DYNAMIC EPISODE ALIGNMENT: SAMPLE-EFFICIENT REINFORCEMENT LEARNING OF VISION-LANGUAGE-ACTION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Online reinforcement learning (RL) can improve vision-language-action (VLA) models beyond imitation learning, but many methods rely on a final success-or-failure reward that does not reveal which actions helped. RL can therefore reinforce hesitations in successes and discourage productive actions in failures, requiring many task attempts to resolve this conflicting credit. We propose **D**ynamic **E**pisode **Al**ignment (DEAL), which uses the policy's own successes to identify productive and wasted steps without an extra scoring model. DEAL uses dynamic programming (DP) to align each new episode with the policy's shortest success in the same scene, matching steps by the policy's own representations. RL then updates only productive steps in successes and wasted steps in failures, so successes no longer reinforce hesitations and failures no longer discourage productive actions. In a tabular model under simplifying assumptions, we prove that selecting steps reduces the episodes needed to reliably reinforce productive actions from linear to constant in the number of subgoals. To support DEAL's updates, we introduce a recurrent Gaussian action head that provides exact action likelihoods. Combining DEAL with PPO and the recurrent Gaussian action head yields **97.6%** success on LIBERO with one demonstration per task and **80.4%** on RoboTwin 2.0, versus SimpleVLA-RL's 96.9% and 66.4%. From the same starting policies on three RoboTwin 2.0 tasks, DEAL improves PPO's sample efficiency and raises mean success by a relative **26%** after equal training. On a humanoid with two dexterous hands, online RL with DEAL exceeds EgoVLA's success by a relative **46.8%** on unseen backgrounds.

1 INTRODUCTION

Vision-language-action (VLA) models map camera images, language instructions, and usually the robot's proprioceptive state (joint or end-effector configuration and gripper state) to actions (Zitkovich et al., 2023; Kim et al., 2024; Black et al., 2024). They typically learn by imitating robot demonstrations (Kim et al., 2024), but a missed grasp can leave the robot in a situation the demonstrations do not cover, where recovery is difficult (Ross et al., 2011). Online reinforcement learning (RL) improves the policy through practice guided by rewards (Guo et al., 2025; Li et al., 2025). Because dense rewards for intermediate progress need task-specific design, many VLA RL methods use only a final binary reward indicating episode success or failure (Li et al., 2025; Tan et al., 2025).

The final reward does not reveal which of an episode's hundreds of actions helped or hurt: this is the credit assignment problem (Sutton & Barto, 2018). VLA methods learning from this reward alone commonly use two approaches. Critic-free methods such as GRPO (Shao et al., 2024) compare episode outcomes and give every action in an episode the same score (Li et al., 2025). Critic-based methods (Guo et al., 2025), such as PPO with GAE (Schulman et al., 2017; 2016), learn a value function that predicts future reward to estimate each action's credit. Neither has direct step-level feedback: success can reinforce hesitations and retries, while failure can discourage productive actions. Averaging out this conflicting credit takes many episodes, each a complete task attempt, so online RL is sample inefficient. For example, SimpleVLA-RL (Li et al., 2025) uses about 114,000 episodes in its RL updates to reach its 91.7% success on LIBERO-Long (Liu et al., 2023) (their

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

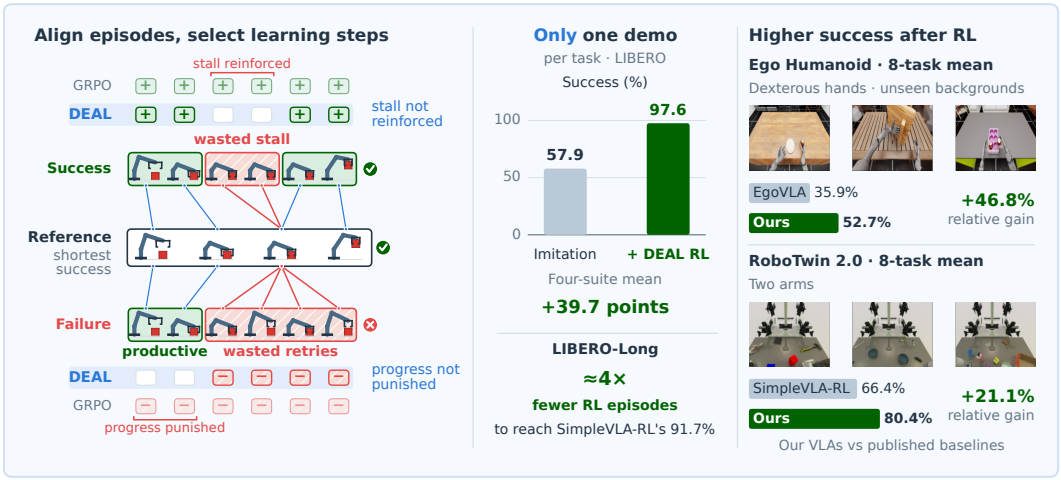


Figure 1: **DEAL avoids reinforcing wasted steps and penalizing productive ones. Left:** DEAL aligns each new episode with the scene's shortest success. Steps sharing one reference step are wasted, except the step that moves on or completes the task; the rest are productive. Unlike GRPO, which gives every step the episode's outcome credit, DEAL updates only productive steps of successes and wasted steps of failures, reducing conflicting credit so the policy learns from far fewer episodes. **Middle:** with one demonstration per task, our VLA reaches 97.6% on LIBERO, and on LIBERO-Long it reaches SimpleVLA-RL's 91.7% success with an estimated 4× fewer RL episodes. **Right:** our VLAs' relative success gains over published EgoVLA and SimpleVLA-RL results, eight tasks each.

Figure 3 and our Appendix C). Additional feedback can resolve conflicting credit but typically requires an extra progress-scoring model (Fei et al., 2025), task-specific stages (Zhang et al., 2024), or human corrections (Chen et al., 2025c). *Can identifying wasted and productive steps instead make online RL for VLAs substantially more sample-efficient?*

To this end, we propose **DEAL** (**D**ynamic **E**pisode **AL**ignment), which identifies productive and wasted steps to improve the sample efficiency of online RL (Figure 1). Hesitations and failed retries lengthen an episode without advancing the task. To locate these wasted steps, DEAL aligns each new episode with the policy's shortest success so far in the same scene (task and initial object arrangement), which serves as the reference. Because episodes reach the same point at different times, dynamic programming matches similar steps in order rather than at equal times. The comparison for this alignment uses the policy's own step embeddings from its vision-language model (VLM) and proprioceptive encoder, so DEAL needs no extra scoring model. Steps that share one reference step are wasted except the one that moves on or completes the task; the rest are productive. RL then updates only productive steps in successes and wasted steps in failures, reducing conflicting credit. Our tabular analysis explains why this reduces the episodes needed for learning (Proposition 1).

Policy-gradient RL needs action log-probabilities to reinforce actions with positive credit and discourage those with negative credit. Flow-matching heads generate an action chunk—a sequence of future actions—without directly providing each action's log-probability (Chen et al., 2025a; Zhang et al., 2025). Token-based heads provide log-probabilities but discretize action coordinates into bins, losing distinctions within a bin (Li et al., 2025; Lu et al., 2025). We introduce a recurrent Gaussian action head that provides exact log-probabilities for continuous actions. It generates one action at a time with fresh proprioception, while a gated recurrent unit (GRU) initialized from VLM embeddings keeps successive actions consistent. Reusing those embeddings over several steps avoids running the costly VLM at every step (Figure 3).

We train VLAs with PPO and DEAL, using the recurrent Gaussian action head of Section 3.4. On LIBERO (Liu et al., 2023), with one demonstration per task, our VLA reaches 97.6% success versus SimpleVLA-RL's 96.9% (Li et al., 2025), and on eight RoboTwin 2.0 tasks (Chen et al., 2025b) it reaches 80.4% versus SimpleVLA-RL's 66.4%. On the Ego Humanoid Manipulation Benchmark (Yang et al., 2025), online RL with DEAL exceeds EgoVLA's success by a relative

36.1% on seen and 46.8% on unseen backgrounds. To isolate DEAL's contribution to RL sample efficiency, we compare GRPO and PPO with and without DEAL from the same starting policy on three RoboTwin tasks: DEAL reaches each baseline's peak with 1.7× and 2.5× fewer episodes, respectively (Figure 4(b, c)). In the tabular comparison, GRPO with DEAL needs up to 8.3× fewer episodes than GRPO to reach 90% success.

Our contributions: **(i) DEAL**, which aligns episodes from the policy's own experience to identify productive and wasted steps, improving the sample efficiency of online RL for VLAs. **(ii) A recurrent Gaussian action head** that keeps successive actions consistent, provides exact log-probabilities for RL, and runs about 1.4× faster than $\pi_{0.5}$'s flow-matching head on the same VLM (Physical Intelligence et al., 2025b). **(iii) A theoretical explanation of DEAL's sample-efficiency gains.** In a tabular model under simplifying assumptions, DEAL reliably reinforces productive actions with a constant number of episodes, whereas a policy gradient with mean-subtracted rewards needs a number that grows linearly with the number of subgoals.

2 Related Work

RL fine-tuning of VLA models. VLA post-training includes policy-gradient updates and advantage-conditioned learning from episode outcomes. SimpleVLA-RL (Li et al., 2025) and RIPT-VLA (Tan et al., 2025) compare binary outcomes of episodes of the same task without a critic, while iRe-VLA (Guo et al., 2025) alternates critic-based RL with supervised learning on successes, and RL4VLA (Liu et al., 2025a) fine-tunes VLAs with PPO and a learned critic. RECAP (Physical Intelligence et al., 2025a) instead trains the policy with supervised learning conditioned on advantages from a separate VLM value model that predicts time to completion.

Step-level credit and selection. An episode's outcome does not distinguish the actions that helped from those that wasted steps. VLA-RL (Lu et al., 2025) trains an additional process reward model. Temporal GRPO (Zhou et al., 2026) compares stage-completion outcomes among episodes entering the same stage, so later failure does not erase credit for completed stages. Detecting stages requires privileged simulator state. SRPO (Fei et al., 2025) scores whole failed episodes against the policy's successes with an additional pretrained video model. Other methods select which steps enter the update: PCM (Bagaria et al., 2026) saves gradient computation by sampling more chunks from gripper-defined phases where successful and failed behavior differs. Self-imitation learning (Oh et al., 2018) reinforces past actions whose returns exceed the value estimate, and GiGPO (Feng et al., 2025) compares actions taken from identical states across episodes. DEAL instead uses alignment to retain productive actions from successes and wasted actions from failures, without a separate scoring model, stage labels or identical states.

Alignment for credit assignment. Alignment with successful behavior can provide dense rewards. Align-RUDDER (Patil et al., 2022) extracts events shared by a few demonstrations and redistributes reward according to how much each step improves alignment with those events. TemporalOT (Fu et al., 2024) rewards matching expert videos. T^2VLA (Chen et al., 2026) adapts at test time, without a success signal, using dynamic time warping (DTW) similarity to high-confidence rollouts as reward; DEAL instead references the scene's shortest success, confirmed by the outcome. ORCA (Huey et al., 2025) shows that DTW imitation rewards can favor repeated matches to early subgoals without covering later ones. DEAL uses repeated matches to identify wasted steps and select actions for learning, keeping the outcome as the only reward.

Action representations for RL. SimpleVLA-RL (Li et al., 2025) replaces OpenVLA-OFT's continuous head with discrete action tokens, which supply action probabilities for RL but lose precision within each bin. OpenVLA-OFT's imitation-learning ablation found continuous actions improved LIBERO success by 5 percentage points over discrete tokens (Kim et al., 2025). Flow-matching policies keep actions continuous but lack direct action likelihoods. ReinFlow and π_{RL}'s Flow-Noise inject noise and use denoising-chain likelihoods for policy optimization (Zhang et al., 2025; Chen et al., 2025a); π_{RL} reports increasing training cost with more denoising steps. Real-Time EXPO-FT (Dong et al., 2026) addresses slow inference by editing VLA-generated chunks using fresh observations. Our recurrent Gaussian head instead generates each action directly from cached VLM features and fresh proprioception, providing exact likelihoods without denoising.

3 METHOD

3.1 SETUP AND OVERVIEW

We use online RL to fine-tune a policy for language-conditioned manipulation, alternating between collecting episodes and updating the policy. At step t, the observation $o_t = (I_t, s_t, \ell)$ contains camera images I_t, proprioceptive state s_t (joint or end-effector configuration and gripper state), and task instruction ℓ. Given the observation history $H_t = (o_1, \ldots, o_t)$, the policy π_ϕ, with parameters ϕ, samples a continuous action $a_t \sim \pi_\phi(\cdot \mid H_t)$. The resulting T-step episode $\tau = ((o_t, a_t))_{t=1}^{T}$ ends in success ($y = 1$) when the robot completes the task, or failure ($y = 0$) when the step limit is reached without completing it. This binary outcome determines the only reward, delivered at the end of the episode: $R(y)$ is positive for success and zero or negative for failure.

This binary outcome does not distinguish a direct grasp from one reached after repeated failed attempts, which waste steps without completing the grasp. DEAL identifies these wasted steps, and the productive steps that advance the task, by aligning the episode with a successful reference from the same scene—the same task and initial object arrangement—as Section 3.2 explains. The RL updates in Section 3.3 then reinforce productive steps in successes and discourage wasted steps in failures, excluding the steps that would receive conflicting credit. These updates need action likelihoods, which we obtain by adapting a vision-language model (VLM) with the recurrent Gaussian action head of Section 3.4.

3.2 DYNAMIC EPISODE ALIGNMENT

To identify wasted steps, we compare each episode with a reference: the scene's shortest success so far, since fewer steps leave less room for hesitation and retries. A successful demonstration from the same scene can initialize the reference; otherwise, all steps enter RL updates until the policy first succeeds. After each collection round, a new success replaces the reference only if it is shorter. A pause can leave an episode approaching an object while the reference already grasps it. We therefore match similar steps in order, rather than at equal times, as panel 2 of Figure 2 illustrates.

