld adds a decision-oriented assessment: reference comparisons measure the utility of declared behaviors before model-demonstration selection.

Adaptive construction and validation. EnvHarness revises interfaces from trajectory diagnostics; SPADE uses privileged-hint return gaps; GenEnv targets learner success rates; and EnvsFORGE uses policy performance and reference solutions (Huang et al., 2026; Liu et al., 2026; Guo et al., 2026; Wu et al., 2026). AutoEnv separates dynamics, observations, and presentation, while VHD-Play starts from solved mechanisms (Zhang et al., 2025; Shen et al., 2026). AdaptWorld holds the family prior and public interface fixed while restricting specified reference behaviors. Freshworld assessment separates recipe selection from admission, and a subsequent outcome filter selects the model responses used for training.

3 LONG-HORIZON TASKS IN DYNAMIC ENVIRONMENTS

Worlds and public information. Domains define dynamics, actions, observations, and scoring; *family recipes* ψ configure them and set priors μ_ψ over hidden parameters. Instances fix parameters and exogenous inputs. We model evolving state and observations as a POMDP with static hidden parameters (Doshi-Velez & Konidaris, 2013; Ross et al., 2007):

$$\theta \sim \mu_\psi, \qquad S_0 \sim p_{0,\psi}(\cdot \mid \theta), \qquad (S_{t+1}, Y_{t+1}) \sim K_\psi(\cdot, \cdot \mid S_t, A_t; \theta). \tag{1}$$

The policy chooses requests from public history $h_t = (Y_0, A_0, \ldots, A_{t-1}, Y_t)$ to maximize episode utility J_ψ (Ross et al., 2007). In finance, θ specifies asset responses; S_t includes market conditions, holdings, and pending settlements; Y_t contains released observations and execution receipts. The Markov state includes pending actions and observations; realized parameters and future events remain private (Appendix A).

Decision requirements and horizon. *Parameter inference* updates beliefs about fixed unknown rules; *state tracking* estimates changing conditions; *intertemporal planning* accounts for delayed consequences; and *active sensing* selects observations according to their cost and remaining decision value (Doshi-Velez & Konidaris, 2013; Ross et al., 2007; Klenske & Hennig, 2016; Duan et al., 2016). Families declare supported requirements and reference interventions. Persistent commitments couple decisions over time. Physical horizon H, commitment span L, response count T, context length C_{ctx}, and execution budget B are distinct: one response can advance several simulated days.

Execution and outcomes. Completion requires the prescribed terminal and final accounting; validity requires rule compliance; success additionally requires the goal. Utility measures outcomes after costs. Budget exhaustion is failure; infrastructure interruptions yield missing outcomes. External evaluation retains benchmark score definitions (Chen et al., 2026; Shi et al., 2026; Backlund & Petersson, 2025) and the stated stopping rules, including the local Vending horizon extension (Section 6.5). Appendix B gives the full outcome contract.

4 CONSTRUCTING AND SCALING TRAINING ENVIRONMENTS

AdaptWorld couples evolving evidence with persistent consequences to construct training tasks (Figure 2). Recipes define decision problems; actions change world states. Inventory illustrates the stages.

4.1 PROPOSE EXECUTABLE RECIPES

A shop stocks two perishable products for 40 days, starting with 3,000 cash units. Orders require immediate payment, arrive after one or two days, expire after eight days, and share storage. Unknown demand and seasonality determine sales. The INV-SALES recipe provides an initial noisy report, then sales observations and unmet-demand flags. Utility is final cash gain minus shortage penalties, normalized by initial cash for reference comparisons.

A language-model author selects program-enumerated legal configurations and varies presentation (Hu et al., 2025; Shi et al., 2025; Cai et al., 2025; Song et al., 2026; Xu et al., 2026). Executable mechanisms supply numerical facts and rules (Sullivan et al., 2025; Song et al., 2026; Wu et al., 2026; Shen et al., 2026; Xu et al., 2026). Selling all five available units reveals demand of at least five; unmet-demand flags distinguish strictly greater demand. Additional stock can reveal demand but ties up cash and risks expiry. Learning matters while orders can still arrive. This coupling between action and information is characteristic of dual control (Klenske & Hennig, 2016).

4.2 USE REFERENCE COMPARISONS TO REVISE RECIPES

The public-information reference estimates demand from reports and sales, ordering according to stock, deliveries, and cash. Its restricted counterpart suppresses subsequent sales updates within the same controller; a simple baseline uses the initial prior. For declared intervention k, the protocol Γ measures

$$\Delta_k(\psi; \Gamma) = \mathbb{E}_{\theta \sim \mu_\psi, \omega} \left[\bar{J}_\psi(\pi^{\text{ref}}; \theta, \omega) - \bar{J}_\psi(\pi^{-k}; \theta, \omega) \right], \tag{2}$$

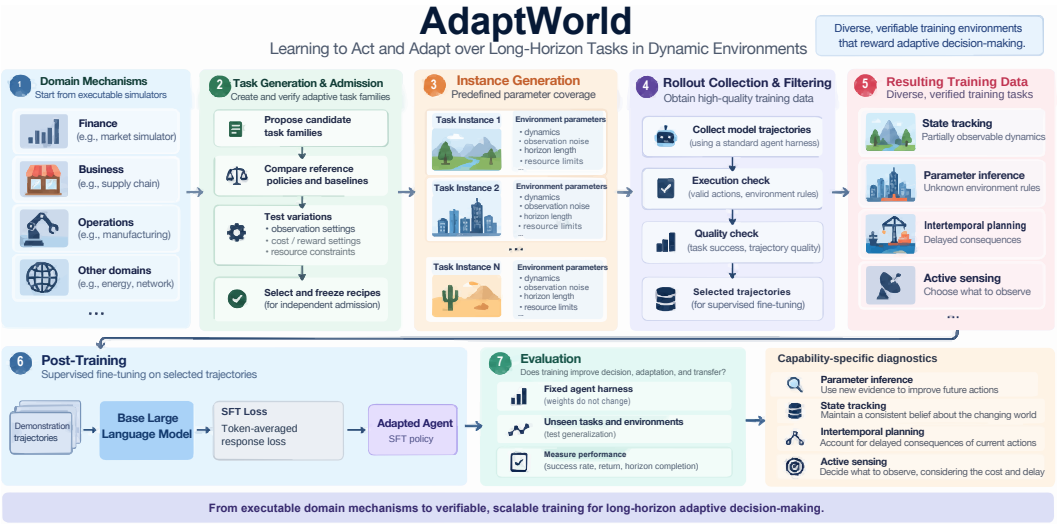


Figure 2: **AdaptWorld workflow.** Domain mechanisms support proposal, reference comparisons, revision, and independent admission. Accepted instances supply rollouts; execution and outcome checks select SFT demonstrations. Diagnostics are domain-specific.

where $\bar{J}_\psi$ uses a fixed domain scale. Both controllers use public information with shared parameters and exogenous randomness ω. Different actions can produce different observations (Ross et al., 2007; Klenske & Hennig, 2016); in inventory, orders change how demand is censored by stockouts.

Bounded edits target shortfalls in these comparisons (Dennis et al., 2020; Parker-Holder et al., 2022; Guo et al., 2026; Wu et al., 2026). For example, multiplying initial-report noise by 1.5, capped at 1.0, changes 0.25 to 0.375 and leaves more uncertainty for later sales to resolve. Search considers report-noise and storage-capacity edits over at most two rounds with two proposals per round. An edit is retained only if it reduces aggregate threshold shortfall. The prior, objective, controllers, and thresholds remain fixed throughout search.

4.3 Freeze, Assess, and Instantiate

Fresh worlds assess each frozen recipe once under the same prior. Admission requires structural validity, reference attainability, and advantages over declared restrictions and simple baselines; unresolved execution failures block admission (Shi et al., 2025; Song et al., 2026; Wu et al., 2026; Shen et al., 2026; Xu et al., 2026). Diagnostics are declared before search according to domain mechanisms. GRID-ID freezes parameter-mixture weights, while GRID-PLAN shortens the reference planner's lookahead. INV-REPORT suppresses both report and sales updates; INV-SALES retains the initial report but suppresses subsequent sales updates. NET-DIAG removes later alarms from a joint parameter–state belief, and NET-PROBE disables paid observations. Finance uses public-information references with outcome and execution checks. Appendices D–E specify statistical units, confidence bounds, and domain coverage.

Let $L_{d,k}$ lower-bound the normalized capability gap in Equation 2, L_b lower-bound the reference advantage over simple baseline b, and L_{succ} lower-bound reference success. Admission requires

$$\text{Admit}(\psi) = \text{Valid}(\psi) \wedge [L_{\text{succ}} \geq \rho] \wedge \bigwedge_{k \in \mathcal{K}_\psi} [L_{d,k} \geq \delta_{d,k}] \wedge \bigwedge_{b \in \mathcal{B}_d} [L_b \geq \delta_b], \tag{3}$$

where the success floor ρ and positive gap thresholds are fixed before search. This criterion concerns expected gains under a family prior; an individual world need not favor every capability.

Production allocates parameter quotas before observing returns. For two products with seven demand rates and three seasonal coefficients each, $(7 \times 3)^2 = 441$ joint settings define the inventory support. Predetermined allocation selects settings; new exogenous inputs provide additional worlds. Public task materials expose rules and possible parameters while keeping realized parameters and

future events private. Reference execution and replay check transitions, accounting, and scoring (Shi et al., 2025; Wang et al., 2026b; Xu et al., 2026; Shen et al., 2026). Coverage is rechecked after filtering, which can change the represented distribution.

A new domain supplies dynamics, tools, verifiers, references, and legal edits while reusing these stages. Finance uses holdings and settlement, grid uses storage and ramp constraints, inventory uses replenishment and expiry, and network recovery uses delayed probes and repairs (Table 5). Changes to parameter support or mechanisms trigger renewed family assessment.

Admission tests decision requirements under the family prior; filtering selects demonstrations. Informative families can yield failed rollouts, and successful rollouts alone cannot establish adaptation value. Both stages shape the training distribution.

