How Should LLMs Teach RL Agents? Action Assistance and Competence-Adaptive Release

Anonymous authors
Paper under double-blind review

Abstract

Reinforcement learning (RL) in complex decision-making tasks often requires extensive environment interaction. Large language models (LLMs) can provide state-conditioned action advice, but success with teacher assistance does not imply that an RL agent can solve the task independently. We propose a competence-adaptive framework for LLM-guided RL that distinguishes two roles of teacher guidance: action intervention, which changes the trajectories collected from the environment, and demonstration supervision, which supports RL policy updates. During training, the framework executes the LLM teacher's action with an adaptive probability. State–action pairs from actual interventions are stored and replayed to provide supervision alongside the RL objective. A fuzzy controller adjusts the intervention probability based on policy uncertainty, teacher-free evaluation success, and agreement between the RL agent's proposed actions and the teacher's actions, while retaining a positive base weight for demonstration supervision. We evaluate the framework on three MiniGrid task families, DoorKey, UnlockPickup, and MultiRoom, and Env2. Across six principal MiniGrid configurations, median final teacher-free success ranges from 90% to 100%. On DoorKey 5×5, the median number of environment steps to first reach 95% teacher-free success decreases from 64,000 to 6,000. Ablations show that persistent teacher-action takeover can impair independent performance, and that removing demonstration supervision slows learning on some tasks.

1 Introduction

Reinforcement learning (RL) learns sequential decision-making policies through interaction with the environment Sutton & Barto (1998). In these tasks, the consequences of an action may become apparent only after many transitions Pignatelli et al. (2023). When successful outcomes are observed mainly at the end of an episode, early policies rarely reach the goal, and many collected trajectories provide little information about which actions should be improved Andrychowicz et al. (2017). As the decision horizon grows, acquiring informative experience can require increasingly many environment interactions. Improving training efficiency under a limited interaction budget while learning a policy that can complete the task is a central challenge Uchendu et al. (2023).

To improve training efficiency, prior work has explored reward shaping Ng et al. (1999); Skalse et al. (2022), hierarchical reinforcement learning (HRL) Kulkarni et al. (2016), and imitation learning (IL) Mandlekar et al. (2021). Reward shaping guides exploration with denser feedback, but poorly designed shaping may change the optimal policy for the original task Skalse et al. (2022). HRL uses subgoals or temporal abstractions to simplify long-horizon decisions, yet some methods rely on task-specific hierarchies that are difficult to transfer Kulkarni et al. (2016). IL uses expert demonstrations to improve early policy learning, but offline approaches based on precollected demonstrations face substantial data acquisition costs, and their fixed datasets may not cover states visited by the learner Mandlekar et al. (2021). These limitations motivate us to investigate guidance that can be provided in response to the learner's current state and reused in subsequent policy updates, with the aim of substantially improving the training efficiency of RL under a limited interaction budget.

Large language models (LLMs) offer a potential source of such guidance Cao et al. (2024). Through large-scale training, they acquire linguistic capabilities and task-relevant priors Huang et al. (2022). Given a task description and a structured representation of the current state, an LLM can propose

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.

an action for a state the learner actually encounters, without requiring successful trajectories to be collected in advance for every possible situation Brohan et al. (2023). The model's prior knowledge can thus inform suggestions across different training states, while the resulting state–action pairs can be stored as demonstrations and reused in subsequent policy updates Zhou et al. (2023). However, the usefulness of these suggestions depends on the state representation, prompt design, and action feasibility; they cannot be assumed to constitute a reliable expert policy Sclar et al. (2024); Valmeekam et al. (2023). *The key challenge is to use this online guidance both to help the learner acquire informative experience and to develop a policy that can act independently without continued access to the LLM*. Deploying the language model itself as the controller is impractical: a model of this size has too many parameters to serve at every step, whereas the trained RL policy is small enough for lightweight deployment once guidance is withdrawn.