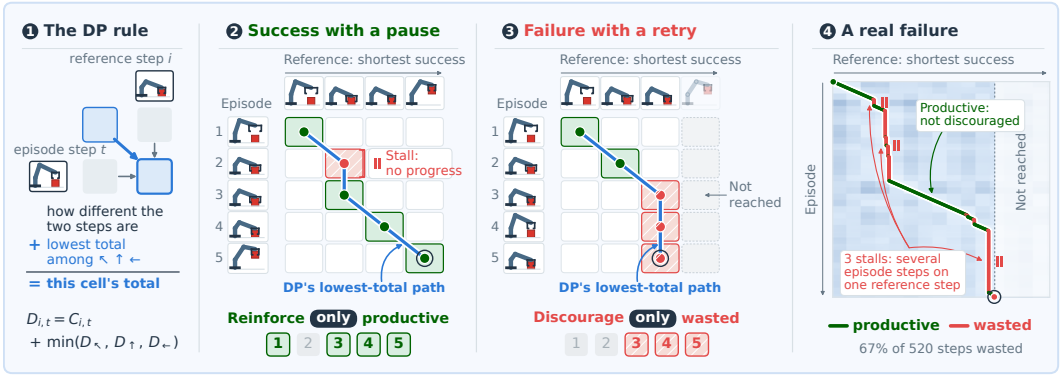


Figure 2: **Dynamic episode alignment.** Successful and failed episodes are aligned to the scene's shortest success (Eq. 1). Several steps sharing one reference step are wasted, except the one that moves on; the rest are productive. Panel 4 applies the same rule to a real LIBERO-Long failure, where stalls appear as vertical runs of the path; Appendix B shows two successes and two failures.

To compare episode and reference steps, we reuse two representations from the policy, so alignment needs no separate scoring model. The action head embeds the proprioceptive state s_t as e_t at each step. The VLM encodes images, the instruction and, where supported, proprioception; we average its output tokens to obtain v_t. The policy calls the VLM once every n steps and reuses v_t between calls; Section 3.4 explains this reuse. We concatenate the two representations into the step feature $z_t = [e_t; v_t]$. If a demonstration reference lacks VLM features, both episodes use only the proprioceptive embeddings for the comparison. For a reference of T_{ref} steps, z_i^{ref} denotes its feature at step i. Matching reference step i with episode step t costs their Euclidean distance $C_{i,t} = \|z_i^{\text{ref}} - z_t\|_2$.

Dynamic time warping (Sakoe & Chiba, 1978) uses the pairwise costs $C_{i,t}$ to find an alignment path: an ordered sequence of matched episode and reference steps with minimum total cost. Let $D_{i,t}$ be the minimum cumulative cost from the first step pair $(1, 1)$ to reference step i and episode step t:

$$D_{i,t} = C_{i,t} + \min\{D_{i-1,t-1},\, D_{i-1,t},\, D_{i,t-1}\}. \tag{1}$$

Panel 1 of Figure 2 shows the three predecessor choices in Eq. 1. A diagonal move matches the next step of each sequence. A move from the upper cell matches another episode step to the same reference step, as during a pause. A move from the left cell matches another reference step to the same episode step, as during faster execution. The path never moves backward in either sequence. A success completes the task, so the path ends at (T_{ref}, T), including the last step of both episodes. A failure does not complete the task, so its last step need not match the reference's final step. We choose the failure's endpoint $(i^\star, T)$, where $i^\star = \arg\min_i D_{i,T}$ minimizes the total cost of matching all T steps. Tracing the minimizing predecessors back from the endpoint recovers the paths in panels 2–4 of Figure 2. Alignment costs $O(T_{\mathrm{ref}}T)$ operations.

Let M_i contain the episode-step indices matched to reference step i by the alignment path. For $|M_i| > 1$, the episode spends several steps where the reference spends one. We label these steps wasted, except the one whose action moves the episode past reference step i, and collect all wasted steps in $\mathcal{W}$; the remaining steps are productive, $\mathcal{P} = \{1, \ldots, T\} \setminus \mathcal{W}$. These labels describe progress relative to the reference. A wasted group need not prevent later progress: in panel 2 of Figure 2, the wasted step 2 is followed by productive steps 3–5, and the episode succeeds. Real episodes contain such stalls. The LIBERO-Long failure in panel 4 makes progress between stalls and stops at reference step 171 of 229. DEAL discourages only its stalls, not the progress between them. Because the endpoint $i^\star$ reflects how far a failure progressed, its number of wasted steps is not set by its length (Appendix B).

Repeated reference matches also identify wasted motion. A robot may reach toward an object, withdraw, and repeat the approach. Each withdrawal undoes the reach, so the retries add steps without completing the grasp. The ordered path cannot match a repeated approach back to reference steps it has already passed. When retries share one reference step until failure, all these steps enter $\mathcal{W}$, as steps 3–5 do in panel 3 of Figure 2. One limitation is the assumption that an episode and its reference follow the same subgoal order; Appendix D discusses order-invariant alignment as future work. Appendix A gives details of reference initialization and alignment.

3.3 Learning from selected steps

The labels determine which actions enter RL updates. Critic-free GRPO (Shao et al., 2024) compares the outcomes of episodes of the same task and gives every action in an episode one score, its advantage. Positive advantages reinforce actions and negative ones discourage them, so one score for the whole episode can reinforce wasted steps in successes and discourage productive steps in failures. DEAL instead restricts learning to the productive steps $\mathcal{P}$ of successes and the wasted steps $\mathcal{W}$ of failures. The reference determines this selection, but supplies no imitation targets: the policy learns from the collected episode's own actions and outcome.

We analyze a tabular Markov decision process (MDP) with N subgoals to show why DEAL's step selection saves episodes. The policy must complete the subgoals in order within a step limit, and each action either completes the current subgoal or wastes a step.

Proposition 1 (Informal)**.** *Under the assumptions in Appendix E, alignment matches each episode step to its corresponding reference step. DEAL's selected updates then increase or preserve the probability of choosing actions that complete subgoals, so task success probability never decreases. Without DEAL's selection, a policy gradient using the final binary reward minus the mean reward can penalize productive actions when an episode fails. An episode's success depends mostly on its other subgoals, so reliably reinforcing a productive action requires averaging over a number of episodes proportional to N. DEAL needs only a constant number of episodes, independent of N.*

The analysis considers critic-free updates, but DEAL's step selection also applies to PPO (Schulman et al., 2017) with a learned critic. For VLA fine-tuning, we estimate advantages only on the selected steps, kept in their original order with episode indices $t_1 < \cdots < t_K$, where K is the number selected. The terminal reward $R(y)$ from Section 3.1 goes on the last selected step: $r_K = R(y)$ and $r_j = 0$ for $j < K$. The critic V_ψ, with parameters ψ, predicts the remaining discounted reward from

the observation history, and its prediction at selected step j is $V_j = V_\psi(H_{t_j})$. With reward discount γ, generalized advantage estimation (GAE) (Schulman et al., 2016) computes temporal-difference errors δ_j and advantages A_j backward along the selected sequence:

$$\delta_j = r_j + \gamma V_{j+1} - V_j, \qquad A_j = \delta_j + \gamma\lambda A_{j+1}, \tag{2}$$

where $V_{K+1} = A_{K+1} = 0$ at termination and λ controls the weight of later errors. Equation 2 discounts once between successive selected steps. In a success, excluding wasted steps avoids reducing a productive action's share of the terminal reward just because wasted steps followed it.

PPO uses the advantages A_j from Eq. 2 to maximize this objective over the selected actions:

$$J(\phi) = \mathbb{E}_j[\min(w_j A_j, \text{clip}(w_j, 1-\epsilon_{\text{low}}, 1+\epsilon_{\text{high}})A_j)], \quad w_j = \frac{\pi_\phi(a_{t_j} \mid H_{t_j})}{\pi_{\text{old}}(a_{t_j} \mid H_{t_j})}. \tag{3}$$

The ratio w_j compares the selected action's likelihood under the current policy and the collecting policy π_{old}, both conditioned on its recorded observation history H_{t_j}. The expectation averages selected steps across the batch. Clipping to $[1-\epsilon_{\text{low}}, 1+\epsilon_{\text{high}}]$, with thresholds ϵ_{low} and ϵ_{high}, limits the incentive for large likelihood changes. The critic fits the same selected steps to targets $A_j + V_j$, computed before the update and held fixed. Section 4.3 also shows sample-efficiency gains when critic-free GRPO receives its credit from DEAL's alignment. Appendix A gives implementation details.

3.4 Recurrent Gaussian action head

The likelihood ratio in Eq. 3 requires the likelihood of each continuous action. Flow-matching heads do not provide it directly (Chen et al., 2025a; Zhang et al., 2025). Token-based heads provide probabilities for discretized action bins (Li et al., 2025; Lu et al., 2025), preventing RL from favoring finer adjustments within a bin. We instead use a Gaussian action head that predicts a mean and standard deviation for each coordinate, providing exact likelihoods without discretizing the actions.

Running the multi-billion-parameter VLM at every step is costly, so we reuse its output for n steps, the same idea as action chunking (Bharadhwaj et al., 2024). Between VLM calls, the head must keep actions consistent while incorporating fresh proprioception. An action chunk, however, fixes its actions when it predicts them, so they cannot use later feedback. Our head instead predicts each action from recurrent memory and current proprioception. The left panel of Figure 3 shows how the memory connects successive predictions.

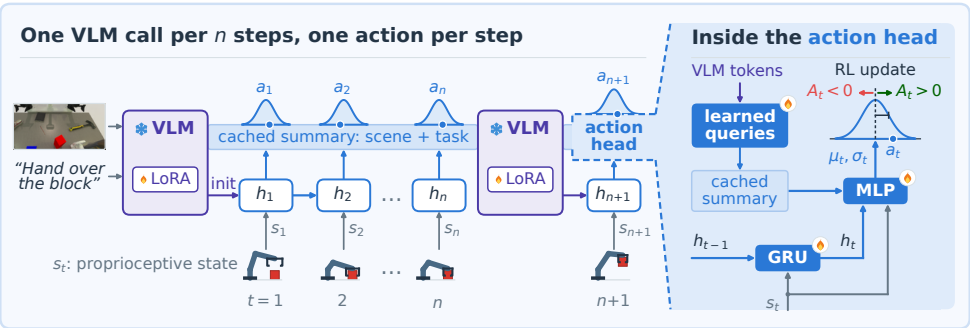


Figure 3: **Recurrent Gaussian action head. Left:** one VLM call supports n actions, each computed with fresh proprioceptive feedback. **Right:** the MLP combines the cached summary, proprioception and recurrent memory to predict a Gaussian action distribution. For selected step $t = t_j$, the RL panel's A_t is the advantage A_j in Eq. 2.

The right panel of Figure 3 shows how the head combines cached VLM information with the robot's changing proprioception. At each VLM call, learned queries, trainable vectors inside the head, cross-attend to the VLM's output tokens to form a cached summary, reused until the next call. The head initializes the memory h_t of a gated recurrent unit (GRU) (Cho et al., 2014) from the mean-pooled VLM representation v_t, defined in Section 3.2. At every action step, the GRU updates the memory using the new proprioceptive state s_t. A multilayer perceptron (MLP) combines the cached summary, the proprioceptive embedding e_t from Section 3.2, and a joint embedding of h_t and v_t

Table 1: Ego benchmark: success (%) on eight tasks, grouped by horizon as in EgoVLA; Δ: gain from RL over imitation. Published baselines are from EgoVLA. NoPretrain omits human-video pretraining; 50% uses half the robot demonstrations and is not reported with unseen backgrounds.

	Short-horizon			Long-horizon					
Method	Stack Can	Flip Mug	Open Drawer	Unload Cans	Insert Cans	Stack-Can-Into-Drawer	Insert-And-Unload-Cans	Sort Cans	Mean
Seen backgrounds									
ACT (Zhao et al., 2023)	22.2	7.4	18.5	0.0	0.0	11.1	0.0	0.0	7.4
EgoVLA-NoPretrain	55.6	3.7	59.3	63.0	11.1	0.0	7.4	51.9	31.5
EgoVLA (50% demos)	44.4	22.2	22.2	0.0	7.4	29.6	0.0	0.0	15.7
EgoVLA	77.8	59.3	59.3	66.7	22.2	40.7	44.4	55.6	53.2
Our BC	81.5	50.0	62.0	68.5	15.7	34.3	42.6	50.0	50.6
Our RL (DEAL)	**94.4**	**73.1**	**98.1**	**78.7**	**51.9**	**72.2**	**50.9**	**60.2**	**72.4**
Δ	+12.9	+23.1	+36.1	+10.2	+36.2	+37.9	+8.3	+10.2	+21.8
Unseen backgrounds									
ACT (Zhao et al., 2023)	9.1	0.0	24.2	1.5	0.0	1.5	0.0	0.0	4.5
EgoVLA-NoPretrain	57.6	4.7	46.2	34.9	6.1	0.0	0.0	15.2	20.6
EgoVLA	62.1	30.8	50.0	50.0	15.2	28.8	31.8	18.2	35.9
Our BC	65.2	39.4	43.9	45.5	16.7	22.7	27.3	15.2	34.5
Our RL (DEAL)	**71.2**	**54.5**	**81.8**	**66.7**	**40.9**	**47.0**	**39.4**	**19.7**	**52.7**
Δ	+6.0	+15.1	+37.9	+21.2	+24.2	+24.3	+12.1	+4.5	+18.2

to predict the action mean μ_t and per-coordinate standard deviations σ_t. The resulting Gaussian policy provides the likelihood used in Eq. 3: $\pi_\phi(a_t \mid H_t) = \mathcal{N}(a_t; \mu_t, \mathrm{diag}(\sigma_t^2))$. Separate standard deviations allow different action coordinates to explore on different scales. A separate MLP predicts the critic value $V_\psi(H_t)$ from v_t and the action MLP's last hidden features. Critic training updates only this MLP.