5 Learning from Verified Demonstrations

Select useful completed episodes. Models interact through public observations and native tools. Even admitted tasks can yield unsuccessful rollouts, so completed trajectories undergo execution replay and outcome filtering (Shi et al., 2025; Wang et al., 2026b; Xu et al., 2026; Wu et al., 2026; Shen et al., 2026). For grid, inventory, and network, selection requires valid completion, a business floor, and improvement over the strongest predefined simple baseline on the same world:

$$q_\psi(\xi) = \frac{J_\psi(\xi) - \max_{b \in \mathcal{B}_d} J_\psi(b; \theta, \omega)}{g_\psi} \geq 0.2, \qquad g_\psi > 0. \tag{4}$$

The scale g_ψ is the smallest frozen family-level mean reference advantage over these baselines. Business floors require nonnegative inventory utility, grid service of at least 0.8, and network service of at least 0.5; finance uses its domain-specific selection rule. Train/evaluation splits precede collection.

Preserve the context of each supervised response. Token-averaged SFT uses sampled assistant responses with their original public decision contexts (Ouyang et al., 2022; Shi et al., 2025; Song et al., 2026; Wang et al., 2026b). System, user, tool, and copied history provide context without duplicate target loss. Segmentation preserves response identities, assigning each eligible target once before repeated exposure. Grid, inventory, and network contexts are checked against collection requests; finance contexts are reconstructed from native CLI records. Appendix F gives the loss and masking rules.

6 Experiments

We evaluate learning to act over long horizons through downstream performance and sustained execution. Construction and training-distribution analyses examine how task supply, source, coverage, and difficulty shape these outcomes.

6.1 Experimental setup

Training and controls. The shared training set contains 5,120 selected episodes from 4,226 tasks: 512 finance episodes and 1,536 each from grid, inventory, and network recovery. Its 8,266 segments contain 43,446,732 eligible target tokens per exposure. We fine-tune Qwen3.6-27B and Qwen3.6-35B-A3B with two exposures, giving an 86.89M-token supervision target. Within each backbone, source and composition comparisons hold the optimizer and execution settings fixed while varying the designated data factor. Teacher composition and task-grouped selection are detailed in Appendix G.

Evaluation. CEO, Merchant, and local Vending measure external business performance in USD, CNY, and USD, respectively (Chen et al., 2026; Shi et al., 2026; Backlund & Petersson, 2025). External evaluations follow the settings of the original benchmarks, with the horizon extensions described in Section 6.5. Each backbone–training-condition–benchmark combination contains 256 episodes; the reported score is the arithmetic mean of terminal outcomes, including legal early terminals. We also evaluate 639 internal tasks: 125 grid, 264 inventory, and 250 network. Internal

Table 1: **External benchmark performance by training source.** Each score is the arithmetic mean over 256 evaluation episodes. SFT conditions share an 86.89M supervised-token target. Bold identifies the highest score within each backbone and benchmark; the last row compares four-domain SFT with Base. Units appear in the header. Appendix G gives full-precision scores and observed min–max ranges.

Training source	Qwen3.6-27B			Qwen3.6-35B-A3B		
	CEO (k USD)	Merchant (k CNY)	Vending (USD)	CEO (k USD)	Merchant (k CNY)	Vending (USD)
Base	106.69	1,299.15	1,235.85	94.74	2,179.98	593.75
Finance SFT	154.89	2,879.25	2,595.53	113.26	1,906.34	639.08
Grid SFT	114.71	2,916.44	2,360.32	116.41	2,172.21	1,160.21
Inventory SFT	181.60	3,957.66	2,694.31	108.64	**2,857.74**	1,270.10
Network SFT	163.36	2,171.78	2,029.04	**118.51**	2,138.85	852.93
Four-domain SFT	**202.04**	**4,679.07**	**4,562.90**	115.23	2,272.28	**1,298.16**
Change vs. Base	+89.4%	+260.2%	+269.2%	+21.6%	+4.2%	+118.6%

success requires valid completion, replay verification, and the frozen goal predicate. The Base and four-domain SFT conditions are shared across analyses; single-domain comparisons change the training source.

6.2 External performance and source specialization

Four-domain training improves both backbones. Table 1 shows higher four-domain scores than Base in all six comparisons. For 27B, gains are 89.4% on CEO, 260.2% on Merchant, and 269.2% on Vending; for 35B, they are 21.6%, 4.2%, and 118.6%. Thus the benefit extends across benchmark objectives, with the largest relative gain on Vending for both models. The smaller 35B Merchant gain is especially informative: training effectiveness depends on the combination of backbone and target task, even when the demonstration set and supervision target are shared.

Diversity and specialization have different benefits. For 27B, four-domain training exceeds the strongest single-domain source by 11.3% on CEO, 18.2% on Merchant, and 69.4% on Vending. Because the supervision target is matched, these gains favor broader training content over repeated exposure to the best individual source. Inventory is the strongest single source on all three 27B benchmarks, yet its demonstrations alone leave a substantial gap on Vending.

For 35B, specialization remains competitive. Network-only training reaches 118.51k USD on CEO, compared with 115.23k for the mixture. Inventory-only training reaches 2,857.74k CNY on Merchant, 25.8% above the four-domain score. On Vending, four-domain training is strongest but exceeds inventory-only by just 2.2%. Finance-only training also lowers 35B Merchant performance by 12.6% relative to Base. These results reveal a practical trade-off: the shared mixture improves every evaluated target, while concentrating supervision on a matched source can maximize a particular target. They support choosing source composition jointly with the backbone and evaluation objective.

6.3 Recipe revision expands the training-task supply

Table 2 compares the same pool of 128 candidates, with 32 per domain, before and after bounded revision. Accepted recipes increase from 44 to 93: a net gain of 49 families and a 2.11-fold increase in accepted yield. Acceptance rises by 38.3 percentage points under unchanged domain-specific checks. The search therefore expands usable task diversity by modifying the decision problem within its allowed configuration range, while keeping the prior and admission thresholds fixed.

Reference controllers succeed on 118 of 128 evaluation worlds (92.2%), showing that the goals are attainable under public information. This rate complements the recipe count: attainability concerns controller outcomes, whereas admission also tests the declared decision requirements. Instantiating 64 worlds per accepted family produces 5,952 task instances. Subsequent rollout filtering selects

Table 2: **Environment construction yield.** The same acceptance checks are applied before and after bounded recipe revision. Reference success uses 32 evaluation worlds per domain; its denominator counts worlds rather than candidate recipes.

Measure	Aggregate over four domains
Candidate recipes	128 (32 per domain)
Accepted recipes	44/128 → 93/128
Acceptance rate	34.4% → 72.7% (**+38.3 pp**)
Generated instances	5,952 (64 per accepted recipe)
Reference success	118/128 (92.2%; 32 worlds per domain)

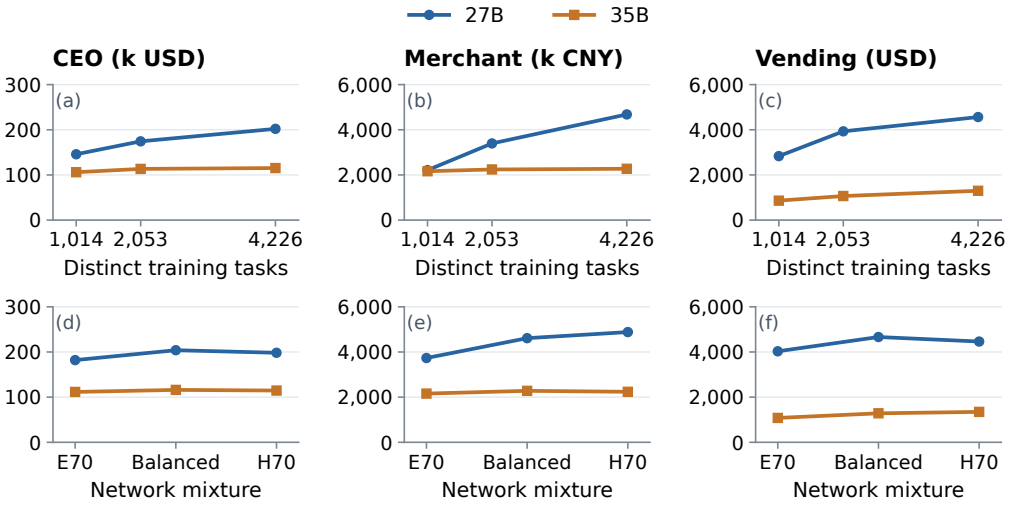


Figure 3: **Training-distribution sensitivity at a common supervision target.** Top: distinct training task count. Bottom: network easy/medium/hard mixture, with other domains fixed: E70 is easy-heavy (70/20/10), balanced is equal thirds, and H70 is hard-heavy (10/20/70). Colors and markers distinguish backbones; columns retain each benchmark's native score units.

5,120 episodes from 4,226 tasks. These stages separate the size of the generated task supply from the amount of usable model supervision; multiple selected episodes may come from the same task.

6.4 Coverage and difficulty shape transfer

More distinct tasks help at a matched supervision target. The top row of Figure 3 increases coverage from 1,014 to 2,053 and 4,226 tasks. From the smallest to the largest set, 27B scores improve by 38.6% on CEO, 111.6% on Merchant, and 61.1% on Vending. The corresponding 35B gains are 8.7%, 5.0%, and 50.7%. Task diversity therefore has the largest effect on 27B Merchant and benefits Vending for both backbones. The matched token target makes this a comparison of coverage and repeated exposure, rather than additional supervision.

The response to additional coverage also differs across targets. Increasing from 2,053 to 4,226 tasks adds only 1.6% to 35B CEO and 1.3% to 35B Merchant, but still adds 21.8% to 35B Vending. The relatively flat CEO and Merchant curves suggest diminishing benefit from further coverage within the current distribution, while Vending continues to benefit from additional task diversity.

