PACE: POLICY-NATIVE ADAPTIVE DECISION TIMING FOR LONG-HORIZON REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-horizon reasoning requires deciding not only *what* actions to take, but *how many* to execute open-loop before replanning. This number—the *execution depth*—balances replanning cost against compounding execution errors. Most current systems either fix the execution depth as a hand-tuned scalar or adjust it at inference time with heuristic rules decoupled from the policy; we argue both can be suboptimal. In this work, we treat the execution depth as a learnable, history-conditioned variable of the policy itself, and propose PACE, a *model-native* VLM policy that jointly predicts what to execute and for how long under a hard decision budget. We evaluate PACE on four long-horizon environments spanning fully and partially observable settings and visual and textual modalities: Sliding Puzzle, Sokoban, ALFWorld, and ScienceWorld. Across all settings, PACE improves success rate by 3.1 $\sim$ 15.6 percentage points, while using fewer decisions on average, achieving empirical success–decision Pareto dominance.

1 INTRODUCTION

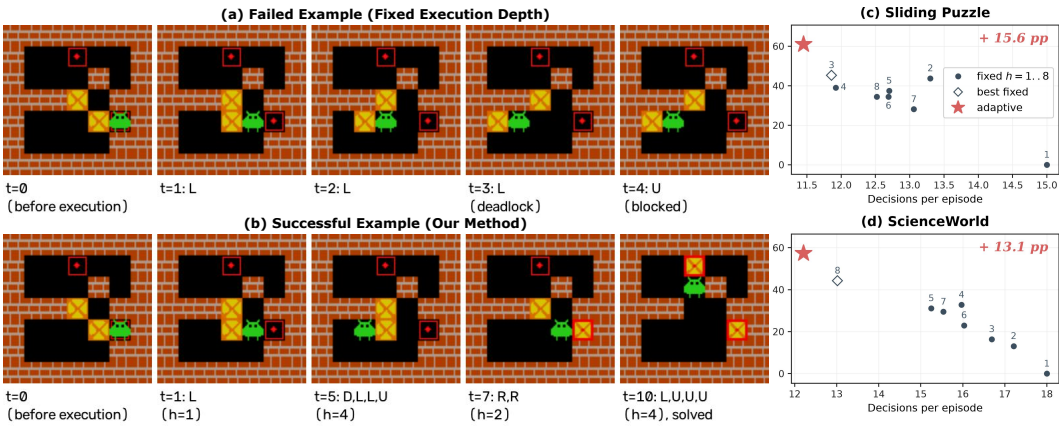

Figure 1: **Fixed execution depth versus adaptive execution depth on Sokoban game.** The player needs to push the two boxes onto the goal using four actions: left (L), right (R), up (U), down (D). (a, b) depict failed and successful examples on the same Sokoban game. (a) shows failed fixed execution depth ($h = 4$) execution on Sokoban. The model decides `L, L, L, U` without seeing any intermediate state. Its internal simulation miscounts the pushes, so the third `L` drives the lower box into a corner at $t = 3$, an irreversible deadlock, and the final `U` walks into a wall. (b) shows the same puzzle with adaptive execution depths $h \in \{1, 2, 4, 8\}$. After the single risky push, the agent re-observes ($h = 1$). It then executes longer depth of actions through safe moves and solves the puzzle in 4 decisions, within the 6-decision budget. (c, d) depict success rate versus mean decisions per episode on Sliding Puzzle and ScienceWorld. The adaptive policy (red star) Pareto-dominates every fixed horizon $h \in \{1, \ldots, 8\}$: it reaches higher success with fewer decisions (+15.6 and +13.1 pp over the best fixed h).

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.

Foundation-models are increasingly asked to solve long-horizon tasks, such as multi-step puzzles, household manipulation, and multi-stage scientific procedures, yet they persistently struggle as the horizon grows. These tasks require an agent to repeatedly decide what to do, execute the actions, observe the outcome, and decide again.

Classical work on temporal abstraction (Sutton et al., 1999; Bacon et al., 2017) addresses long horizons with closed-loop *options* whose termination conditions are task-semantic. Modern long-horizon systems (Zhao et al., 2023; Chi et al., 2025; Kim et al., 2025; Intelligence et al., 2025) take a simpler route: a plan-then-act loop, in which the policy generates a chunk of actions, executes it *open-loop*—without re-observing the environment—and then re-observes and replans. We call the number of primitive actions executed between two consecutive replans the *execution depth* h. The execution depth induces a trade-off: a short h spends more replans, each a full backbone forward pass over a fresh observation, while a long h accumulates compounding error as the imagined and realized states diverge.

Because every backbone invocation is expensive, the number of replans per episode is a genuine resource. We make it explicit as a hard *decision budget* K: each replan costs one unit regardless of its execution depth, and the episode ends once K decisions are exhausted. Under the same budget, we evaluate Pareto improvements in success versus actual decision counts and, separately, success versus environment steps. The central question is then how to spend this budget: at each decision, how many actions should be executed before looking again? In short, *when to replan*?

We formulate this as a budget-constrained decision problem: the objective is task success; the decision variable at decision k is the pair $(h_k, \mathbf{a}_k)$ of an execution depth and an action chunk of that length; and the constraint is that at most K decisions are made per episode. Existing systems largely sidestep this problem by fixing h as a single hand-tuned scalar per task. Such as, action chunking in robot learning (Zhao et al., 2023; Chi et al., 2025), modern vision–language–action models (Kim et al., 2025; Intelligence et al., 2025), step-bounded reasoning in coding, web, and embodied VLM systems (Yao et al., 2022; Yang et al., 2024; Koh et al., 2024). This is the suboptimality we argue against. In this work, we empirically show that on long-horizon visual reasoning tasks, the optimal h should be state-, task-, and budget-dependent, and a scalar cannot track a target that varies across states.

We propose **PACE**, a policy-native method with adaptive decision timing that enhances long-horizon reasoning. A lightweight depth head and a compact action decoder share a single 7B vision–language backbone (Fig. 2). The action decoder predicts an action sequence, while the depth head selects how many actions to execute before observing and planning again. The two heads are trained jointly: first an SFT warm start on counterfactual macro-action data, then a GRPO-style RL objective (Shao et al., 2024) under the hard budget K. The rollout procedure enforces the budget: episodes that use all K decisions without completing the task receive zero task-success reward. RL therefore trains the policy to succeed within this limit, without providing K or a remaining-budget counter as input. To support controlled measurement, we also release a vision–language puzzle environment with an exact-solver-grounded macro-action data pipeline (App. C).

We evaluate PACE across four environments and seven settings, covering visual puzzles and text-based embodied tasks. Using the same trained model and decision budget, PACE improves success rate by 3.1–15.6 percentage points over the best-performing fixed depth while using fewer decisions in every setting, achieving success–decision Pareto dominance. On Sokoban, it also reduces environment steps by 25.5% (30.1 vs. 40.4 per episode). The success rate advantage persists at budgets different from training, without retraining (§4).

Our contributions are summarized in three folds:

- **(C1) Problem formulation.** We formulate execution-depth selection as a decision process under a hard budget, with actions executed open-loop between observations. With the action-generation mechanism fixed, we establish a sufficient condition for a history-conditioned depth policy to strictly outperform the best fixed depth in the same admissible set (§3.1, Prop. 1; proof in App. B).

- **(C2) Unified adaptive policy and infrastructure.** We propose a single policy-native VLM with a depth head and a compact action head sharing one backbone, trained jointly under one objective (§3), along with our self-built vision–language puzzle environment and an automatic macro-action data pipeline (App. C).

- **(C3) Success–decision Pareto dominance.** Across seven settings in four environments, PACE improves success by 3.1–15.6 percentage points with fewer decisions than the best-performing fixed depth under the hard budget (Table 2, Fig. 3).

2 Related Work

Reinforcement learning for long-horizon agents. RL for LLM agents improves credit assignment and training stability through group-relative optimization (Shao et al., 2024), anchor-state grouping (Feng et al., 2025), milestone-based credit assignment (Wang et al., 2026), and related methods (Li et al., 2026b; Xu et al., 2025; Wang et al., 2025). Our focus is orthogonal: PACE learns how many actions to execute before re-observing under a hard decision budget. Integrating these training advances is left to future work.

Adaptive action chunking in robotics and control. Fixed-length action chunks are used in imitation learning (Zhao et al., 2023) and TD learning (Li et al., 2026a). Subsequent work adjusts execution through inference-time rules (Liu et al., 2025; Black et al., 2026; Liang et al., 2026; Xu et al., 2026; Nie et al., 2026), learned values (Shin et al., 2026; Gireesh et al., 2026), or an RL-trained horizon head over a frozen action policy (Zhao et al., 2026). PACE focuses on allocating a hard decision budget: it jointly learns actions and execution depth, balancing extended execution against opportunities for feedback across fully and partially observable environments.

Adaptive execution in foundation-model agents. Concurrent work SPACE (Yang et al., 2026b) combines multi-action execution with skill distillation and execution-time controls. HPL (Gao et al., 2026) uses multi-step groups as preference supervision units, while early-exit methods (Lu et al., 2025) route decisions between models. PACE jointly learns actions and execution depth through compact heads within a unified policy under a hard decision budget, without skill distillation, skill libraries, a separate preference-optimization objective, or mid-execution guards.

Temporal abstraction and costly observation. Options establish temporal abstraction and policy improvement through interruption (Sutton et al., 1999); option-critic learns termination (Bacon et al., 2017), while deliberation costs penalize frequent decisions (Harb et al., 2018). Costly-observation methods study when to acquire feedback (Nam et al., 2021; Krale et al., 2023), and adaptive computation allocates reasoning effort (Graves, 2016; Banino et al., 2021; Yang et al., 2026a). PACE learns how many actions to execute between observations, studying the trade-off between decision cost and task success under a hard budget. Proposition 1 specializes one-step policy improvement to this budgeted open-loop setting (App. B).

3 Method

PACE jointly learns actions and execution depth under a hard decision budget.

3.1 Problem Formulation

Environments and observability. An episodic environment is a tuple $(\mathcal{S}, \mathcal{A}, P, \Omega, O, r)$; when O reveals the state it is a Markov decision process (MDP), otherwise a partially observable Markov decision process (POMDP) (Kaelbling et al., 1998). Our puzzles have deterministic dynamics and fully observable states, whereas ALFWorld and ScienceWorld are partially observable: new observations can provide information about previously unobserved aspects of the environment.

The budget-K decision process. A decision is one planning round, including the initial plan and subsequent replanning rounds. Given context x—the history of observations and executed actions—the policy $\pi = (\pi^a, \pi^h)$ outputs an action chunk and an execution depth $h \in \mathcal{H} = \{1, 2, 4, 8\}$. The first h actions execute open-loop before the next observation and decision, unless the episode terminates. Each decision consumes one unit of a hard budget K, regardless of depth; exhausting K before completion is budget failure. Decision count therefore measures planning rounds, rather than wall-clock time or total inference compute. The objective is success probability $J_K(\pi)$; a fixed-depth

policy $\pi^{(h)}$ uses a constant depth with the same action mechanism. The budget is omitted from policy inputs; formal definitions appear in App. B.

Why learn the depth?

Proposition 1 (When adaptive depth improves success)**.** *Fix the action-generation mechanism and a fixed-depth policy $\pi^{(h)}$. Let $Q^{\pi}(x, d)$ denote the success probability when selecting depth d at decision history x and following π thereafter, with all budget and termination rules enforced. If a nonterminal history x^* is reached with probability $\rho > 0$ under $\pi^{(h)}$ and some $h' \in \mathcal{H}$ satisfies $\Delta = Q^{\pi^{(h)}}(x^*, h') - Q^{\pi^{(h)}}(x^*, h) > 0$, then changing depth only at x^* yields a policy π' with*

$$J_K(\pi') - J_K(\pi^{(h)}) \;=\; \rho\,\Delta \;>\; 0.$$

If this condition holds for a best fixed-depth policy in $\mathcal{H}$, the constructed policy strictly outperforms every fixed depth in that set under the same action-generation mechanism (Cor. 1). The proof specializes standard one-step policy improvement to budgeted open-loop execution. It shares the policy-improvement principle of option interruption (Sutton et al., 1999), but selects execution depth at decision boundaries rather than revising termination during execution. Proof and scope appear in App. B.

3.2 Automated Macro-Action Data Pipeline

We use exact-solver optimal paths for the puzzles and built-in task-completing demonstrations, without optimality guarantees, for ALFWorld and ScienceWorld (App. C). For each reference trajectory $(a_1, \ldots, a_L)$, every starting position u yields examples $\big(x_u, (a_u, \ldots, a_{u+m-1}), m\big)$ for $m \in \mathcal{H} = \{1, 2, 4, 8\}$ with $u + m - 1 \le L$, where x_u is the context before action a_u and m is the target action-chunk length. Each chunk is predicted from x_u without intermediate observations, matching open-loop execution.

3.3 Model Architecture

As shown in Fig. 2, PACE uses a shared Qwen2.5-VL-7B-Instruct backbone with a lightweight depth head and a compact action decoder. At decision t, the backbone encodes the interaction context into z_t, which both branches reuse. PACE predicts depth and actions directly, without generating or supervising explicit chain-of-thought text.Decoder specifications appear in App. D.

Factorized depth and action prediction. The depth head predicts a distribution over $\mathcal{H} = \{1, 2, 4, 8\}$. The action decoder predicts primitive actions from a domain-specific vocabulary $\mathcal{A}_e$ (Table 6; vocabulary sizes in Table 11), conditioned on z_t and previously generated actions. Depth selects the executed prefix and is not a decoder input: $\pi_\theta(h_t, a_t^{1:h_t} \mid x_t) = \pi_\theta^h(h_t \mid z_t) \prod_{j=1}^{h_t} \pi_\theta^a(a_t^j \mid z_t, a_t^{<j})$. The decoder generates up to $h_{\max} = 8$ actions, of which the first h_t are executed open-loop.

Compact output parameterization. Table 1 compares output parameterizations. A flat classifier requires $\sum_{h \in \mathcal{H}} |\mathcal{A}_e|^h$ depth–chunk classes, whereas our factorization uses $|\mathcal{H}|$ depth logits and $|\mathcal{A}_e|$ action logits per decoded position. It composes known actions into chunks without enumerating complete chunks as separate classes. Each action output identifies a direction or a complete command, avoiding token-by-token command generation and free-text parsing.

3.4 Training Protocol

Both SFT and RL adapt the shared backbone through LoRA, keeping its pretrained weights frozen. LoRA configurations and training hyperparameters are provided in App. D.

Action-supervised warm start. SFT updates the backbone LoRA and action decoder using teacher-forced cross-entropy on the multi-length examples from §3.2, while keeping the depth head frozen. Depth selection is learned during RL. We train three SFT models: one shared by the two puzzles, one across the five ALFWorld task types, and one across all ScienceWorld horizon groups.

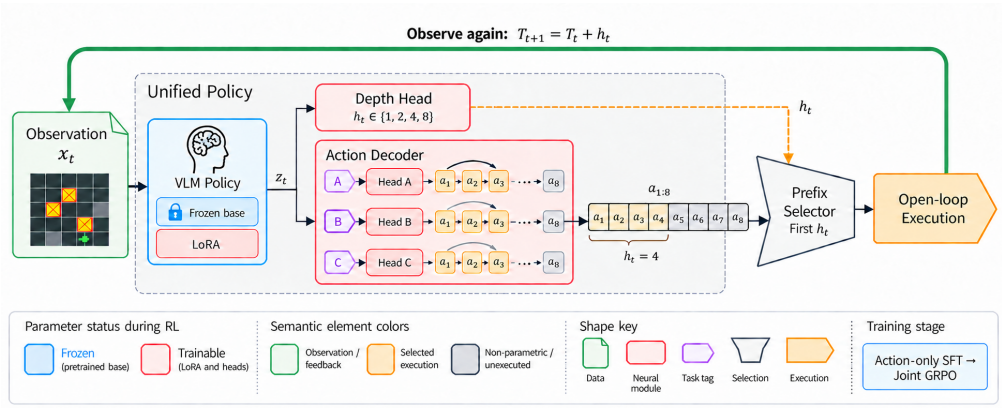

Figure 2: **PACE architecture.** The backbone representation z_t feeds a depth head and an autoregressive action decoder. The decoder generates eight actions independently of the selected depth h_t; the first h_t execute without intermediate feedback. RL jointly updates backbone LoRA and both heads.

Joint reinforcement learning. GRPO jointly updates the backbone LoRA, action decoder, and depth head under a hard decision budget. Each puzzle is trained separately; ALFWorld shares one model across five task types, and ScienceWorld shares one across horizon groups with group-specific budgets. Candidates are compared within each decision point using the rewards discussed in §4.1. Policy inputs omit budget and step counters. During training, rollouts stop after at most K decisions; unfinished episodes receive zero task-success reward, while any progress rewards remain as specified in §4.1. These budget-constrained outcomes provide the RL signal for learning action and depth choices that complete tasks within the available decisions. In both text environments, only the first action is masked to admissible commands; subsequent actions execute without intermediate feedback or updated masks. Nonterminal invalid actions do not interrupt execution.

4 Experiments

4.1 Setup and Evaluation Protocol

Tasks and benchmarks. We evaluate four environments in seven settings: Sliding Puzzle, Sokoban, ALFWorld seen/unseen (Shridhar et al., 2021), and ScienceWorld short/medium/long (Wang et al., 2022). Our visual puzzle benchmarks pair observations with exact-solver diagnostics; Sliding Puzzle is reversible, whereas Sokoban permits irreversible deadlocks. The text environments provide partially observable interaction. Domain properties and construction details appear in App. C (Table 6).

Task selection and horizon coverage. We retain 12 ScienceWorld physics and chemistry task types based on vocabulary size and reference-trajectory compatibility with the 150-step evaluation cap. Short, medium, and long groups use separate budgets to examine timing across its broad horizon range; these groups describe reference length, not calibrated difficulty. Other environments evaluate mixed-length instances under a common budget. ALFWorld retains five families, excluding Pick2, whose post-SFT success remains below 10% even under a relaxed budget. Integrating complementary

Table 1: **Output parameterization per decision.** Counts refer to output logits, not total computation. $T(h)$ is the number of text tokens used to serialize depth and h commands; V is the LM token vocabulary, and $\mathcal{A}_e$ is the domain-specific action vocabulary. PACE selects depth from $\mathcal{H} = \{1, 2, 4, 8\}$, decodes $h_{\max} = 8$ actions, and executes the selected prefix of length h.

Scheme	Prediction target	Output logits
Autoregressive LM	Depth and commands as text	$T(h)\|V\|$
Flat joint classifier	One class per depth–sequence pair	$\sum_{d\in\mathcal{H}} \|\mathcal{A}_e\|^d$
PACE	Depth and $h_{\max}$ autoregressive actions	$\|\mathcal{H}\| + h_{\max}\|\mathcal{A}_e\|$

credit-assignment advances (Feng et al., 2025; Wang et al., 2026) to improve Pick2 is outside this study's compute scope. Selection and grouping details appear in App. C.3.5 and App. C.4.5.