3.5 Training

Behavior cloning (BC) initializes online exploration from demonstrated behavior. We fit the Gaussian policy of Section 3.4 to robot demonstrations by maximum likelihood, minimizing

$$\mathcal{L}_{\mathrm{BC}}(\phi) = -\mathbb{E}_{(H_t, a_t)\sim\mathcal{D}}[\log \pi_\phi(a_t \mid H_t)], \tag{4}$$

where $\mathcal{D}$ contains demonstrated actions and their recorded observation histories. As during execution, one VLM call supports several consecutive action predictions, each using the demonstration's available proprioceptive history. BC fits both means and standard deviations, with the learned standard deviations setting the initial exploration scales for online RL. During online RL, PPO fine-tunes the action head of Figure 3, including the standard deviations, with the objective of Eq. 3, while low-rank adapters (LoRA) (Hu et al., 2022) adapt the VLM. Appendix A gives implementation details.

4 Experiments

We evaluate the task success of our VLAs from single-arm robots to a humanoid with two dexterous hands. We then isolate DEAL's effect on RL sample efficiency through a comparison from the same starting policy on RoboTwin 2.0 (Chen et al., 2025b). The tabular analysis of Proposition 1 explains why selecting steps helps. Finally, we measure how fast the recurrent Gaussian action head of Section 3.4 generates actions.

On the Ego Humanoid Manipulation Benchmark (Yang et al., 2025), we adapt EgoVLA's VLM pretrained on human videos with our recurrent Gaussian action head to set 38 joint-position targets for the humanoid's arms and hands. We also adapt the VLM of the base $\pi_{0.5}$ checkpoint (Physical Intelligence et al., 2025b), before any LIBERO or RoboTwin fine-tuning, with the same head for dual-arm RoboTwin 2.0 with 14-dimensional actions and single-arm LIBERO with 7-dimensional actions. We first train each policy to imitate robot demonstrations through behavior cloning (BC), following EgoVLA with 100 successful demonstrations per Ego task and SimpleVLA-RL (Li et al., 2025) with 1,000 per RoboTwin task. On LIBERO, we follow SimpleVLA-RL's two settings: all 50 demonstrations per task, and a separate experiment with only one demonstration per task to test learning when demonstrations are scarce. We then fine-tune the policies online with PPO and DEAL, unless stated otherwise.

Table 2: RoboTwin 2.0: success (%) on eight tasks; Δ: gain from RL over imitation. Published baselines are from SimpleVLA-RL. Appendix C reports RL episode counts.

Method	Pick dual bottles	Place phone stand	Move can to pot	Place A2B left	Place empty cup	Handover mic	Handover block	Stack two bowls	Mean
π_0 (Black et al., 2024)	50.0	22.0	41.0	38.0	60.0	96.0	39.0	53.0	49.9
RDT (Liu et al., 2025b)	18.0	13.0	33.0	21.0	42.0	95.0	26.0	42.0	36.3
OpenVLA-OFT (Kim et al., 2025)	29.7	17.1	28.1	37.5	77.3	45.3	33.1	40.6	38.6
SimpleVLA-RL	68.3	39.6	61.2	45.3	94.2	89.2	57.8	75.8	66.4
Δ	+38.6	+22.5	+33.1	+7.8	+16.9	+43.9	+24.7	+35.2	+27.8
Our BC	42.0	17.0	33.0	42.0	76.0	33.0	41.0	38.0	40.3
Our RL (DEAL)	**81.0**	**58.0**	**70.0**	**78.0**	**98.0**	**99.0**	**80.0**	**79.0**	**80.4**
Δ	+39.0	+41.0	+37.0	+36.0	+22.0	+66.0	+39.0	+41.0	+40.1

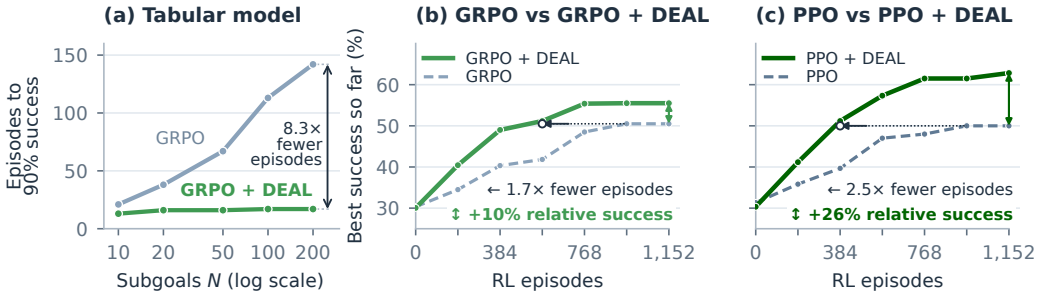


Figure 4: **DEAL reaches the same success with fewer RL episodes.** (a) Tabular GRPO: episodes to 90% success, with separately tuned step sizes, references learned from successes and 30 held-out seeds (Appendix E.6). (b, c) Best success so far, evaluated on 100 held-out scenes per task and averaged over three RoboTwin 2.0 tasks (Handover block, Handover mic and Place phone stand) from the same starting policy per task (Appendix C). Arrows mark how many fewer episodes each DEAL variant needs to reach its baseline's peak, and its relative gain at 1,152 episodes.

At evaluation, we execute the policy's mean action, which makes LIBERO and RoboTwin 2.0 evaluations deterministic. Following SimpleVLA-RL, we evaluate LIBERO's 50 scenes per task, which RL also trains on, and, in RoboTwin 2.0's domain-randomized setting, 100 evaluation seeds per task. Each RoboTwin seed fixes one scene's objects and randomization, and no imitation or RL episode uses these seeds. On Ego, we run EgoVLA's rollouts: 27 on three seen backgrounds, repeated four times (108), and 66 on 22 unseen backgrounds. For the LIBERO and RoboTwin sample-efficiency comparisons, training-episode counts include all rounds of RL and exclude evaluation.

4.1 Online RL with two dexterous hands

Coordinating two arms and two dexterous hands is difficult because the policy must control many joints as the hands make and break contact with objects (Rajeswaran et al., 2018). Ego tasks chain several manipulations using 14 arm and 24 hand joints. Because robot demonstrations of such control are costly, EgoVLA (Yang et al., 2025) pretrains on human videos before imitating 100 robot demonstrations per task. Neither EgoVLA nor its reported baselines use online RL. We test how much online RL with DEAL improves the imitation-learned policy without new demonstrations.

Online RL with DEAL raises mean success from 50.6% to **72.4%** on seen backgrounds and from 34.5% to **52.7%** on unseen backgrounds (Table 1). Neither imitation learning nor RL uses the unseen backgrounds, so the **18.2**-point gain there shows that RL with DEAL generalizes to backgrounds never seen in training. In both settings, our VLA matches or exceeds every success rate reported by EgoVLA on all eight evaluated tasks, including all five long-horizon tasks. EgoVLA identifies four of the eight as requiring fine-grained manipulation: Stack-Can, Flip-Mug, Sort-Cans and Insert-And-Unload-Cans (Yang et al., 2025). Without new demonstrations, the roughly **22**-point seen-background gain is comparable to EgoVLA's gain from human-video pretraining (31.5% to 53.2%).

4.2 HIGHER SUCCESS ON LIBERO AND ROBOTWIN 2.0

To test whether the gains extend to another VLM and other robots, we adapt $\pi_{0.5}$'s VLM with the recurrent Gaussian action head of Section 3.4 on LIBERO and RoboTwin 2.0. Scarce demonstrations leave much room for RL: with one per LIBERO task, our VLA raises success from 57.9% after imitation to **97.6%** (Table 4). This matches SimpleVLA-RL's 96.9% (Li et al., 2025) in the same setting within noise. On LIBERO-Long, our VLA reaches SimpleVLA-RL's 91.7% success with an estimated **4**× fewer RL episodes (Appendix C). We omit SRPO (Fei et al., 2025), which adds rewards from a pretrained video model; DEAL uses only the binary outcome. With all 50 demonstrations per task, our VLA raises success from 96.2% to **99.6%**, the highest mean in Table 3.

Table 3: LIBERO with all 50 demonstrations per task: success (%) on the four suites. Published results are from SimpleVLA-RL.

Method	Spatial	Object	Goal	Long	Mean
Octo (Ghosh et al., 2024)	78.9	85.7	84.6	51.1	75.1
OpenVLA (Kim et al., 2024)	84.7	88.4	79.2	53.7	76.5
Nora (Hung et al., 2025)	92.2	95.4	89.4	74.6	87.9
π_0+FAST (Pertsch et al., 2025)	96.4	96.8	88.6	60.2	85.5
π_0 (Black et al., 2024)	96.8	98.8	95.8	85.2	94.2
UniVLA (Bu et al., 2025)	96.5	96.8	95.6	92.0	95.2
OpenVLA-OFT (Kim et al., 2025)	91.6	95.3	90.6	86.5	91.0
SimpleVLA-RL	**99.4**	99.1	99.2	98.5	99.1
Our BC	98.0	97.8	95.8	93.2	96.2
Our RL (DEAL)	**99.4**	**100.0**	**99.6**	**99.4**	**99.6**

Table 4: LIBERO with one demonstration per task: success (%) on the four suites. Published results are from SimpleVLA-RL.

Method	Spatial	Object	Goal	Long	Mean
OpenVLA-OFT	63.6	54.9	59.6	17.3	48.9
SimpleVLA-RL	98.2	98.7	98.8	91.7	96.9
Our BC	62.2	81.4	55.6	32.4	57.9
Our RL (DEAL)	**98.8**	**100.0**	**99.0**	**92.6**	**97.6**

We extend the evaluation to coordinated dual-arm tasks on RoboTwin 2.0. Our VLA raises mean success from 40.3% after imitation to **80.4%** after RL, a **40.1**-point gain versus 27.8 for SimpleVLA-RL from a similar start (38.6%), and leads Table 2 on all eight tasks. Across the five with released checkpoints, it matches their measured success with an estimated **5.6**× fewer RL episodes in total (Appendix C).

4.3 WHY DEAL NEEDS FEWER RL EPISODES

Proposition 1 explains DEAL's sample-efficiency advantage in a tabular model: step selection prevents failed episodes from discouraging productive actions. We test whether this improves learning with tabular GRPO (Figure 4(a)). As tasks grow from 10 to 200 subgoals, GRPO needs 21 to 142 episodes to reach 90% success, whereas GRPO with DEAL needs only **13 to 17**: up to **8.3**× fewer.

To isolate DEAL's contribution to VLA sample efficiency, we fine-tune our VLA with PPO, with and without DEAL, from the same imitation-learned policy on each of three RoboTwin 2.0 tasks. We repeat the comparison with GRPO; within each algorithm, the only change is DEAL's credit assignment. Averaged over the three tasks, PPO with DEAL reaches PPO's peak success with **2.5**× fewer episodes, and after 1,152 episodes its success is a relative **26%** higher (Figure 4(c)). With GRPO, DEAL needs **1.7**× fewer episodes and ends a relative 10% higher (Figure 4(b)).

Faster action generation with exact likelihoods. With the same $\pi_{0.5}$ VLM on one L40S GPU, generating 20 actions takes **191 ms** with the recurrent Gaussian action head of Section 3.4 and 264 ms with $\pi_{0.5}$'s flow-matching head at batch size one, including the VLM call. Action generation is **1.37–1.41**× faster across batch sizes 1–16.

5 CONCLUSION

DEAL improves RL sample efficiency by aligning each episode with the policy's shortest success in the same scene. The alignment lets RL update only productive steps in successes and wasted steps in failures. Tabular analysis explains why selecting steps reduces conflicting credit, and VLA experiments show that DEAL improves sample efficiency for both GRPO and PPO. Across three benchmarks, our VLAs achieve higher success than EgoVLA on humanoid manipulation with two dexterous hands and than SimpleVLA-RL on RoboTwin 2.0, and match it on LIBERO. Together, these results show how the policy's own successes can guide credit assignment and reduce the interaction cost of learning from sparse rewards.

Ethics Statement

This work studies reinforcement learning for robot manipulation policies. All experiments run in simulation on public benchmarks (LIBERO, RoboTwin 2.0 and the Ego Humanoid Manipulation Benchmark), and we collect no data from human participants. On the Ego benchmark we adapt EgoVLA's publicly released vision-language model and use no additional human data. Online RL on physical robots relies on many trial-and-error attempts that can damage hardware or surroundings. By reducing the number of RL episodes, DEAL can lower this cost, but deploying RL-trained policies on real robots still requires safeguards such as human supervision and workspace limits.

Reproducibility Statement

Section 3 describes DEAL. Appendix A gives the implementation settings, including the action-head architecture, imitation settings and PPO hyperparameters. Appendix C explains how we count RL episodes and derives the LIBERO-Long and RoboTwin 2.0 estimates and the controlled sample-efficiency ratios. Appendix E states the assumptions of the tabular analysis and proves each result, and Appendix E.6 describes the numerical check. All benchmarks are public, and Section 4 states the demonstrations and evaluation protocol we follow on each. We will release our code and trained checkpoints upon publication.

AI Use Statement

We used large language models (LLMs) to improve the clarity and readability of the manuscript and to assist with code development. The authors reviewed the AI-assisted text and code and take responsibility for the final manuscript, implementation, and reported results.

References

Vaidehi Bagaria, Nikshep Grampurohit, and Pulkit Verma. Learn where outcomes diverge: Efficient VLA RL via probabilistic chunk masking. *CoRR*, 2026. doi: 10.48550/ARXIV.2605.16154.

Homanga Bharadhwaj, Jay Vakil, Mohit Sharma, Abhinav Gupta, Shubham Tulsiani, and Vikash Kumar. RoboAgent: Generalization and efficiency in robot manipulation via semantic augmentations and action chunking. In *ICRA*, pp. 4788–4795. IEEE, 2024.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. $\pi 0$: A vision-language-action flow model for general robot control. *CoRR*, 2024. doi: 10.48550/ARXIV.2410.24164.

Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. UniVLA: Learning to act anywhere with task-centric latent actions. *CoRR*, 2025. doi: 10.48550/ARXIV.2505.06111.

Remi Cadene, Simon Alibert, Alexander Soare, Quentin Gallouedec, Adil Zouitine, Steven Palma, Pepijn Kooijmans, Michel Aractingi, Mustafa Shukor, Dana Aubakirova, Martino Russi, Francesco Capuano, Caroline Pascal, Jade Choghari, Jess Moss, and Thomas Wolf. LeRobot: State-of-the-art machine learning for real-world robotics in PyTorch. `https://github.com/huggingface/lerobot`, 2024.

Kang Chen, Zhihao Liu, Tonghe Zhang, Zhen Guo, Si Xu, Hao Lin, Hongzhi Zang, Quanlu Zhang, Zhaofei Yu, Guoliang Fan, Tiejun Huang, Yu Wang, and Chao Yu. πRL: Online RL fine-tuning for flow-based vision-language-action models. *CoRR*, 2025a. doi: 10.48550/ARXIV.2510.25889.

Siyao Chen, Jiakang Yuan, Jiaxin Wang, and Tao Chen. Trust your instincts: Confidence-driven test-time RL for vision-language-action models. *CoRR*, 2026. doi: 10.48550/ARXIV.2606.29892.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Qiwei Liang, Zixuan Li, Xianliang Lin, Yiheng Ge, Zhenyu Gu, Weiliang Deng, Yubin Guo, Tian Nian, Xuanbing Xie, Qiangyu

Chen, Kailun Su, Tianling Xu, Guodong Liu, Mengkang Hu, Huan ang Gao, Kaixuan Wang, Zhixuan Liang, Yusen Qin, Xiaokang Yang, Ping Luo, and Yao Mu. RoboTwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *CoRR*, 2025b. doi: 10.48550/ARXIV.2506.18088.

Yuhui Chen, Shuai Tian, Shugao Liu, Yingting Zhou, Haoran Li, and Dongbin Zhao. ConRFT: A reinforced fine-tuning method for VLA models via consistency policy. *CoRR*, 2025c. doi: 10.48550/ARXIV.2502.05450.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *EMNLP*, pp. 1724–1734. ACL, 2014.

Perry Dong, Kuo-Han Hung, Dorsa Sadigh, and Chelsea Finn. Reinforcement learning for real-time vision-language-action policies, 2026. URL https://arxiv.org/abs/2609.18207.

Senyu Fei, Siyin Wang, Li Ji, Ao Li, Shiduo Zhang, Liming Liu, Jinlong Hou, Jingjing Gong, Xianzhong Zhao, and Xipeng Qiu. SRPO: Self-referential policy optimization for vision-language-action models. *CoRR*, 2025. doi: 10.48550/ARXIV.2511.15605.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for LLM agent training. *CoRR*, 2025. doi: 10.48550/ARXIV.2505.10978.

Yuwei Fu, Haichao Zhang, Di Wu, Wei Xu, and Benoit Boulet. Robot policy learning with temporal optimal transport reward. In *NeurIPS*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/dcd297696d0bb304ba426b3c5a679c37-Abstract-Conference.html.

Dibya Ghosh, Homer Rich Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Quan Vuong, Ted Xiao, Pannag R. Sanketi, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *Robotics: Science and Systems*, 2024. doi: 10.15607/RSS.2024.XX.090.

Yanjiang Guo, Jianke Zhang, Xiaoyu Chen, Xiang Ji, Yen-Jen Wang, Yucheng Hu, and Jianyu Chen. Improving vision-language-action model with online reinforcement learning. In *ICRA*, pp. 15665–15672, 2025. doi: 10.1109/ICRA55743.2025.11127299.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *ICLR*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

William Huey, Huaxiaoyue Wang, Anne Wu, Yoav Artzi, and Sanjiban Choudhury. Imitation learning from a single temporally misaligned video. In *ICML*, 2025. URL https://proceedings.mlr.press/v267/huey25a.html.

Chia-Yu Hung, Qi Sun, Pengfei Hong, Amir Zadeh, Chuan Li, U-Xuan Tan, Navonil Majumder, and Soujanya Poria. NORA: A small open-sourced generalist vision language action model for embodied tasks. *CoRR*, 2025. doi: 10.48550/ARXIV.2504.19854.

Zeyu Jia, Alexander Rakhlin, and Tengyang Xie. Do we need to verify step by step? rethinking process supervision from a theoretical perspective. In *ICML*, 2025. URL https://proceedings.mlr.press/v267/jia25f.html.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Paul Foster, Pannag R. Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. In *CoRL*, pp. 2679–2713, 2024. URL https://proceedings.mlr.press/v270/kim25c.html.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *CoRR*, 2025. doi: 10.48550/ARXIV.2502.19645.

Haozhan Li, Yuxin Zuo, Jiale Yu, Yuhao Zhang, Zhaohui Yang, Kaiyan Zhang, Xuekai Zhu, Yuchen Zhang, Tianxing Chen, Ganqu Cui, Dehui Wang, Dingxiang Luo, Yuchen Fan, Youbang Sun, Jia Zeng, Jiangmiao Pang, Shanghang Zhang, Yu Wang, Yao Mu, Bowen Zhou, and Ning Ding. SimpleVLA-RL: Scaling VLA training via reinforcement learning. *CoRR*, 2025. doi: 10.48550/ARXIV.2509.09674.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *NeurIPS*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/8c3c666820ea055a77726d66fc7d447f-Abstract-Datasets_and_Benchmarks.html.

Jijia Liu, Feng Gao, Bingwen Wei, Xinlei Chen, Qingmin Liao, Yi Wu, Chao Yu, and Yu Wang. What can RL bring to VLA generalization? an empirical study. In *NeurIPS*, 2025a. URL http://papers.nips.cc/paper_files/paper/2025/hash/8c0fabe372177d2aded596be2d3b4544-Abstract-Conference.html.

Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. In *ICLR*, 2025b. URL https://openreview.net/forum?id=yAzN4tz7oI.

Guanxing Lu, Wenkai Guo, Chubin Zhang, Yuheng Zhou, Haonan Jiang, Zifeng Gao, Yansong Tang, and Zi wei Wang. VLA-RL: Towards masterful and general robotic manipulation with scalable reinforcement learning. *CoRR*, 2025. doi: 10.48550/ARXIV.2505.18719.

Junhyuk Oh, Yijie Guo, Satinder Singh, and Honglak Lee. Self-imitation learning. In *ICML*, pp. 3878–3887, 2018. URL https://proceedings.mlr.press/v80/oh18b.html.

Vihang Patil, Markus Hofmarcher, Marius-Constantin Dinu, Matthias Dorfer, Patrick M. Blies, Johannes Brandstetter, José Antonio Arjona-Medina, and Sepp Hochreiter. Align-RUDDER: Learning from few demonstrations by reward redistribution. In *ICML*, pp. 17531–17572, 2022. URL https://proceedings.mlr.press/v162/patil22a.html.

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. FAST: Efficient action tokenization for vision-language-action models. *CoRR*, 2025. doi: 10.48550/ARXIV.2501.09747.

Physical Intelligence, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Kevin Black, Ken Conley, Grace Connors, James Darpinian, Karan Dhabalia, Jared DiCarlo, Danny Driess, Michael Equi, Adnan Esmail, Yunhao Fang, Chelsea Finn, Catherine Glossop, Thomas Godden, Ivan Goryachev, Lachy Groom, Hunter Hancock, Karol Hausman, Gashon Hussein, Brian Ichter, Szymon Jakubczak, Rowan Jen, Tim Jones, Ben Katz, Liyiming Ke, Chandra Kuchi, Marinda Lamb, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Yao Lu, Vishnu Mano, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Charvi Sharma, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, Will Stoeckle, Alex Swerdlow, James Tanner, Marcel Torne, Quan Vuong, Anna Walling, Haohuan Wang, Blake Williams, Sukwon Yoo, Lili Yu, Ury Zhilinsky, and Zhiyuan Zhou. π*0.6: a VLA that learns from experience. *CoRR*, 2025a. doi: 10.48550/ARXIV.2511.14759.

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. π0.5: a vision-language-action model with open-world generalization. *CoRR*, 2025b. doi: 10.48550/ARXIV.2504.16054.

Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. In *Robotics: Science and Systems*, 2018. doi: 10.15607/RSS.2018.XIV.049.

Stéphane Ross, Geoffrey J. Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *AISTATS*, pp. 627–635, 2011. URL http://proceedings.mlr.press/v15/ross11a/ross11a.pdf.

H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49, 1978. doi: 10.1109/tassp.1978.1163055.

John Schulman, Philipp Moritz, Sergey Levine, Michael I. Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *ICLR (Poster)*, 2016. URL http://arxiv.org/abs/1506.02438.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, 2017. URL http://arxiv.org/abs/1707.06347.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *CoRR*, 2024. doi: 10.48550/ARXIV.2402.03300.

Richard S. Sutton and Andrew G. Barto. *Reinforcement learning - an introduction, 2nd Edition*. 2018. URL http://www.incompleteideas.net/book/the-book-2nd.html.

Shuhan Tan, Kairan Dou, Yue Zhao, and Philipp Krähenbühl. Interactive post-training for vision-language-action models. *CoRR*, 2025. doi: 10.48550/ARXIV.2505.17016.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, pp. 229–256, 1992. doi: 10.1007/BF00992696.

Ruihan Yang, Qinxi Yu, Yecheng Wu, Rui Yan, Borui Li, An-Chieh Cheng, Xueyan Zou, Yunhao Fang, Xuxin Cheng, Ri-Zhao Qiu, Hongxu Yin, Sifei Liu, Song Han, Yao Lu, and Xiaolong Wang. EgoVLA: Learning vision-language-action models from egocentric human videos. *CoRR*, 2025. doi: 10.48550/ARXIV.2507.12440.

Tonghe Zhang, Chao Yu, Sichang Su, and Yu Wang. ReinFlow: Fine-tuning flow matching policy with online reinforcement learning. In *NeurIPS*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/98d6f928497d9eaad9f320fb2db3040d-Abstract-Conference.html.

Zijian Zhang, Kaiyuan Zheng, Zhaorun Chen, Joel Jang, Yi Li, Chaoqi Wang, Mingyu Ding, Dieter Fox, and Huaxiu Yao. GRAPE: Generalizing robot policy via preference alignment. *CoRR*, 2024. doi: 10.48550/ARXIV.2411.19309.

Tingting Zhao, Hirotaka Hachiya, Gang Niu, and Masashi Sugiyama. Analysis and improvement of policy gradient estimation. In *NIPS*, pp. 262–270, 2011. URL https://proceedings.neurips.cc/paper/2011/hash/85d8ce590ad8981ca2c8286f79f59954-Abstract.html.

Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Robotics: Science and Systems*, 2023. doi: 10.15607/RSS.2023.XIX.016.

Yao Zhou, Hang Gao, Fengge Wu, Changwen Zheng, and Wenwen Qiang. Temporal GRPO: Beyond trajectory-level credit in vision-language-action reinforcement learning. *CoRR*, 2026. doi: 10.48550/ARXIV.2608.13026.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong T. Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R. Sanketi, Grecia Salazar, Michael S. Ryoo, Krista Reymann, Kanishka Rao, Karl Pertsch, Igor Mordatch, Henryk Michalewski, Yao Lu, Sergey Levine, Lisa Lee, Tsang-Wei Edward Lee, Isabel Leal, Yuheng Kuang, Dmitry Kalashnikov, Ryan Julian, Nikhil J. Joshi, Alex Irpan, Brian Ichter, Jasmine Hsu, Alexander Herzog, Karol Hausman, Keerthana Gopalakrishnan, Chuyuan Fu, Pete Florence, Chelsea Finn, Kumar Avinava Dubey, Danny Driess, Tianli Ding, Krzysztof Marcin Choromanski, Xi Chen,

Yevgen Chebotar, Justice Carbajal, Noah Brown, Anthony Brohan, Montserrat Gonzalez Arenas, and Kehang Han. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *CoRL*, pp. 2165–2183, 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html.

A Implementation details

Alignment. Each scene, identified by its task and initial object arrangement, keeps its own reference. Unless a successful demonstration from the scene initializes it, the scene's first success does. After each collection round, the scene's shortest new success replaces the reference only if it is shorter. The step features of Section 3.2 concatenate the action head's 64-dimensional proprioceptive embedding e_t with the average v_t of the VLM's output tokens.

Action head and imitation. We describe the configuration used on LIBERO as an example. Four learned queries cross-attend to the VLM's output tokens through three attention layers with eight heads. The GRU memory has 128 dimensions, the action MLP has hidden widths 1024, 1024, 512, 512 and 256, and the critic MLP has hidden widths 1024, 512 and 256. During imitation, one VLM call supports 20 consecutive actions. Imitation uses AdamW with batch size 64, learning rate 3×10^{-5}, weight decay 0.01, 500 warm-up steps and cosine decay to 10^{-6}.

RL updates. Table A1 lists the PPO settings. Each collection round gathers 64 episodes, and each update makes two passes over the selected steps of the round. Successes receive the terminal reward $R(1) = 5$ and failures $R(0) = 0$.

Table A1: PPO settings on all three benchmarks, except where noted.