Source composition and task coverage answer complementary training questions. On 35B Merchant, inventory-only training exceeds the four-domain mixture by 25.8%, whereas increasing mixed-data coverage from 2,053 to 4,226 tasks adds 1.3%. This contrast suggests that source choice deserves particular attention for this target. For Vending, both backbones continue to benefit from broader coverage, and the full mixture gives the highest source-comparison scores. Together, the comparisons help identify whether to prioritize additional task variety or a more targeted source distribution.

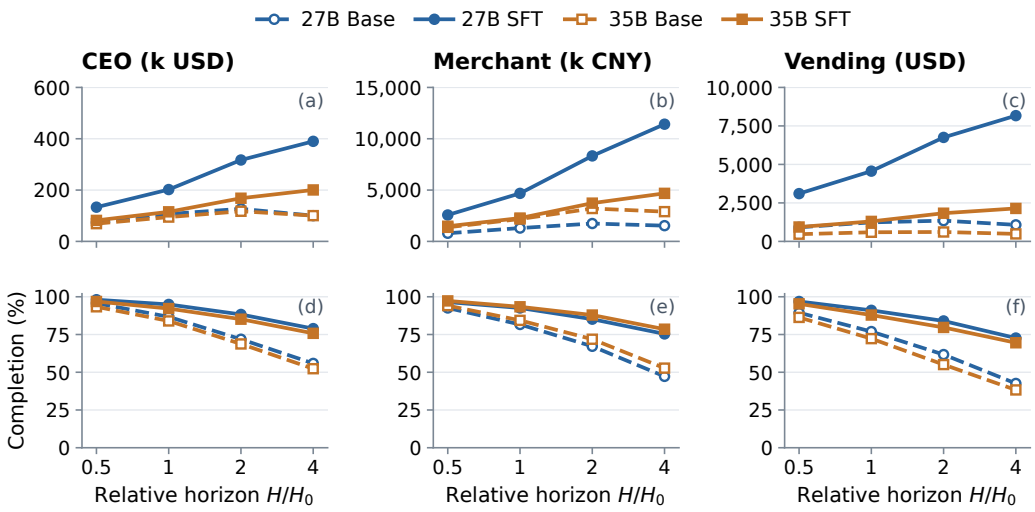


Figure 4: **Performance as physical horizon increases.** Columns correspond to CEO, Merchant, and local Vending. Top: benchmark score in native units. Bottom: fraction reaching the required horizon, including early failures in the denominator. Color denotes backbone; open/dashed and filled/solid marks denote Base and SFT. Reference durations H_0 are 365, 30, and 365 simulated days, respectively. Each condition contains 256 episodes.

A higher fraction of hard tasks is not uniformly better. The bottom row changes only the network easy/medium/hard mixture: E70 is 70/20/10, balanced uses equal thirds, and H70 is 10/20/70. The other domains, quality criteria, and teacher composition remain fixed. Balanced outperforms easy-heavy in all six comparisons. Hard-heavy, however, achieves the highest score in only two: 27B Merchant and 35B Vending, improving over balanced by 5.9% and 4.9%, respectively. Balanced is strongest in the remaining four comparisons. Difficulty labels come from prespecified 27B Base evaluations, so this pattern concerns a particular policy and budget. It favors retaining a range of task demands and selecting the mixture for the target objective, rather than treating hardness as a universal measure of training value.

6.5 Completion gains increase with physical horizon

Figure 4 evaluates $H/H_0 \in \{0.5, 1, 2, 4\}$, with $H_0 = 365$ simulated days for CEO, 30 for Merchant, and 365 for local Vending. Response and wall-time budgets scale with duration, while context and per-response limits remain fixed. Each condition contains 256 episodes. Horizon completion counts episodes that reach the required duration; legal early adverse terminals remain in the denominator and retain their settled scores. Local Vending uses this duration-based extension.

SFT sustains execution as the horizon grows. Completion declines for both Base and SFT, but their gap widens in every model–benchmark pair. At $4H_0$, 27B completion rises from 55.9% to 78.9% on CEO, 47.3% to 75.4% on Merchant, and 42.6% to 72.7% on Vending. For 35B, the corresponding changes are 52.3% to 75.8%, 52.7% to 78.5%, and 38.3% to 69.5%. Across the six comparisons, SFT completes 59–80 additional episodes out of 256. The advantage therefore persists as the task demands more physical steps under the same context limit.

Completion clarifies the cumulative-score curves. Scores and completion measure different aspects of long-horizon performance. More simulated time provides additional opportunities to accumulate returns, while completion measures whether the agent sustains execution through that time. At $4H_0$, SFT reaches 69.5–78.9% completion, compared with 38.3–55.9% for Base. Higher completion alongside higher scores shows that the gains extend to sustained operation, beyond the reference-duration score comparison. The widening completion gap is consistent with improved management of repeated decisions and persistent constraints.

The score curves also reveal a duration-dependent failure pattern. Every Base score peaks at $2H_0$ and then falls at $4H_0$, while every SFT score continues to increase. At $0.5H_0$, the SFT completion advantage is 2.3–9.0 percentage points; at $4H_0$, it reaches 23.0–31.3 points. The growing separation makes longer evaluations useful for distinguishing policies whose differences are smaller over short runs. Because duration and total execution budgets increase together, this comparison measures robustness to extended operation with a fixed context limit.

6.6 Internal success and demonstration structure

Improvements are largest where Base has more room to improve. Table 3 evaluates held-out tasks from the generated environments. Network recovery has the lowest Base success and the largest absolute gains: 12.0 percentage points for 27B and approximately 25.9 for 35B. Inventory improves by 9.85 and 15.53 points. Grid is already at 96.0% for 27B, leaving limited headroom, and rises to 97.6%; 35B rises from 88.0% to 100.0%. Thus the mixture improves demanding domains while maintaining strong performance on the easier internal setting.

Table 3: **Internal task success (%).** Parentheses give successes/evaluable tasks. Base and four-domain SFT follow the same model conditions as Table 1. The 35B Base network denominator excludes one unresolved outcome; the text gives planned-denominator bounds.

Weight	Grid	Inventory	Network
27B Base	96.00 (120/125)	81.06 (214/264)	48.40 (121/250)
27B four-domain SFT	97.60 (122/125)	90.91 (240/264)	60.40 (151/250)
35B Base	88.00 (110/125)	73.48 (194/264)	27.31 (68/249)
35B four-domain SFT	100.00 (125/125)	89.02 (235/264)	53.20 (133/250)

The single unresolved 35B Base network outcome places planned-denominator success at 27.2–27.6%, compared with 53.2% after SFT. Its treatment therefore leaves a 25.6–26.0-point improvement. These internal results complement the external benchmarks by testing the generated tasks' own execution and goal criteria.

Temporal structure extends beyond transcript length. The selected episodes have median lengths of 11–15 model replies, with maxima of 75 replies and 78 tool calls (Appendix G.3, Figure 6). Their long-horizon demands arise from physical transitions and delayed commitments within tool interactions. Finance has a longer output-token tail than network recovery: P90 is 18.2k versus 8.8k tokens. Consequently, episode shares and supervision shares differ: finance, grid, inventory, and network contribute 13.8%, 34.8%, 29.8%, and 21.6% of target tokens. This accounting motivates matching supervised tokens when comparing task sources and coverage.

7 Discussion and Conclusion

AdaptWorld supports learning to act and adapt over long-horizon tasks by coupling evolving evidence with persistent consequences. Reference comparisons, family assessment, and demonstration filtering connect tasks to training. Bounded recipe revision expands accepted task supply, while broader coverage and mixed sources improve downstream performance. Source specialization makes training composition an empirical design choice. SFT also sustains execution over longer physical horizons, with completion gains reaching 23–31 percentage points at $4H_0$.

Training tasks should reward consequential decisions; evaluation should measure terminal outcomes and sustained execution. Reference diagnostics establish decision value within the constructed environment; downstream comparisons measure the benefit to the trained policy. Policy interventions and matched admission/filtering ablations would further isolate component contributions.

AI USE STATEMENT

We used AI tools to aid and polish writing, draft manuscript sections, retrieve and discover related work, support research ideation and execution, and generate synthetic data. Writing assistance included preparing initial prose, reorganizing sections, refining language, and checking consistency across the manuscript. Retrieval assistance included finding relevant studies and organizing references. AI also supported the development and refinement of research ideas, inspection of project materials, and implementation of analysis and figure-generation code.

For synthetic data generation, Kimi K3 supported constrained task authoring by selecting candidate configurations and varying task presentation within the program-defined design space. Executable environment mechanisms supplied numerical facts, transition rules, and scoring. Model rollouts in these environments supplied candidate demonstrations, which underwent the execution and outcome checks described in the paper. The authors are responsible for the final manuscript, research decisions, and reported results.

REPRODUCIBILITY STATEMENT

Appendices A–D specify the task contract, execution semantics, reference interventions, and admission criteria. Appendix E details domain-specific diagnostic coverage. Appendix G reports the training cohort, optimizer settings, evaluation protocol, and supplementary results. The accompanying source includes plotting inputs in `plot_inputs/` and scripts for reproducing the figures.

REFERENCES

Axel Backlund and Lukas Petersson. Vending-bench: A benchmark for long-term coherence of autonomous agents, 2025. URL https://arxiv.org/abs/2502.15840.

Shihao Cai, Runnan Fang, Jialong Wu, Baixuan Li, Xinyu Wang, Yong Jiang, Liangcai Su, Liwen Zhang, Wenbiao Yin, Zhen Zhang, Fuli Feng, Pengjun Xie, and Xiaobin Wang. Autoforge: Automated environment synthesis for agentic reinforcement learning, 2025. URL https://arxiv.org/abs/2512.22857.

Haozhe Chen, Karthik Narasimhan, and Zhuang Liu. CEO-Bench: Can Agents Play the Long Game? arXiv preprint arXiv:2606.18543, 2026. URL https://arxiv.org/abs/2606.18543v1.