Data splits and vocabulary coverage. Puzzles use fresh evaluation instances from the training generators; ALFWorld uses official seen/unseen splits; ScienceWorld uses disjoint training and evaluation (task, variation) pairs within each horizon group. Text-domain vocabularies use training data only. All evaluation instances remain in the main results, including those with out-of-vocabulary commands in their reference paths, and are scored by actual rollouts (App. D.5).

Reward settings. All domains use sparse success, supplemented by solver-derived progress in the puzzles and official-score changes in ScienceWorld. ALFWorld uses sparse success alone. We add no further task-specific shaping, testing both progress-augmented and sparse-only rewards.

Decision budgets. Main training and evaluation budgets match: $K = 15$ for Sliding Puzzle, 6 for Sokoban, 9 for ALFWorld, and $6/9/18$ for ScienceWorld short/medium/long. Reference lengths inform these choices to impose decision pressure while permitting longer depths to extend action capacity to at most Kh. All timing policies share the budget within a setting; constraint severity need not match across settings. Budget generalization increases evaluation budgets without retraining (§4.3.1; Table 10).

Implementation and compute. PACE uses Qwen2.5-VL-7B-Instruct (Bai et al., 2025). Training takes 10–48 hours per domain on eight NVIDIA A100 80 GB GPUs. Training configurations and compute appear in App. D.2; additional zero-shot model configurations appear in App. E.2.

Baselines and depth coverage. Fixed-depth and heuristic policies share PACE's RL-trained checkpoint. Fixed policies execute prefixes of length $h \in \{1, \ldots, 8\}$ from the eight-action decoder, directly testing depths outside $\mathcal{H} = \{1, 2, 4, 8\}$. We select the highest-success fixed depth in hindsight per setting, retaining ties. SFT-only instead reports the best fixed depth over the same range using the pre-RL checkpoint, without its untrained depth head.

Heuristics include Repeat-last, Random-h (uniform over $\mathcal{H}$), and cumulative-entropy and per-action entropy-threshold rules inspired by AAC (Liang et al., 2026). The best-τ threshold is selected on the evaluation set per setting, favoring this baseline within its tested family. More details are discussed in App. D.4. We also evaluate the zero-shot backbone.

Metrics and Pareto dominance. We report success rate $\bar{y}$ and mean executed decisions $\bar{D}$ and environment steps $\bar{S}$ over all episodes, including failures. For $c \in \{\bar{D}, \bar{S}\}$, policy π_1 Pareto-dominates π_2 when $\bar{y}_{\pi_1} \geq \bar{y}_{\pi_2}$ and $c_{\pi_1} \leq c_{\pi_2}$, with at least one strict inequality. Success–decision and success–interaction dominance are assessed separately under matched protocols.

4.2 Main Results

Success across settings. PACE achieves the highest observed success across all seven settings (Table 2), improving over the hindsight-best fixed depth by 3.1–15.6 percentage points. It also outperforms random, repeat-last, and entropy-based timing rules evaluated with the same trained checkpoint. The corresponding success–interaction Pareto plots appear in App. D.6 (Fig. 10). On Sokoban, the modest 3.1-percentage-point success gain accompanies 25.5% fewer environment steps per episode (30.1 versus 40.4) than the best fixed depth. ScienceWorld short has lower observed success than medium and long despite shorter reference trajectories. Because the groups contain different instances and use different budgets, these rates do not isolate the effect of horizon length.

Success–decision trade-off. Figure 3 relates success to the mean number of executed decisions per episode. Compared with the highest-success fixed-depth policies on the same RL checkpoint, PACE achieves higher success with fewer decisions across all seven settings. This reduces replanning frequency and backbone context-encoding passes along the executed trajectory, with replanning cost measured in decision rounds.

Table 2: **Success rates (%) across seven settings.** PACE achieves the highest observed success in every setting. Joint-RL fixed-depth and heuristic timing policies share PACE's trained checkpoint; SFT-only uses the checkpoint before RL. For each checkpoint and setting, best fixed-h is selected in hindsight over $h \in \{1, \ldots, 8\}$. Δ compares PACE with the joint-RL best fixed depth. †The best-τ threshold is selected separately for each setting on the evaluation set, providing an optimistic comparison within the tested threshold family.

			ALFWorld		ScienceWorld		
Method	**Sliding Puzzle**	**Sokoban**	Seen	Unseen	Short	Medium	Long
Qwen2.5-VL-7B (Bai et al., 2025)	0.0	0.0	5.2	3.4	0.0	1.9	1.7
SFT-only (best fixed-h)	48.4	34.4	37.9	35.0	27.5	54.5	37.7
Repeat-last	0.0	3.1	18.1	10.3	0.0	0.0	0.0
Random-h	35.9	15.6	44.0	40.2	34.1	54.5	23.6
Entropy (cumulative)	46.9	17.2	37.9	29.9	26.1	41.8	16.4
Entropy (threshold, best-τ)†	37.5	29.7	44.8	35.0	34.8	47.3	4.9
Joint RL (best fixed-h)	45.3 (h=3)	32.8 (h=8)	45.7 (h=4/7)	38.5 (h=7/8)	34.8 (h=6)	58.2 (h=5/7)	44.3 (h=8)
PACE (ours)	**60.9**	**35.9**	**54.3**	**48.7**	**44.9**	**69.1**	**57.4**
Δ vs. Joint RL (best fixed) (pp)	+15.6	+3.1	+8.6	+10.3	+10.1	+10.9	+13.1

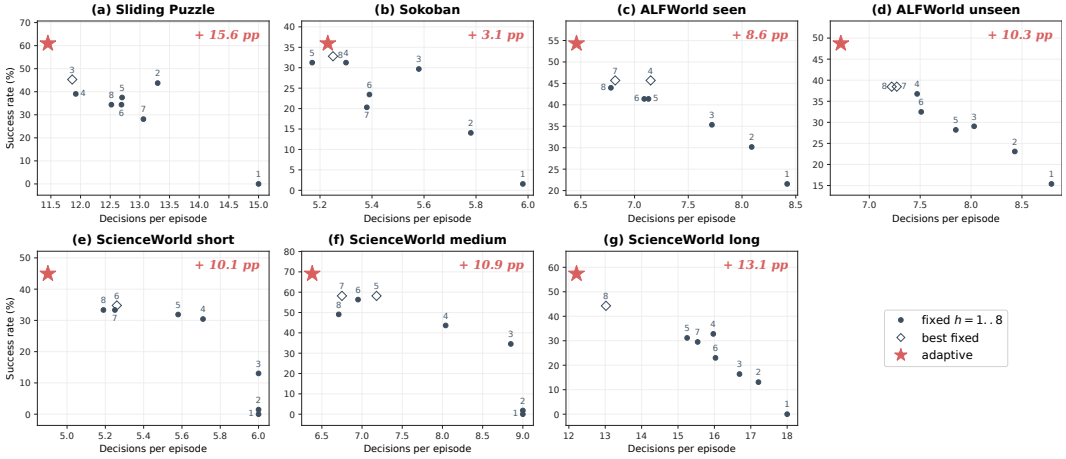

Figure 3: **Success–decision trade-offs across seven settings.** Stars denote PACE; diamonds mark the highest-success fixed depths, including ties; circles show the remaining tested fixed depths. Annotations give success gains over the best fixed depth in percentage points. Decision counts are averaged over all evaluation episodes, including failures. Axis ranges vary by setting.

4.3 Ablation Study

4.3.1 Budget Generalization without Retraining

Without retraining or changes to the input format, PACE exceeds the same-checkpoint hindsight-best fixed depth by 4.7–15.0 percentage points across all seven larger-budget settings (Table 3). The gain reaches 14.5 points on ALFWorld unseen, showing that the advantage persists on unseen layouts at a budget different from training.

4.3.2 Action-Decoder Freezing versus Joint Optimization

We compare joint optimization with a variant that freezes the action decoder at its SFT weights while updating the depth head and backbone LoRA (Table 4). Both variants use adaptive depth. Because the shared representation remains trainable, action predictions can still change; this ablation does not isolate the contribution of timing alone.

Freezing reduces success by 14.1 and 15.6 percentage points on Sliding Puzzle and Sokoban, respectively, but improves observed success by 2.6–8.2 points across all five text-domain settings. One hypothesis is that the puzzles benefit more from decoder adaptation, whereas retaining SFT

Table 3: **Budget generalization without retraining.** Each policy is evaluated at a larger decision budget than used during training, with unchanged policy inputs. Best fixed-h is selected in hindsight over $h \in \{1, \ldots, 8\}$ at the evaluation budget, using the same checkpoint as PACE. Success rates are percentages; Δ denotes PACE minus best fixed in percentage points.

Setting	$K_{\text{train}} \to K_{\text{eval}}$	PACE (ours)	Best fixed-h	Δ (pp)
Sliding Puzzle	$5 \to 15$	**56.2**	43.8	$+12.5$
Sokoban	$3 \to 4$	**28.1**	23.4	$+4.7$
ALFWorld seen	$5 \to 9$	**50.9**	43.1	$+7.8$
ALFWorld unseen	$5 \to 9$	**42.7**	28.2	$+14.5$
ScienceWorld short	$6 \to 12$	**64.2**	58.2	$+6.0$
ScienceWorld medium	$9 \to 18$	**72.2**	63.0	$+9.3$
ScienceWorld long	$18 \to 36$	**63.3**	48.3	$+15.0$

decoder weights provides useful regularization in the text domains. We leave controlled experiments investigating the mechanisms underlying these differences to future work.

Table 4: **Depth-only versus joint RL optimization.** Depth-only updates the depth head and backbone LoRA while freezing the SFT action decoder; joint optimization additionally updates the action decoder. Both use adaptive execution depth. Success rates are percentages; Δ is Depth-only minus Joint in percentage points, computed before rounding. Bold marks the higher observed success within each setting.

Setting	Depth-only	Joint (PACE)	Δ (pp)
Sliding Puzzle	46.9	**60.9**	-14.1
Sokoban	20.3	**35.9**	-15.6
ALFWorld seen	**56.9**	54.3	$+2.6$
ALFWorld unseen	**53.8**	48.7	$+5.1$
ScienceWorld short	**49.3**	44.9	$+4.3$
ScienceWorld medium	**74.5**	69.1	$+5.5$
ScienceWorld long	**65.6**	57.4	$+8.2$

4.4 Zero-Shot Depth Selection

We compare prompted self-selected depth with fixed depths $h \in \{1, 2, 4, 8\}$ within each zero-shot model. Self-selection does not consistently outperform the best tested constant: GPT-5.6-Terra achieves 57.8% versus 43.8% on Sokoban, but 46.3% versus 58.2% on ScienceWorld short. Thus, prompting variable execution depth alone does not ensure effective timing. These results provide supplementary context; PACE's gains are established through same-checkpoint comparisons. Full results and model-specific protocols appear in App. E.

5 Efficiency

Decision and interaction costs. Beyond the per-episode trade-offs (Fig. 3; App. D.6), PACE reduces aggregate decisions per success in all seven settings and steps per success in six relative to every highest-success fixed-depth comparator, including costs from failed episodes (Table 14). On ScienceWorld medium, steps per success lie between the two tied comparators.

Backbone amortization. Reusing one backbone representation per decision gives $D/S = 1/\bar{\ell}$ encodings per executed action, where D and S count decisions and actions, and $\bar{\ell} = S/D$ is mean executed depth. This reduces encoding work per action relative to one-action decisions, with potential latency and throughput benefits.

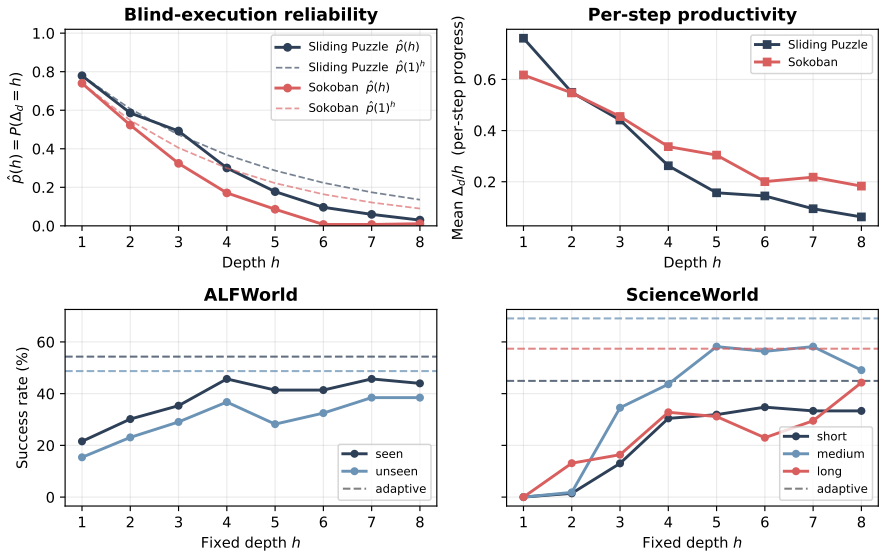

Figure 4: **Execution reliability and depth–success frontiers.** Top: puzzle reliability $\hat{p}(h)$ versus $\hat{p}(1)^h$ and per-step solver progress. Bottom: fixed-depth versus PACE success in ALFWorld and ScienceWorld using the same checkpoint and decision budget within each setting.

6 Discussion

Execution reliability versus decision cost. Longer execution saves decisions but reduces puzzle optimal-path reliability below $\hat{p}(1)^h$ at larger depths, with generally lower per-step solver progress (Fig. 4). Shallow execution instead limits action capacity under the budget. In text domains, no tested fixed depth matches PACE's success on the same checkpoint.

State-conditioned depth allocation. In both puzzles, mean selected depth decreases from near-goal to far-from-goal states under matched budgets (Fig. 5) and relaxed evaluation budgets (App. D.7). PACE uses all four depths in every setting (Fig. 12). On ScienceWorld medium, it selects $h = 8$ in 82.9% of decisions yet exceeds fixed-8 success by 20 percentage points (Fig. 4), showing that a dominant depth preference can coexist with adaptive gains. These patterns do not identify which individual depth choices cause the improvement. Training dynamics and failure cases in App. D.3.

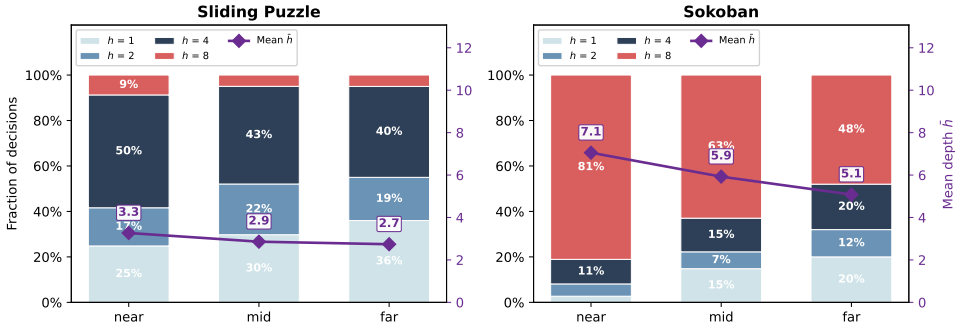

Figure 5: **State-conditioned depth under matched training and evaluation budgets.** Bars show selected-depth distributions by remaining solver distance; purple lines show mean depth. Both puzzles exhibit a near-deep/far-shallow pattern.

7 Conclusion

We introduced PACE, which jointly learns actions and execution depth under a hard decision budget. Across four environments and seven settings, it improves success over the same-checkpoint best fixed depth while using fewer decisions, with gains persisting under budget generalization. State-conditioned depth patterns further characterize the learned policy. These results support policy-native decision timing: learning what actions to take and when to reason again.

REFERENCES

Anthropic. Claude, 9 2026. URL https://www.anthropic.com. Large language model.

Pierre-Luc Bacon, Jean Harb, and Doina Precup. The option-critic architecture. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025. URL https://arxiv.org/abs/2502.13923.

Andrea Banino, Jan Balaguer, and Charles Blundell. Pondernet: Learning to ponder. *arXiv preprint arXiv:2107.05407*, 2021.

Kevin Black, Manuel Galliker, and Sergey Levine. Real-time execution of action chunking flow policies. *Advances in Neural Information Processing Systems*, 38:33383–33407, 2026.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

Marc-Alexandre Côté, Ákos Kádár, Xingdi Yuan, Ben Kybartas, Tavian Barnes, Emery Fine, James Moore, Ruo Yu Tao, Matthew Hausknecht, Layla El Asri, Mahmoud Adada, Wendy Tay, and Adam Trischler. Textworld: A learning environment for text-based games, 2019. URL https://arxiv.org/abs/1806.11532.

DeepSeek. Deepseek-v4-flash-vision-exp. https://huggingface.co/deepseek-ai/DeepSeek-V4-Flash-Vision-Exp, 2026. Accessed: 2026-09-24.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for llm agent training, 2025. URL https://arxiv.org/abs/2505.10978.

Heyang Gao, Zexu Sun, Erxue Min, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, and Xu Chen. Solving the granularity mismatch: Hierarchical preference learning for long-horizon llm agents. In *International Conference on Learning Representations*, volume 2026, pp. 67081–67110, 2026.

Nandiraju Gireesh, Yuanliang Ju, and He Wang. Adaptive q-chunking for offline-to-online reinforcement learning, 2026. URL https://arxiv.org/abs/2605.05544.