Setting	Value
Policy learning rate	10^{-5} (LIBERO, Ego); 5×10^{-6} (RoboTwin 2.0)
Critic learning rate	$10\times$ the policy learning rate
Discount γ	0.99
GAE λ	0.95
Clipping range $[1 - \epsilon_{\text{low}},\ 1 + \epsilon_{\text{high}}]$	$[0.8,\ 1.2]$
PPO epochs per round	2
Episodes per round	64
PPO minibatch	2,048 steps
Maximum episode length	LIBERO: 280, 280, 300, 520 (Spatial, Object, Goal, Long), following LeRobot (Cadene et al., 2024) RoboTwin 2.0: 100–400 per task, following SimpleVLA-RL Ego: the benchmark's per-task limits

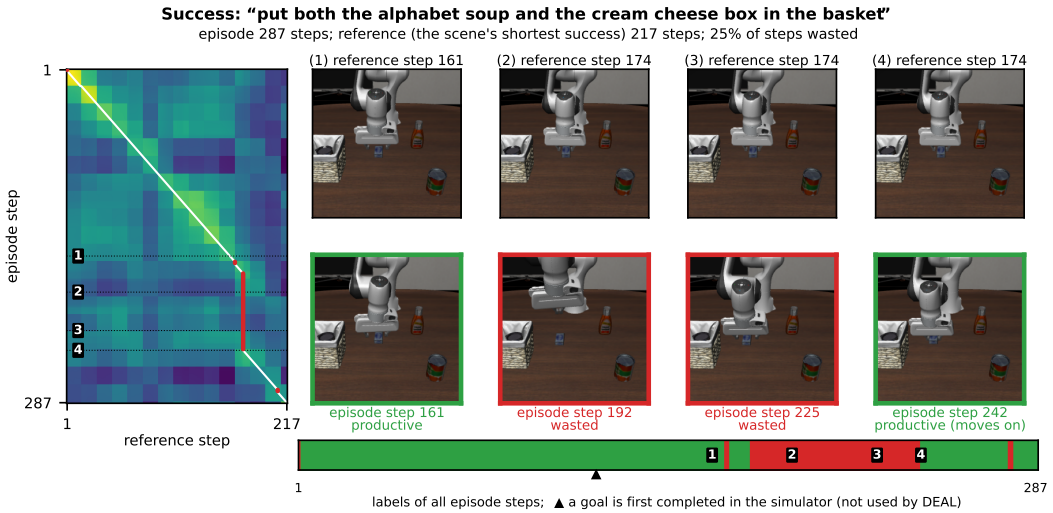
B Alignment examples

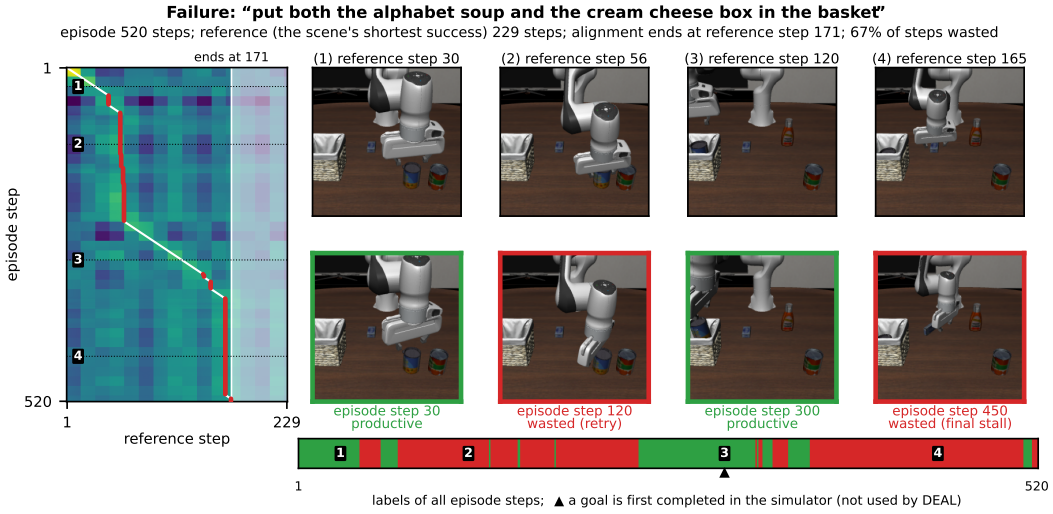
The panels below show DEAL's labels on a successful and a failed episode of each of two LIBERO-Long tasks. We collect eight episodes in each scene from our trained policy, sampling actions as in RL, and the scene's shortest success among them is the reference. The features and alignment are those used in RL (Section 3.2). Left: alignment costs between the reference (columns) and the episode (rows), as in Figure 2; bright is low cost, the white path is the alignment, and red marks wasted steps. A vertical run of the path is a stall: several episode steps matched to one reference step. Right: reference (top) and episode (bottom) frames at the four marked steps; each border shows the episode step's label. Below: the labels of all episode steps; triangles mark where the simulator registers a completed goal, which DEAL does not use.

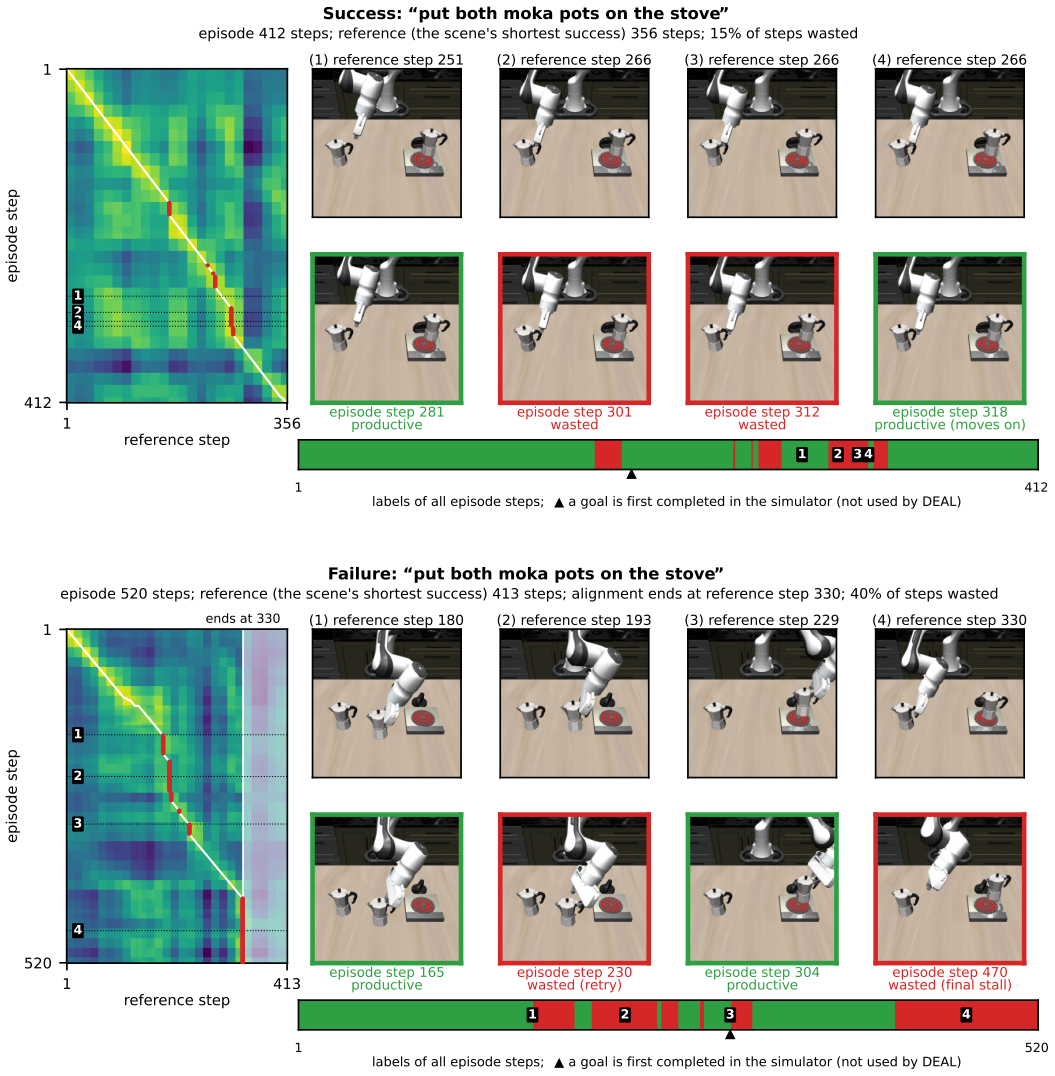
A success's alignment ends at the reference's final step, so at least the episode's extra length is wasted, and the alignment decides where. The basket success spends 287 steps where its reference spends 217, and the moka-pot success 412 where its reference spends 356; their wasted steps, 25% and 15%, fall mostly where the gripper hovers over the cream cheese or the second moka pot before grasping it.

A failure's alignment has no fixed end. It stops at the reference step that best matches the episode's last step, so the alignment decides both how many steps are wasted and which. Both failures run to the 520-step limit, as every LIBERO-Long failure does, yet one ends at reference step 171 of 229 with 67% of its steps wasted, and the other at step 330 of 413 with 40% wasted. Their wasted steps are stalls: repeated failed grasps of the soup can, repeated attempts to lift the first moka pot, and the final attempts on the cream cheese and the second pot. The productive steps between these stalls put the soup in the basket and the first moka pot on the stove. The failure's update leaves them out, so the failure does not discourage them. Under outcome-only credit, a failure's single negative score already discourages all of its steps, retries included; DEAL only removes the productive steps from this update and adds no negative credit of its own.

The supplementary material contains videos of four LIBERO-Long episodes collected in the same way: two failures, including the basket failure below, and two successes. Each video plays the episode beside the matched reference frames and the alignment.







C RL EPISODE COUNTS

Counting RL episodes. Our counts include every training episode collected from the start of RL and exclude evaluation episodes.

SimpleVLA-RL (Li et al., 2025) filters episodes before updating the policy, so the number collected can exceed the number used for learning. Its released code collects eight episodes for each of 64 scenes and removes groups in which all eight attempts fail or all eight succeed. When the success rate is low, many groups fail all eight attempts; when it is high, many succeed on all eight. Either way, a large share of the collected episodes can be discarded. Collection continues until an update has 64 retained groups, or $64 \times 8 = 512$ episodes. An RL step therefore accounts for 512 retained episodes; discarded episodes also contribute to the interaction cost.

LIBERO. SimpleVLA-RL's published one-demonstration LIBERO-Long curve (their Figure 3) first reaches its reported 91.7% at approximately step 223, corresponding to about $223 \times 512 = 114{,}176$ retained episodes. With one demonstration per task, our VLA reaches 92.6% on LIBERO-Long within 26,880 collected episodes, so it meets 91.7% no later. Dividing these counts gives approximately 4.2; we conservatively report an estimated $4\times$ comparison between the two methods. The baseline count excludes discarded episodes. The other suites' learning curves are not published, so we report no ratio for them.

RoboTwin 2.0. For five of our eight RoboTwin tasks, SimpleVLA-RL's public model collection provides one RL checkpoint per task and no later checkpoints. We evaluate these checkpoints with SimpleVLA-RL's released code on 128 held-out seeds; each scores below its published success rate. A checkpoint's training step gives the episodes used for its updates, 512 per step, for 148,480 episodes across the five checkpoints. Counting our RL episodes up to the first evaluation that matches or exceeds each checkpoint's measured success gives 26,624 episodes in total, an estimated 5.6× fewer. Excluding the baseline's discarded episodes makes this comparison conservative.

Ego. Our RL collects 2,304 to about 23,000 training episodes per Ego task. EgoVLA uses no online RL, so there is no episode count to compare.

Controlled VLA comparison. For Figure 4(b, c), we average best-so-far success rates over the three RoboTwin tasks and count RL episodes per task, up to 1,152 per task. On each task, all four methods start from the same imitation-learned policy, with 41%, 33% and 17% success on Handover block, Handover mic and Place phone stand. GRPO first reaches its peak of 50.5% after 960 episodes, and GRPO with DEAL reaches that success after 576; the episode ratio is $960/576 \approx 1.7$. PPO first reaches its 50.0% peak after 960 episodes, and PPO with DEAL reaches that success after 384, giving $960/384 = 2.5$.

D LIMITATIONS

DEAL's alignment assumes that an episode and its reference complete subgoals in the same order. When a task permits another valid order, an episode can make progress that the ordered alignment cannot match to the corresponding reference steps, causing productive actions to be labeled wasted. Extending DEAL to tasks with interchangeable subgoals therefore calls for order-invariant alignment, for example by matching subgoal segments in any order while preserving the order of steps within each segment; we leave this to future work.

E WHY ALIGNMENT-BASED STEP SELECTION HELPS: A TABULAR ANALYSIS

This appendix states and proves the results summarized by Proposition 1. We use a tabular task in which every action either completes the current subgoal or wastes a step, so that the productive and wasted steps are known exactly. Three questions are answered in order. First, does the alignment of Section 3.2 recover exactly the wasted steps (Theorem 1)? Second, in which direction does each kept step move the policy, with and without selection (Proposition 2 and Theorem 2)? Third, how many episodes does an update that gives every action the same credit need before it reliably favors the productive action at one subgoal (Theorem 3)? Proposition 1 combines Theorems 1 and 2 with Proposition 2 and Theorem 3(d–f).

E.1 SETTING

Definition 1 (Progress chain)**.** A task has $N \geq 1$ ordered subgoal states $\bar{x}_1, \ldots, \bar{x}_N$ and a goal $\bar{x}_{N+1}$. Every state x has a level $\kappa(x) \in \{1, \ldots, N+1\}$, with $\kappa(\bar{x}_k) = k$. At $\bar{x}_k$ the policy chooses between *advance*, which moves to $\bar{x}_{k+1}$, and *waste*, whose effect is fixed per subgoal: either a pause, which stays at $\bar{x}_k$ ($d_k = 1$), or a detour of $d_k - 1 \geq 1$ forced steps through side states of level k that return to $\bar{x}_k$. Side states involve no choice. An episode $x_1, a_1, \ldots, x_T, a_T$ starts at $\bar{x}_1$ and has $T \geq 1$ steps. It is a success ($y = 1$) if $x_{T+1} = \bar{x}_{N+1}$ with $T \leq T_{\max}$, and a failure ($y = 0$) if $T = T_{\max}$ steps have been taken without reaching the goal, where $T_{\max} \geq N$ is the step limit. Step t is *productive* if $\kappa(x_{t+1}) = \kappa(x_t) + 1$ and *wasted* otherwise. The chain with its step limit is a finite-horizon tabular Markov decision process: the state is x together with the number of steps left, and the reward is y at termination. The *pause-only model* has $d_k = 1$ for every k.