Sagar Deb and Ashwanth Krishnan. STOCKTAKE: Measuring the Gap Between Perception and Action in LLM Agents with a Fair Oracle. arXiv preprint arXiv:2607.13618, 2026. URL https://arxiv.org/abs/2607.13618.

Michael Dennis, Natasha Jaques, Eugene Vinitsky, Alexandre Bayen, Stuart Russell, Andrew Critch, and Sergey Levine. Emergent Complexity and Zero-shot Transfer via Unsupervised Environment Design. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://arxiv.org/abs/2012.02096.

Finale Doshi-Velez and George Konidaris. Hidden Parameter Markov Decision Processes: A Semiparametric Regression Approach for Discovering Latent Task Parametrizations. arXiv preprint arXiv:1308.3513, 2013. URL https://arxiv.org/abs/1308.3513.

Yan Duan, John Schulman, Xi Chen, Peter L. Bartlett, Ilya Sutskever, and Pieter Abbeel. RL^2: Fast Reinforcement Learning via Slow Reinforcement Learning. arXiv preprint arXiv:1611.02779, 2016. URL https://arxiv.org/abs/1611.02779v2.

Wei Fan, Xinjie Shen, Xudong Guo, Jianhong Tu, Yang Su, Yinger Zhang, Lianghao Deng, Fengyu Wang, Baohua Dong, Yangqiu Song, and Dayiheng Liu. E-commerce bench: Evaluating llm agents on long-horizon autonomous business operation, 2026. URL https://arxiv.org/abs/2608.30730.

Jiacheng Guo, Ling Yang, Peter Chen, Qixin Xiao, Yinjie Wang, Xinzhe Juan, Jiahao Qiu, Ke Shen, and Mengdi Wang. GenEnv: Difficulty-Aligned Co-Evolution Between LLM Agents and Environment Simulators. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=5vsYRklFJf.

Mengkang Hu, Pu Zhao, Can Xu, Qingfeng Sun, Jian-Guang Lou, Qingwei Lin, Ping Luo, and Saravan Rajmohan. AgentGen: Enhancing Planning Abilities for Large Language Model based Agent via Environment and Task Generation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1*, 2025. doi: 10.1145/3690624.3709321. URL https://doi.org/10.1145/3690624.3709321.

Xavier Hu, Jinxiang Xia, Shengze Xu, Kangqi Song, Yishuo Yuan, Guibin Zhang, JinCheng Ren, Boyu Feng, Li Lu, Tieyong Zeng, Jiaheng Liu, Minghao Liu, He Zhu, Yuchen Eleanor Jiang, Wei Wang, and Wangchunshu Zhou. Ecogym: Evaluating llms for long-horizon plan-and-execute in interactive economies, 2026. URL https://arxiv.org/abs/2602.09514.

Chengsong Huang, Zifeng Wang, Rujun Han, Jun Yan, Yanfei Chen, Zoey CuiZhu, Ke Jiang, Peng Xia, Han Yu, Yufan Zhuang, Yifei Ming, Jiaqi Pan, Bhavana Dalvi Mishra, Jiaxin Huang, Burak Gokturk, Tomas Pfister, and Chen-Yu Lee. Envharness: Awakening static worlds for agent learning, 2026. URL https://arxiv.org/abs/2608.19880.

Minbyul Jeong and Chanwoong Yoon. AgentMercury: Your Agent Can Synthesize Verifiable Environments for Business Scenarios at scale. arXiv preprint arXiv:2608.20634, 2026. URL https://arxiv.org/abs/2608.20634.

Edgar D. Klenske and Philipp Hennig. Dual Control for Approximate Bayesian Reinforcement Learning. arXiv preprint arXiv:1510.03591, 2016. URL https://arxiv.org/abs/1510.03591v2.

Junlong Li, Wenshuo Zhao, Jian Zhao, Weihao Zeng, Haoze Wu, Xiaochen Wang, Rui Ge, Yuxuan Cao, Yuzhen Huang, Wei Liu, Junteng Liu, Zhaochen Su, Yiyang Guo, Fan Zhou, Lueyang Zhang, Juan Michelini, Xingyao Wang, Xiang Yue, Shuyan Zhou, Graham Neubig, and Junxian He. The Tool Decathlon: Benchmarking Language Agents for Diverse, Realistic, and Long-Horizon Task Execution. In *International Conference on Learning Representations*, 2026. URL https://mlanthology.org/iclr/2026/li2026iclr-tool/.

Bo Liu, Simon Yu, Yiding Jiang, Ao Qu, Andrew Zhao, Zichen Liu, Junsu Kim, Zijian Zhou, Seungone Kim, Tongzheng Ren, Mickel Liu, Hanfei Yu, Zhaorun Chen, Weiyan Shi, Paul Pu Liang, Luke Zettlemoyer, Yejin Choi, and Natasha Jaques. Spade: Self-play in adaptive synthetic executable environments, 2026. URL https://arxiv.org/abs/2608.19197.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744. Curran Associates, Inc., 2022. doi: 10.52202/068431-2011. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf.

Yijun Pan, Yukun Lian, Kunyu Shi, Junbo Li, Hongwei Xue, Sicong Xie, Guannan Zhang, and Xiaoying Xing. Business arena: Benchmarking llm agents in a realistic marketplace, 2026. URL https://arxiv.org/abs/2608.08621.

Yash Pandya, Sahil Gupta, Sarthak Harne, Archana Yadav, Kavyansh Chourasia, Hussein Mozannar, Vibhav Vineet, Sara Abdali, Corby Rosset, Yash Lara, Ahmed Awadallah, Ece Kamar, and Akshay Nambi. Echoverse: Deep, evolving environments for training computer-use agents at scale, 2026. URL https://arxiv.org/abs/2607.28074.

Jack Parker-Holder, Minqi Jiang, Michael Dennis, Mikayel Samvelyan, Jakob Foerster, Edward Grefenstette, and Tim Rocktäschel. Evolving Curricula with Regret-Based Environment

Design. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pp. 17473–17498, 2022. URL https://proceedings.mlr.press/v162/parker-holder22a.html.

Pouya Pezeshkpour and Estevam Hruschka. From Task Solving to Robust Real-World Adaptation in LLM Agents. arXiv preprint arXiv:2602.02760, 2026. URL https://arxiv.org/abs/2602.02760.

Stephane Ross, Brahim Chaib-draa, and Joelle Pineau. Bayes-Adaptive POMDPs. In *Advances in Neural Information Processing Systems*, volume 20, 2007. URL https://papers.nips.cc/paper_files/paper/2007/hash/3b3dbaf68507998acd6a5a5254ab2d76-Abstract.html.

Xinjie Shen, Wei Fan, Xudong Guo, Jianhong Tu, Yang Su, Chuqiao Kuang, Yinger Zhang, Lianghao Deng, and Dayiheng Liu. Verifiable Hidden Dynamics Play: Generating Agentic RL Environments from Solved Mechanisms. arXiv preprint arXiv:2609.27321, 2026. URL https://arxiv.org/abs/2609.27321.

Dingfeng Shi, Jingyi Cao, Qianben Chen, Weichen Sun, Weizhen Li, Hongxuan Lu, Fangchen Dong, Tianrui Qin, King Zhu, Minghao Liu, Jian Yang, Ge Zhang, Jiaheng Liu, Changwang Zhang, Jun Wang, Yuchen Eleanor Jiang, and Wangchunshu Zhou. Taskcraft: Automated generation of agentic tasks, 2025. URL https://arxiv.org/abs/2506.10055.

Qiming Shi, Yulong Tao, Linbo Jin, Zhaolu Kang, Yibo Dou, Jiawen Zhu, Tianjun Pan, Shaokang Fu, Chengyu Wang, Siyue Li, Yaping Cheng, Di Weng, and Chengfu Huo. Merchantbench: Benchmarking llm agents for long-term coherence in e-commerce operations, 2026. URL https://arxiv.org/abs/2607.28956.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: language agents with verbal reinforcement learning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 8634–8652. Curran Associates, Inc., 2023. doi: 10.52202/075280-0377. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/1b44b878bb782e6954cd888628510e90-Paper-Conference.pdf.

Xiaoshuai Song, Haofei Chang, Guanting Dong, Yutao Zhu, Ji-Rong Wen, and Zhicheng Dou. EnvScaler: Scaling tool-interactive environments for LLM agent via programmatic synthesis. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 8326–8357, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.407. URL https://aclanthology.org/2026.findings-acl.407/.

Issa Sugiura, Daichi Hattori, Kazuo Araragi, Keita Ogawa, Shota Onose, Taro Makino, Teppei Usuki, and Takashi Ishida. Coffeebench: Benchmarking long-horizon llm agents in heterogeneous multi-agent economies, 2026. URL https://arxiv.org/abs/2606.16613.

Michael Sullivan, Mareike Hartmann, and Alexander Koller. Procedural environment generation for tool-use agents. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 18544–18562, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.936. URL https://aclanthology.org/2025.emnlp-main.936/.

Fahim Tajwar, Yiding Jiang, Abitha Thankaraj, Sumaita Sadia Rahman, J Zico Kolter, Jeff Schneider, and Russ Salakhutdinov. Training a generally curious agent. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 58227–58281, 2025. URL https://proceedings.mlr.press/v267/tajwar25a.html.

Harsh Trivedi, Tushar Khot, Mareike Hartmann, Ruskin Manku, Vinty Dong, Edward Li, Shashank Gupta, Ashish Sabharwal, and Niranjan Balasubramanian. AppWorld: A controllable world of

apps and people for benchmarking interactive coding agents. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16022–16076, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.850. URL https://aclanthology.org/2024.acl-long.850/.

Dunwei Tu, Hongyan Hao, Hansi Yang, Yihao Chen, Yi-Kai Zhang, Zhikang Xia, Yu Yang, Yueqing Sun, Xingchen Liu, Furao Shen, Qi Gu, Hui Su, and Xunliang Cai. Scaleenv: Scaling environment synthesis from scratch for generalist interactive tool-use agent training, 2026. URL https://arxiv.org/abs/2602.06820.