GLM-5-Team, :, Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, Chenzheng Zhu, Congfeng Yin, Cunxiang Wang, Gengzheng Pan, Hao Zeng, Haoke Zhang, Haoran Wang, Huilong Chen, Jiajie Zhang, Jian Jiao, Jiaqi Guo, Jingsen Wang, Jingzhao Du, Jinzhu Wu, Kedong Wang, Lei Li, Lin Fan, Lucen Zhong, Mingdao Liu, Mingming Zhao, Pengfan Du, Qian Dong, Rui Lu, Shuang-Li, Shulin Cao, Song Liu, Ting Jiang, Xiaodong Chen, Xiaohan Zhang, Xuancheng Huang, Xuezhen Dong, Yabo Xu, Yao Wei, Yifan An, Yilin Niu, Yitong Zhu, Yuanhao Wen, Yukuo Cen, Yushi Bai, Zhongpei Qiao, Zihan Wang, Zikang Wang, Zilin Zhu, Ziqiang Liu, Zixuan Li, Bojie Wang, Bosi Wen, Can Huang, Changpeng Cai, Chao Yu, Chen Li, Chengwei Hu, Chenhui Zhang, Dan Zhang, Daoyan Lin, Dayong Yang, Di Wang, Ding Ai, Erle Zhu, Fangzhou Yi, Feiyu Chen, Guohong Wen, Hailong Sun, Haisha Zhao, Haiyi Hu, Hanchen Zhang, Hanrui Liu, Hanyu Zhang, Hao Peng, Hao Tai, Haobo Zhang, He Liu, Hongwei Wang, Hongxi Yan, Hongyu Ge, Huan Liu, Huanpeng Chu, Jia'ni Zhao, Jiachen Wang, Jiajing Zhao, Jiamin Ren, Jiapeng Wang, Jiaxin Zhang, Jiayi Gui, Jiayue Zhao, Jijie Li, Jing An, Jing Li, Jingwei Yuan, Jinhua Du, Jinxin Liu, Junkai Zhi, Junwen Duan, Kaiyue Zhou, Kangjian Wei, Ke Wang, Keyun Luo, Laiqiang Zhang, Leigang Sha, Liang Xu, Lindong Wu, Lintao Ding, Lu Chen, Minghao Li, Nianyi Lin, Pan Ta, Qiang Zou, Rongjun Song, Ruiqi Yang, Shangqing Tu, Shangtong Yang, Shaoxiang Wu, Shengyan Zhang, Shijie Li, Shuang Li, Shuyi Fan, Wei Qin, Wei Tian, Weining Zhang, Wenbo Yu, Wenjie Liang, Xiang Kuang, Xiangmeng Cheng, Xiangyang Li, Xiaoquan Yan, Xiaowei Hu, Xiaoying Ling, Xing Fan, Xingye Xia, Xinyuan Zhang, Xinze Zhang, Xirui Pan, Xu Zou, Xunkai Zhang, Yadi Liu, Yandong Wu, Yanfu Li, Yidong Wang, Yifan Zhu, Yijun Tan, Yilin Zhou, Yiming Pan, Ying Zhang, Yinpei Su,

Yipeng Geng, Yong Yan, Yonglin Tan, Yuean Bi, Yuhan Shen, Yuhao Yang, Yujiang Li, Yunan Liu, Yunqing Wang, Yuntao Li, Yurong Wu, Yutao Zhang, Yuxi Duan, Yuxuan Zhang, Zezhen Liu, Zhengtao Jiang, Zhenhe Yan, Zheyu Zhang, Zhixiang Wei, Zhuo Chen, Zhuoer Feng, Zijun Yao, Ziwei Chai, Ziyuan Wang, Zuzhou Zhang, Bin Xu, Minlie Huang, Hongning Wang, Juanzi Li, Yuxiao Dong, and Jie Tang. Glm-5: from vibe coding to agentic engineering, 2026. URL https://arxiv.org/abs/2602.15763.

Alex Graves. Adaptive computation time for recurrent neural networks. *arXiv preprint arXiv:1603.08983*, 2016.

Jean Harb, Pierre-Luc Bacon, Martin Klissarov, and Doina Precup. When waiting is not an option: Learning options with a deliberation cost. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization, 2025. URL https://arxiv.org/abs/2504.16054.

Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025.

Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Russ Salakhutdinov, and Daniel Fried. Visualwebarena: Evaluating multimodal agents on realistic visual web tasks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 881–905, 2024.

Merlijn Krale, Thiago D Simao, and Nils Jansen. Act-then-measure: Reinforcement learning for partially observable environments with active measuring. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 33, pp. 212–220, 2023.

Qiyang Li, Zhiyuan Paul Zhou, and Sergey Levine. Reinforcement learning with action chunking. *Advances in Neural Information Processing Systems*, 38:55518–55553, 2026a.

Yuanfan Li, Qi Zhou, Wenjing Duan, and Lu Chen. When denser credit is not enough: Evidence-calibrated policy optimization for long-horizon llm agent training. *arXiv preprint arXiv:2606.05885*, 2026b.

Yuanchang Liang, Xiaobo Wang, Kai Wang, Shuo Wang, Xiaojiang Peng, Haoyu Chen, David Kim Huat Chua, and Prahlad Vadakkepat. Adaptive action chunking at inference-time for vision-language-action models. *arXiv preprint arXiv:2604.04161*, 2026.

Yuejiang Liu, Jubayer Hamid, Annie Xie, Yoonho Lee, Max Du, and Chelsea Finn. Bidirectional decoding: Improving action chunking via guided test-time sampling. In *International Conference on Learning Representations*, volume 2025, pp. 4594–4627, 2025.

Qingyu Lu, Liang Ding, Siyi Cao, Xuebo Liu, Kanjian Zhang, Jinxia Zhang, and Dacheng Tao. Runaway is ashamed, but helpful: On the early-exit behavior of large language model-based agents in embodied environments. In *EMNLP (Findings)*, pp. 24014–24027, 2025.

HyunJi Alex Nam, Scott Fleming, and Emma Brunskill. Reinforcement learning with state observation costs in action-contingent noiselessly observable markov decision processes. *Advances in Neural Information Processing Systems*, 34:15650–15666, 2021.

Junnan Nie, Jiayi Li, Jiachen Zhang, Junyi Lao, Chenghao Liu, Tianle Zhang, and Songfang Huang. Pace: Phase-aware chunk execution for robot policies with action chunking, 2026. URL https://arxiv.org/abs/2606.00537.

Qwen Team. Qwen3.8-Flash-Next: A new architecture, towards ultimate cost-efficiency, August 2026a. URL https://qwen.ai/blog?id=qwen3.8-flash-next.

Qwen Team. Qwen3.8-Max: A new bar for coding and cowork, August 2026b. URL https://qwen.ai/blog?id=qwen3.8.

Max-Philipp B. Schrader. gym-sokoban. https://github.com/mpSchrader/gym-sokoban, 2018.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Sahil Sharma, Aravind Srinivas, and Balaraman Ravindran. Learning to repeat: Fine grained action repetition for deep reinforcement learning. *arXiv preprint arXiv:1702.06054*, 2017.

Yongjae Shin, Jongseong Chae, Seongmin Kim, Jongeui Park, and Youngchul Sung. Adaptive action chunking via multi-chunk q value estimation. *arXiv preprint arXiv:2605.10044*, 2026.

Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. Alfred: A benchmark for interpreting grounded instructions for everyday tasks, 2020. URL https://arxiv.org/abs/1912.01734.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. Alfworld: Aligning text and embodied environments for interactive learning, 2021. URL https://arxiv.org/abs/2010.03768.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, Akshay Nathan, Alan Luo, Alec Helyar, Aleksander Madry, Aleksandr Efremov, Aleksandra Spyra, Alex Baker-Whitcomb, Alex Beutel, Alex Karpenko, Alex Makelov, Alex Neitz, Alex Wei, Alexandra Barr, Alexandre Kirchmeyer, Alexey Ivanov, Alexi Christakis, Alistair Gillespie, Allison Tam, Ally Bennett, Alvin Wan, Alyssa Huang, Amy McDonald Sandjideh, Amy Yang, Ananya Kumar, Andre Saraiva, Andrea Vallone, Andrei Gheorghe, Andres Garcia Garcia, Andrew Braunstein, Andrew Liu, Andrew Schmidt, Andrey Mereskin, Andrey Mishchenko, Andy Applebaum, Andy Rogerson, Ann Rajan, Annie Wei, Anoop Kotha, Anubha Srivastava, Anushree Agrawal, Arun Vijayvergiya, Ashley Tyra, Ashvin Nair, Avi Nayak, Ben Eggers, Bessie Ji, Beth Hoover, Bill Chen, Blair Chen, Boaz Barak, Borys Minaiev, Botao Hao, Bowen Baker, Brad Lightcap, Brandon McKinzie, Brandon Wang, Brendan Quinn, Brian Fioca, Brian Hsu, Brian Yang, Brian Yu, Brian Zhang, Brittany Brenner, Callie Riggins Zetino, Cameron Raymond, Camillo Lugaresi, Carolina Paz, Cary Hudson, Cedric Whitney, Chak Li, Charles Chen, Charlotte Cole, Chelsea Voss, Chen Ding, Chen Shen, Chengdu Huang, Chris Colby, Chris Hallacy, Chris Koch, Chris Lu, Christina Kaplan, Christina Kim, CJ Minott-Henriques, Cliff Frey, Cody Yu, Coley Czarnecki, Colin Reid, Colin Wei, Cory Decareaux, Cristina Scheau, Cyril Zhang, Cyrus Forbes, Da Tang, Dakota Goldberg, Dan Roberts, Dana Palmie, Daniel Kappler, Daniel Levine, Daniel Wright, Dave Leo, David Lin, David Robinson, Declan Grabb, Derek Chen, Derek Lim, Derek Salama, Dibya Bhattacharjee, Dimitris Tsipras, Dinghua Li, Dingli Yu, DJ Strouse, Drew Williams, Dylan Hunn, Ed Bayes, Edwin Arbus, Ekin Akyurek, Elaine Ya Le, Elana Widmann, Eli Yani, Elizabeth Proehl, Enis Sert, Enoch Cheung, Eri Schwartz, Eric Han, Eric Jiang, Eric Mitchell, Eric Sigler, Eric Wallace, Erik Ritter, Erin Kavanaugh, Evan Mays, Evgenii Nikishin, Fangyuan Li, Felipe Petroski Such, Filipe de Avila Belbute Peres, Filippo Raso, Florent Bekerman, Foivos Tsimpourlas, Fotis Chantzis, Francis Song, Francis Zhang, Gaby Raila, Garrett McGrath, Gary Briggs, Gary Yang, Giambattista Parascandolo, Gildas Chabot, Grace Kim, Grace Zhao, Gregory Valiant, Guillaume Leclerc, Hadi Salman, Hanson Wang, Hao Sheng, Haoming Jiang, Haoyu Wang, Haozhun Jin, Harshit Sikchi, Heather Schmidt, Henry Aspegren, Honglin Chen, Huida Qiu, Hunter Lightman, Ian Covert, Ian Kivlichan, Ian Silber, Ian Sohl, Ibrahim Hammoud, Ignasi Clavera, Ikai Lan, Ilge Akkaya, Ilya Kostrikov, Irina Kofman, Isak Etinger, Ishaan Singal, Jackie Hehir, Jacob Huh, Jacqueline Pan, Jake Wilczynski, Jakub Pachocki, James Lee, James Quinn, Jamie Kiros, Janvi Kalra, Jasmyn Samaroo, Jason Wang, Jason Wolfe,

Jay Chen, Jay Wang, Jean Harb, Jeffrey Han, Jeffrey Wang, Jennifer Zhao, Jeremy Chen, Jerene Yang, Jerry Tworek, Jesse Chand, Jessica Landon, Jessica Liang, Ji Lin, Jiancheng Liu, Jianfeng Wang, Jie Tang, Jihan Yin, Joanne Jang, Joel Morris, Joey Flynn, Johannes Ferstad, Johannes Heidecke, John Fishbein, John Hallman, Jonah Grant, Jonathan Chien, Jonathan Gordon, Jongsoo Park, Jordan Liss, Jos Kraaijeveld, Joseph Guay, Joseph Mo, Josh Lawson, Josh McGrath, Joshua Vendrow, Joy Jiao, Julian Lee, Julie Steele, Julie Wang, Junhua Mao, Kai Chen, Kai Hayashi, Kai Xiao, Kamyar Salahi, Kan Wu, Karan Sekhri, Karan Sharma, Karan Singhal, Karen Li, Kenny Nguyen, Keren Gu-Lemberg, Kevin King, Kevin Liu, Kevin Stone, Kevin Yu, Kristen Ying, Kristian Georgiev, Kristie Lim, Kushal Tirumala, Kyle Miller, Lama Ahmad, Larry Lv, Laura Clare, Laurance Fauconnet, Lauren Itow, Lauren Yang, Laurentia Romaniuk, Leah Anise, Lee Byron, Leher Pathak, Leon Maksin, Leyan Lo, Leyton Ho, Li Jing, Liang Wu, Liang Xiong, Lien Mamitsuka, Lin Yang, Lindsay McCallum, Lindsey Held, Liz Bourgeois, Logan Engstrom, Lorenz Kuhn, Louis Feuvrier, Lu Zhang, Lucas Switzer, Lukas Kondraciuk, Lukasz Kaiser, Manas Joglekar, Mandeep Singh, Mandip Shah, Manuka Stratta, Marcus Williams, Mark Chen, Mark Sun, Marselus Cayton, Martin Li, Marvin Zhang, Marwan Aljubeh, Matt Nichols, Matthew Haines, Max Schwarzer, Mayank Gupta, Meghan Shah, Melody Y. Guan, Melody Huang, Meng Dong, Mengqing Wang, Mia Glaese, Micah Carroll, Michael Lampe, Michael Malek, Michael Sharman, Michael Zhang, Michele Wang, Michelle Pokrass, Mihai Florian, Mikhail Pavlov, Miles Wang, Ming Chen, Mingxuan Wang, Minnia Feng, Mo Bavarian, Molly Lin, Moose Abdool, Mostafa Rohaninejad, Nacho Soto, Natalie Staudacher, Natan LaFontaine, Nathan Marwell, Nelson Liu, Nick Preston, Nick Turley, Nicklas Ansman, Nicole Blades, Nikil Pancha, Nikita Mikhaylin, Niko Felix, Nikunj Handa, Nishant Rai, Nitish Keskar, Noam Brown, Ofir Nachum, Oleg Boiko, Oleg Murk, Olivia Watkins, Oona Gleeson, Pamela Mishkin, Patryk Lesiewicz, Paul Baltescu, Pavel Belov, Peter Zhokhov, Philip Pronin, Phillip Guo, Phoebe Thacker, Qi Liu, Qiming Yuan, Qinghua Liu, Rachel Dias, Rachel Puckett, Rahul Arora, Ravi Teja Mullapudi, Raz Gaon, Reah Miyara, Rennie Song, Rishabh Aggarwal, RJ Marsan, Robel Yemiru, Robert Xiong, Rohan Kshirsagar, Rohan Nuttall, Roman Tsiupa, Ronen Eldan, Rose Wang, Roshan James, Roy Ziv, Rui Shu, Ruslan Nigmatullin, Saachi Jain, Saam Talaie, Sam Altman, Sam Arnesen, Sam Toizer, Sam Toyer, Samuel Miserendino, Sandhini Agarwal, Sarah Yoo, Savannah Heon, Scott Ethersmith, Sean Grove, Sean Taylor, Sebastien Bubeck, Sever Banesiu, Shaokyi Amdo, Shengjia Zhao, Sherwin Wu, Shibani Santurkar, Shiyu Zhao, Shraman Ray Chaudhuri, Shreyas Krishnaswamy, Shuaiqi, Xia, Shuyang Cheng, Shyamal Anadkat, Simón Posada Fishman, Simon Tobin, Siyuan Fu, Somay Jain, Song Mei, Sonya Egoian, Spencer Kim, Spug Golden, SQ Mah, Steph Lin, Stephen Imm, Steve Sharpe, Steve Yadlowsky, Sulman Choudhry, Sungwon Eum, Suvansh Sanjeev, Tabarak Khan, Tal Stramer, Tao Wang, Tao Xin, Tarun Gogineni, Taya Christianson, Ted Sanders, Tejal Patwardhan, Thomas Degry, Thomas Shadwell, Tianfu Fu, Tianshi Gao, Timur Garipov, Tina Sriskandarajah, Toki Sherbakov, Tomek Korbak, Tomer Kaftan, Tomo Hiratsuka, Tongzhou Wang, Tony Song, Tony Zhao, Troy Peterson, Val Kharitonov, Victoria Chernova, Vineet Kosaraju, Vishal Kuo, Vitchyr Pong, Vivek Verma, Vlad Petrov, Wanning Jiang, Weixing Zhang, Wenda Zhou, Wenlei Xie, Wenting Zhan, Wes McCabe, Will DePue, Will Ellsworth, Wulfie Bain, Wyatt Thompson, Xiangning Chen, Xiangyu Qi, Xin Xiang, Xinwei Shi, Yann Dubois, Yaodong Yu, Yara Khakbaz, Yifan Wu, Yilei Qian, Yin Tat Lee, Yinbo Chen, Yizhen Zhang, Yizhong Xiong, Yonglong Tian, Young Cha, Yu Bai, Yu Yang, Yuan Yuan, Yuanzhi Li, Yufeng Zhang, Yuguang Yang, Yujia Jin, Yun Jiang, Yunyun Wang, Yushi Wang, Yutian Liu, Zach Stubenvoll, Zehao Dou, Zheng Wu, and Zhigang Wang. Openai gpt-5 system card, 2026. URL https://arxiv.org/abs/2601.03267.

Richard S Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2):181–211, 1999.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy Lillicrap, Angeliki Lazaridou, Orhan Firat, James Molloy, Michael Isard, Paul R. Barham, Tom Hennigan, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, Ryan Doherty, Eli Collins, Clemens Meyer, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, Jack Krawczyk, Cosmo Du, Ed Chi, Heng-Tze Cheng, Eric Ni, Purvi Shah, Patrick Kane, Betty Chan, Manaal Faruqui, Aliaksei Severyn, Hanzhao Lin, YaGuang Li, Yong Cheng, Abe Ittycheriah, Mahdis Mahdieh, Mia Chen, Pei Sun, Dustin Tran, Sumit Bagri, Balaji Lakshminarayanan, Jeremiah Liu, Andras Orban, Fabian Güra, Hao Zhou, Xinying Song, Aurelien Boffy, Harish Ganapathy, Steven Zheng, HyunJeong Choe, Ágoston Weisz, Tao Zhu, Yifeng Lu, Siddharth