Prior work has explored teacher advice as a means of assisting RL. Action advising and the reuse of advice through imitation have been studied in RL Zhou et al. (2023). In the LLM-teacher setting, LLM4Teach Zhou et al. (2023) uses the model to select among pretrained high-level skills based on the current state, producing a soft teacher distribution over actions. RL executes its own actions in the environment and learns from this distribution through an auxiliary soft-policy distillation loss alongside the RL objective. The weight of this loss decreases according to a predefined training schedule until teacher supervision is withdrawn. Thus, emphasizing guidance early in training and reducing reliance on the teacher later are already established ideas. However, annealing the distillation weight primarily controls the teacher's influence on policy updates. It does not examine how executing teacher-recommended actions affects the trajectories collected during training, or whether demonstration-based supervision should continue after teacher action intervention is withdrawn.

This leads to the questions addressed in this paper. For an LLM teacher that provides action suggestions based on the current state, when should a suggested action be executed in the environment, and when should it serve only as a supervision label for the RL policy? As training progresses, should teacher control over actions, demonstration-based supervision, or both be withdrawn?

Our central insight is that teacher guidance operates through two distinct channels. First, teacher-action intervention changes the actions executed in the environment, thereby affecting subsequent state visitation and opportunities to receive feedback. Second, teacher actions serve as supervisory labels for policy updates, allowing the RL agent to gradually learn useful behavior. The former can help collect more informative trajectories early in training, but high success rates in teacher-assisted episodes do not establish that the student has mastered the task independently. The latter allows limited guidance to be reused across multiple updates. We adopt a competence-adaptive guidance schedule that adjusts the probability of teacher-action intervention based on policy uncertainty, teacher-free success, and student–teacher action agreement. It provides more assistance when teacher-free performance is weak and retains replay-based supervision as action control is reduced.

The main contributions of this paper are as follows:

(1) Distinguishing the roles of teacher guidance. We distinguish how LLM teacher guidance affects action execution and policy updates.

(2) Gradual handoff of action control. The probability of executing the teacher's action is recomputed from the student's uncertainty, its success in teacher-free evaluation, and its match to the teacher's actions. This adjusts LLM guidance for the RL agent throughout training.

(3) Evaluation across environments. We conduct experiments on three MiniGrid task families and on Env2. We assess training efficiency, component effects, and student performance without teacher assistance.

2 Related Work

LLM Guidance for RL. LLMs have been used to provide rewards and exploration goals for RL. Kwon *et al.* Kwon et al. (2023) and Ma *et al.* Ma et al. (2024) use an LLM for reward design, while Klissarov *et al.* Klissarov et al. (2024) derive intrinsic rewards from preferences over event descriptions. Du *et al.* Du et al. (2023) use an LLM to generate or relabel exploration goals. These methods primarily influence the feedback or goals available to the learner. Our work examines action

guidance for states the student encounters, allowing teacher suggestions to affect both collected trajectories and policy updates.

Teacher Guidance for Action and Policy Learning. Teacher agents can guide the actions taken during interaction or provide information for policy learning. Torrey and Taylor Torrey & Taylor (2013) study action advice under a limited budget, while Xue *et al.* Xue et al. (2023) study teacher intervention during data collection. Schmitt *et al.* Schmitt et al. (2018) combine policy distillation with RL, and Ilhan *et al.* Ilhan et al. (2021) reuse advice through an auxiliary behavior cloning model. LLM4Teach Zhou et al. (2023) trains a student on its own trajectories using a soft teacher policy and gradually reduces the distillation weight. We combine teacher action intervention with demonstration replay and examine the gradual withdrawal of action control while retaining supervision from stored demonstrations.