Definition 2 (Tabular policy)**.** The policy depends on x only and advances at $\bar{x}_k$ with probability $p_k = 1/(1 + e^{-\vartheta_k})$, where $\vartheta \in \mathbb{R}^N$. Let $q(\vartheta)$ denote the success probability. In one episode, m_k counts the decisions at $\bar{x}_k$, and $\xi_k \in \{0, 1\}$ records whether one of them was an advance. We realize an episode by running the chain without a step limit and truncating it after $T_{\max}$ steps. In the pause-only model, $L_k \sim \mathrm{Geom}_0(p_k)$ then denotes the number of pauses at $\bar{x}_k$ in the untruncated run, $\Pr(L_k = l) = (1 - p_k)^l p_k$, and the L_k are independent. Let $S_{-k} = \sum_{j \neq k}(L_j + 1)$ be the time

spent at the other subgoals and $B_k = T_{\max} - S_{-k}$ the steps left for subgoal k. The episode succeeds if and only if $L_k + 1 \le B_k$.

Definition 3 (Reference, features and selection)**.** Each state has a feature $f(x)$, and step t has $z_t = f(x_t)$. The reference is a success with features $z_1^{\mathrm{ref}}, \dots, z_{T_{\mathrm{ref}}}^{\mathrm{ref}}$. An *alignment path* is a sequence of cells that starts at $(1,1)$ and moves by $(+1,+1)$, $(+1,0)$ or $(0,+1)$; its cost is the sum of $C_{i,t}$ over its cells. The alignment is Eq. 1 with $D_{1,1} = C_{1,1}$ and $D = +\infty$ outside the grid, so that $D_{i,t}$ is the minimum cost of a path ending at (i,t). Tracing minimizing predecessors back from (T_{ref}, T) for a success and from $(i^\star, T)$ for a failure, with $i^\star$ the smallest minimizer of $D_{i,T}$, returns a path of minimum cost to that endpoint. M_i collects the steps matched to reference step i. As in Section 3.2, a step in a repeated group is wasted unless its action moves the episode past that reference step:

- for $t < T$, step t is wasted if and only if steps t and $t+1$ are matched to a common reference step;
- a success's last step completes the task and is productive;
- a failure's last step is wasted if and only if $T \ge 2$ and it shares a reference step with step $T-1$, that is, the failure never left its last repeated group.

The first item labels wasted the steps of $M_i \setminus \{\max M_i\}$ for every i with $|M_i| > 1$, because each M_i is a set of consecutive steps and the path leaves a group only through its last step. The third item is the rule of Section 3.2 for the group $M_{i^\star}$: since $i^\star$ is the smallest minimizer and costs are nonnegative, the traced path cannot enter $(i^\star, T)$ from $(i^\star - 1, T)$, so step T is matched to $i^\star$ only and the item says $|M_{i^\star}| > 1$. DEAL selects the productive steps of successes and the wasted steps of failures. An episode with no selected step makes no update.

Definition 4 (Outcome-signed updates)**.** Each episode τ receives an advantage A_τ whose sign is its outcome. We cover (R) a fixed baseline, $A_\tau = y - b$ with $b \in (0,1)$, and (G) GRPO, $A_\tau = (y - \bar{y})/\mathrm{sd}(y)$ over a group of $G \ge 2$ episodes of one scene, with groups whose outcomes are all equal dropped. A batch update with step size $\alpha > 0$ and positive per-step weights $\omega_{\tau,t}$ is

$$\Delta\vartheta = \alpha \sum_{\tau} A_\tau \sum_{t \in \mathcal{S}_\tau} \omega_{\tau,t}\, \nabla_\vartheta \log \pi(a_t \mid x_t),$$

where $\mathcal{S}_\tau$ contains every step for the *outcome-only* update and the selected steps for DEAL. GRPO's per-episode normalization is one such weight.

Assumption 1. (A1) Every transition changes the level by 0 or $+1$. (A2) For every visitable non-goal state x and every $j \in \{1, \dots, N\}$ with $j \ne \kappa(x)$, $\|f(x) - f(\bar{x}_{\kappa(x)})\| < \|f(x) - f(\bar{x}_j)\|$. (A3) The reference is the length-N success, $z_i^{\mathrm{ref}} = f(\bar{x}_i)$. (A4) The weight $\omega_{\tau,t} = \omega_\tau$ is constant within each episode. (A5) $p_k \ge \frac{1}{2}$ for every k.

(A1) restates the transition structure of Definition 1. (A2) says that a detour looks more like the subgoal it left than like any other subgoal, and that subgoal features are distinct. (A3) holds once the policy has produced a success without waste; Section E.5 treats references that pause. (A4) and (A5) are used only for the last group of a failure; Remark 2 explains why they are needed and what holds without them.

E.2 Alignment recovers the wasted steps

Lemma 1 (DTW returns the true matching)**.** *Let $C \in \mathbb{R}_{\ge 0}^{T_{\mathrm{ref}} \times T}$, and let $u : \{1, \dots, T\} \to \{1, \dots, T_{\mathrm{ref}}\}$ satisfy $u(1) = 1$ and $u(t+1) - u(t) \in \{0, 1\}$. Suppose every column is minimized strictly at u: $C_{u(t),t} < C_{i,t}$ for all t and all $i \ne u(t)$. Then:*

(a) among all alignment paths that end in column T, the path $\Gamma_u = ((u(t), t))_{t=1}^{T}$ is the unique one of minimum cost;

(b) $u(T)$ is the unique minimizer of $D_{\cdot,T}$, and tracing back from $(u(T), T)$ returns Γ_u under any tie-breaking rule;

(c) if $u(T) = T_{\mathrm{ref}}$, tracing back from (T_{ref}, T) returns Γ_u under any tie-breaking rule;

(d) in both cases $M_i = u^{-1}(i)$.

Proof. By induction on $i+t$, $D_{i,t}$ in Eq. 1 is the minimum cost of a path ending at (i,t), and tracing back through minimizing predecessors returns such a path. (a) The column index of a path starts at 1, ends at T and grows by at most one per move, so every path ending in column T has at least one cell in every column. If the cells of a path in column t are exactly $\{(u(t),t)\}$, that column contributes $C_{u(t),t}$. Otherwise the column contains some (i,t) with $i \neq u(t)$, and since costs are nonnegative it contributes at least $C_{i,t} > C_{u(t),t}$. Summing over columns, every path costs at least $\mathrm{cost}(\Gamma_u)$, with equality only if every column holds exactly the cell $(u(t),t)$; a path is determined by its set of cells, so equality holds only for Γ_u, which is itself a path since its moves are diagonal or episode-only. (b) Every path ending at (i,T) with $i \neq u(T)$ differs from Γ_u, so by (a) it costs more. Finitely many paths exist, hence $D_{i,T} > D_{u(T),T}$. The minimum-cost path to $(u(T),T)$ is unique, so every tie-breaking rule returns it. (c) is the same argument restricted to paths ending at (T_{ref},T). (d) follows from (b) and (c). □

Theorem 1 (The alignment labels exactly the wasted steps)**.** *Under (A1)–(A3), with the labels of Definition 3 and any tie-breaking rule:*

(a) in a success, the wasted steps are labeled wasted and its N productive steps are labeled productive;

(b) in a failure, a step $t < T$ is labeled wasted if and only if it is wasted, and $i^\star = \kappa(x_T)$. For $T \geq 2$, the last step is labeled wasted if and only if x_{T-1} and x_T have the same level; the result of its action is not observed.

Proof. Let $u(t) = \kappa(x_t)$. By (A1), $u(1) = 1$ and $u(t+1) - u(t) \in \{0,1\}$. A success has $x_{T+1} = \bar{x}_{N+1}$, so $u(T) = N = T_{\mathrm{ref}}$. By (A3), $C_{i,t} = \|f(\bar{x}_i) - f(x_t)\|$, so (A2) is exactly the strict column condition of Lemma 1. The alignment path is therefore Γ_u for successes and failures, with $i^\star = \kappa(x_T)$ for a failure, and step t is matched to reference step $u(t)$ only. For $t < T$, steps t and $t+1$ share a reference step if and only if $u(t+1) = u(t)$, that is, if and only if step t did not raise the level. A success's last step enters the goal, which is productive, and Γ_u has $N-1$ diagonal moves, so a success has N productive steps. The statement about a failure's last step restates the rule with $u(T) = u(T-1)$. □

Remark 1. Strict separation is needed only between each step and the other subgoals; nearest-neighbor matching would also recover the matching under it. What the ordering built into Eq. 1 adds when separation fails is not analyzed here. A condition that compares each step only with the neighboring subgoals is not enough: for reference features $(-0.56, -0.77, 0.00)$ and episode features $(-0.60, 0.04, 0.01, -1.77, -0.17, -0.21)$ with true levels $(1,1,1,2,3,3)$, the local condition holds, yet the alignment path costs 2.41 against 2.59 for the true matching. Without (A2), a detour that resembles the next subgoal can be matched to it and mislabeled: with $f(\bar{x}_1) = (0,0)$, $f(\bar{x}_2) = (1,0)$ and a side state with feature $(0.95, 0.1)$ at level 1, the success that takes one three-step detour (two side-state steps) before advancing gets a path of cost 1.22 against 1.91 for the true matching, and its waste decision is labeled productive; with a two-step detour the true matching is recovered.

E.3 Direction of the credit

Lemma 2 (Scores and true advantages)**.** *(i) $\partial_{\vartheta_k} \log \pi(\textit{advance} \mid \bar{x}_k) = 1 - p_k$ and $\partial_{\vartheta_k} \log \pi(\textit{waste} \mid \bar{x}_k) = -p_k$. All other coordinates, and all forced side steps, have zero score. Hence, under (A4), one episode's outcome-only update at subgoal k is $\alpha\omega_\tau A_\tau(\xi_k - m_k p_k)$. (ii) Let $U_c(k)$ be the probability of reaching the goal within c steps from $\bar{x}_k$, with $U_c(N+1) = 1$ for $c \geq 0$ and $U_c(k) = 0$ for $c \leq 0$ and $k \leq N$. With $c \geq 1$ steps left at $\bar{x}_k$, the true advantages are $(1-p_k)\,\Delta U_c(k)$ for advance and $-p_k\,\Delta U_c(k)$ for waste, where $\Delta U_c(k) = U_{c-1}(k+1) - U_{c-d_k}(k) \geq 0$. Advances are never worse than waste, and both advantages are zero when $c = 1$ and $k < N$.*

Proof. (i) $\frac{d}{d\vartheta} \log \frac{1}{1+e^{-\vartheta}} = 1 - p$ and $\frac{d}{d\vartheta} \log \frac{e^{-\vartheta}}{1+e^{-\vartheta}} = -p$. Summing over the decisions at $\bar{x}_k$ gives $\xi_k(1-p_k) - (m_k - \xi_k)p_k$. (ii) The action values are $U_{c-1}(k+1)$ and $U_{c-d_k}(k)$, and the state value is their p_k-weighted mean, which gives both formulas. $U_c(k)$ is nondecreasing in c, and also in k:

realize the untruncated chain once, so that the time to the goal from $\bar{x}_{k+1}$ is at most that from $\bar{x}_k$. Hence $U_{c-d_k}(k) \le U_{c-1}(k) \le U_{c-1}(k+1)$. For $c = 1$ and $k < N$ both values are 0. □

Proposition 2 (Outcome-only credit can point the wrong way)**.** *Consider one episode with (A4) and outcome-signed A_τ.*

- *(a) A success lowers p_k if and only if it made more than $1/p_k$ decisions at $\bar{x}_k$. For $p_k > \frac{1}{2}$ one waste suffices.*
- *(b) A failure lowers p_k at every subgoal it left after fewer than $1/p_k$ decisions, in particular at every subgoal it passed without waste. It raises p_k at a subgoal where it decided but never advanced; such a subgoal exists unless its last action is an advance.*
- *(c) Step by step, the advantage assigned to an advance in a success and to a waste in a failure never opposes the true advantage; the advantage assigned to a waste in a success and to an advance in a failure opposes it whenever that advantage is nonzero. Under (A1)–(A3), DEAL keeps only steps of the first two kinds and, in a failure, possibly its last action if that action is an advance; such an advance is taken with one step left below level N, so its true advantage is zero (Theorem 1, Lemma 2(ii)).*

Proof. (a) and (b) follow from the sign of $A_\tau(\xi_k - m_k p_k)$ in Lemma 2(i), with $\xi_k = 1$ in (a) and at the subgoals a failure left, and $\xi_k = 0$, $m_k \ge 1$ at a subgoal where it decided but never advanced; a failure's last action is an advance exactly when $\xi_k = 1$ at every subgoal where it decided. (c) compares the signs of A_τ with those in Lemma 2(ii). By Theorem 1, a success keeps only advances and a failure keeps, before T, only wasted steps; its last step, if kept, is taken with one step left at a level below N, where both advantages vanish, or has level N, where a failure's last action is a waste or a forced side step. □

Lemma 3 (More advancing means more success)**.** *q is nondecreasing in every p_k. In the pause-only model,*

$$\frac{\partial q}{\partial \vartheta_k} = \mathbb{E}\big[p_k\, B_k (1-p_k)^{B_k}\, \mathbf{1}\{B_k \ge 1\}\big] > 0,$$

so $\vartheta' \ge \vartheta$ componentwise with $\vartheta' \ne \vartheta$ implies $q(\vartheta') > q(\vartheta)$. Moreover, $q < 1$ for every finite ϑ, and $q \to 1$ if and only if every $p_k \to 1$.