Gopal, Jarrod Kahn, Maciej Kula, Jeff Pitman, Rushin Shah, Emanuel Taropa, Majd Al Merey, Martin Baeuml, Zhifeng Chen, Laurent El Shafey, Yujing Zhang, Olcan Sercinoglu, George Tucker, Enrique Piqueras, Maxim Krikun, Iain Barr, Nikolay Savinov, Ivo Danihelka, Becca Roelofs, Anaïs White, Anders Andreassen, Tamara von Glehn, Lakshman Yagati, Mehran Kazemi, Lucas Gonzalez, Misha Khalman, Jakub Sygnowski, Alexandre Frechette, Charlotte Smith, Laura Culp, Lev Proleev, Yi Luan, Xi Chen, James Lottes, Nathan Schucher, Federico Lebron, Alban Rrustemi, Natalie Clay, Phil Crone, Tomas Kocisky, Jeffrey Zhao, Bartek Perz, Dian Yu, Heidi Howard, Adam Bloniarz, Jack W. Rae, Han Lu, Laurent Sifre, Marcello Maggioni, Fred Alcober, Dan Garrette, Megan Barnes, Shantanu Thakoor, Jacob Austin, Gabriel Barth-Maron, William Wong, Rishabh Joshi, Rahma Chaabouni, Deeni Fatiha, Arun Ahuja, Gaurav Singh Tomar, Evan Senter, Martin Chadwick, Ilya Kornakov, Nithya Attaluri, Iñaki Iturrate, Ruibo Liu, Yunxuan Li, Sarah Cogan, Jeremy Chen, Chao Jia, Chenjie Gu, Qiao Zhang, Jordan Grimstad, Ale Jakse Hartman, Xavier Garcia, Thanumalayan Sankaranarayana Pillai, Jacob Devlin, Michael Laskin, Diego de Las Casas, Dasha Valter, Connie Tao, Lorenzo Blanco, Adrià Puigdomènech Badia, David Reitter, Mianna Chen, Jenny Brennan, Clara Rivera, Sergey Brin, Shariq Iqbal, Gabriela Surita, Jane Labanowski, Abhi Rao, Stephanie Winkler, Emilio Parisotto, Yiming Gu, Kate Olszewska, Ravi Addanki, Antoine Miech, Annie Louis, Denis Teplyashin, Geoff Brown, Elliot Catt, Jan Balaguer, Jackie Xiang, Pidong Wang, Zoe Ashwood, Anton Briukhov, Albert Webson, Sanjay Ganapathy, Smit Sanghavi, Ajay Kannan, Ming-Wei Chang, Axel Stjerngren, Josip Djolonga, Yuting Sun, Ankur Bapna, Matthew Aitchison, Pedram Pejman, Henryk Michalewski, Tianhe Yu, Cindy Wang, Juliette Love, Junwhan Ahn, Dawn Bloxwich, Kehang Han, Peter Humphreys, Thibault Sellam, James Bradbury, Varun Godbole, Sina Samangooei, Bogdan Damoc, Alex Kaskasoli, Sébastien M. R. Arnold, Vijay Vasudevan, Shubham Agrawal, Jason Riesa, Dmitry Lepikhin, Richard Tanburn, Srivatsan Srinivasan, Hyeontaek Lim, Sarah Hodkinson, Pranav Shyam, Johan Ferret, Steven Hand, Ankush Garg, Tom Le Paine, Jian Li, Yujia Li, Minh Giang, Alexander Neitz, Zaheer Abbas, Sarah York, Machel Reid, Elizabeth Cole, Aakanksha Chowdhery, Dipanjan Das, Dominika Rogozińska, Vitaliy Nikolaev, Pablo Sprechmann, Zachary Nado, Lukas Zilka, Flavien Prost, Luheng He, Marianne Monteiro, Gaurav Mishra, Chris Welty, Josh Newlan, Dawei Jia, Miltiadis Allamanis, Clara Huiyi Hu, Raoul de Liedekerke, Justin Gilmer, Carl Saroufim, Shruti Rijhwani, Shaobo Hou, Disha Shrivastava, Anirudh Baddepudi, Alex Goldin, Adnan Ozturel, Albin Cassirer, Yunhan Xu, Daniel Sohn, Devendra Sachan, Reinald Kim Amplayo, Craig Swanson, Dessie Petrova, Shashi Narayan, Arthur Guez, Siddhartha Brahma, Jessica Landon, Miteyan Patel, Ruizhe Zhao, Kevin Villela, Luyu Wang, Wenhao Jia, Matthew Rahtz, Mai Giménez, Legg Yeung, James Keeling, Petko Georgiev, Diana Mincu, Boxi Wu, Salem Haykal, Rachel Saputro, Kiran Vodrahalli, James Qin, Zeynep Cankara, Abhanshu Sharma, Nick Fernando, Will Hawkins, Behnam Neyshabur, Solomon Kim, Adrian Hutter, Priyanka Agrawal, Alex Castro-Ros, George van den Driessche, Tao Wang, Fan Yang, Shuo yiin Chang, Paul Komarek, Ross McIlroy, Mario Lučić, Guodong Zhang, Wael Farhan, Michael Sharman, Paul Natsev, Paul Michel, Yamini Bansal, Siyuan Qiao, Kris Cao, Siamak Shakeri, Christina Butterfield, Justin Chung, Paul Kishan Rubenstein, Shivani Agrawal, Arthur Mensch, Kedar Soparkar, Karel Lenc, Timothy Chung, Aedan Pope, Loren Maggiore, Jackie Kay, Priya Jhakra, Shibo Wang, Joshua Maynez, Mary Phuong, Taylor Tobin, Andrea Tacchetti, Maja Trebacz, Kevin Robinson, Yash Katariya, Sebastian Riedel, Paige Bailey, Kefan Xiao, Nimesh Ghelani, Lora Aroyo, Ambrose Slone, Neil Houlsby, Xuehan Xiong, Zhen Yang, Elena Gribovskaya, Jonas Adler, Mateo Wirth, Lisa Lee, Music Li, Thais Kagohara, Jay Pavagadhi, Sophie Bridgers, Anna Bortsova, Sanjay Ghemawat, Zafarali Ahmed, Tianqi Liu, Richard Powell, Vijay Bolina, Mariko Iinuma, Polina Zablotskaia, James Besley, Da-Woon Chung, Timothy Dozat, Ramona Comanescu, Xiance Si, Jeremy Greer, Guolong Su, Martin Polacek, Raphaël Lopez Kaufman, Simon Tokumine, Hexiang Hu, Elena Buchatskaya, Yingjie Miao, Mohamed Elhawaty, Aditya Siddhant, Nenad Tomasev, Jinwei Xing, Christina Greer, Helen Miller, Shereen Ashraf, Aurko Roy, Zizhao Zhang, Ada Ma, Angelos Filos, Milos Besta, Rory Blevins, Ted Klimenko, Chih-Kuan Yeh, Soravit Changpinyo, Jiaqi Mu, Oscar Chang, Mantas Pajarskas, Carrie Muir, Vered Cohen, Charline Le Lan, Krishna Haridasan, Amit Marathe, Steven Hansen, Sholto Douglas, Rajkumar Samuel, Mingqiu Wang, Sophia Austin, Chang Lan, Jiepu Jiang, Justin Chiu, Jaime Alonso Lorenzo, Lars Lowe Sjösund, Sébastien Cevey, Zach Gleicher, Thi Avrahami, Anudhyan Boral, Hansa Srinivasan, Vittorio Selo, Rhys May, Konstantinos Aisopos, Léonard Hussenot, Livio Baldini Soares, Kate Baumli, Michael B. Chang, Adrià Recasens, Ben Caine, Alexander Pritzel, Filip Pavetic, Fabio Pardo, Anita Gergely, Justin Frye, Vinay Ramasesh, Dan Horgan, Kartikeya Badola, Nora Kassner, Subhrajit Roy, Ethan Dyer, Víctor Campos Campos, Alex Tomala, Yunhao Tang, Dalia El Badawy, Elspeth White, Basil Mustafa, Oran Lang, Abhishek Jindal,

Sharad Vikram, Zhitao Gong, Sergi Caelles, Ross Hemsley, Gregory Thornton, Fangxiaoyu Feng, Wojciech Stokowiec, Ce Zheng, Phoebe Thacker, Çağlar Ünlü, Zhishuai Zhang, Mohammad Saleh, James Svensson, Max Bileschi, Piyush Patil, Ankesh Anand, Roman Ring, Katerina Tsihlas, Arpi Vezer, Marco Selvi, Toby Shevlane, Mikel Rodriguez, Tom Kwiatkowski, Samira Daruki, Keran Rong, Allan Dafoe, Nicholas FitzGerald, Keren Gu-Lemberg, Mina Khan, Lisa Anne Hendricks, Marie Pellat, Vladimir Feinberg, James Cobon-Kerr, Tara Sainath, Maribeth Rauh, Sayed Hadi Hashemi, Richard Ives, Yana Hasson, Eric Noland, Yuan Cao, Nathan Byrd, Le Hou, Qingze Wang, Thibault Sottiaux, Michela Paganini, Jean-Baptiste Lespiau, Alexandre Moufarek, Samer Hassan, Kaushik Shivakumar, Joost van Amersfoort, Amol Mandhane, Pratik Joshi, Anirudh Goyal, Matthew Tung, Andrew Brock, Hannah Sheahan, Vedant Misra, Cheng Li, Nemanja Rakićević, Mostafa Dehghani, Fangyu Liu, Sid Mittal, Junhyuk Oh, Seb Noury, Eren Sezener, Fantine Huot, Matthew Lamm, Nicola De Cao, Charlie Chen, Sidharth Mudgal, Romina Stella, Kevin Brooks, Gautam Vasudevan, Chenxi Liu, Mainak Chain, Nivedita Melinkeri, Aaron Cohen, Venus Wang, Kristie Seymore, Sergey Zubkov, Rahul Goel, Summer Yue, Sai Krishnakumaran, Brian Albert, Nate Hurley, Motoki Sano, Anhad Mohananey, Jonah Joughin, Egor Filonov, Tomasz Kepa, Yomna Eldawy, Jiawern Lim, Rahul Rishi, Shirin Badiezadegan, Taylor Bos, Jerry Chang, Sanil Jain, Sri Gayatri Sundara Padmanabhan, Subha Puttagunta, Kalpesh Krishna, Leslie Baker, Norbert Kalb, Vamsi Bedapudi, Adam Kurzrok, Shuntong Lei, Anthony Yu, Oren Litvin, Xiang Zhou, Zhichun Wu, Sam Sobell, Andrea Siciliano, Alan Papir, Robby Neale, Jonas Bragagnolo, Tej Toor, Tina Chen, Valentin Anklin, Feiran Wang, Richie Feng, Milad Gholami, Kevin Ling, Lijuan Liu, Jules Walter, Hamid Moghaddam, Arun Kishore, Jakub Adamek, Tyler Mercado, Jonathan Mallinson, Siddhinita Wandekar, Stephen Cagle, Eran Ofek, Guillermo Garrido, Clemens Lombriser, Maksim Mukha, Botu Sun, Hafeezul Rahman Mohammad, Josip Matak, Yadi Qian, Vikas Peswani, Pawel Janus, Quan Yuan, Leif Schelin, Oana David, Ankur Garg, Yifan He, Oleksii Duzhyi, Anton Älgmyr, Timothée Lottaz, Qi Li, Vikas Yadav, Luyao Xu, Alex Chinien, Rakesh Shivanna, Aleksandr Chuklin, Josie Li, Carrie Spadine, Travis Wolfe, Kareem Mohamed, Subhabrata Das, Zihang Dai, Kyle He, Daniel von Dincklage, Shyam Upadhyay, Akanksha Maurya, Luyan Chi, Sebastian Krause, Khalid Salama, Pam G Rabinovitch, Pavan Kumar Reddy M, Aarush Selvan, Mikhail Dektiarev, Golnaz Ghiasi, Erdem Guven, Himanshu Gupta, Boyi Liu, Deepak Sharma, Idan Heimlich Shtacher, Shachi Paul, Oscar Akerlund, François-Xavier Aubet, Terry Huang, Chen Zhu, Eric Zhu, Elico Teixeira, Matthew Fritze, Francesco Bertolini, Liana-Eleonora Marinescu, Martin Bölle, Dominik Paulus, Khyatti Gupta, Tejasi Latkar, Max Chang, Jason Sanders, Roopa Wilson, Xuewei Wu, Yi-Xuan Tan, Lam Nguyen Thiet, Tulsee Doshi, Sid Lall, Swaroop Mishra, Wanming Chen, Thang Luong, Seth Benjamin, Jasmine Lee, Ewa Andrejczuk, Dominik Rabiej, Vipul Ranjan, Krzysztof Styrc, Pengcheng Yin, Jon Simon, Malcolm Rose Harriott, Mudit Bansal, Alexei Robsky, Geoff Bacon, David Greene, Daniil Mirylenka, Chen Zhou, Obaid Sarvana, Abhimanyu Goyal, Samuel Andermatt, Patrick Siegler, Ben Horn, Assaf Israel, Francesco Pongetti, Chih-Wei "Louis" Chen, Marco Selvatici, Pedro Silva, Kathie Wang, Jackson Tolins, Kelvin Guu, Roey Yogev, Xiaochen Cai, Alessandro Agostini, Maulik Shah, Hung Nguyen, Noah Ó Donnaile, Sébastien Pereira, Linda Friso, Adam Stambler, Adam Kurzrok, Chenkai Kuang, Yan Romanikhin, Mark Geller, ZJ Yan, Kane Jang, Cheng-Chun Lee, Wojciech Fica, Eric Malmi, Qijun Tan, Dan Banica, Daniel Balle, Ryan Pham, Yanping Huang, Diana Avram, Hongzhi Shi, Jasjot Singh, Chris Hidey, Niharika Ahuja, Pranab Saxena, Dan Dooley, Srividya Pranavi Potharaju, Eileen O'Neill, Anand Gokulchandran, Ryan Foley, Kai Zhao, Mike Dusenberry, Yuan Liu, Pulkit Mehta, Ragha Kotikalapudi, Chalence Safranek-Shrader, Andrew Goodman, Joshua Kessinger, Eran Globen, Prateek Kolhar, Chris Gorgolewski, Ali Ibrahim, Yang Song, Ali Eichenbaum, Thomas Brovelli, Sahitya Potluri, Preethi Lahoti, Cip Baetu, Ali Ghorbani, Charles Chen, Andy Crawford, Shalini Pal, Mukund Sridhar, Petru Gurita, Asier Mujika, Igor Petrovski, Pierre-Louis Cedoz, Chenmei Li, Shiyuan Chen, Niccolò Dal Santo, Siddharth Goyal, Jitesh Punjabi, Karthik Kappaganthu, Chester Kwak, Pallavi LV, Sarmishta Velury, Himadri Choudhury, Jamie Hall, Premal Shah, Ricardo Figueira, Matt Thomas, Minjie Lu, Ting Zhou, Chintu Kumar, Thomas Jurdi, Sharat Chikkerur, Yenai Ma, Adams Yu, Soo Kwak, Victor Ähdel, Sujeevan Rajayogam, Travis Choma, Fei Liu, Aditya Barua, Colin Ji, Ji Ho Park, Vincent Hellendoorn, Alex Bailey, Taylan Bilal, Huanjie Zhou, Mehrdad Khatir, Charles Sutton, Wojciech Rzadkowski, Fiona Macintosh, Roopali Vij, Konstantin Shagin, Paul Medina, Chen Liang, Jinjing Zhou, Pararth Shah, Yingying Bi, Attila Dankovics, Shipra Banga, Sabine Lehmann, Marissa Bredesen, Zifan Lin, John Eric Hoffmann, Jonathan Lai, Raynald Chung, Kai Yang, Nihal Balani, Arthur Bražinskas, Andrei Sozanschi, Matthew Hayes, Héctor Fernández Alcalde, Peter Makarov, Will Chen, Antonio Stella, Liselotte Snijders, Michael Mandl, Ante Kärrman, Paweł Nowak, Xinyi Wu, Alex Dyck, Krishnan Vaidyanathan,

Raghavender R, Jessica Mallet, Mitch Rudominer, Eric Johnston, Sushil Mittal, Akhil Udathu, Janara Christensen, Vishal Verma, Zach Irving, Andreas Santucci, Gamaleldin Elsayed, Elnaz Davoodi, Marin Georgiev, Ian Tenney, Nan Hua, Geoffrey Cideron, Edouard Leurent, Mahmoud Alnahlawi, Ionut Georgescu, Nan Wei, Ivy Zheng, Dylan Scandinaro, Heinrich Jiang, Jasper Snoek, Mukund Sundararajan, Xuezhi Wang, Zack Ontiveros, Itay Karo, Jeremy Cole, Vinu Rajashekhar, Lara Tumeh, Eyal Ben-David, Rishub Jain, Jonathan Uesato, Romina Datta, Oskar Bunyan, Shimu Wu, John Zhang, Piotr Stanczyk, Ye Zhang, David Steiner, Subhajit Naskar, Michael Azzam, Matthew Johnson, Adam Paszke, Chung-Cheng Chiu, Jaume Sanchez Elias, Afroz Mohiuddin, Faizan Muhammad, Jin Miao, Andrew Lee, Nino Vieillard, Jane Park, Jiageng Zhang, Jeff Stanway, Drew Garmon, Abhijit Karmarkar, Zhe Dong, Jong Lee, Aviral Kumar, Luowei Zhou, Jonathan Evens, William Isaac, Geoffrey Irving, Edward Loper, Michael Fink, Isha Arkatkar, Nanxin Chen, Izhak Shafran, Ivan Petrychenko, Zhe Chen, Johnson Jia, Anselm Levskaya, Zhenkai Zhu, Peter Grabowski, Yu Mao, Alberto Magni, Kaisheng Yao, Javier Snaider, Norman Casagrande, Evan Palmer, Paul Suganthan, Alfonso Castaño, Irene Giannoumis, Wooyeol Kim, Mikołaj Rybiński, Ashwin Sreevatsa, Jennifer Prendki, David Soergel, Adrian Goedeckemeyer, Willi Gierke, Mohsen Jafari, Meenu Gaba, Jeremy Wiesner, Diana Gage Wright, Yawen Wei, Harsha Vashisht, Yana Kulizhskaya, Jay Hoover, Maigo Le, Lu Li, Chimezie Iwuanyanwu, Lu Liu, Kevin Ramirez, Andrey Khorlin, Albert Cui, Tian LIN, Marcus Wu, Ricardo Aguilar, Keith Pallo, Abhishek Chakladar, Ginger Perng, Elena Allica Abellan, Mingyang Zhang, Ishita Dasgupta, Nate Kushman, Ivo Penchev, Alena Repina, Xihui Wu, Tom van der Weide, Priya Ponnapalli, Caroline Kaplan, Jiri Simsa, Shuangfeng Li, Olivier Dousse, Fan Yang, Jeff Piper, Nathan Ie, Rama Pasumarthi, Nathan Lintz, Anitha Vijayakumar, Daniel Andor, Pedro Valenzuela, Minnie Lui, Cosmin Paduraru, Daiyi Peng, Katherine Lee, Shuyuan Zhang, Somer Greene, Duc Dung Nguyen, Paula Kurylowicz, Cassidy Hardin, Lucas Dixon, Lili Janzer, Kiam Choo, Ziqiang Feng, Biao Zhang, Achintya Singhal, Dayou Du, Dan McKinnon, Natasha Antropova, Tolga Bolukbasi, Orgad Keller, David Reid, Daniel Finchelstein, Maria Abi Raad, Remi Crocker, Peter Hawkins, Robert Dadashi, Colin Gaffney, Ken Franko, Anna Bulanova, Rémi Leblond, Shirley Chung, Harry Askham, Luis C. Cobo, Kelvin Xu, Felix Fischer, Jun Xu, Christina Sorokin, Chris Alberti, Chu-Cheng Lin, Colin Evans, Alek Dimitriev, Hannah Forbes, Dylan Banarse, Zora Tung, Mark Omernick, Colton Bishop, Rachel Sterneck, Rohan Jain, Jiawei Xia, Ehsan Amid, Francesco Piccinno, Xingyu Wang, Praseem Banzal, Daniel J. Mankowitz, Alex Polozov, Victoria Krakovna, Sasha Brown, MohammadHossein Bateni, Dennis Duan, Vlad Firoiu, Meghana Thotakuri, Tom Natan, Matthieu Geist, Ser tan Girgin, Hui Li, Jiayu Ye, Ofir Roval, Reiko Tojo, Michael Kwong, James Lee-Thorp, Christopher Yew, Danila Sinopalnikov, Sabela Ramos, John Mellor, Abhishek Sharma, Kathy Wu, David Miller, Nicolas Sonnerat, Denis Vnukov, Rory Greig, Jennifer Beattie, Emily Caveness, Libin Bai, Julian Eisenschlos, Alex Korchemniy, Tomy Tsai, Mimi Jasarevic, Weize Kong, Phuong Dao, Zeyu Zheng, Frederick Liu, Fan Yang, Rui Zhu, Tian Huey Teh, Jason Sanmiya, Evgeny Gladchenko, Nejc Trdin, Daniel Toyama, Evan Rosen, Sasan Tavakkol, Linting Xue, Chen Elkind, Oliver Woodman, John Carpenter, George Papamakarios, Rupert Kemp, Sushant Kafle, Tanya Grunina, Rishika Sinha, Alice Talbert, Diane Wu, Denese Owusu-Afriyie, Cosmo Du, Chloe Thornton, Jordi Pont-Tuset, Pradyumna Narayana, Jing Li, Saaber Fatehi, John Wieting, Omar Ajmeri, Benigno Uria, Yeongil Ko, Laura Knight, Amélie Héliou, Ning Niu, Shane Gu, Chenxi Pang, Yeqing Li, Nir Levine, Ariel Stolovich, Rebeca Santamaria-Fernandez, Sonam Goenka, Wenny Yustalim, Robin Strudel, Ali Elqursh, Charlie Deck, Hyo Lee, Zonglin Li, Kyle Levin, Raphael Hoffmann, Dan Holtmann-Rice, Olivier Bachem, Sho Arora, Christy Koh, Soheil Hassas Yeganeh, Siim Põder, Mukarram Tariq, Yanhua Sun, Lucian Ionita, Mojtaba Seyedhosseini, Pouya Tafti, Zhiyu Liu, Anmol Gulati, Jasmine Liu, Xinyu Ye, Bart Chrzaszcz, Lily Wang, Nikhil Sethi, Tianrun Li, Ben Brown, Shreya Singh, Wei Fan, Aaron Parisi, Joe Stanton, Vinod Koverkathu, Christopher A. Choquette-Choo, Yunjie Li, TJ Lu, Abe Ittycheriah, Prakash Shroff, Mani Varadarajan, Sanaz Bahargam, Rob Willoughby, David Gaddy, Guillaume Desjardins, Marco Cornero, Brona Robenek, Bhavishya Mittal, Ben Albrecht, Ashish Shenoy, Fedor Moiseev, Henrik Jacobsson, Alireza Ghaffarkhah, Morgane Rivière, Alanna Walton, Clément Crepy, Alicia Parrish, Zongwei Zhou, Clement Farabet, Carey Radebaugh, Praveen Srinivasan, Claudia van der Salm, Andreas Fidjeland, Salvatore Scellato, Eri Latorre-Chimoto, Hanna Klimczak-Plucińska, David Bridson, Dario de Cesare, Tom Hudson, Piermaria Mendolicchio, Lexi Walker, Alex Morris, Matthew Mauger, Alexey Guseynov, Alison Reid, Seth Odoom, Lucia Loher, Victor Cotruta, Madhavi Yenugula, Dominik Grewe, Anastasia Petrushkina, Tom Duerig, Antonio Sanchez, Steve Yadlowsky, Amy Shen, Amir Globerson, Lynette Webb, Sahil Dua, Dong Li, Surya Bhupatiraju, Dan Hurt, Haroon Qureshi, Ananth Agarwal, Tomer Shani, Matan Eyal, Anuj Khare, Shreyas Rammohan Belle, Lei Wang, Chetan Tekur, Mihir Sanjay