Tianyou Wang, Chongyang Gao, Kezhen Chen, Dong Chen, Yinghao He, Donghan Li, Wangcheng Xu, Hongjiu Zhang, and Chi Li. FM-Bench: A Benchmark for Long-Horizon Management with Competing Agents. arXiv preprint arXiv:2608.18423, 2026a. URL https://arxiv.org/abs/2608.18423.

Zhaoyang Wang, Yiming Liang, Xuchao Zhang, Qianhui Wu, Siwei Han, Anson Bastos, Rujia Wang, Chetan Bansal, Baolin Peng, Jianfeng Gao, Saravan Rajmohan, and Huaxiu Yao. SynthAgent: Adapting web agents with synthetic supervision. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15730–15752, San Diego, California, United States, July 2026b. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.716. URL https://aclanthology.org/2026.acl-long.716/.

Zhaoyang Wang, Canwen Xu, Boyi Liu, Yite Wang, Siwei Han, Zhewei Yao, Huaxiu Yao, and Yuxiong He. Agent world model: Infinity synthetic environments for agentic reinforcement learning, 2026c. URL https://arxiv.org/abs/2602.10090.

Xiaojun Wu, Cehao Yang, Honghao Liu, Xueyuan Lin, Zhichao Shi, Hao Zhou, Xuhui Jiang, Chengjin Xu, Jia Li, and Jian Guo. Envs-FORGE: Frontier-Optimized Reward-Grounded Environment Synthesis for Agent RL. arXiv preprint arXiv:2608.14312, 2026. URL https://arxiv.org/abs/2608.14312.

Zhiheng Xi, Jixuan Huang, Chenyang Liao, Baodai Huang, Honglin Guo, Jiaqi Liu, Rui Zheng, Junjie Ye, Jiazheng Zhang, Wenxiang Chen, Wei He, Yiwen Ding, Guanyu Li, Zehui Chen, Zhengyin Du, Xuesong Yao, Yufei Xu, Jiecao Chen, Tao Gui, Zuxuan Wu, Qi Zhang, Xuanjing Huang, and Yu-Gang Jiang. AgentGym-RL: An Open-Source Framework to Train LLM Agents for Long-Horizon Decision Making via Multi-Turn RL. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=ZgCCDwcGwn.

Minrui Xu, Zilin Wang, Mengyi DENG, Zhiwei Li, Zhicheng Yang, Xiao Zhu, Yinhong Liu, Boyu Zhu, Baiyu Huang, Chao Chen, Heyuan Deng, Fei Mi, Lifeng Shang, Xingshan Zeng, and Zhijiang Guo. Envfactory: Scaling tool-use agents via executable environments synthesis and robust rl, 2026. URL https://arxiv.org/abs/2605.18703.

Yirong Zeng, Shen You, Jinhang Feng, Yufei Liu, Xiao Ding, Yutai Hou, Hao Cong, Yuxian Wang, Wu Ning, Wang Xu, and Bibo Cai. Envcraft: Synthesizing executable environments in agentic rl for claw-like agent, 2026. URL https://arxiv.org/abs/2609.05576.

Jiayi Zhang, Yiran Peng, Fanqi Kong, Cheng Yang, Yifan Wu, Zhaoyang Yu, Jinyu Xiang, Jianhao Ruan, Jinlin Wang, Maojia Song, HongZhang Liu, Xiangru Tang, Bang Liu, Chenglin Wu, and Yuyu Luo. AutoEnv: Automated Environments for Measuring Cross-Environment Agent Learning. arXiv preprint arXiv:2511.19304, 2025. URL https://arxiv.org/abs/2511.19304.

Linghua Zhang, Jun Wang, Jingtong Wu, and Zhisong Zhang. Retailbench: Evaluating long-horizon autonomous decision-making and strategy stability of llm agents in realistic retail environments, 2026. URL https://arxiv.org/abs/2603.16453.

Yuhang Zhou, Kai Zheng, Haoling Li, Dengyun Peng, Can Xu, and Jingjing Chen. TurnOPD: Making On-Policy Distillation Turn-Aware for Efficient Long-Horizon Agent Training. arXiv preprint arXiv:2607.05804, 2026a. URL https://arxiv.org/abs/2607.05804v1.

Zijian Zhou, Ao Qu, Zhaoxuan Wu, Sunghwan Kim, Alok Prakash, Daniela Rus, Jinhua Zhao, Bryan Kian Hsiang Low, and Paul Pu Liang. MEM1: Learning to Synergize Memory and Reasoning for Efficient Long-Horizon Agents. In *International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=XY8AaxDSLb.

A Complete Contract and Information Boundary

A recipe defines a distribution over executable worlds. Its identity includes the domain implementation version, mechanism configuration, parameter support and weights, initialization, public interface, goals, and utility. An episode additionally fixes a parameter draw and exogenous random streams. Presentation identity is recorded separately. Changing a delay creates a new recipe even if its parameter prior is unchanged.

Full family contract and sufficient state. A domain d determines state, action, feedback, and utility semantics. A recipe ψ fixes a mechanism configuration and parameter prior. Its complete contract is

$$\mathcal{T}_\psi = (\mathcal{S}_d, \mathcal{A}_d, \mathcal{Y}_d, \Theta_\psi, \mu_\psi, p_{0,\psi}, O_{0,\psi}, K_\psi, \mathcal{A}^{\text{legal}}_\psi, H_\psi, \text{Stop}_\psi, J_\psi, G_\psi). \tag{5}$$

The entries specify state, request, feedback, and hidden-parameter spaces; the prior; state initialization and initial disclosure; the joint transition–feedback kernel; legal executed actions; physical horizon; stopping; utility; and the goal predicate (Doshi-Velez & Konidaris, 2013; Ross et al., 2007). Hidden parameters affect decisions through uncertain rules, dynamic hidden state through evolving conditions, commitments through delayed consequences, and selectable observations through acquisition costs and release times (Doshi-Velez & Konidaris, 2013; Ross et al., 2007; Klenske & Hennig, 2016). Each family declares which of these decision requirements its mechanisms support and binds them to the admission diagnostics in Section 4.3.

The full generator state is

$$S_t = (\tau_t, X_t, Z_t, Q^{\text{act}}_t, Q^{\text{obs}}_t, M_t), \tag{6}$$

Here τ_t is physical time, X_t resources, Z_t evolving hidden state, Q^{act}_t pending actions, Q^{obs}_t pending observations, and M_t cumulative settlement quantities. Pending observations retain the sampled value and release time; pending actions retain their future completion conditions. Any pre-sampled calendar needed for future transitions also belongs to private state. Thus (S_t, θ) is Markov even when past actions have delayed consequences (Doshi-Velez & Konidaris, 2013; Ross et al., 2007).

Public-history policies. Public feedback includes initial materials, released measurements, rewards, errors, action masks, execution receipts, and termination information. With request history and the joint hidden-state belief,

$$h_t = (Y_0, A_0, \ldots, A_{t-1}, Y_t), \quad A_t \sim \pi(\cdot \mid h_t), \quad b_t(s, \theta) = \Pr(S_t = s, \theta \mid h_t). \tag{7}$$

The utility objective is $\max_\pi \mathbb{E}_{\theta,\omega,\pi}[J_\psi]$ over public-history policies that validly settle almost surely under functioning execution, where ω is exogenous randomness. The belief represents uncertainty over both fixed rules and current state (Doshi-Velez & Konidaris, 2013; Ross et al., 2007). This belief formalizes the information available to a public-history policy; a language model can use that information without explicitly representing the distribution. Evaluation additionally measures completion and success under a finite execution budget.

Initial disclosure and filtering. The initial kernel is $O_{0,\psi}(y \mid s, \theta)$. Bayes filtering for the joint hidden parameter and state is (Ross et al., 2007; Doshi-Velez & Konidaris, 2013)

$$b_0(s, \theta) \propto O_{0,\psi}(Y_0 \mid s, \theta) p_{0,\psi}(s \mid \theta) \mu_\psi(\theta), \tag{8}$$

$$b_{t+1}(s', \theta) = \frac{\int K_\psi(s', Y_{t+1} \mid s, A_t; \theta) b_t(s, \theta)\, ds}{\int\int\int K_\psi(\tilde{s}, Y_{t+1} \mid s, A_t; \vartheta) b_t(s, \vartheta)\, ds\, d\tilde{s}\, d\vartheta}. \tag{9}$$

Sums replace integrals for discrete variables. Rewards, tool errors, action masks, receipts, and terminal flags jointly form the observation, with likelihoods determined by the physical transition.

A controller that uses only some channels is an approximation to this information problem (Ross et al., 2007).

Ongoing hidden-state uncertainty. State tracking requires private disturbances to introduce decision-relevant uncertainty after initialization, with public evidence capable of updating it (Ross et al., 2007). Deterministic drift of an unknown initial condition is insufficient. A pre-sampled private calendar remains part of the Markov state, but its future entries are excluded from the initial physical variables when applying this structural criterion. The requirement concerns the information process, independently of when the simulator draws random numbers.

Legal actions and stopping. The POMDP interaction index t and the state clock τ_t are distinct. At each interaction the policy submits $A_t = A_t^{\mathrm{req}}$ in the public request space. The validator produces an executed action A_t^{exec} in $\mathcal{A}_\psi^{\mathrm{legal}}(S_t)$ or a rejection outcome. The kernel is defined on the entire request space and includes validation, projection, time advancement, and feedback. The receipt reports the executed action or declared rejection. If exact legality depends on private state, this receipt is part of the observation law. A rejected request follows its declared transition, including time and budget charges. A legal no-op or fallback is available wherever the protocol requires a decision. Reading an existing public record may leave τ_t unchanged; such a call remains an interaction and consumes its declared model budget. Physical transitions, queue deadlines, and horizon stopping use τ_t.