3 METHOD

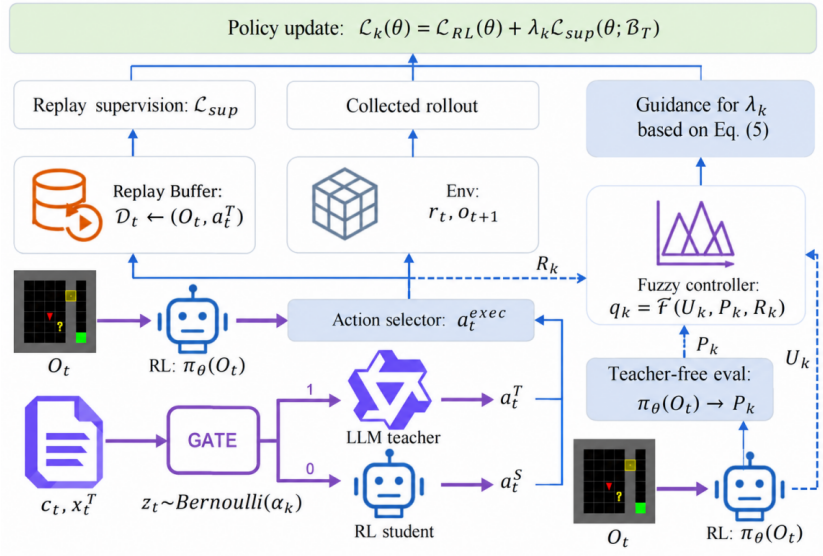


Figure 1: Overview of the proposed competence-adaptive LLM-guided RL framework. At each step, the RL student proposes an action, and a Bernoulli gate determines whether to request an action from the LLM teacher. The selected action is executed in the environment; observations and teacher actions from actual interventions are stored for replay supervision. The student policy is updated using the RL loss and a weighted supervision loss. A fuzzy controller adjusts the intervention probability and supervision weight based on policy uncertainty, teacher-free evaluation success, and student–teacher action agreement.

3.1 PROBLEM FORMULATION

We consider an episodic Markov decision process (MDP) Sutton & Barto (1998) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, p, r, \rho_0, \gamma, H)$. At step t, the RL agent, referred to as the student, receives an observation o_t derived from the environment state s_t and proposes an action $a_t^S \sim \pi_\theta(\cdot \mid o_t)$. Our goal is to maximize the student's expected return when it acts without LLM assistance, subject to a fixed training-interaction budget:

$$\begin{aligned} \max_{\theta} \quad & J_{\text{free}}(\theta) = \mathbb{E}_{\tau \sim p_{\pi_\theta}} \left[\sum_{t=0}^{H-1} \gamma^t r_t \right] \\ \text{s.t.} \quad & N_{\text{env}}^{\text{train}} \leq B, \end{aligned} \tag{1}$$

where B is the maximum number of environment interactions used for training, and trajectories in J_{free} are generated by the RL policy alone.

During training, a teacher module receives a task or phase instruction c_t and a structured state description x_t^T, and proposes a low-level action $a_t^T \in \mathcal{A}$. The LLM description may contain task-specific information beyond the student's observation. The module includes the LLM prompt, action parsing, and a fallback for unavailable or unparsable responses; its actions are not assumed to be optimal. We next describe LLM-guided action intervention during experience collection, replay-based supervision of the RL student, and competence-adaptive regulation of teacher intervention.

3.2 Action Intervention and Supervision Replay

LLM teacher guidance affects both experience collection and policy updates. At step t, the RL student proposes $a_t^S \sim \pi_\theta(\cdot \mid o_t)$. The executed action is determined by:

$$z_t \sim \text{Bernoulli}(\alpha_k), \quad a_t^{\text{exec}} = \begin{cases} a_t^T, & z_t = 1, \\ a_t^S, & z_t = 0. \end{cases} \tag{2}$$

Here, α_k is the intervention probability at controller update k. The teacher is queried only when $z_t = 1$, and in that case the teacher's action is executed. Otherwise, the RL student's action is executed. Because the executed action determines the next transition, intervention changes the states and rewards encountered in subsequent experience collection. For each step with $z_t = 1$, we store (o_t, a_t^T) in a fixed-capacity replay buffer $\mathcal{D}_T$. At each policy update, we sample a minibatch $\mathcal{B}_T$ uniformly from the buffer and define the demonstration-supervision loss as:

$$\mathcal{L}_{\text{sup}}(\theta; \mathcal{B}_T) = -\frac{1}{|\mathcal{B}_T|} \sum_{(o, a^T) \in \mathcal{B}_T} \log \pi_\theta(a^T \mid o). \tag{3}$$

The buffer contains only actions that the teacher actually executed, and its stored pairs can continue to supervise the RL student after action intervention stops. In Eq. (3), $\pi_\theta(a^T \mid o)$ denotes an action probability in MiniGrid and an action probability density in the continuous-control task.