Kale, Jinliang Wei, Ruoxin Sang, Brennan Saeta, Tyler Liechty, Yi Sun, Yao Zhao, Stephan Lee, Pandu Nayak, Doug Fritz, Manish Reddy Vuyyuru, John Aslanides, Nidhi Vyas, Martin Wicke, Xiao Ma, Evgenii Eltyshev, Nina Martin, Hardie Cate, James Manyika, Keyvan Amiri, Yelin Kim, Xi Xiong, Kai Kang, Florian Luisier, Nilesh Tripuraneni, David Madras, Mandy Guo, Austin Waters, Oliver Wang, Joshua Ainslie, Jason Baldridge, Han Zhang, Garima Pruthi, Jakob Bauer, Feng Yang, Riham Mansour, Jason Gelman, Yang Xu, George Polovets, Ji Liu, Honglong Cai, Warren Chen, XiangHai Sheng, Emily Xue, Sherjil Ozair, Christof Angermueller, Xiaowei Li, Anoop Sinha, Weiren Wang, Julia Wiesinger, Emmanouil Koukoumidis, Yuan Tian, Anand Iyer, Madhu Gurumurthy, Mark Goldenson, Parashar Shah, MK Blake, Hongkun Yu, Anthony Urbanowicz, Jennimaria Palomaki, Chrisantha Fernando, Ken Durden, Harsh Mehta, Nikola Momchev, Elahe Rahimtoroghi, Maria Georgaki, Amit Raul, Sebastian Ruder, Morgan Redshaw, Jinhyuk Lee, Denny Zhou, Komal Jalan, Dinghua Li, Blake Hechtman, Parker Schuh, Milad Nasr, Kieran Milan, Vladimir Mikulik, Juliana Franco, Tim Green, Nam Nguyen, Joe Kelley, Aroma Mahendru, Andrea Hu, Joshua Howland, Ben Vargas, Jeffrey Hui, Kshitij Bansal, Vikram Rao, Rakesh Ghiya, Emma Wang, Ke Ye, Jean Michel Sarr, Melanie Moranski Preston, Madeleine Elish, Steve Li, Aakash Kaku, Jigar Gupta, Ice Pasupat, Da-Cheng Juan, Milan Someswar, Tejvi M., Xinyun Chen, Aida Amini, Alex Fabrikant, Eric Chu, Xuanyi Dong, Amruta Muthal, Senaka Buthpitiya, Sarthak Jauhari, Nan Hua, Urvashi Khandelwal, Ayal Hitron, Jie Ren, Larissa Rinaldi, Shahar Drath, Avigail Dabush, Nan-Jiang Jiang, Harshal Godhia, Uli Sachs, Anthony Chen, Yicheng Fan, Hagai Taitelbaum, Hila Noga, Zhuyun Dai, James Wang, Chen Liang, Jenny Hamer, Chun-Sung Ferng, Chenel Elkind, Aviel Atias, Paulina Lee, Vít Listík, Mathias Carlen, Jan van de Kerkhof, Marcin Pikus, Krunoslav Zaher, Paul Müller, Sasha Zykova, Richard Stefanec, Vitaly Gatsko, Christoph Hirnschall, Ashwin Sethi, Xingyu Federico Xu, Chetan Ahuja, Beth Tsai, Anca Stefanoiu, Bo Feng, Keshav Dhandhania, Manish Katyal, Akshay Gupta, Atharva Parulekar, Divya Pitta, Jing Zhao, Vivaan Bhatia, Yashodha Bhavnani, Omar Alhadlaq, Xiaolin Li, Peter Danenberg, Dennis Tu, Alex Pine, Vera Filippova, Abhipso Ghosh, Ben Limonchik, Bhargava Urala, Chaitanya Krishna Lanka, Derik Clive, Yi Sun, Edward Li, Hao Wu, Kevin Hongtongsak, Ianna Li, Kalind Thakkar, Kuanysh Omarov, Kushal Majmundar, Michael Alverson, Michael Kucharski, Mohak Patel, Mudit Jain, Maksim Zabelin, Paolo Pelagatti, Rohan Kohli, Saurabh Kumar, Joseph Kim, Swetha Sankar, Vineet Shah, Lakshmi Ramachandruni, Xiangkai Zeng, Ben Bariach, Laura Weidinger, Tu Vu, Alek Andreev, Antoine He, Kevin Hui, Sheleem Kashem, Amar Subramanya, Sissie Hsiao, Demis Hassabis, Koray Kavukcuoglu, Adam Sadovsky, Quoc Le, Trevor Strohman, Yonghui Wu, Slav Petrov, Jeffrey Dean, and Oriol Vinyals. Gemini: A family of highly capable multimodal models, 2025. URL https://arxiv.org/abs/2312.11805.

Gemma Team. Gemma 4 technical report, 2026. URL https://arxiv.org/abs/2607.02770.

Jiawei Wang, Jiacai Liu, Yuqian Fu, Yingru Li, Xintao Wang, Yuan Lin, Yu Yue, Lin Zhang, Yang Wang, and Ke Wang. Harnessing uncertainty: Entropy-modulated policy gradients for long-horizon llm agents. *arXiv preprint arXiv:2509.09265*, 2025.

Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. Scienceworld: Is your agent smarter than a 5th grader?, 2022. URL https://arxiv.org/abs/2203.07540.

Zixuan Wang, Yuchen Yan, Hongxing Li, Teng Pan, Dingming Li, Ruiqing Zhang, Weiming Lu, Jun Xiao, Yueting Zhuang, and Yongliang Shen. Milestone-guided policy learning for long-horizon language agents, 2026. URL https://arxiv.org/abs/2605.06078.

Runze Xu, Xiaolong Shan, Shuang Dai, Yu Wang, and Jincheng Yu. Knowing when to stop: Adaptive action chunking via internal cross-attention dynamics in vlas. *arXiv preprint arXiv:2609.00908*, 2026.

Wujiang Xu, Wentian Zhao, Zhenting Wang, Yu-Jhe Li, Can Jin, Mingyu Jin, Kai Mei, Kun Wan, and Dimitris N Metaxas. Epo: Entropy-regularized policy optimization for llm agents reinforcement learning. *arXiv preprint arXiv:2509.22576*, 2025.

John Yang, Carlos E Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. Swe-agent: Agent-computer interfaces enable automated software engineering. *Advances in Neural Information Processing Systems*, 37:50528–50652, 2024.

Ruihan Yang, Fanghua Ye, Xiang We, Ruoqing Zhao, Kang Luo, Xinbo Xu, Bo Zhao, Ruotian Ma, Shanyi Wang, Zhaopeng Tu, et al. Think fast and slow: Step-level cognitive depth adaptation for llm agents. *arXiv preprint arXiv:2602.12662*, 2026a.

Yanting Yang, Can Jin, Jinman Zhao, Jiahao Wu, Yang Zhou, Zhepeng Wang, Zhendong Wang, Mu Zhou, and Dimitris N. Metaxas. Act more, decide less: Skill-guided adaptive action chunking for long-horizon llm agents, 2026b. URL https://arxiv.org/abs/2609.02042.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.

Yuchi Zhao, Miroslav Bogdanovic, Arjun Sohal, Liyu Tao, Kourosh Darvish, Alán Aspuru-Guzik, Florian Shkurti, and Animesh Garg. Dynamic execution horizon prediction for chunk-based robot policies, 2026. URL https://arxiv.org/abs/2606.11408.

Helen Zhou, Audrey Huang, Kamyar Azizzadenesheli, David Childers, and Zachary Lipton. Timing as an action: Learning when to observe and act. In Sanjoy Dasgupta, Stephan Mandt, and Yingzhen Li (eds.), *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pp. 3979–3987. PMLR, 02–04 May 2024. URL https://proceedings.mlr.press/v238/zhou24c.html.

A Notation

Table 5 summarizes the notation used in our formulation, architecture, training, and evaluation. The theoretical history includes decision boundaries for budget accounting; policy inputs omit explicit budget and step counters. Selected depth and mean executed depth are distinct quantities.

Table 5: Notation. Index meanings are local to their context; probabilities and percentage-point differences use distinct units as specified below.

Symbol	Meaning
Environment and decision protocol	
$\mathcal{S},\ s$	Environment state space and an individual state.
$\mathcal{A},\ a$	Primitive action space and an individual primitive action.
$P(s' \mid s, a)$	Environment transition kernel.
$\Omega,\ O$	Observation space and state-to-observation mapping.
r	Binary terminal task-success reward in the problem formulation; distinct from shaped training rewards.
$x,\ x_t$	Decision history, and the history at decision t. In the proofs, it includes observations, executed actions, and past decision boundaries.
T_t	Environment step at which decision t begins; $T_{t+1} = T_t + h_t$ unless the episode terminates earlier.
K	Hard decision budget: each policy invocation costs one unit, irrespective of execution depth.
$K_{\text{train}},\ K_{\text{eval}}$	Decision budgets enforced during training and evaluation.
$\mathcal{H}$	Learned depth-selection set, $\{1, 2, 4, 8\}$. Main fixed-depth baselines additionally test $3, 5, 6, 7$.
$h,\ h_t,\ h_{\max}$	Selected execution depth, its value at decision t, and the maximum decoded length ($h_{\max} = 8$). Termination can shorten the executed prefix.
$\bar{h}$	Mean selected depth over a set of decision points (Figs. 11 and 5); distinct from the mean executed depth $\bar{\ell}$.
Policy and architecture	
$\pi = (\pi_a, \pi_h)$	Policy comprising the action-generation mechanism and the depth-selection rule.

Continued on next page

Table 5 (continued)

Symbol	Meaning
$\theta,\ \pi_\theta$	Parameters and joint policy of the learned model.
z_t	Backbone representation of the available policy context at decision t, reused by both output branches.
$\pi_\theta^h,\ \pi_\theta^a$	Parameterized depth distribution and action distribution, respectively. The selected depth is not an input to the action decoder.
$a_t^j,\ a_t^{<j},\ a_t^{1:h_t}$	Action at position j within decision t, preceding action prefix, and the first h_t predicted actions.
$\mathcal{A}_e$	Fixed decoder action vocabulary for domain e; its entries represent complete primitive actions.
$\mathcal{A}_{\mathrm{adm}}(s)$	Actions admissible at state s. In the text domains, first-action masking uses $\mathcal{A}_{\mathrm{adm}}(s) \cap \mathcal{A}_e$.
Budgeted success and improvement result	
$J_K(\pi)$	Episode success probability under policy π and budget K, with the task distribution and other limits fixed.
$\pi^{(h)},\ f_h$	Fixed-depth policy using the shared action mechanism; f_h is its shorthand in the evaluation tables.
$Q^\pi(x, d)$	Conditional success probability after choosing depth d at history x, using the shared action mechanism, and following π thereafter.
$x^*,\ \rho$	Nonterminal history at which a deviation is considered, and its positive arrival probability under $\pi^{(h)}$.
$h',\ \pi'$	Alternative depth and the policy that uses it only at x^*, otherwise following $\pi^{(h)}$.
Δ (Proposition)	Continuation-value gap $Q^{\pi^{(h)}}(x^*, h') - Q^{\pi^{(h)}}(x^*, h)$, measured in probability units.
$\Pi_{\mathcal{H}},\ h^\star$	Full-history depth-policy class over $\mathcal{H}$ with the action mechanism fixed, and a depth maximizing $J_K(\pi^{(h)})$ over $h \in \mathcal{H}$.
Supervision and training rewards	
L	Length of a reference trajectory $(a_1, \ldots, a_L)$.
x_u, a_u, m (SFT)	Reference context, reference action at trajectory position u, and supervised depth $d \in \mathcal{H}$; d is not a depth-head label.
G	GRPO group size during RL training.
Diagnostics and efficiency	
δ	Solver-defined distance to the goal in the puzzles.
$\hat{p}(h)$	Empirical solver-defined optimal-path reliability of a length-h open-loop action chunk in the puzzles.
$\hat{p}(1)^h$	Independent-step reference constructed from one-step empirical reliability.
$\Delta d,\ \mathbb{E}[\Delta d/h]$	Reduction in solver distance to the goal and the length-normalized progress statistic used in Fig. 4. Here d denotes solver distance.
$D_i,\ S_i,\ y_i$	Executed decisions, executed environment steps, and binary success for evaluation episode i. Step counts include invalid attempts.
$D,\ S$	Decision and environment-step counts over the same specified trajectory or collection of trajectories.
$\bar{\ell}$	Mean executed depth across decisions: $\sum_i S_i / \sum_i D_i$. Its reciprocal is encodings per executed action under one backbone encoding per decision.
$C_{\mathrm{dec}},\ C_{\mathrm{step}}$	Aggregate executed cost per observed success: $\sum_i D_i / \sum_i y_i$ and $\sum_i S_i / \sum_i y_i$, respectively. Both include failed-episode costs.
c	Generic cost in Pareto comparisons, $c \in \{\bar{D}, \bar{S}\}$.
π_1, π_2	Generic policies in the Pareto-dominance definition.
τ	Threshold used by an entropy-based timing rule.
Δ (results tables)	Success-rate difference in percentage points; its direction is specified by the table.
n (puzzle geometry)	Side length of an $n \times n$ Sliding Puzzle board; unrelated to the evaluation sample size.

B PROOFS

B.1 SETUP

An episodic environment is a tuple $(\mathcal{S}, \mathcal{A}, P, \Omega, O, r)$ as in §3.1. A *decision history* x is the full interaction record at a decision boundary: all observations and executed primitive actions so far, including the placement of past decision boundaries and the number of decisions taken. Histories therefore encode the budget accounting: with hard budget K, the number of remaining decisions is a function of x. All comparisons below fix the initial task distribution, the decision budget K, environment step limits, and termination rules.

Definition 1 (Budget-K decision process)**.** At each decision boundary with history x, a policy $\pi = (\pi_a, \pi_h)$ selects an execution depth $h \in \mathcal{H}$ via $\pi_h(\cdot \mid x)$ and an action chunk via π_a; the first h actions are executed open-loop, yielding the next decision history. Every decision consumes one unit of the budget regardless of h. An episode ends in success (terminal reward 1), in environment-side termination, or in *budget failure* when K decisions are exhausted before success. The objective is the success probability $J_K(\pi)$. For $h \in \mathcal{H}$, the *fixed-depth arm* $\pi^{(h)}$ selects depth h at every decision, with the same action mechanism π_a.

For a policy π and non-terminal decision history x, $Q^\pi(x, d)$ is as in Proposition 1:

$$Q^\pi(x, d) \;=\; \Pr\big[\text{episode succeeds} \;\big|\; x;\ \text{depth } d \text{ now, actions from } \pi_a,\ \pi \text{ thereafter}\big],$$

where the conditional law includes the budget accounting of Definition 1, environment step limits, and termination rules.

Assumption 1 (Shared action mechanism)**.** At history x, the action chunk is sampled from the fixed kernel $\pi_a(\cdot \mid x)$; the selected depth determines the executed prefix and is not an input to the action decoder (§3.3). All policies compared below share this kernel and differ only in depth selection.

B.2 PROOF OF PROPOSITION 1

Proof. A decision history contains its entire prefix, so x^* occurs at most once along any trajectory, and π' is well defined. The two policies agree at every history other than x^*; in particular they agree on the event that x^* is never reached, and they induce the same distribution over trajectory prefixes up to and including arrival at x^*, so the arrival probability is ρ under both. Conditioning on arrival, the success probability is $Q^{\pi^{(h)}}(x^*, h)$ under $\pi^{(h)}$ and $Q^{\pi^{(h)}}(x^*, h')$ under π': the deviation is taken once and both policies thereafter coincide with $\pi^{(h)}$, and any change in the downstream visitation distribution induced by the deviation is part of the conditional law defining $Q^{\pi^{(h)}}(x^*, h')$, not an omission from it. The law of total probability over the arrival event gives the identity, and positivity follows from $\rho > 0$ and the premise. □

Corollary 1 (Dominance over fixed depths)**.** *Let $\Pi_{\mathcal{H}}$ be the class of history-conditioned depth policies over $\mathcal{H}$ under Assumption 1, and let $h^\star \in \arg\max_{h \in \mathcal{H}} J_K(\pi^{(h)})$ (the maximum exists; $\mathcal{H}$ is finite). Then*

$$\sup_{\pi \in \Pi_{\mathcal{H}}} J_K(\pi) \;\geq\; \max_{h \in \mathcal{H}} J_K(\pi^{(h)}),$$

since every fixed arm belongs to $\Pi_{\mathcal{H}}$. If, in addition, the premise of Proposition 1 holds for a best fixed-depth arm $\pi^{(h^\star)}$, the constructed deviation policy $\pi' \in \Pi_{\mathcal{H}}$ satisfies

$$J_K(\pi') \;>\; \max_{h \in \mathcal{H}} J_K(\pi^{(h)}) :$$

a history-conditioned depth policy strictly outperforms every fixed depth in $\mathcal{H}$.