The stopping rule specifies horizon termination and any earlier absorbing state, such as insolvency. It also specifies whether outstanding commitments mature, receive salvage value, incur cancellation charges, or expire. No post-horizon resource or observation is implicitly available. Terminal settlement runs once and produces a signed or hashed record binding the trajectory identity. The mathematical utility objective ranges over public-history policies that validly settle almost surely. The diagnostic controllers must satisfy this execution condition through bounded solvers and legal fallbacks; a controller failure blocks family admission. Learned agents are additionally evaluated on their probability of completing this contract.

Information access. The generator and trusted verifier may read private state. The agent receives only h_t through the declared tools and task files. Reference controllers use the same public information, with their own explicitly allocated computation. File separation and replay are necessary implementation checks; proving isolation additionally requires adversarial access tests. A shared seed identifies a world for evaluators and is not exposed as a way to reconstruct private randomness.

B Executable Completion, Success, and Quality

For a trusted generated-task execution record ξ, contract completion and success are

$$C_\psi(\xi) = \mathbf{1}[\text{prescribed terminal reached and settled}], \qquad \mathrm{Succ}_\psi(\xi) = C_\psi(\xi)V_\psi(\xi)G_\psi(\xi). \quad (10)$$

A prescribed terminal includes a legal early absorbing state. Horizon completion C_H in Figure 4 instead requires reaching the designated physical duration: an early insolvency can have $C_\psi = 1$ and $C_H = 0$. Here V_ψ verifies execution and G_ψ is the predeclared goal. Terminal utility J_ψ evaluates quality on valid completed episodes. Verification checks the actual trajectory, including legal rejections; it does not require matching one reference sequence (Shi et al., 2025; Wu et al., 2026; Shen et al., 2026; Xu et al., 2026).

A verifier reconstructs executed actions from the trusted log and replays the versioned simulator using the private world identity (Shi et al., 2025; Wu et al., 2026; Shen et al., 2026; Xu et al., 2026). It checks action normalization, resource constraints, queue order, disclosures, termination, and settlement, with predeclared numerical tolerances. An action that the interface legally rejects is not itself a corrupt execution. An infrastructure error that prevents verification leaves the outcome unresolved.

The validity of a partial prefix can be checked even when $C = 0$; it does not confer success. Completion C, valid completion CV, and success CVG use explicit evaluation denominators. Utility

Table 4: Outcome semantics. A dash indicates an unavailable value, not zero utility.

Outcome	C	V	Success	Utility treatment
Valid terminal satisfying goal	1	1	1	Report terminal J
Valid adverse terminal or goal miss	1	1	0	Report terminal J
Model budget exhausted before terminal	0	0/1	0	Terminal J unavailable
Proven agent-caused invalid execution	0/1	0	0	No valid terminal J
Infrastructure interruption or missing log	–	–	–	Missing; separate audit

statistics describe valid completions and are accompanied by their selection rate. Reference returns are achievable comparator values, not optimal bounds.

Frozen goal predicates. The domain goal predicates are: finance, nonnegative net terminal investment profit without insolvency; grid, served-demand fraction at least 0.8; inventory, nonnegative net profit after private unfilled-demand penalties; network, served-time fraction at least 0.5. Completion and execution validity are separate gates. Resource constraints are enforced by the simulator, while continuous utility distinguishes cost, waste, and fees among successful outcomes. Each task freezes its goal predicate before collection and evaluation.

For cross-domain diagnostics, utility scales are fixed from public quantities: finance uses net terminal profit divided by initial capital, grid uses negative operating cost divided by $50H$ times total public base demand, inventory uses net penalized profit divided by initial cash, and network uses service steps minus fees divided by H. These scales support within-domain reference comparisons. Reported results retain their domain-specific units.

Missingness and failures. Let $N = S+F+M$ be planned model evaluations, comprising verified successes, model failures, and infrastructure-missing outcomes. The unresolved outcomes place the finite-cohort success fraction in

$$[S/N, (S + M)/N]. \tag{11}$$

This range captures unresolved outcomes; sampling uncertainty is a separate quantity. Complete-case success $S/(S + F)$ is descriptive and cannot alone assign difficulty. With missing cases, confidence sets take the envelope of bounds treating all missing outcomes as failures or successes. Difficulty validation requires the resulting set to lie inside its target band. Family-reference acceptance requires complete technical verification; unresolved missing comparisons prevent admission.

Transport recovery follows a frozen policy that retains original attempts, retry causes, and their correspondence to planned evaluations. Legal low scores are retained. Timeouts caused by the declared model budget count as model failures; transport outages are diagnosed separately, and unresolved causes remain missing.

C Mechanism Examples and Controller Interventions

Financial evidence and persistent commitments. The financial runtime exposes public price histories, macro signals, regime-warning information, account state, and reports under configured disclosure rules. Reports contain repeated noisy estimates and additional fields; their length increases extraction and context demands. Holdings, transaction fees, lockups, and pending cash settlements affect future choices. An optional dividend mechanism tracks qualifying holding streaks and pays according to configured eligibility, period, and rate; a substantial position reduction can reset eligibility. Finance emphasizes inference from released information and decisions under persistent commitments (Doshi-Velez & Konidaris, 2013; Ross et al., 2007; Klenske & Hennig, 2016). Selectable paid observations are evaluated in the network domain.

Finance tasks span approximately 110–130 market days. Domain recipes specify their own physical duration; shorter adapter configurations use 30, 40, or 60 days. A programmatic loop can advance multiple days within one model response. We therefore distinguish physical days, model responses, context tokens, and commitment spans. These task-level durations describe the training environments, while Figure 4 scales the external evaluation horizon.

Inventory evidence depends on action. The examples use physical time τ, distinct from interface index t. For available inventory I_τ and demand D_τ, sales are $V_\tau = \min(D_\tau, I_\tau)$. When the interface reports the strict stockout indicator $U_\tau = \mathbf{1}[D_\tau > I_\tau]$, the observation likelihood is

$$p(V_\tau = v, U_\tau = u \mid I_\tau, \theta, Z_\tau) = \begin{cases} p(D_\tau = v \mid \theta, Z_\tau), & u = 0,\ v \le I_\tau, \\ \Pr(D_\tau > I_\tau \mid \theta, Z_\tau), & u = 1,\ v = I_\tau, \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

The equality case is distinguished from strict unmet demand. Earlier orders change I_τ, so they change both future reward and the censoring of future evidence (Klenske & Hennig, 2016). The inventory tasks use static demand parameters and periodic variation. Their report and sales interventions diagnose parameter inference (Doshi-Velez & Konidaris, 2013), while order delays and expiry create persistent consequences. The likelihood notation permits dynamic state Z_τ when present in a domain; it does not require a hidden demand regime in every inventory recipe.

Network sensing has an action window. A probe at τ measures pre-transition health and releases that stored measurement at $\tau + \ell_o$. A repair selected then occupies a crew for ℓ_a steps. Its service benefit can begin at $\tau + \ell_o + \ell_a$, subject to the adapter's within-step service order. If that time reaches the terminal horizon, the probe has no repair-mediated service window. It may still inform routing, so the timing experiment must either preserve and measure that channel or use a declared intervention that holds routing fixed. Information value is the net difference in full utility, including query fees, repair opportunity costs, and future faults (Klenske & Hennig, 2016; Ross et al., 2007).

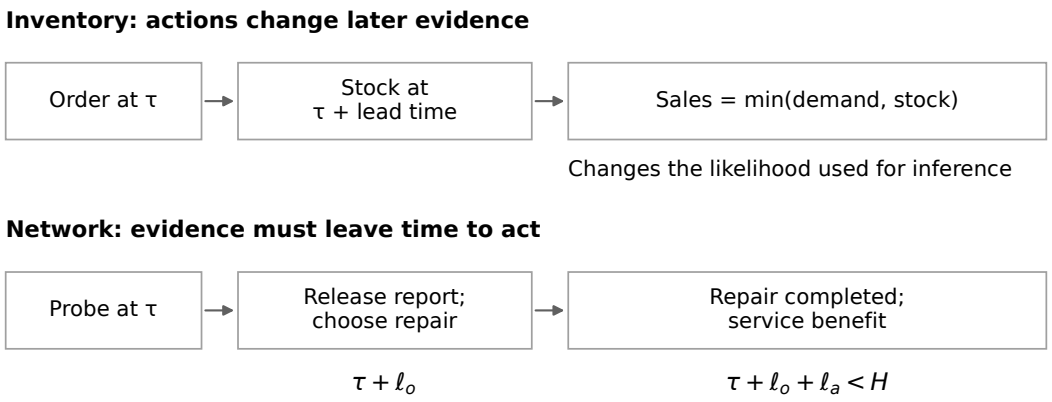


Figure 5: **Evidence and consequences share a clock.** Ordering changes later demand censoring. Network evidence is actionable only through a remaining control channel and its delay. The diagrams illustrate the timing relationships in the task mechanisms.

Four declared interventions. Represent a public reference as a parameter estimator, conditional state filter, finite-horizon planner, and sensing selector (Doshi-Velez & Konidaris, 2013; Ross et al., 2007; Klenske & Hennig, 2016). For a factorized implementation, $b_t(\theta, z) = w_t(\theta) q_t(z \mid \theta)$.

1. $-\theta$: keep parameter weights at their initialized public-evidence values while retaining conditional state propagation and filtering. The initial disclosure and prior are identical across controllers.

2. $-z$: retain parameter-weight updates but propagate each conditional dynamic-state estimate without incorporating subsequent state-measurement innovations. Parameter updates use the explicitly declared marginalized likelihood. Any indirect state information through parameters remains documented.
3. $-\mathrm{plan}$: replace the reference lookahead by one physical time step, rather than one tool request or model response, while keeping estimators, feasible actions, and objective units. Terminal values, fallback rules, and solver work are recorded. A compute-matched sensitivity control separates shorter foresight from lower computation.
4. $-\mathrm{sense}$: remove selectable paid queries from the policy while retaining all passive feedback and the same environment. Avoided fees remain in its utility. Automatic information obtained from ordinary actions is not silently removed.