Proof. Represent the number of wastes at $\bar{x}_k$ in the untruncated run as $\lfloor \ln \Upsilon_k / \ln(1-p_k) \rfloor$ with Υ_k uniform on $(0,1)$; it is then nonincreasing in p_k, and so is the total time to the goal, which gives the first claim in both models. In the pause-only model, given S_{-k}, success means $L_k \le B_k - 1$, which has probability $1-(1-p_k)^{B_k}$ for $B_k \ge 1$ and 0 otherwise. Differentiating in p_k and multiplying by $dp_k/d\vartheta_k = p_k(1-p_k)$ gives the formula. The derivative is positive because $B_k = T_{\max} - N + 1 \ge 1$ whenever every other subgoal is passed without a pause, which has positive probability. Integrating the positive gradient along the segment from ϑ to ϑ' gives the strict inequality. Finally, $q \le 1 - (1-p_k)^{T_{\max}-N+1}$ for every k, and $q \ge \prod_k p_k$. □

Theorem 2 (DEAL's updates never make advancing less likely)**.** *Assume (A1)–(A4) and outcome-signed advantages. For every batch and every step size $\alpha > 0$, DEAL's update satisfies $\Delta\vartheta_k \ge 0$ for every k with $p_k \ge \frac{1}{2}$, hence for every k under (A5). Consequently, under (A1)–(A5):*

- *(a) the success probability never decreases, and in the pause-only model it strictly increases whenever $\Delta\vartheta \ne 0$;*
- *(b) every mixed GRPO group, and under (R) every success, raises every ϑ_k strictly;*
- *(c) (A5) is preserved by the update, so if it holds initially, (a) and (b) hold at every update.*

Proof. By Theorem 1(a), a success keeps exactly its N advances, one per subgoal, and each adds $\alpha\omega_\tau A_\tau(1-p_k) > 0$ to its own coordinate (Lemma 2(i)). By Theorem 1(b), every step $t < T$ that a failure keeps is wasted: a waste decision adds $\alpha\omega_\tau |A_\tau| p_k > 0$ and a forced side step adds 0. It remains to handle a failure's last step when it is selected, that is, when $T \ge 2$ and x_{T-1} and x_T have the same level k. If x_T is a side state or a_T is a waste, its contribution is nonnegative. If

$x_T = \bar{x}_k$ and a_T is an advance, it contributes $-\alpha\omega_\tau|A_\tau|(1-p_k)$, and the same group contains a selected waste decision at $\bar{x}_k$: either $x_{T-1} = \bar{x}_k$, whose decision kept the level, or x_{T-1} is a side state of a detour that started with such a decision, at a step whose successor also has level k. With the common weight of (A4), coordinate k therefore receives at least $\alpha\omega_\tau|A_\tau|\,(p_k - (1-p_k)) \geq 0$ when $p_k \geq \frac{1}{2}$. Summing over episodes gives the sign of $\Delta\vartheta_k$. Lemma 3 gives (a). For (b), a mixed group contains a success, whose contribution is positive at every coordinate. (c) holds because no coordinate decreases. □

Remark 2 (The two conditions). Both (A4) and (A5) are needed for the pathwise statement, and only for a failure's last group. With $N = 2$, $T_{\max} = 3$ and $p = (0.2, 0.6)$, the failure (waste, waste, advance) stays at $\bar{x}_1$, and its selected group gives $\Delta\vartheta_1 \propto 3p_1 - 1 < 0$. With $p_1 = 0.6$ in the same chain, the terminal group (waste, waste, advance) with weights $(1, 1, 10)$ gives $0.6 + 0.6 - 4 < 0$. Without (A5), $\mathbb{E}[\Delta\vartheta_k] \geq 0$ still holds for every k, provided the weights and advantages are determined by the batch's state sequences and outcomes, as they are under (R) with constant weights and under GRPO with its per-episode normalization. Indeed, the outcome of a failure whose last state is below level N does not depend on its last action, and neither does the selection of its last step, so given the state sequences and the other episodes the score of that action has mean zero; a failure whose last state is $\bar{x}_N$ must end with a waste, whose contribution is positive; every other selected step contributes a nonnegative amount. A weight that grows with the last action breaks this: weighting failures that end with an advance a hundred times more turns the mean update at $\bar{x}_1$ in the example above negative. Nothing is claimed in expectation for q itself.

Corollary 1 (Convergence)**.** *Assume (A1)–(A5), with (A5) holding initially, and (R). If $\alpha_t\omega_\tau \geq c_0 > 0$ at every update, then $\vartheta_k \to \infty$ for every k almost surely, so $q \to 1$. Step selection biases the update, but it does not move the optimum.*

Proof. By Theorem 2, q stays at least its initial value $q_0 > 0$, so every batch contains a success with conditional probability at least q_0 given the past, and by Lévy's extension of the Borel–Cantelli lemma, infinitely many batches contain a success almost surely. Fix k and an integer $\bar{\vartheta}$. On the event $\{\vartheta_k \leq \bar{\vartheta}$ forever$\}$, each such batch raises ϑ_k by at least $c_0(1-b)(1-p_k) \geq c_0(1-b)/(1+e^{\bar{\vartheta}})$ and no batch lowers it, so $\vartheta_k \to \infty$ on that event, which therefore has probability zero. A union over integers $\bar{\vartheta}$ gives $\vartheta_k \to \infty$, and $q \geq \prod_k p_k \to 1$. □

E.4 How many episodes outcome-only credit needs

We now fix the policy and ask how many episodes it takes for the averaged update at one subgoal to favor advancing. Work in the pause-only model with (R) and unit weights. Let $X_k^{\text{out}} = (y-b)(\xi_k - m_k p_k)$ be one episode's outcome-only update at subgoal k divided by α, and let X_k^{sel} be DEAL's. Let $\nu_k = \mathbb{E}X_k^{\text{out}}$, $\text{sd}_k^2 = \text{Var}X_k^{\text{out}}$, $\text{SNR}_k = \nu_k/\text{sd}_k$, $\beta = \min(b, 1-b)$, $\bar{\beta} = \max(b, 1-b)$ and $\chi_b = \pi[(1-b)^2 + b^2]$.

Why the mean is small. Informally, whether an episode succeeds depends on the time it spends at all N subgoals. A single decision at $\bar{x}_k$ changes that time by one step, so it changes the outcome only when the time spent at the other $N-1$ subgoals lands within about one step of the limit, which has probability of order $1/\sqrt{N}$. Every episode that visits $\bar{x}_k$ still moves ϑ_k by an amount of order one, and the sign of that movement is set by the outcome, which the other subgoals mostly decide. A success that paused at $\bar{x}_k$ but was fast elsewhere reinforces the pause, and a failure that advanced immediately at $\bar{x}_k$ but was slow elsewhere discourages the advance. The mean update is the small remainder after these movements cancel.

Theorem 3 (Averaging cost of outcome-only credit)**.** *In the pause-only model with (R):*

(a) $\nu_k = \partial q/\partial\vartheta_k > 0$. *If all $p_j = p$, then $\nu_k = \binom{T_{\max}}{N} p^N (1-p)^{T_{\max}-N+1}$ for every k.*

(b) Let $N \geq 2$ and $\Lambda_j = (1-p_j)/(2-p_j)^2$. Then $\nu_k \leq \frac{1-p_k}{p_k}\sqrt{\pi/(8\sum_{j\neq k}\Lambda_j)}$. If $p_j \leq p_{\text{hi}} < 1$ for all $j \neq k$, then $\nu_k = O(1/\sqrt{N})$ for fixed p_k and p_{hi}.

(c) $\beta^2 p_k(1-p_k)\, q \leq \mathbb{E}\big[(X_k^{\text{out}})^2\big] \leq \bar{\beta}^2(1-p_k)$.

(d) Let all $p_j = p \in (0,1)$, $T_{\max} = \lceil N/p \rceil$, and $k = k_N$ with $k_N/N \to \bar{k} \in [0,1)$. As $N \to \infty$: $q \to \frac{1}{2}$; $\sqrt{N}\,\nu_k \to \sqrt{(1-p)/(2\pi)}$; $\mathrm{sd}_k^2 \to (1-p)[(1-b)^2+b^2]/2$; and $N\,\mathrm{SNR}_k^2 \to 1/\chi_b$, which is $2/\pi$ at $b = \frac{1}{2}$. Both $\Pr(X_k^{\mathrm{out}} < 0)$ and $\Pr(X_k^{\mathrm{out}} > 0)$ tend to $\frac{1}{2}\Pr(mp \neq 1)$ with $m \sim \mathrm{Geom}_1(p)$, which is $\frac{1}{2}$ when $1/p$ is not an integer: the sign of one episode's update at a subgoal becomes a fair coin.

(e) Let $\bar{X}_Q$ average Q independent episodes. For every Q, $\Pr(\bar{X}_Q \le 0) \le 1/(Q\,\mathrm{SNR}_k^2)$. In the regime of (d), if $Q = Q_N$ with $Q_N/N \to \varkappa > 0$, then

$$\Pr(\bar{X}_Q \le 0) \to \Phi\Big(-\sqrt{\varkappa/\chi_b}\Big),$$

which is at most η if and only if $\varkappa \ge \chi_b\,\Phi^{-1}(1-\eta)^2$. At $b = \frac{1}{2}$, the limit of the mean reward in this regime, linear growth is also necessary: if $\Pr(\bar{X}_{Q_N} > 0) \ge 1-\eta > \frac{1}{2}$ for all large N, then $\liminf_N Q_N/N \ge \frac{\pi}{2}\Phi^{-1}(1-\eta)^2$, about $4.25N$ episodes for $\eta = 0.05$. For other baselines a bounded number of episodes can suffice: with $p = 0.999$ and $b = 0.05$, ten episodes give a positive average with limiting probability 0.994.

(f) Under (A1)–(A5), $X_k^{\mathrm{sel}} \ge 0$ surely, and every success gives $X_k^{\mathrm{sel}} = (1-b)(1-p_k) > 0$. Hence $\Pr(\bar{X}_Q^{\mathrm{sel}} > 0) \ge 1-(1-q)^Q$. In the regime of (d), any fixed $Q > \log_2(1/\eta)$ suffices for every large N; for example, five episodes give $1 - 2^{-5} \approx 96.9\%$ in the limit.

Proof. (a) Lemma 2(i) gives $\partial_{\vartheta_k} \log \Pr_\vartheta(\tau) = \xi_k - m_k p_k$, because transitions are deterministic and the episode space is finite. This score has mean zero, and $\mathbb{E}[y\,\partial_{\vartheta_k} \log \Pr_\vartheta(\tau)] = \partial q/\partial \vartheta_k$ (Williams, 1992), so the baseline adds no bias; positivity is Lemma 3. For equal p_j, extend each episode with fictitious coin flips after it succeeds. Then $y = \mathbf{1}\{\mathrm{Bin}(T_{\max}, p) \ge N\}$ and $\frac{d}{dp}\Pr(\mathrm{Bin}(T_{\max},p) \ge N) = T_{\max}\binom{T_{\max}-1}{N-1} p^{N-1}(1-p)^{T_{\max}-N}$. Multiplying by $p(1-p)$ gives the derivative for a parameter shared by all subgoals, which is $\sum_k \partial q/\partial \vartheta_k$. Since q is a symmetric function of $(p_1,\dots,p_N)$, all these partial derivatives are equal at $p_j \equiv p$, so each is $1/N$ of the total, and $T_{\max}\binom{T_{\max}-1}{N-1} = N\binom{T_{\max}}{N}$.

(b) By Lemma 3, $\nu_k = \sum_l \Pr(S_{-k} = l)\, p_k (T_{\max} - l)(1-p_k)^{T_{\max}-l}\mathbf{1}\{l \le T_{\max}-1\}$. Since $\sum_{c \ge 1} p_k c (1-p_k)^c = (1-p_k)/p_k$, $\nu_k \le \frac{1-p_k}{p_k}\max_l \Pr(S_{-k} = l)$. A $\mathrm{Geom}_1(p)$ variable has characteristic function of modulus $(1+x)^{-1/2}$ with $x = 2(1-p)(1-\cos\zeta)/p^2 \le 4(1-p)/p^2$. Using $\log(1+x) \ge x/(1+x_{\max})$, the modulus is at most $\exp(-\Lambda(1-\cos\zeta))$ with $\Lambda = (1-p)/(2-p)^2$. Multiplying over $j \neq k$, bounding $1-\cos\zeta \ge 2\zeta^2/\pi^2$ on $[-\pi,\pi]$, and inverting gives $\max_l \Pr(S_{-k} = l) \le \frac{1}{2\pi}\int e^{-2\zeta^2 \sum_{j\neq k}\Lambda_j/\pi^2} d\zeta = \sqrt{\pi/(8\sum_{j\neq k}\Lambda_j)}$. The function $(1-p)/(2-p)^2$ is decreasing in p, so $\sum_{j\neq k}\Lambda_j \ge (N-1)\Lambda_{\mathrm{hi}}$ with $\Lambda_{\mathrm{hi}} = (1-p_{\mathrm{hi}})/(2-p_{\mathrm{hi}})^2$, which gives the last claim.

(c) Condition on the arrival time at $\bar{x}_k$. The decisions there are independent Bernoulli(p_k) draws, $\xi_k - m_k p_k$ is the sum of their centered values over a bounded stopping time, and Wald's second identity gives $\mathbb{E}(\xi_k - m_k p_k)^2 = p_k(1-p_k)\mathbb{E}m_k$. Moreover $(y-b)^2 \in [\beta^2, \bar{\beta}^2]$, and $q \le \mathbb{E}m_k \le 1/p_k$ because every success decides at least once at $\bar{x}_k$.