B.3 SCOPE AND INTERPRETATION

Remark 1 (Scope of the depth set). The guarantee is internal to $\mathcal{H} = \{1, 2, 4, 8\}$: weak dominance follows from class inclusion over the *same* depth set, and the strict statement compares against the best arm in $\mathcal{H}$. Our experiments additionally evaluate fixed depths $3, 5, 6, 7$; comparisons against these are empirical results outside the class covered by Corollary 1.

Remark 2 (Existence versus realizability). The result establishes the existence of an improving history-conditioned depth policy under the stated condition. It does not guarantee that the chosen policy parameterization can represent this policy or that training will recover it; the performance of the learned policy is assessed empirically.

Remark 3 (Relation to existing guarantees). The proof specializes a standard one-step policy-improvement argument to budgeted open-loop execution. It shares the principle underlying option interruption (Sutton et al., 1999), but selects depth at the decision boundary rather than changing an ongoing option's termination rule, and uses no intermediate observations during execution. AQC studies related adaptive-length dominance under discounted return with advantage-based depth selection (Gireesh et al., 2026); timing-as-an-action has also been formalized explicitly (Zhou et al., 2024). Our result gives a sufficient condition for improvement in the budget-K success objective over full-history depth policies, without claiming a new policy-improvement principle or a guarantee for the learned network.

C Task illustrations and instance generation

Table 6: **Domain coverage and training reward settings.**

	Visual environments		**Text environments**	
Property	**Sliding Puzzle**	**Sokoban**	**ALFWorld**	**ScienceWorld**
Observability	Full	Full	Partial	Partial
Reversibility	Reversible	Irreversible pushes; deadlocks	Task-dependent	Irreversible outcomes possible
Action vocabulary	Four directions	Four directions	5,698 commands	658 commands
Solver diagnostics	Available	Available	Not used	Not used
Horizon coverage	Mixed horizons	Mixed horizons	Mixed horizons	Short / medium / long
Reward setting	Solver progress + sparse success	Solver progress + sparse success	Sparse success only	Official-score change + sparse success

Notes. ALFWorld's vocabulary is the union of admissible commands over the training set; ScienceWorld's is built from the training split of the 12 selected tasks. Solver diagnostics measure open-loop action quality.

We show the four task families used throughout our experiments: Sliding Puzzle, Sokoban, ScienceWorld, and ALFWorld. Fig. 7 shows the start and goal states of one representative instance from each image based task family, Fig. 8 shows the simplified codes for each text based task family. Each task family covers different properties and attributes, detailed in Table 6 We also show the data collection and generation engine and our environment pipeline in Figure 6

C.1 Sliding Puzzle

C.1.1 Mechanics.

A Sliding Puzzle instance is a square grid of size $n \times n$ with $n^2 - 1$ numbered tiles and one empty cell. At each primitive timestep the policy selects an action from $\{\text{up}, \text{down}, \text{left}, \text{right}\}$, sliding a tile from the indicated direction into the empty cell. A solved instance is one whose tile configuration matches a fixed target (canonical row-major: tiles $1, 2, \ldots, n^2 - 1$ in left-to-right, top-to-bottom order, with the empty cell in the bottom-right).

C.1.2 Reversibility.

Every primitive action is reversible by the inverse direction; configurations are reachable iff their permutation has even parity (half of the $(n^2)!$ configurations). This rules out unrecoverable states from primitive errors: a wrongly emitted long chunk only *wastes* primitive actions, never causes the instance to become impossible.

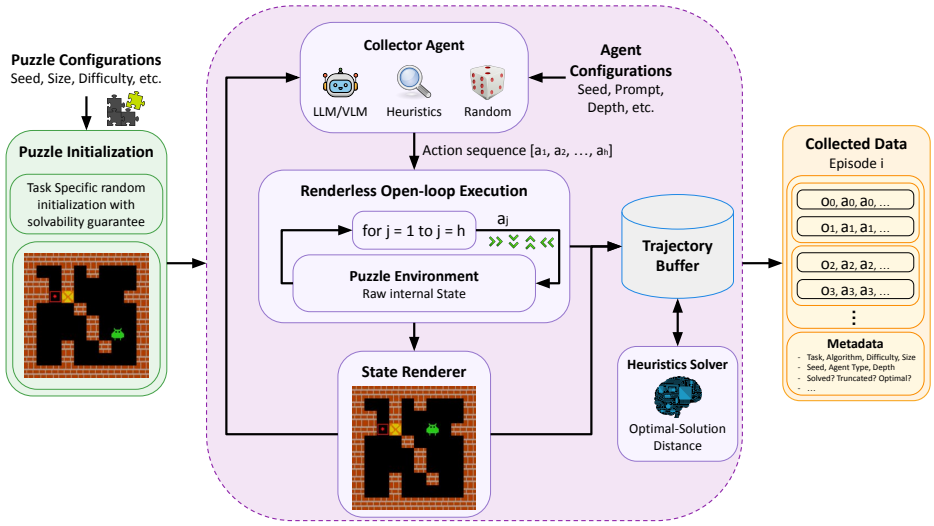

Figure 6: **Detailed pipeline for data generation and collection.** For each episode, a reproducible and configurable random environment and a selected collector agent are initialized. The agent predicts a chunk for open-loop execution and renders the next state for prediction through a separate renderer. All chunks and states are saved and compared against the heuristics solver for ground-truth-based signal.

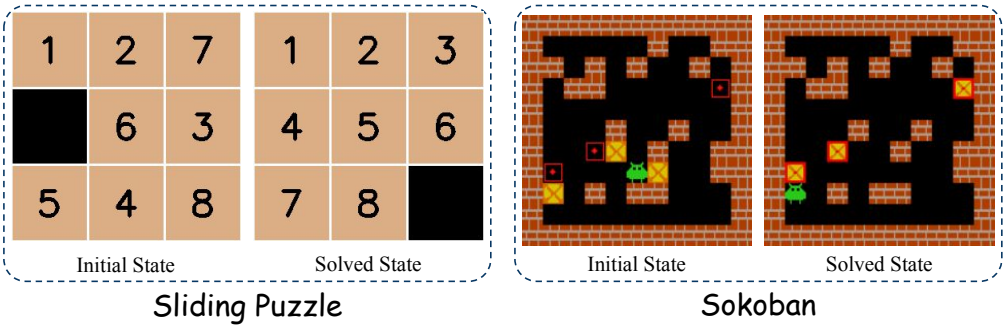

Figure 7: **Task illustrations.** For each task we show one representative instance, with the *initial state* on the left and the *goal state* on the right of each pair. *Left pair:* Sliding Puzzle (3×3). Tiles must be rearranged into canonical row-major order (goal) by sliding tiles one at a time into the empty cell; every primitive action is reversible. *Right pair:* Sokoban. The agent (Player) must push every box (B) onto a goal cell (•); pushes are not reversible by the agent, so a wrongly emitted long chunk can render the instance permanently unsolvable. Both tasks require long-horizon planning under known dynamics, but differ in reversibility: Sliding Puzzle moves are reversible, whereas Sokoban pushes can create irreversible deadlocks.

C.1.3 Difficulty parameter.

We characterize instance difficulty by the optimal solution length: the minimum number of primitive actions needed to reach the goal, computed by an exact A* solver with the linear-conflict heuristic. Longer optimal solutions require more actions, but do not necessarily imply greater difficulty for every policy.

C.1.4 Instance generation.

To the best of our knowledge, there are no existing standardized Sliding Puzzle benchmark that allows generating random setup with designated difficulty, we implement and evaluate on our own Sliding Puzzle environment. Each instance is generated by random walking from the goal configuration

ScienceWorld

```
env = ScienceWorldEnv(envStepLimit=10**6)
env.load(task, variation, simplificationStr="")
templates = env.get_possible_actions()
obs, info = env.reset()
msgs = [system_prompt(env.taskdescription(),
                      templates)]

for _ in range(K):
    msgs.append(user(render_state(obs, info)))
    reply = model(msgs, schema)
    msgs.append(reply)
    for a in resolve(reply):
        obs, r, done, info = env.step(a)
        if done: break
    if done: break

success = info["score"] == 100
```

ALFWorld

```
infos = textworld.EnvInfos(won=True,
            admissible_commands=True)
env_id = textworld.gym.register_games(
    [game_file], infos, batch_size=1,
    max_episode_steps=150,
    wrappers=[AlfredDemangler(), AlfredInfos])
env = textworld.gym.make(env_id)
(obs,), info = env.reset()
msgs = [system_prompt(task_of(obs), templates)]

for _ in range(K):
    msgs.append(user(render_state(obs, info)))
    reply = model(msgs, schema)
    msgs.append(reply)
    for a in resolve(reply):
        (obs,),_,(done,),info = env.step([a])
        if done: break
    if done: break

success = info["won"][0]
```

Figure 8: **Simplified data collection loop on the official ScienceWorld and ALFWorld packages.** At each decision point the model returns a chunk, which is executed open-loop; only the state after its last action is rendered into the next user turn.

for the requested number of primitive steps and verifying via A* that the shortest path back has the requested depth. Train and test instances are distinct samples drawn from the same generator with the same shortest-path-depth filter.

C.2 Sokoban

C.2.1 Mechanics.

A Sokoban instance is a 2D grid containing walls, free cells, the agent (player), one or more boxes, and one or more goal locations (one per box). At each primitive timestep the agent selects from $\{\text{up}, \text{down}, \text{left}, \text{right}\}$. A move into a free cell or onto a goal cell shifts the agent; a move into a box that is itself adjacent to a free cell pushes the box one step in the same direction; a move into a box adjacent to a wall or another box is no-op. A solved instance is one in which every box occupies a goal cell.

C.2.2 Irreversibility.

Box pushes are not reversible by the agent: there is no `pull` action. Pushing a box into a corner against two walls produces an unrecoverable deadlock. This asymmetry is the source of Sokoban's "planning-dominant" character: a wrongly emitted long chunk can render the instance permanently unsolvable, in contrast to Sliding Puzzle where mistakes are reversible.

C.2.3 Difficulty parameter.

We parameterise difficulty by *box count*: instances with more boxes have larger combinatorial state spaces and longer optimal plans on average. Optimal plans are computed by a domain-specific Sokoban solver (based on iterative deepening with deadlock detection); we filter generated instances to those whose optimal-plan length lies within a range appropriate for our main and budget-generalization decision budgets (Table 10).

C.2.4 Instance generation.

To the best of our knowledge, there are no existing standardized Sokoban benchmark that allows generating random setup with designated difficulty, we implement and evaluate on our own Sokoban environment. Each instance is generated through reverse pull scrambling with randomized retries, and walls are additionally greedily added to the map at locations at shuffled cells. Visualization assets are partially borrowed from Schrader (2018).

Table 7: Evaluated task types in ScienceWorld and ALFWorld. Selection criteria are detailed in App. C.3.5 and App. C.4.5, respectively.

Benchmark	Category / Family	Tasks / Official Label
ScienceWorld	Phase change	`boil`, `melt`, `freeze`, `change-the-state-of-matter-of`
	Measurement	`use-thermometer`, `measure-melting-point-known-substance`, `measure-melting-point-unknown-substance`
	Electricity	`power-component`, `power-component-renewable-vs-nonrenewable-energy`, `test-conductivity`, `test-conductivity-of-unknown-substances`
	Chemistry	`chemistry-mix`
ALFWorld	Pick & Place	`pick_and_place_simple`
	Examine in Light	`look_at_obj_in_light`
	Clean & Place	`pick_clean_then_place_in_recep`
	Heat & Place	`pick_heat_then_place_in_recep`
	Cool & Place	`pick_cool_then_place_in_recep`

C.3 SCIENCEWORLD

C.3.1 MECHANICS.

A ScienceWorld (Wang et al., 2022) instance is an elementary science task set in a text-simulated environment of interconnected rooms (kitchen, workshop, greenhouse, . . .) populated with objects, containers, and devices governed by simple thermodynamic, electrical, and chemical rules. An instance is identified by a (task, variation) pair; the variation fixes the target substance or object, its location, and the room layout. At each primitive timestep the agent issues a free-form English command instantiating one of the task's action templates (e.g. `open OBJ`, `move OBJ to OBJ`, `connect OBJ to OBJ`, `focus on OBJ`, `wait`), with objects named as they appear in the observations. A command the parser does not accept leaves the state unchanged but still consumes a primitive step. We use the unsimplified environment: doors must be opened and there is no teleport action. Each task defines an ordered sequence of subgoals and a score in $[0, 100]$; a solved instance is one in which every subgoal has been met (score 100).

C.3.2 IRREVERSIBILITY.

ScienceWorld is the only environment in our suite in which an episode can fail before its budget is exhausted: `focus on` an object other than the one the task designates ends the episode immediately with score -100. Many physical transformations (mixing substances, melting or boiling a sample, using up the only instance of an object) likewise cannot be undone. A wrongly emitted long chunk can therefore either terminate the episode outright or remove the object the task depends on.

C.3.3 REFERENCE HORIZONS.

We group instances by the length of the simulator's reference action chunk for each variation. Observed reference lengths span 6–22 actions in the short group, 23–32 in medium, and 35–153 in long, with decision budgets $K = 6$, 9, and 18, respectively. These groups describe reference horizons rather than calibrated difficulty levels. Larger budgets provide greater action capacity for the longer-horizon groups.

C.3.4 INSTANCE GENERATION.

We use official ScienceWorld package for data collection. Instances are not generated procedurally: each is a (task, variation) pair drawn from ScienceWorld evaluation set, split by horizon length. The three buckets span 12 task types (Table 7).

C.3.5 SCIENCEWORLD TASK SELECTION

Three cuts to twelve. Selection accounts for grounded vocabulary size and compatibility with our execution limit. (i) *Plant/genetics (4 removed).* In our integration audit, these tasks generate entities at runtime and contribute approximately 550k of the 599k enumerated commands (92%). With a 512-dimensional decoder output, a 600k-way classification layer alone would require approximately 307M weights, making this inventory impractical for our training setup. (ii) *Inclined-plane (3 removed).* Sampled gold trajectories reach 333 actions, exceeding our 150-step evaluation cap. We exclude these tasks because their reference trajectories are not consistently compatible with this cap; shorter solutions may nevertheless exist. (iii) *Remaining biology and paint-mixing (11 removed).* We further restrict the study to the laboratory procedures in Table 7 to control vocabulary growth. The paint-mixing pair alone adds approximately 14.9k commands. The decoder vocabulary constructed from the training split of the 12 retained task types contains 658 commands (Table 11).

Scope of evaluation. Our factorized decoder avoids enumerating complete action chunks, while its output layer scales linearly with the grounded action vocabulary (§3.3). Task selection keeps this inventory manageable under our architecture and training resources. Accordingly, our ScienceWorld findings apply to the selected 12 task types under the stated action interface and execution cap; they do not establish performance on the full 30-task benchmark.

Horizon stratification. Episodes are grouped by reference-trajectory length. Held-out median lengths are 18, 25, and 57 for the short, medium, and long groups, respectively. Reference lengths are used for grouping and budget configuration, never as policy inputs.

Task list. Table 7 lists the twelve retained task types by category.

C.4 ALFWORLD

C.4.1 MECHANICS.

An ALFWorld (Shridhar et al., 2021) instance is a TextWorld (Côté et al., 2019) game aligned with a household task from the ALFRED benchmark (Shridhar et al., 2020). The room contains numbered receptacles (`drawer 3`, `countertop 1`, ...) and objects placed in or on them; object locations are hidden until the agent visits, and if necessary opens, the receptacle holding them. At each primitive timestep the agent issues a command from a fixed set of templates, such as `go to`, `open`, `close`, `take ... from`, `move ... to`, `heat`, `cool`, `clean`, and `use`. The agent must go to a receptacle before interacting with it and carries at most one object at a time. A command the game does not accept is answered with "Nothing happens." and still consumes a primitive step. Success is binary with no partial credit: a solved instance is one in which the task's goal condition holds, e.g. a cooled apple placed on the countertop.

C.4.2 REVERSIBILITY.

Some action errors are recoverable: misplaced objects can often be retrieved and moved elsewhere. However, recovery, unsuccessful searches, and invalid commands consume environment steps and may prevent completion within the decision budget.

C.4.3 DIFFICULTY PARAMETER.

ALFWorld has no explicit difficulty parameter, and all instances share the decision budget $K{=}9$, which is enforced but not disclosed to the model. We instead report results separately on the two evaluation splits: *valid-seen*, whose room layouts appear in the ALFWorld training games, and *valid-unseen*, whose layouts do not. As a reference solution length we record the length of the plan returned by ALFWorld's PDDL planner; since the planner has full observability of object locations, this is a lower bound that an agent which must search for objects cannot generally attain.

C.4.4 INSTANCE GENERATION.

We use official alfworld package for data collection. Instances are the fixed ALFWorld evaluation games rather than generated ones: we take every game in the valid-seen and valid-unseen splits except those of the `pick_two_obj_and_place` task type (Table 7).

C.4.5 ALFWORLD TASK SELECTION

We evaluate five of the six official task families under the official seen/unseen splits (Table 7), excluding `pick_two_obj_and_place` (Pick2) for the following reasons.

Limited post-SFT performance. Pick2 success remains below 10% after supervised warm start, even under step-wise control with $K = 150$. This low starting success motivates prioritizing timing comparisons on the retained families. Published prompting baselines also vary substantially: BEACON reports Pick2 success of 0–3.7% with Qwen2.5-1.5B-Instruct and 3.2–23.8% with Qwen2.5-7B-Instruct (Wang et al., 2026). These provide context rather than matched comparisons with our model.

Training investment and study scope. Prior work demonstrates that Pick2 benefits from finer-grained credit assignment. With Qwen2.5-7B-Instruct, BEACON reports Pick2 success of 64.7%, 79.2%, and 90.0% for GRPO, GiGPO, and BEACON, respectively (Wang et al., 2026). GiGPO's ALFWorld configuration uses 150 iterations with 16×8 trajectories per iteration, approximately 19,200 training trajectories (Feng et al., 2025). This is a benchmark-wide training allocation, not a Pick2-specific requirement. These advances are complementary to our focus on decision timing. Under our compute budget, we prioritize the retained families and leave integrating existing training techniques to improve Pick2 performance to future work. Our ALFWorld results therefore concern the five selected families.

Horizon coverage. Reference trajectories in the five retained task families have a median length of 14 actions and a 90th percentile of approximately 33 actions.