Each adapter specifies the update changed by its intervention. For coupled estimators, a joint-evidence ablation measures the combined update. The resulting gaps quantify controller-level decision value; sensitivity to the reference, computational budget, or intervention can be assessed with corresponding variants.

D Family Admission and Statistical Units

Let $\mathcal{K}_\psi$ index the reference interventions declared for family ψ. Each intervention targets parameter inference, state tracking, planning, active sensing, or an explicitly specified joint evidence update (Doshi-Velez & Konidaris, 2013; Ross et al., 2007; Klenske & Hennig, 2016). A family combines structural validity, attainable goals, and reference advantages for its specified decision requirements. Domain mechanisms and intervention semantics fix $\mathcal{K}_\psi$ before search (Appendix E).

Freeze before independent assessment. The protocol fixes the prior, goals, utility scale, controllers, intervention semantics, simple baselines, legal edits, thresholds, and execution budget. Development worlds support bounded recipe editing; fresh worlds assess the frozen candidate once (Dennis et al., 2020; Parker-Holder et al., 2022). Rejected recipes remain outside the admitted set, and unresolved execution failures block admission. A revised recipe receives a new identity and new final data.

Equation 3 combines the reference-success bound with the declared capability and baseline comparisons. The following definitions specify their statistical units and confidence bounds.

Paired utility comparisons. Reference and restricted controllers receive the same parameter draw and exogenous randomness, but their observations can differ because their actions differ (Ross et al., 2007; Klenske & Hennig, 2016). For small finite priors, an independent block covers the support with prior weights. For larger priors, independent parameter draws define clusters, with repeated paths averaged within each cluster. Stratified estimates retain the declared weights. Individual paths within a cluster are not independent statistical units.

For independent block or cluster means $m_{r,c}$ in comparison c, let $\widehat{m}_c = n^{-1}\sum_r m_{r,c}$ and let s_c denote their sample standard deviation. Under an approximate normal model for these means, a multiplicity-adjusted lower bound is

$$L_c = \widehat{m}_c - t_{1-\alpha/q,n-1}\, s_c/\sqrt{n}. \tag{13}$$

Here q includes all comparisons in the admission decision. The primary comparison, threshold, and confidence procedure are fixed before assessment. Within-family error control does not imply error control across an adaptive sequence of attempted families.

Attainability and difficulty. Reference success uses a separate set of independent prior-drawn worlds. For x successes out of n, its exact one-sided binomial lower bound is

$$L_{\mathrm{succ}} = \begin{cases} 0, & x = 0, \\ \mathrm{Beta}^{-1}(\alpha/q; x, n-x+1), & x > 0. \end{cases} \tag{14}$$

Family-level success bounds use the independent reference worlds assigned to that recipe. Table 2 reports a domain-balanced reference evaluation with 32 worlds per domain. Its denominator counts evaluation worlds, rather than candidate recipes or repeated paths in a utility cluster.

For policy π with execution budget B, difficulty is

$$D_\psi(\pi, B) = 1 - \Pr[\mathrm{Succ}_\psi = 1 \mid \pi, B]. \tag{15}$$

It depends on the model, interface, and resources. The empirical network labels used in Figure 3 are defined in Appendix G.3; they are distinct from independently confirming a predeclared difficulty band with fresh worlds. Infrastructure missingness follows Equation 11.

Construction cost and coverage. All candidate evaluations, final tests, failed searches, controller transitions, and authoring calls belong to the construction budget. Construction compute includes the work per candidate as well as the number of candidates. Predetermined parameter quotas describe instance production, while outcome-filtered demonstration coverage describes the supervision available for training. Recipe acceptance, reference success, and demonstration yield therefore measure distinct stages.

E Domain Mechanisms and Capability Diagnostics

The four domains share construction, assessment, rollout, and selection stages. Dynamics, likelihoods, legal actions, utilities, and reference controllers are supplied by each domain. Candidate recipes vary these mechanism configurations within declared ranges; named diagnostics identify the reference comparisons used to assess them.

E.1 Mechanisms across domains

Finance combines unknown asset responses and evolving market conditions with holdings, fees, lockups, and settlement. Grid control combines hidden load responses and changing demand with storage, procurement, feeder, and ramp constraints. Inventory combines unknown demand coefficients, censored sales, replenishment queues, cash constraints, and perishability. Network recovery combines uncertain failure rates and component health with routing, delayed probes, repairs, and crew occupancy. Each adapter defines its public interface, observation likelihoods, terminal goal, and utility.

Table 5: **Domain mechanisms.** Each runtime defines private quantities, public feedback, and persistent action consequences. Capability admission is a separate family-level test.

Domain	Hidden quantities	Public evidence	Actions and consequences
Finance	Asset responses; market factors	Prices, reports, market signals	Holdings, fees, lockups, settlement
Grid	Load responses; demand components	Sensors, ledgers, meter readings	Charging, procurement, storage, ramp limits
Inventory	Demand rates; seasonal effects	Reports, sales, unmet-demand flags	Orders, arrivals, expiry, cash, capacity
Network	Failure rates; component health	Alarms, diagnostic reports	Routing, repair delays, crew occupancy, costs

E.2 Diagnostic coverage

Table 6 relates the domain mechanisms to their reference diagnostics. Capability requirements follow the mechanisms available in each family. In particular, an intervention on joint parameter–state beliefs measures joint evidence use, while a paid-probe restriction tests active sensing.

Six named diagnostic configurations specify interventions for grid, inventory, and network. GRID-ID freezes parameter-mixture weights; GRID-PLAN shortens an eight-step planner. INV-REPORT suppresses report and sales parameter updates; INV-SALES suppresses subsequent sales updates. NET-DIAG removes later alarms from a joint parameter–health posterior; NET-PROBE disables paid probing. Candidate recipes vary legal mechanism settings under these diagnostics, so the number of named diagnostics differs from the number of candidate recipes in Table 2.

Recipe revision searches a code-defined finite neighborhood with at most two rounds and two evaluated edits per round. A selected recipe freezes its prior, utility, references, diagnostic set, and

Table 6: **Domain mechanisms and diagnostic scope.** Recipes use the checks associated with their declared decision requirements.

Domain	Decision mechanisms	Reference and diagnostic scope
Finance	Hidden market dynamics, holdings, settlement	Public-information investment references; outcome and execution checks
Grid	Hidden load responses, dynamic filtering, storage	Parameter-mixture updates and multistep planning
Inventory	Unknown demand, censored sales, delayed orders	Initial-report inference and continued sales-based updates
Network	Joint health/rate belief, paid probes, delayed repairs	Alarm-based evidence updates and selective paid probing

thresholds before independent assessment. The same declared checks apply before and after revision. Model-demonstration filtering then evaluates episode-level execution and outcome quality.

E.3 Reusing the construction interface

A new adapter supplies private sufficient state, transition and disclosure kernels, legal actions, stopping and settlement rules, public goals, utility units, references, and versioned recipe edits. Replay and information-access checks precede model collection. Construction freezes a recipe before independent assessment, and production allocates parameter quotas before observing returns. Changes to the mechanisms or diagnostic set require renewed assessment under the updated recipe.

F Trajectory Selection and Supervised Targets

Collection and selection. Trajectories retain task and world identities, public messages, native tool schemas, sampled assistant responses, execution receipts, tool results, termination, and verification outcomes (Wang et al., 2026b; Shi et al., 2025; Xu et al., 2026). Reference-controller actions serve as diagnostics rather than language-model demonstrations. The grid, inventory, and network quality rule is Equation 4; finance retains its domain-specific selection protocol. The 80-response internal evaluation budget in Appendix G.2 is not a cap on the collected training episodes.

If a trajectory is retained according to event F, selection changes its parameter distribution, provided $\Pr(F \mid \psi) > 0$, to

$$\mu_\psi(\theta \mid F) = \frac{\mu_\psi(\theta)\Pr(F \mid \theta, \psi)}{\mathbb{E}_{\vartheta\sim\mu_\psi}\Pr(F \mid \vartheta, \psi)}. \tag{16}$$

Outcome filtering therefore reweights the production prior according to the probability of retention. The selected cohort is described by domain, task, episode, and supervised-token counts in Appendix G.1.

Context and target masks. Each supervised response keeps the context available when it was generated (Wang et al., 2026b). Grid, inventory, and network exports are checked against collection requests; finance contexts are reconstructed from native CLI records. Segmentation preserves response identities and assigns target loss to each eligible response once before repeated exposure. System, user, tool, and copied historical assistant tokens receive no target loss. No genuine context-compaction examples occur in the current cohort.

For model parameters ϕ, target token $x_{i,u}$, its recorded context $c_{i,u}$, and binary eligibility mask $m_{i,u}$, standard token-averaged SFT minimizes (Wang et al., 2026b)

$$\mathcal{L}_{\mathrm{SFT}}(\phi) = -\frac{\sum_{i,u} m_{i,u} \log \pi_\phi(x_{i,u} \mid c_{i,u})}{\sum_{i,u} m_{i,u}}. \tag{17}$$

The denominator counts unique eligible target tokens across data-parallel replicas and accumulated microbatches. Sequence-parallel partitions contribute each target token once. This normalization weights eligible target tokens equally, so longer responses contribute more loss. Eligible targets per exposure, the exposure schedule, and the tokens consumed by optimizer updates define the corresponding data and training counts.

G Cohort Protocols and Supplementary Results

G.1 Data and training configuration

The training export contains 4,226 distinct tasks, 5,120 episodes, and 8,266 segments (Table 7). An episode is a collected model interaction, whereas a segment is one exported training sequence. The selection seed is 20260921. The 27B and 35B teacher sources contribute 3,189 and 1,931 episodes, respectively, to the shared training set used by both backbones. The held-out validation export contains 224 episodes: 32 finance and 64 in each other domain, totaling 352 segments and 1,868,490 supervised tokens. Finance reserves 10% of task identities for validation; added-domain splits retain the frozen task assignments. No validation episode contributes target tokens to the training export.

Table 7: **Training-export accounting by domain.** Tokens are eligible supervised targets per full exposure. Segment counts differ from episode counts because an episode can supply multiple training sequences.