3.3 Competence-Adaptive Regulation of LLM Guidance

We adjust LLM guidance according to the student's current behavior. At controller update k, we measure policy uncertainty U_k, teacher-free success P_k, and action agreement with the teacher R_k:

$$\begin{aligned} U_k &= \mathcal{N}(\bar{h}_k), \qquad P_k = \tfrac{1}{M} \textstyle\sum_{j=1}^{M} y_j^{\text{free}}, \\ R_k &= \tfrac{1}{|\mathcal{I}_k|} \textstyle\sum_{t \in \mathcal{I}_k} \text{agr}(a_t^S, a_t^T), \end{aligned} \tag{4}$$

where $\bar{h}_k$ is the mean entropy of the student's action distribution on the final observation batch of the latest rollout, and $\mathcal{N}$ normalizes it to $[0, 1]$. The variable M is the number of teacher-free evaluation episodes, and $y_j^{\text{free}} \in \{0, 1\}$ records success in episode j. The set $\mathcal{I}_k$ contains the most recent batch of steps at which an LLM teacher action was executed. For each such step, a_t^S is the student's proposed action and a_t^T is the teacher's action. The function agr measures exact action matching in discrete environments and rescaled cosine similarity in continuous environments. The controller retains the latest P_k between evaluations and the latest R_k when no new action pairs are available. A fixed, rule-based Mamdani fuzzy system combines these signals into an intervention level $q_k \in [0, 1]$ Lin & Lee (1994). We then set:

$$\begin{aligned} q_k &= \mathcal{F}(U_k, P_k, R_k), \qquad \alpha_k = \alpha_{\max} q_k, \\ \lambda_k &= \lambda_{\text{base}} + (1 - \lambda_{\text{base}}) \alpha_k (1 - R_k). \end{aligned} \tag{5}$$

The controller combines triangular membership functions and the fixed rules in Table 1 through min–max inference and centroid defuzzification. We set $\alpha_{\max} = 0.1$ (Table 5). Because q_k can equal one, this cap is what keeps intervention sparse: at most one environment step in ten executes the teacher action. Most transitions therefore stay on the student policy, and LLM queries stay bounded. The opposite choice, executing the teacher at every step, is the $\alpha = 1$ row in Section 4.3; it lowers teacher-free success. In the intervention setting, α_k specifies the probability of invoking the LLM-based teacher interface. $\lambda_{\text{base}} > 0$ denotes the base supervision weight, which maintains replay-based policy supervision from stored teacher action labels as the intervention probability

Table 1: Fixed Mamdani rule base for LLM teacher intervention. A dash indicates that the rule does not depend on that input. The consequent denotes a fuzzy set rather than an exact value of q_k.

Rule	U_k	P_k	R_k	Consequent
1	–	Low	–	High
2	High	Medium	–	Medium
3	–	Medium	Low	Medium
4	–	Medium	High	Low
5	–	High	Low	Low
6	–	High	High	Zero
7	Low	High	–	Zero
8	Low	Low	–	Medium

declines. If the interface returns an executable action, that action replaces the RL student's proposed action. The coefficient λ_k weights supervision on teacher-action pairs sampled from the replay buffer. When the buffer is nonempty, the RL student minimizes:

$$\mathcal{L}_k(\theta) = \mathcal{L}_{\mathrm{RL}}(\theta) + \lambda_k \mathcal{L}_{\mathrm{sup}}(\theta; \mathcal{B}_T), \tag{6}$$

where $\mathcal{L}_{\mathrm{RL}}$ denotes the RL training loss, including applicable auxiliary terms. At each rollout boundary, the controller updates U_k and updates R_k if the latest policy batch contains intervened actions; otherwise, it retains the previous value of R_k. A new teacher-free evaluation updates P_k. The controller recomputes α_k from the available signals. The intervention probability follows that update for the rest of training, declining or rising with the measured signals, while replay supervision continues.

3.4 Training Procedure and Intervention-Free Evaluation

Training alternates between rollout collection and updates of the RL student. During collection, a_t^{exec} in Eq. (2) generates the transition used for RL training. Only executed LLM teacher actions contribute pairs (o_t, a_t^T) to $\mathcal{D}_T$. After each rollout, the controller updates the available signals and recomputes α_k and λ_k according to Eq. (5). The student then updates θ using the rollout and, when $\mathcal{D}_T$ is nonempty, a minibatch $\mathcal{B}_T$ as specified in Eq. (6).