Task list. Table 7 lists the five retained ALFWorld task families and their official labels.

D MORE EXPERIMENT DETAILS

D.1 MODEL ARCHITECTURE DETAILS AND TRAINING HYPERPARAMETERS

D.2 MODEL ARCHITECTURE AND TRAINING HYPERPARAMETERS

D.2.1 MODEL ARCHITECTURE

All four environments, including both puzzles, use Qwen2.5-VL-7B-Instruct (Bai et al., 2025) with LoRA adapters (Hu et al., 2021) in the language backbone. Pretrained backbone weights, including the vision encoder, remain frozen. SFT updates the adapters and action decoder while freezing the depth head; joint RL updates all three.

The depth head predicts $h_t \in \mathcal{H} = \{1, 2, 4, 8\}$ from the shared representation z_t. The action decoder autoregressively generates eight actions from z_t and preceding action embeddings, independently of the selected depth. Each environment uses an output projection over its action vocabulary $\mathcal{A}_e$.

D.2.2 RL TRAINING HYPERPARAMETERS

D.3 TRAINING DYNAMICS

Figure 9 tracks action and depth entropy during RL rollouts. Depth entropy remains relatively steady in the puzzles, declines gradually in ALFWorld, and decreases initially before stabilizing in ScienceWorld. Neither mean action nor mean depth entropy approaches zero over the displayed interval.

Table 8: **Architecture and LoRA configuration.** Settings are shared across all four environments, except for action-vocabulary size.

Parameter	Value
Backbone adaptation	
LoRA rank / alpha	16 / 32
LoRA dropout	0.05
LoRA targets	Language-model attention and MLP projections
Precision	bf16
Depth head	
Architecture	Linear: $3{,}584 \to 4$
Depth menu $\mathcal{H}$	$\{1, 2, 4, 8\}$
Autoregressive action decoder	
Layers / attention heads	2 / 4
Hidden / FFN dimension	512 / 1,024
Dropout / activation	0.1 / GELU
Normalization	Pre-norm
Context projection	Linear: $3{,}584 \to 512$
Action embedding dimension	512
Maximum depth	8 actions
Output vocabulary size $\lvert\mathcal{A}_e\rvert$	
Sliding Puzzle / Sokoban	4 / 4
ALFWorld / ScienceWorld	5,698 / 658

Table 9: **Main-experiment RL hyperparameters.** Backbone learning rates apply to LoRA adapters. ScienceWorld budgets correspond to short, medium, and long episodes.

Hyperparameter	Sliding	Sokoban	ALFWorld	ScienceWorld
Training cycles	50–100	50–100	50–100	50–100
Effective batch size	128	128	128	128
GRPO group size	8	8	8	8
Training budget K_{train}	15	6	9	6 / 9 / 18
Depth menu $\mathcal{H}$	$\{1, 2, 4, 8\}$			
GRPO clip ϵ	0.05	0.05	0.20	0.20
KL coefficient β	0.05	0.20	0.02	0.02
Depth entropy coefficient α_h	0.01	0.01	0.01	0.01
Action entropy coefficient α_a	0.001	0.001	0.001	0.003
Backbone LoRA learning rate	5×10^{-7}	2×10^{-7}	5×10^{-6}	5×10^{-6}
Head learning rate	7×10^{-6}	3×10^{-6}	1×10^{-4}	1×10^{-4}

D.4 Baselines and Heuristics

Repeat-last. This baseline follows the dynamic action-repetition idea of FiGAR (Sharma et al., 2017), applied at evaluation time to our trained policy without retraining. At each decision, the depth head selects $h \in \mathcal{H}$ as usual, but only the first proposed action is kept and repeated h times open-loop. Everything else (checkpoint, decision budget, and evaluation episodes) matches the main evaluation. Unlike FiGAR, which learns the repetition count, we reuse the trained depth head for this purpose. The baseline therefore keeps PACE's timing but removes the variety of actions within a chunk, so the gap to PACE reflects the value of what is executed rather than when the policy re-observes.

Entropy (threshold, best-τ). This baseline shortens execution when the policy is uncertain: the executed depth is truncated once the predictive entropy exceeds a threshold τ, so uncertain states lead to earlier re-observation. Thresholds are set as fractions of the maximum possible entropy, so they do

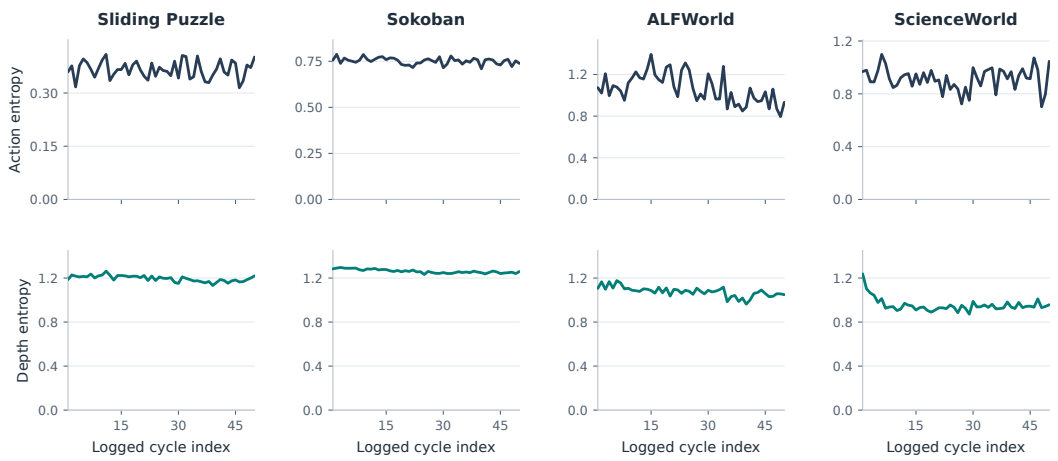

Figure 9: **Action and depth entropy during RL rollouts.** Curves show logged mean action entropy (top) and depth entropy (bottom) for the first 50 cycle records in each environment, without smoothing. The horizontal axis indexes records in file order, rather than absolute training-cycle IDs. Action-entropy scales vary by environment; depth-entropy scales are shared.

Table 10: **Hard decision budgets.** Main experiments match training and evaluation budgets. Budget-generalization experiments increase the evaluation budget without retraining. ALFWorld uses the same five selected task families in training and evaluation.

Setting	Main K (train = eval)	Generalization (train $\rightarrow$ eval)
Sliding Puzzle	15	$5 \rightarrow 15$
Sokoban	6	$3 \rightarrow 4$
ALFWorld	9	$5 \rightarrow 9$
ScienceWorld short	6	$6 \rightarrow 12$
ScienceWorld medium	9	$9 \rightarrow 18$
ScienceWorld long	18	$18 \rightarrow 36$

not depend on a particular checkpoint. For each setting, we report the best τ from the tested grid, chosen in hindsight on the evaluation set (marked † in Table 2). This choice favors the baseline, so PACE's advantage over it is a conservative comparison.

D.5 Training Details and Decision Budgets

Decision-budget configuration. Table 10 lists the main and budget-generalization configurations. Budget choices were informed by reference-trajectory lengths to impose decision pressure while allowing longer execution depths to extend action capacity. A fixed depth h permits at most Kh primitive actions, subject to termination and environment limits; this capacity does not guarantee task completion. ALFWorld uses one model and a common budget across its five retained task families. ScienceWorld uses one model trained on all three horizon groups, with a separate budget for each group.

Action vocabularies and output heads. Table 11 reports the fixed action vocabulary $\mathcal{A}_e$ used by each domain's decoder. The puzzles use four movement directions. ALFWorld uses 5,698 commands, obtained by taking the union of admissible commands over the training set. ScienceWorld uses 658 commands constructed from the training split of the 12 selected task types. Each action-decoder output has $|\mathcal{A}_e|$ logits.

Checkpoint sharing. Table 12 summarizes model sharing. SFT updates the backbone LoRA and action decoder while keeping the depth head frozen. RL jointly updates the backbone LoRA, action decoder, and depth head; pretrained backbone weights remain frozen throughout both stages. For

Table 11: **Per-domain action vocabulary sizes**. The puzzle domains use the four move directions. The ALFWorld vocabulary is the union of admissible commands over the training set; the ScienceWorld vocabulary is built from the training split of the 12 selected tasks.

Setting	$\lvert\mathcal{A}_e\rvert$
Sliding Puzzle / Sokoban	4
ALFWorld	5,698
ScienceWorld (12 tasks)	658

Table 12: Model sharing during supervised and reinforcement learning.

Domain	SFT	RL
Sliding Puzzle	Shared with Sokoban	Separate puzzle-specific model
Sokoban	Shared with Sliding Puzzle	Separate puzzle-specific model
ALFWorld	One model across five task types	One model across the same five task types
ScienceWorld	One model across three horizon groups	One model with mixed-budget training

ALFWorld, both stages use the same five task types: Pick & Place, Clean & Place, Heat & Place, Cool & Place, and Examine in Light. Pick2 is excluded from both stages.

Out-of-vocabulary handling. Action vocabularies are constructed from training data only. Held-out episodes are flagged when their expert demonstration contains a command outside the vocabulary; Table 13 reports the incidence within the evaluated task families. All main-table results use the full evaluation sets: flagged episodes remain in the denominator and are scored by their actual rollouts, with no score imputed.

A flag indicates incomplete coverage of the expert path, not necessarily an unsolvable task.

Across the audited configurations, 9 of the 17 flagged ALFWorld episodes and 5 of the 13 flagged ScienceWorld episodes are solved by at least one configuration through an in-vocabulary trajectory. Our results assess adaptive timing under the fixed action inventories and reference-path coverage reported here. The puzzles use a closed four-action vocabulary and have no OOV cases.

Table 13: Held-out episodes whose expert demonstrations contain out-of-vocabulary commands. Main results retain all episodes, scored by actual rollouts. Solved counts are aggregated across audited configurations, not success rates for a single policy.

Split	Flagged fraction	Solved via in-vocab alternatives
ALFWorld seen	3.4%	4/4
ALFWorld unseen	11.1%	5/13
ScienceWorld held-out	7.0%	5/13

D.6 Efficiency

Environment interaction steps. Figure 10 complements the decision-count analysis in Fig. 3. PACE achieves higher observed success than every tested fixed depth, while interaction-cost savings vary by setting. On Sokoban, it uses 30.1 environment steps per episode versus 40.4 for the highest-success fixed depth. Decision counts and environment steps therefore capture distinct aspects of efficiency.

Table 14: **Aggregate executed cost per success.** Lower is better. Fixed-depth comparators are selected by highest success, not lowest cost per success. For tied depths, slash-separated values follow the order in the h column. Bold indicates that PACE has lower cost than every listed fixed-depth comparator in that setting. Costs include both successful and failed episodes.

		Decisions per success		Steps per success	
Setting	Fixed h	Fixed	PACE	Fixed	PACE
Sliding Puzzle	3	26.17	**18.79**	77.31	**57.54**
Sokoban	8	16.00	**14.57**	123.19	**83.61**
ALFWorld seen	4 / 7	15.64 / 14.92	**11.89**	60.75 / 100.28	**44.30**
ALFWorld unseen	7 / 8	18.91 / 18.78	**13.79**	128.56 / 145.60	**84.00**
ScienceWorld short	6	15.13	**10.90**	88.62	**65.00**
ScienceWorld medium	5 / 7	12.34 / 11.59	**9.24**	59.81 / 77.84	61.95
ScienceWorld long	8	29.41	**21.29**	231.19	**130.26**

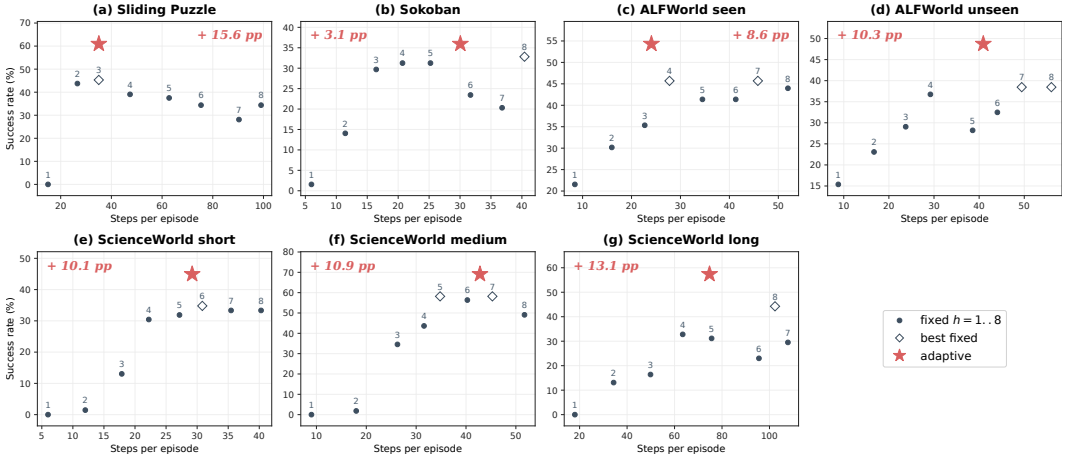

Figure 10: **Success–interaction trade-offs across seven settings.** Stars denote PACE; diamonds mark the highest-success fixed depths, including ties; circles show the remaining tested fixed depths. All fixed-depth policies share PACE's RL-trained checkpoint. Annotations give success gains over the best fixed depth in percentage points. Environment steps are averaged over all evaluation episodes, including failures, and include invalid action attempts. Axis ranges vary by setting.

Cost per success. We report total executed cost per observed success: $C_{\text{dec}} = \sum_i D_i / \sum_i y_i$ and $C_{\text{step}} = \sum_i S_i / \sum_i y_i$, where D_i, S_i, and y_i denote episode-level decision count, environment-step count, and binary success. Both ratios include costs incurred by failed episodes. Against the highest-success fixed depths, PACE uses fewer decisions per success in all seven settings and fewer steps per success in six (Table 14). On ScienceWorld medium, its steps per success lie between the two tied fixed-depth policies.

D.7 Depth Allocation under Relaxed Budgets

Figure 11 complements the matched-budget analysis in Fig. 5. With evaluation budgets exceeding training budgets, both puzzles again exhibit greater mean selected depth near the goal than farther away. The distance-conditioned allocation pattern therefore appears under both budget configurations.

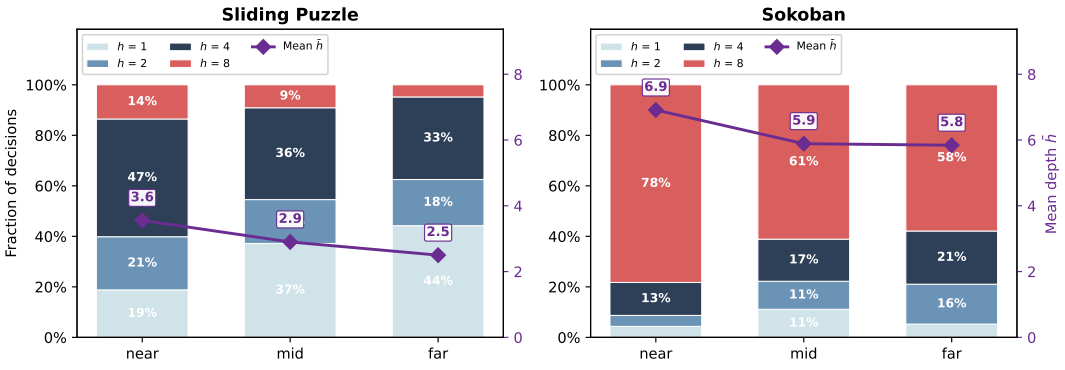

Figure 11: Chosen commitment depth versus remaining solver distance for the adaptive policy on Sliding and Sokoban, evaluated under a relaxed decision budget: the budget K at test time is larger than the budget used during training (budget-transfer setting; see the evaluation protocol in the paper). Stacked bars give the share of decisions at each chosen depth h within each distance bucket (near / mid / far, bucketed by solver distance at decision time); purple diamonds give the mean depth per bucket. Buckets with fewer than 15 decisions are hatched. The policy commits deep when close to the goal and shortens its commitments as the remaining distance grows.

D.8 Depth distributions across settings

Figure 12 summarizes selected-depth frequencies under the main evaluation configuration. All four depths occur in every setting, although their frequencies vary substantially. ALFWorld seen distributes selections relatively broadly, whereas ALFWorld unseen and all three ScienceWorld groups favor $h = 8$. These aggregate distributions establish the use of multiple depths; the puzzle analysis above additionally relates depth choices to task progress.

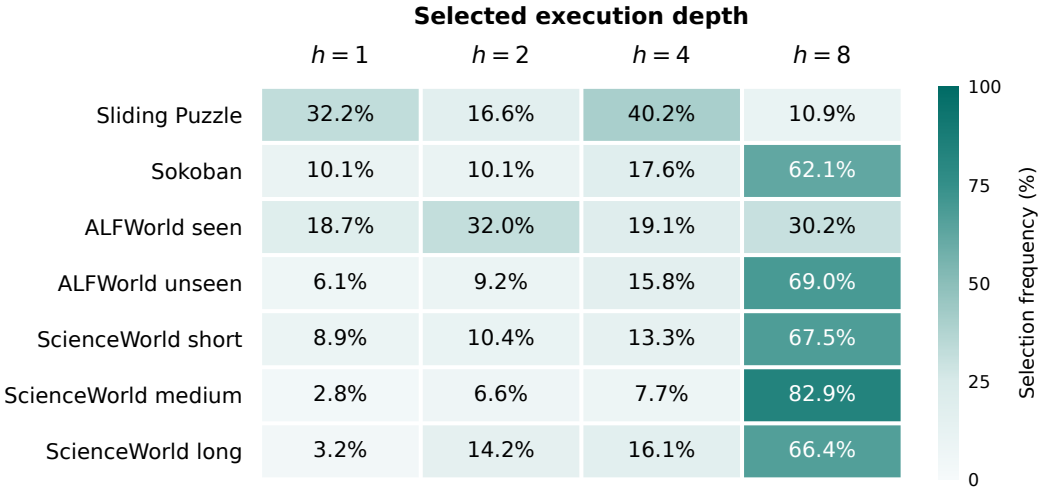

Figure 12: **Selected execution-depth distributions across seven settings.** Cells report selection frequencies (%) for $h \in \{1, 2, 4, 8\}$ under the main adaptive policy; darker shading indicates higher frequency. Values describe selected depths, not execution lengths truncated by episode termination. Percentages may sum to 99.9 or 100.1 due to rounding.