(d) Here $|pT_{\max} - N| < 1$, so the De Moivre–Laplace theorem gives $q \to \frac{1}{2}$, and its local version applied to (a) gives the limit of $\sqrt{N}\nu_k$. Fix $l \ge 0$. Given $L_k = l$, the episode succeeds if and only if $S_{-k} \le T_{\max} - l - 1$; S_{-k} has mean $(N-1)/p$ and standard deviation of order $\sqrt{N}$, and the shift $l+1$ is bounded, so this probability tends to $\frac{1}{2}$ by the central limit theorem. The arrival time at $\bar{x}_k$ has mean $(k-1)/p$ and standard deviation of order $\sqrt{N}$, while $T_{\max} - (k-1)/p$ grows linearly in N because $\bar{k} < 1$, so $\Pr(\xi_k = 1 \mid L_k = l) \to 1$ by Chebyshev's inequality. Because L_k has geometric tails and $|X_k^{\mathrm{out}}| \le \bar{\beta}(1 + (L_k+1)p)$, dominated convergence over l gives $\mathbb{E}(X_k^{\mathrm{out}})^2 \to \frac{1}{2}[(1-b)^2+b^2]\,\mathbb{E}(1-mp)^2$, and $\mathbb{E}(1-mp)^2 = p^2\mathrm{Var}(m) = 1-p$; since $\nu_k \to 0$, sd_k^2 has the same limit. The sign limits follow in the same way from $\{X_k^{\mathrm{out}} < 0\} = \{y = 1,\, mp > 1\} \cup \{y = 0,\, \xi_k = 1,\, mp < 1\}$ with $m = m_k$.

(e) Chebyshev's inequality gives $\Pr(\bar{X}_Q \le 0) \le \Pr(|\bar{X}_Q - \nu_k| \ge \nu_k) \le \mathrm{sd}_k^2/(Q\nu_k^2)$. For the limit, $\mathbb{E}|X_k^{\mathrm{out}}|^3$ is bounded uniformly in N by the bound in (d) and sd_k has a positive

limit, so the Lyapunov central limit theorem applies to the triangular array $(X^{\mathrm{out}}_{k,i})_{i\le Q_N}$ whenever $Q_N \to \infty$, and Pólya's theorem makes the convergence of the distribution functions uniform. Since $\sqrt{Q_N}\,\nu_k/\mathrm{sd}_k \to \sqrt{\varkappa/\chi_b}$ by (d), $\Pr(\bar X_Q \le 0) \to \Phi(-\sqrt{\varkappa/\chi_b})$, and $\Phi(-\sqrt{\varkappa/\chi_b}) \le \eta$ if and only if $\varkappa \ge \chi_b \Phi^{-1}(1-\eta)^2$; at $b = \frac{1}{2}$, $\chi_b = \pi/2$ and $(\pi/2)\Phi^{-1}(0.95)^2 = 4.25$. For necessity at $b = \frac{1}{2}$, fix Q. Given $L_{k,1}, \dots, L_{k,Q}$, the outcomes $y_1, \dots, y_Q$ are independent, and by the argument of (d) their conditional success probabilities tend to $\frac{1}{2}$ uniformly on $\{\max_i L_{k,i} \le M\}$, an event of probability at least $1 - Q(1-p)^{M+1}$ for every N. Flipping every outcome maps $\sum_i X^{\mathrm{out}}_{k,i}$ to its negative on $\{\xi_{k,i} = 1 \text{ for all } i\}$, an event of probability tending to one, so for fixed Q, $\Pr(\bar X_Q > 0) \le \frac{1}{2} + o(1)$. If instead $Q_N \to \infty$ with $Q_N/N \to 0$, the limit above gives $\Pr(\bar X_Q \le 0) \to \Phi(0) = \frac{1}{2}$. Hence, along any subsequence with $Q_N/N \to \varkappa' < \frac{\pi}{2}\Phi^{-1}(1-\eta)^2$, $\limsup \Pr(\bar X_Q \le 0) \ge \min\{\frac{1}{2}, \Phi(-\sqrt{2\varkappa'/\pi})\} > \eta$, which contradicts $\Pr(\bar X_{Q_N} > 0) \ge 1 - \eta$. The example with $b = 0.05$ is a direct computation under the limit law of (d), in which $X^{\mathrm{out}}_k = (y - b)(1 - mp)$ with y a fair coin independent of $m \sim \mathrm{Geom}_1(p)$.

(f) The sign is Theorem 2 with unit weights. A success keeps one advance at every subgoal, so $X^{\mathrm{sel}}_k = (1-b)(1-p_k)$. The average is therefore positive whenever the batch contains a success, which has probability $1 - (1-q)^Q$, and $q \to \frac{1}{2}$ by (d). □

Parts (d) and (e) concern the regime in which success is uncertain. Away from it, the mean in (a) is exponentially small, and the sign can still be wrong often: for example, with $T_{\max} = 2N$ and $p = 0.6$, $\Pr(X^{\mathrm{out}}_k < 0) \to (1-p)^{\lfloor 1/p \rfloor} = 0.4$, because $y \to 1$ and $\xi_k \to 1$ in probability. Returning to the regime of (d), the outcome-only update summed over subgoals is not diluted in this symmetric chain: its mean $N\nu_k$ is of order $\sqrt{N}$ by (d), while by Wald's second identity its second moment is at most $\bar\beta^2 p(1-p)T_{\max}$, of order N. The dilution concerns the part of the credit specific to one subgoal, which matters when different subgoals need different corrections. The next corollary isolates one learned subgoal, so that the count applies to the success probability itself.

Corollary 2 (One subgoal being learned)**.** *Let only ϑ_k be learned, and let the environment set the time spent at every other subgoal to $1 + L_j$ with $L_j \sim \mathrm{Geom}_0(p)$ independent of ϑ, so that those steps carry no score. With (R), unit weights and Q episodes per update, the outcome-only update lowers q if and only if $\sum_{i=1}^{Q} X^{\mathrm{out}}_{k,i} < 0$. With $p_k = p$ in the regime of Theorem 3(d) and $Q/N \to \varkappa > 0$, this probability tends to $\Phi(-\sqrt{\varkappa/\chi_b})$, which is 21% for $Q = N$ and 5.5% for $Q = 4N$ at $b = \frac{1}{2}$. With exact features that are distinct across subgoals, if the reference makes one decision at $\bar x_k$ and $p_k \ge \frac{1}{2}$, DEAL's update never lowers q.*

Proof. By Lemma 3, whose proof applies verbatim to this chain, q is strictly increasing in ϑ_k, and the update is $\alpha \sum_i X^{\mathrm{out}}_{k,i}$. The law of $X^{\mathrm{out}}_{k,i}$ depends only on L_k, the arrival time at $\bar x_k$ and S_{-k}, so it is the law of Theorem 3. For DEAL, the reference may pause at other subgoals, so Proposition 3 applies: a success keeps its advance at $\bar x_k$ and, since $\rho_k = 1$, no pause there, contributing $(1 - b)(1 - p_k) > 0$; a failure's kept steps at $\bar x_k$ contribute a nonnegative amount by Proposition 3(a) with $p_k \ge \frac{1}{2}$; kept steps at other subgoals carry zero score. □

E.5 References with pauses

The reference is the shortest success so far, so early in training it may itself pause. The next result covers the pause-only model with exact features, $f(x) = f(\bar x_{\kappa(x)})$ distinct across subgoals, and a reference that is a success of the same chain with $\rho_k \ge 1$ steps at level k.

Proposition 3 (Selection with a pausing reference)**.** *Under (A4) and any tie-breaking rule:*

- *(a) every step $t < T$ that a failure keeps is a pause, and a failure's last group consists of all its steps at its final level. The update of a failure is therefore nonnegative at every subgoal k with $p_k \ge \frac{1}{2}$.*
- *(b) a success keeps its N advances and at most $\rho_k - 1$ pauses at $\bar x_k$. Its update there, $\alpha\omega_\tau A_\tau(1 - p_k\hat m_k)$ with $1 \le \hat m_k \le \min(m_k, \rho_k)$ kept decisions, is at least $\alpha\omega_\tau A_\tau(1 - p_k\rho_k)$. It is nonnegative wherever $\rho_k \le 1/p_k$, in particular wherever the reference did not*

pause, and otherwise its negative part is bounded by the reference's pauses, whereas the outcome-only update $\alpha\omega_\tau A_\tau(1 - p_k m_k)$ is unbounded in m_k.

(c) for a fixed reference and a fresh success aligned to it, $\mathbb{E}[1 - p_k\hat{m}_k \mid y = 1] \geq (1-p_k)^{\rho_k} > 0$.

Proof. A cell costs zero if and only if its row and column have the same level, and a zero-cost path to each endpoint exists, so every optimal path stays in same-level blocks and crosses from level k to level $k + 1$ by the diagonal move between the blocks' corners, because a move changes each index by at most one. Inside block k, a path with N_{dg} diagonal, N_{rf} reference-only and N_{ep} episode-only moves satisfies $N_{\mathrm{dg}} + N_{\mathrm{rf}} = \rho_k - 1$ and $N_{\mathrm{dg}} + N_{\mathrm{ep}} = m_k - 1$. (a) A step labeled wasted before T leaves its column by an episode-only move inside a block, so it and the next step have the same level: it paused. For a failure, $D_{i,T} = 0$ exactly on the rows of its final level, $i^\star$ is the first of them, and the path stays in that row across the final block, so the final block forms one group; the last-group argument of Theorem 2 applies, and every other kept step is a pause. (b) A success's steps in block k that leave their column diagonally are kept; there are $\hat{m}_k = 1 + N_{\mathrm{dg}} \leq \min(m_k, \rho_k)$ of them, including the advance, and Lemma 2(i) gives the update. (c) The function $g(L_k) = 1 - p_k \min(L_k + 1, \rho_k)$ is a lower bound on $1 - p_k\hat{m}_k$, because $m_k = L_k + 1$ in a success, and is nonincreasing in L_k; so is $\mathbf{1}\{y = 1\}$ given the other subgoals. By Chebyshev's association inequality applied conditionally on the other subgoals, $\mathbb{E}[g\,\mathbf{1}\{y = 1\}] \geq \mathbb{E}[g]\,q$, and $\mathbb{E}\min(L_k + 1, \rho_k) = (1 - (1 - p_k)^{\rho_k})/p_k$ gives $\mathbb{E}g = (1 - p_k)^{\rho_k}$. □

Lemma 4 (The reference becomes free of waste)**.** *If $p_k \geq p_{\min} > 0$ for every k at all times, each episode is the length-N success with conditional probability at least $p_{\min}^N$. The reference, which is replaced only by a strictly shorter success, therefore satisfies (A3) after Q episodes with probability at least $1 - (1 - p_{\min}^N)^Q$. This bound guarantees (A3) only after a number of episodes exponential in N.*

Proof. An episode advances at every decision with conditional probability $\prod_k p_k \geq p_{\min}^N$ given the past. The length-N success is strictly shorter than every other success and has features $f(\bar{x}_i)$, so it becomes the reference when first collected and is never replaced. The probability that none of Q episodes is such a success is at most $(1 - p_{\min}^N)^Q$. □

E.6 Scope and numerical check

The guarantees concern binary tabular policies on a deterministic chain of ordered subgoals, in which waste delays completion without enabling later progress and step features are closest to the corresponding reference subgoal. Exact recovery (Theorem 1) assumes a reference without wasted steps; monotone improvement (Theorem 2) additionally assumes weights constant within each episode and an advance probability of at least one half at a failure's final group. In practice these assumptions may not hold: learned VLA features need not separate subgoals, parameters are shared across states, transitions are stochastic, and early references pause. Proposition 3 describes what remains with a pausing reference, and Remark 2 what remains without the weight and advance-probability conditions. The averaging count of Theorem 3(e) is proved as a necessary count for the baseline at the mean reward, which GRPO's centering approximates; for other fixed baselines it gives the limit for batches proportional to N only. The analysis compares two update rules for outcome-signed advantages, with and without step selection. It does not cover learned critics, and it is not a lower bound on what outcome supervision can achieve in general: Jia et al. (2025) show that outcome supervision is not statistically harder than process supervision up to polynomial factors in the horizon, and our factor N is polynomial. Variance of the likelihood-ratio gradient that grows with trajectory length was analyzed by Zhao et al. (2011); Theorem 3 gives the exact per-subgoal constant for the outcome-only update.

Exhaustive enumeration of small chains ($N \leq 4$, $T_{\max} \leq N + 4$, pauses and detours of length up to three, six feature maps satisfying (A2), all tie-breaking orders) confirms Theorems 1 and 2: every label agrees with Theorem 1, including its convention for a failure's last step, and no selected update is negative at any subgoal where $p_k \geq \frac{1}{2}$. Table A2 evaluates Theorem 3 by exact dynamic programming.

Table A2: Outcome-only credit at subgoal $k = N/2$, computed exactly ($p = 0.6$, $b = \frac{1}{2}$, $T_{\max} = \lceil N/p \rceil$). DEAL's update is never negative (Theorem 3(f)).

N	ν_k	$\sqrt{N}\nu_k$	$N\,\mathrm{SNR}_k^2$	$\Pr(X_k^{\mathrm{out}} < 0)$	$\Pr(X_k^{\mathrm{sel}} < 0)$
10	0.0771	0.244	0.633	0.381	0
50	0.0352	0.249	0.628	0.442	0
200	0.0178	0.251	0.634	0.470	0
2000	0.00564	0.252	0.636	0.491	0
limit	–	0.252	$2/\pi \approx 0.637$	0.5	0

Table A3 runs tabular GRPO in the pause-only chain (groups of $G = 8$, one scene, advantages centered by the group mean and divided by the sample standard deviation, unit per-step weights, step α/G per group, $p_k = \frac{1}{2}$ initially, $T_{\max}$ the smallest limit with initial success probability at least 0.2, 30 held-out seeds) with and without step selection, where DEAL's reference is the shortest success so far and all steps are used until the first success. These runs start before (A3) holds, so Theorem 2 does not apply to them; they illustrate the averaging cost of Theorem 3 rather than a proved episode count.

Table A3: Episodes for tabular GRPO to reach $q \geq 0.9$, without and with DEAL's step selection (mean over 30 held-out seeds). Left: the same step size 0.1. Right: each method's best step size, chosen on separate seeds. Ratios use the unrounded means.

	step size 0.1			best step size		
N	GRPO	GRPO + DEAL	ratio	GRPO	GRPO + DEAL	ratio
10	715	301	2.4	21	13	1.6
20	631	199	3.2	38	16	2.4
50	650	152	4.3	67	16	4.2
100	640	109	5.9	113	17	6.7
200	640	78	8.2	142	17	8.3