Domain	Tasks	Episodes	Segments	Supervised tokens
Finance	102	512	512	6,003,086
Grid	1,536	1,536	2,700	15,111,289
Inventory	1,536	1,536	2,710	12,934,519
Network	1,052	1,536	2,344	9,397,838
Total	4,226	5,120	8,266	43,446,732

Shared training configuration. The four-domain SFT models use two exposures and training seed 42; evaluation uses the step-402 checkpoints. The shared optimizer settings are learning rate 10^{-6}, minimum learning rate 10^{-7}, warmup ratio 0.03, Adam coefficients $(0.9, 0.95)$, weight decay 5×10^{-5}, gradient-norm clipping at 1.0, and BF16 full-parameter updates. Both backbones use eight GPUs, ZeRO-3 with CPU optimizer offload, a per-device microbatch of one, and a maximum sample length of 131,072 tokens. Table 8 gives backbone-specific packing and parallelism settings, held fixed across the training-data comparisons for that backbone. Both configurations have the same nominal packed-token capacity of $786{,}432$ tokens per optimizer update.

Table 8: **Model-specific training settings.** Packed sequences per update, excluding sequence-parallel copies, equal data-parallel replicas times microbatch size times gradient accumulation, before a possible final partial update.

Setting	Qwen3.6-27B	Qwen3.6-35B-A3B
Packing target (tokens)	131,072	98,304
Sequence-parallel degree	4	2
Data-parallel replicas	2	4
Gradient accumulation	3	2
Packed sequences/update	6	8

Supervision budget and comparison controls. One full exposure contains 43,446,732 eligible supervised tokens; two exposures define the 86,893,464-token supervision target. This budget counts eligible assistant targets; context tokens receive no target loss. Training-source comparisons change the contributing domains. Task-coverage comparisons change the number of distinct tasks while matching the supervision target through repeated exposure. The task-grouped subsets are nested, complete episodes are retained, and the subset-selection seed is 20260925. Each reply has at most eight exposures, and teacher-token shares remain within two percentage points of each original domain mixture. Network-mixture comparisons change the easy/medium/hard proportions while retaining the other domains. These controls distinguish task diversity from additional supervision.

G.2 Internal evaluation and uncertainty

The internal split has 125 grid tasks in one parameter group, 264 inventory tasks in 44 groups, and 250 network tasks in one group. Each weight has one scheduled evaluation per task. Decision and summary calls disable thinking; temperature is 0.7, top-p 0.95, and top-k 20. Budgets are 80 responses and 1,800 seconds, with at most 16,384 output tokens per response, a 131,072-token context, and a 100,000-token compaction trigger. Success requires legal completion, independent replay, and the frozen task-specific goal predicate.

Legal low scores and exhausted model budgets count as failures. The 35B Base network evaluation has 68 successes among 249 evaluable tasks out of 250 planned, giving planned-denominator success bounds of 27.2–27.6%. Table 3 reports internal success for the Base and four-domain SFT models evaluated in Table 1. Results are reported separately for grid, inventory, and network recovery.

G.3 Trajectories and data composition

The training cohort contains 5,120 selected episodes, counted before segmentation or repetition. Figure 6 summarizes agent output tokens, model replies, and executed tool calls, excluding summarizer outputs. The maximum counts are 46,043 tokens, 75 replies, and 78 tool calls. Episode output counts total 43,457,061, while eligible supervised targets total 43,446,732 (Table 7); output and target statistics have different inclusion criteria. Table 9 gives the corresponding quantiles. Internal evaluation uses an 80-response budget.

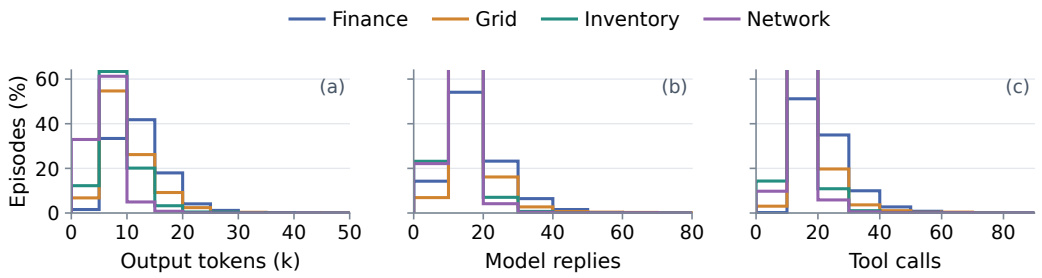


Figure 6: **Training trajectory profiles.** Histograms use 5,000-token bins and 10-reply/call bins. Percentages are normalized within domain; finance contributes 512 episodes and each other domain 1,536. Each episode is counted once before segmentation or repeated exposure.

Table 9: **Episode-level quantiles corresponding to Figure 6.** Trajectory-statistic cells report P50 / P90 over the selected episodes. A turn is a model reply; a reply can contain multiple tool calls.

Domain	Episodes	Output tokens	Turns	Tool calls
Finance	512	11,295.5 / 18,234.1	15.0 / 28.0	19.0 / 33.0
Grid	1536	8,706.5 / 15,944.0	15.0 / 23.0	16.0 / 25.0
Inventory	1536	7,859.5 / 12,710.5	12.0 / 19.0	13.0 / 20.0
Network	1536	5,679.5 / 8,819.0	11.0 / 16.0	12.0 / 17.0

Network difficulty is assigned using the first seven prespecified 27B Base evaluations, excluding demonstration calls and requiring at least five evaluable trials. Success rates at least 0.65 define easy, 0.35 to below 0.65 define medium, and below 0.35 define hard. The assignment is frozen before selecting mixtures. The three conditions share quality criteria and teacher composition, while finance, grid, and inventory exposures remain fixed. Figure 3 compares easy-heavy (70/20/10), balanced (equal thirds), and hard-heavy (10/20/70) mixtures. These labels describe empirical task difficulty under the Base policy and its fixed budget.

G.4 External scores and supplementary ranges

Table 11 gives the Base and four-domain scores underlying Table 1. CEO and local Vending use USD; Merchant uses CNY. The main table divides CEO and Merchant by 10^3 for readability. For Base score s_0 and SFT score s_1, relative change is $100(s_1 - s_0)/s_0$, computed before rounding. Scores are reported separately because benchmark objectives and units differ.

External evaluation uses CEO, Merchant, and a local Vending implementation (Chen et al., 2026; Shi et al., 2026; Backlund & Petersson, 2025). Each backbone–training-condition–benchmark combination contains 256 evaluation episodes. Scores in Table 1 are arithmetic means of terminal scores, retaining legal early adverse terminals. Training conditions share the benchmark implementation and evaluation settings within each comparison. Table 10 gives the minimum and maximum terminal scores across the same evaluation episodes. Vending denotes the local implementation throughout the manuscript.

Table 10: **Single-domain minimum–maximum scores.** Each range spans 256 evaluation episodes. Training conditions correspond to Table 1. Units: CEO, USD thousands; Merchant, CNY thousands; Vending, USD.

Model	Source	CEO	Merchant	Vending
27B	Finance	133.2–223.1	1,778.2–4,376.4	1,244.5–3,040.7
27B	Grid	109.8–183.4	1,366.2–3,343.8	1,644.4–3,904.3
27B	Inventory	124.5–208.9	2,408.7–5,974.6	2,586.9–6,155.0
27B	Network	117.2–195.1	1,291.2–3,159.5	1,415.5–3,476.2
35B	Finance	91.5–131.6	1,729.4–2,848.5	568.9–1,064.7
35B	Grid	86.1–123.6	1,720.0–2,814.4	663.3–1,238.9
35B	Inventory	89.2–128.6	1,752.4–2,882.8	880.4–1,638.5
35B	Network	87.9–125.9	1,717.8–2,808.7	619.4–1,152.9

Table 11: **External benchmark scores and relative improvements.** Values are arithmetic means over 256 evaluation episodes and underlie the Base and four-domain rows of Table 1. CEO and local Vending use USD; Merchant uses CNY. Percentage changes are computed before rounding.

Model	Benchmark	Base	SFT	Change
27B	CEO	106,694.04	202,036.46	+89.3606%
27B	Merchant	1,299,153.51	4,679,074.09	+260.1633%
27B	Vending	1,235.85	4,562.90	+269.2115%
35B	CEO	94,737.87	115,228.98	+21.6293%
35B	Merchant	2,179,983.58	2,272,282.39	+4.2339%
35B	Vending	593.75	1,298.16	+118.6375%

G.5 Experimental comparisons and aggregation

Figure 3 compares task coverage and network difficulty mixtures at the 86,893,464-token supervision target. The full-coverage condition contains 4,226 distinct tasks and corresponds to four-domain SFT in Table 1. Reduced-coverage conditions contain 1,014 and 2,053 tasks. The network-mixture comparison retains the other domains and the common training settings.

Figure 4 evaluates relative physical horizons $H/H_0 = 0.5, 1, 2, 4$. The reference-duration scores at H_0 correspond to Table 1. Horizon completion is $C_H = \mathbf{1}[\text{physical duration } H \text{ reached}]$, whereas contract completion C_ψ in Equation 10 includes legal early terminals with settled outcomes. Each horizon condition contains 256 evaluations; early failures remain in the completion denominator. The reference durations are $H_0 = 365$ simulated days for CEO, 30 days for Merchant, and 365 days for local Vending. Response and wall-time budgets scale with duration, while context and per-response limits remain fixed.

Table 2 reports construction yield. Reference successes are 29, 29, 31, and 29 out of 32 worlds for finance, grid, inventory, and network, respectively. Aggregate acceptance rises from $44/128$ to

93/128, yielding $93 \times 64 = 5{,}952$ instances. The 128 reference worlds constitute the reference-success denominator. Construction yield measures the domain-specific acceptance checks described in Section 4.3.

The accompanying `plot_inputs/` directory contains the score coordinates and episode-level statistics used in the figures. Figure-generation scripts reproduce the displays from these inputs.