D.9 FAILURE CASES

Open-loop simulation errors. Because a chunk is executed without intermediate observations, the model must predict the effect of each of its own actions and plan the rest of the chunk on top of that prediction. The dominant failure in the visual puzzles is a mismatch between this internal simulation and the true dynamics, which compounds over the chunk. In Sokoban, such errors are often irreversible: a single miscounted push can drive a box against a wall or into a corner, after which no solution exists and the rest of the budget is spent in a dead state. Moves into walls are a direct symptom, since they reveal that the position the model assumed for its player differs from the true one. Figure 1 (a) shows a representative case. The initial state is solvable in 10 moves, and the chunk ends with `U`, which pushes the upper box toward the top goal only if the player stands directly below it, i.e., after exactly two `L` moves. The model issues three. The third `L` pushes the lower box into a corner, making the puzzle unsolvable, and the final `U` walks into a wall. The model sees the outcome only after the entire chunk has executed, and all five remaining decisions are spent in an unsolvable state. In the Sliding Puzzle, failures are less abrupt but similar in kind: after a chunk the board frequently drifts away from the goal, and many failed episodes end farther from the solution than they started.

Premature chunk. In ScienceWorld, a task is completed by a terminal action, such as focusing on an object or placing it in one of several answer boxes, and a wrong terminal action ends the episode with failure. Under open-loop execution, models frequently place the evidence-gathering step and the answer in the same chunk, so the answer is fixed before the evidence can be observed. Figure 13 shows such a case under Fixed-8. The model attempts to build a circuit to test whether an apricot conducts electricity, inspects the light bulb, and places the apricot in the box reserved for conductive substances, all within one chunk. None of the five `connect` commands succeeds, because each omits the terminal names the simulator requires. The bulb remains disconnected and off, yet the answer has already been scheduled. The episode ends with a score of -100 after only three of its 18 decisions, so the failure is caused by executing chunk too early, not by budget pressure. The same pattern appears when models focus on a substance and immediately on an answer box before any temperature has been read.

Other failures. The remaining failures mostly exhaust the decision budget without reaching the goal. In ALFWorld, this typically reflects inefficient search, which open-loop execution makes worse. Models often script a fixed routine, such as `go to cabinet 4`, `open cabinet 4`, `take mug from cabinet 4`, before observing the receptacle. The `open` is then rejected because the receptacle is already open or has no door, and the `take` is rejected because the object is not there. Each rejected command still consumes a step. Weaker open models additionally struggle with the response format itself. Because a format-violating chunk is not executed and leaves the observation unchanged, such models tend to resubmit the identical response, consuming the budget without acting.

Task: ... determine if apricot is electrically conductive. …
First, focus on the apricot. If it is electrically conductive, place it in the yellow box. If it is electrically nonconductive, place it in the purple box.
Look: This room is called the workshop. ... a purple box (containing nothing) a table. On the table is: a battery, a black wire, ..., a green light bulb, which is off, ... a yellow box (containing nothing) ...
Inventory: ... a orange apricot ...
Score: 60/100

Model response (decision 3 of 18):

connect battery to black wire
→ Connections must specify specific points of contact on polarized electrical components (e.g. anode, cathode)
connect black wire to apricot
→ Connections must specify specific points of contact on unpolarized electrical components (e.g. terminal 1, terminal 2)
connect apricot to red wire (same error)
connect red wire to green light bulb (same error)
connect green light bulb to battery (same error)
look at green light bulb
→ a green light bulb, which is off. its anode is connected to: nothing. its cathode is connected to: nothing.
move apricot to yellow box
→ You move the apricot to the yellow box. *Score: -100 (task failed)*

Figure 13: **Premature chunk in ScienceWorld**. The top block is an excerpt of the state the model received before its third decision. All actions below were proposed in a single chunk. Each simulator response shown was produced during execution but was not visible to the model when it chose the next action. The circuit is never built, yet the answer is executed in the same chunk. The chunk's eighth action (`look around`) was discarded when the episode ended.

E Zero-Shot Depth Selection: Results and Protocol

E.1 Results

We report success rates for prompted self-selected depth and all four tested fixed depths, including conditions that tie or outperform self-selected depth. Comparisons are within each model and evaluation pool; reasoning configurations and protocol differences are documented in App. E.2.

Table 15: **Zero-shot execution-depth comparisons.** Success rates (%) for prompted self-selected depth and fixed depths $h \in \{1, 2, 4, 8\}$. Self-selected depth is prompted rather than learned through PACE training. Reasoning configurations are model-specific; these are not compute-matched cross-model comparisons.

Model	Setting	Self-selected	Fixed-1	Fixed-2	Fixed-4	Fixed-8
Claude-Sonnet-5	Sliding Puzzle	3.1%	0.0%	0.0%	1.6%	1.6%
(Anthropic, 2026)	Sokoban	6.3%	0.0%	3.1%	7.8%	10.9%
Gemini-3.1-Pro-Preview	Sliding Puzzle	0.0%	0.0%	0.0%	0.0%	0.0%
(Team et al., 2025)	Sokoban	20.3%	0.0%	6.3%	17.2%	18.8%
GPT-5.6-Terra	Sliding Puzzle	1.6%	0.0%	0.0%	4.7%	6.3%
(Singh et al., 2026)	Sokoban	57.8%	1.6%	17.2%	32.8%	43.8%
Qwen2.5-VL-7B-Instruct	Sliding Puzzle	0.0%	0.0%	0.0%	0.0%	0.0%
(Bai et al., 2025)	Sokoban	0.0%	0.0%	0.0%	0.0%	0.0%
Qwen3.8-27B	Sliding Puzzle	4.7%	0.0%	3.1%	3.1%	15.6%
(Qwen Team, 2026b)	Sokoban	18.8%	0.0%	10.9%	21.9%	10.9%
Qwen3.8-Flash-Next	Sliding Puzzle	21.9%	0.0%	4.7%	34.4%	53.1%
(Qwen Team, 2026a)	Sokoban	28.1%	1.6%	17.2%	18.8%	34.4%
GLM-5.3-Flash	Sliding Puzzle	26.6%	0.0%	10.9%	28.1%	17.2%
(GLM-5-Team et al., 2026)	Sokoban	17.2%	1.6%	15.6%	20.3%	26.6%
Gemma-4-31B	Sliding Puzzle	4.7%	0.0%	0.0%	0.0%	0.0%
(Team, 2026)	Sokoban	34.4%	1.6%	12.5%	26.6%	31.2%
DeepSeek-V4-Flash-Vision-Exp	Sliding Puzzle	0.0%	0.0%	0.0%	0.0%	0.0%
(DeepSeek, 2026)	Sokoban	10.9%	0.0%	4.7%	6.2%	10.9%

Table 16: **ALFWorld Zero-shot execution-depth comparisons.** Success rates (%) for prompted self-selected depth and fixed depths $h \in \{1, 2, 4, 8\}$. Self-selected depth is prompted rather than learned through PACE training. Reasoning configurations are model-specific; these are not compute-matched cross-model comparisons.

Model	Setting	Self-selected	Fixed-1	Fixed-2	Fixed-4	Fixed-8
Claude-Sonnet-5	All	94.8%	81.5%	92.7%	92.3%	92.7%
(Anthropic, 2026)	seen	94.8%	85.3%	94.8%	91.4%	89.7%
	unseen	94.9%	77.8%	90.6%	93.2%	95.7%
Gemini-3.1-Pro-Preview	All	97.9%	78.1%	91.4%	97.4%	98.7%
(Team et al., 2025)	seen	96.6%	81.0%	90.5%	94.8%	97.4%
	unseen	99.1%	75.2%	92.3%	100.0%	100.0%
GPT-5.6-Terra	All	91.4%	45.5%	64.8%	85.4%	95.7%
(Singh et al., 2026)	seen	89.7%	44.0%	58.6%	81.9%	95.7%
	unseen	93.2%	47.0%	70.9%	88.9%	95.7%
Qwen2.5-VL-7B-Instruct	All	4.3%	12.9%	11.2%	2.6%	4.3%
(Bai et al., 2025)	seen	5.2%	13.8%	14.7%	4.3%	6.9%
	unseen	3.4%	12.0%	7.7%	0.9%	1.7%
Qwen3.8-27B	All	91.8%	48.1%	79.8%	89.3%	96.1%
(Qwen Team, 2026b)	seen	93.1%	46.6%	77.6%	85.3%	95.7%
	unseen	90.6%	49.6%	82.1%	93.2%	96.6%
Qwen3.8-Flash-Next	All	88.8%	63.9%	79.8%	91.4%	98.3%
(Qwen Team, 2026a)	seen	92.2%	62.9%	83.6%	93.1%	97.4%
	unseen	85.5%	65.0%	76.1%	89.7%	99.1%
GLM-5.3-Flash	All	90.6%	56.2%	78.5%	87.6%	75.1%
(GLM-5-Team et al., 2026)	seen	88.8%	56.0%	75.9%	86.2%	78.4%
	unseen	92.3%	56.4%	81.2%	88.9%	71.8%
Gemma-4-31B	All	73.0%	41.2%	56.7%	62.7%	58.4%
(Team, 2026)	seen	72.4%	41.4%	51.7%	58.6%	53.4%
	unseen	73.5%	41.0%	61.5%	66.7%	63.2%
DeepSeek-V4-Flash-Vision-Exp	All	77.7%	35.6%	48.1%	76.0%	91.8%
(DeepSeek, 2026)	seen	72.4%	31.0%	49.1%	73.3%	88.8%
	unseen	82.9%	40.2%	47.0%	78.6%	94.9%

Table 17: **ScienceWorld Zero-shot execution-depth comparisons.** Success rates (%) for prompted self-selected depth and fixed depths $h \in \{1, 2, 4, 8\}$. Self-selected depth is prompted rather than learned through PACE training. Reasoning configurations are model-specific; these are not compute-matched cross-model comparisons.

Model	Setting	Self-selected	Fixed-1	Fixed-2	Fixed-4	Fixed-8
Claude-Sonnet-5	All	54.1%	36.5%	49.2%	64.6%	61.3%
(Anthropic, 2026)	short	31.3%	10.4%	25.4%	58.2%	56.7%
	medium	51.9%	31.5%	42.6%	53.7%	63.0%
	long	81.7%	70.0%	81.7%	81.7%	65.0%
Gemini-3.1-Pro-Preview	All	78.5%	38.7%	63.0%	77.3%	81.2%
(Team et al., 2025)	short	73.1%	14.9%	40.3%	68.7%	70.1%
	medium	81.5%	38.9%	72.2%	79.6%	87.0%
	long	81.7%	65.0%	80.0%	85.0%	88.3%
GPT-5.6-Terra	All	68.0%	40.9%	56.4%	72.9%	64.1%
(Singh et al., 2026)	short	46.3%	20.9%	31.3%	58.2%	52.2%
	medium	72.2%	33.3%	51.9%	74.1%	66.7%
	long	88.3%	70.0%	88.3%	88.3%	75.0%
Qwen2.5-VL-7B-Instruct	All	1.1%	0.6%	0.6%	0.0%	0.0%
(Bai et al., 2025)	short	0.0%	0.0%	0.0%	0.0%	0.0%
	medium	1.9%	0.0%	1.9%	0.0%	0.0%
	long	1.7%	1.7%	0.0%	0.0%	0.0%
Qwen3.8-27B	All	71.8%	40.9%	60.8%	66.9%	65.2%
(Qwen Team, 2026b)	short	59.7%	10.4%	37.3%	55.2%	49.3%
	medium	75.9%	48.1%	70.4%	63.0%	64.8%
	long	81.7%	68.3%	78.3%	83.3%	83.3%
Qwen3.8-Flash-Next	All	65.7%	36.5%	60.2%	68.5%	64.6%
(Qwen Team, 2026a)	short	49.3%	14.9%	41.8%	53.7%	44.8%
	medium	70.4%	48.1%	63.0%	70.4%	64.8%
	long	80.0%	50.0%	78.3%	83.3%	86.7%
GLM-5.3-Flash	All	40.3%	22.1%	43.1%	53.0%	49.2%
(GLM-5-Team et al., 2026)	short	16.4%	7.5%	29.9%	32.8%	28.4%
	medium	42.6%	22.2%	42.6%	51.9%	48.1%
	long	65.0%	38.3%	58.3%	76.7%	73.3%
Gemma-4-31B	All	49.7%	24.3%	33.7%	58.6%	47.5%
(Team, 2026)	short	35.8%	7.5%	19.4%	52.2%	31.3%
	medium	51.9%	33.3%	31.5%	55.6%	44.4%
	long	63.3%	35.0%	51.7%	68.3%	68.3%
DeepSeek-V4-Flash-Vision-Exp	All	39.8%	27.1%	39.2%	45.3%	49.7%
(DeepSeek, 2026)	short	16.4%	10.4%	20.9%	28.4%	38.8%
	medium	46.3%	31.5%	38.9%	48.1%	50.0%
	long	60.0%	41.7%	60.0%	61.7%	61.7%

E.2 Evaluation Protocol

Purpose and scope. This evaluation asks whether prompting a pretrained model to select its execution depth improves performance over prescribing a constant depth. No task-specific parameter updates are performed. We distinguish prompted self-selected depth from PACE's trained depth head. These experiments provide supplementary evidence about execution timing, rather than a matched ranking against PACE: the systems differ in training, action representation, context handling, and reasoning configuration. Moreover, changing the requested chunk length can change the generated actions themselves. The prompted conditions therefore do not isolate timing in the same way as PACE's same-checkpoint depth overrides.

Depth conditions and decision budgets. Each evaluation entry is tested under five conditions. In the self-selected condition, the model chooses $h \in \mathcal{H} = \{1, 2, 4, 8\}$ at each decision. In Fixed-h, it is instructed to return exactly h actions. The best fixed depth is selected in hindsight from these four values within the reported evaluation pool, rather than the eight depths tested in PACE's main comparisons.

Decision budgets are $K = 15$ for Sliding Puzzle, $K = 6$ for Sokoban, $K = 9$ for ALFWorld, and $K = 6/9/18$ for ScienceWorld short/medium/long, matching the main experiments. Every returned response consumes one decision, including responses rejected for format violations.

Interaction and observations. Each episode is a multi-turn interaction. At a decision boundary, the model receives the current observation and returns an action chunk. The chunk executes open-loop, without intermediate feedback; new information is supplied at the next decision boundary.

The puzzle interface uses rendered board observations. For ALFWorld and ScienceWorld, observations are structured text summaries accompanied by the preceding chunk's actions and environment responses. These responses become available only after execution and cannot alter actions already selected within the chunk. Conversation history is retained across decisions. Provider-supported reasoning content or continuation metadata is preserved in its supported format.

Within each environment, prompt templates share the task description and observation format, while depth instructions and response schemas depend on the condition. Prompts specify action semantics and output requirements without providing solved demonstrations. Examples appear in App. E.3.

Structured responses and validation. The requested JSON schemas are {`"h": int, "actions": [str]`} for self-selected depth, {`"action": str`} for Fixed-1, and {`"actions": [str]`} for other fixed depths. Puzzle actions are drawn from {`U,D,L,R`}; text-domain actions are generated commands whose validity is determined by the environment.

A response is rejected if its action list is missing or ill-typed, if its declared depth disagrees with the list length, or if a fixed-depth response contains the wrong number of actions. Rejected responses execute no actions but still consume a decision. The model receives no additional corrective error message.

For a self-consistent positive depth outside $\mathcal{H}$, the executor retains the prefix of length $\max\{h' \in \mathcal{H} : h' \leq h\}$. Clipping and format violations are logged separately. Observed performance therefore reflects both task execution and compliance with the requested interface.

Execution and termination. Actions in an accepted chunk are passed to the environment in order. Nonterminal invalid commands consume environment steps without triggering replanning. Execution stops when the environment terminates, a configured environment-step limit is reached, or the decision budget is exhausted. Any unexecuted chunk suffix is discarded.

Reasoning and serving configurations. Closed models use their providers' low reasoning-effort settings to limit evaluation cost. Open-weight models use the highest reasoning setting available in the evaluated serving configuration. These settings are not calibrated to equal token, latency, or computational budgets. The results therefore do not establish a compute-matched ranking across models or between closed and open-weight systems.

Within each model, the reasoning and decoding configuration is held fixed across depth conditions. Closed models use provider-native structured-output interfaces; open-weight models use the serving framework's guided decoding. These mechanisms share the requested schemas but need not induce identical decoding behavior.

Vocabulary coverage. Episodes flagged because their expert demonstrations contain commands outside PACE's vocabulary remain in the primary zero-shot evaluation pools and are scored by actual rollout outcomes. For text-generating models, this flag describes reference-path coverage relative to PACE's vocabulary; it does not restrict their own command support.

Aggregation and interpretation. Where results are pooled across splits or horizon groups, success is computed from pooled success counts and episode counts. The best fixed depth for a pooled result

is selected after pooling each fixed-depth condition. This differs from selecting a separate best fixed depth within every split or horizon group.

All evaluated models and depth conditions are reported, including ties, near-zero results, and cases where self-selected depth underperforms fixed depth. High or low success under this protocol does not, by itself, identify the contribution of pretraining, reasoning effort, or interface constraints.

Fault handling and logging. Rate-limit rejections are retried without consuming a decision. Infrastructure failures that prevent obtaining a response trigger recollection of the episode; parseable responses that violate the action schema are charged under the validation rules above.

Logs retain model identifiers, inference configurations, prompts, raw responses, requested and executed actions, validation outcomes, environment feedback, token usage, and latency. These records distinguish failures of the evaluated policy interface from serving-infrastructure failures.

E.3 Prompt Templates

The prompt template examples in Fig. 14 specify the task, observation format, action semantics, and execution-depth condition.

System Prompt:
You are an expert in solving Sokoban games interactively.
Given a scrambled Sokoban state, you should decide a short sequence of 1, 2, 4, or 8 best actions, then the user will execute the moves and provide you with the new puzzle state. You should then decide the next sequence of actions based on the new state, and repeat this process until the puzzle is solved.
You may dynamically choose a different depth each time for each sequence, but each sequence you proposed must comply with the depth requirements.

[Action Format]
Respond with a JSON object with two fields, "h" and "actions". "h" is an integer number of actions performed in a sequence without seeing the next state; "actions" is a list of move strings, each chosen from ["U", "D", "L", "R"].
U means to move the player up, D means to move the player down, L means to move the player left, and R means to move the player right. If the player is adjacent to a box in the direction of movement, the box is pushed in the same direction. Choose a depth h adaptively from [1, 2, 4, 8].

[State Format]
The state is represented as a rendered image of the Sokoban grid, where each cell is drawn as a distinct tile.
- Wall: A brick-patterned block.
- Empty: A solid black cell.
- Player: A green alien on a black cell.
- Box: An orange border yellow crate with an orange "X" pattern.
- Goal: A red border black cell with a small red diamond.
- Box on Goal: A red border yellow crate with an red "X" pattern.
- Player on Goal: A green alien on a red border black cell.

The puzzle is solved when all boxes are placed on goal cells.

User Prompt:
<image>
The image represents the current rendered state. Please provide the next sequence of actions.

Figure 14: **Example system prompt for zero-shot vision-language model baseline on Sokoban (image state).** The puzzle name and its state and action format descriptions are task-dependent and are different for different tasks.

F Limitations

Our experiments use Qwen2.5-VL-7B-Instruct (Bai et al., 2025); evaluation with other backbones remains future work. We also focus on discrete-action environments. The principle of learned

execution depth extends conceptually to continuous control, but applying PACE to broader robotic learning settings requires adapting the action decoder, backbone integration, and training recipe. We leave these adaptations and their empirical validation to future work.