To assess the student's independent action selection, we disable LLM action intervention and evaluate π_{θ_k} over N_{eval} episodes:

$$\begin{aligned} \tau_i^{(k)} &\sim p_{\pi_{\theta_k}} \quad \text{with } z_t = 0, \\ P_k &= \frac{1}{N_{\mathrm{eval}}} \sum_{i=1}^{N_{\mathrm{eval}}} \mathrm{Succ}\left(\tau_i^{(k)}\right). \end{aligned} \tag{7}$$

Here, $\mathrm{Succ}(\tau)$ equals one if the episode satisfies the task's success criterion and zero otherwise. The measured P_k is retained until the next evaluation and used in subsequent controller updates.

4 Experiment

4.1 Experimental setup

Research questions. Our experiments address four questions.

RQ1: Overall effectiveness. How much does the proposed method improve sample efficiency and teacher-free performance compared with classical RL Schulman et al. (2017) and a teacher-supervision baseline Zhou et al. (2023)?

RQ2: Contributions of the guidance mechanisms. What are the respective effects of teacher-action intervention, replay supervision, and competence-adaptive regulation on student learning?

RQ3: Sensitivity to rewards and teacher models. How do the magnitude of process rewards and the choice of LLM teacher affect the resulting student policy?

RQ4: Effect of task difficulty. How do the gains in training efficiency brought by LLM change?

Table 2: Episode length, return, final teacher-free success, and steps to 75% success on Env2 and six MiniGrid tasks. Bold marks the most favorable median within each task.

Task	Method	Ep. length ↓	Return ↑	Success ↑	T_{75} (k) ↓
Env2	Ours	**1216.7 [1186.6, 1526.4]**	214.9 [135.6, 348.0]	0.90 [0.70, 1.00]	**25 [25, 30]**
Env2	No teacher	1527.0 [1133.9, 1972.3]	203.5 [−174.0, 329.9]	0.65 [0.05, 0.90]	3113 [1263, 8310]†
Env2	LLM4Teach-style	1437.5 [1085.7, 2000]	**272.8 [−86.1, 332.6]**	**0.95 [0.00, 1.00]**	1078 [943, > 2507]†
DoorKey 5 × 5	Ours	90.5 [46.1, 115.9]	0.67 [0.58, 0.83]	0.95 [0.85, 1.00]	**5 [3, 8]**
DoorKey 5 × 5	No teacher	**64.8 [29.7, 103.4]**	**0.764 [0.618, 0.893]**	0.95 [0.90, 1.00]	64 [6, 298]
DoorKey 5 × 5	LLM4Teach-style	80.6 [56.2, 84.3]	0.710 [0.697, 0.798]	**1.00**	24.1 [21.4, 33.7]†
DoorKey 6 × 6	Ours	92.4 [88.3, 93.4]	0.764 [0.762, 0.779]	**1.00 [1.00, 1.00]**	**10 [10, 15]**
DoorKey 6 × 6	No teacher	**75.2 [20.3, 360]**	**0.807 [0.000, 0.949]**	0.95 [0.00, 1.00]	3110 / 1210 / No@5M
DoorKey 6 × 6	LLM4Teach-style	77.1 [72.2, 95.4]	**0.807 [0.762, 0.820]**	**1.00 [1.00, 1.00]**	15 [15, 15]
DoorKey 8 × 8	Ours	**149.3 [143.3, 160.0]**	**0.790 [0.770, 0.799]**	**1.00 [0.95, 1.00]**	**20 [20, 25]**
DoorKey 8 × 8	No teacher	640 [640, 640]	0.00 [0.00, 0.00]	0.00	No@5M
DoorKey 8 × 8	LLM4Teach-style	640 [19.3, 640]	0.00 [0.00, 0.756]	0.00 [0.00, 1.00]	No@500k / No@500k / 20
DoorKey 16 × 16	Ours	**1155.5 [600.6, 1540.9]**	**0.58 [0.43, 0.79]**	**0.90 [0.75, 1.00]**	**75 [75, 95]**
DoorKey 16 × 16	No teacher	2560 [2524.5, 2560]	0.000 [0.000, 0.017]	0.00 [0.00, 0.05]	No@5M
DoorKey 16 × 16	LLM4Teach-style	2560 [2560, 2560]	0.00 [0.00, 0.00]	0.00 [0.00, 0.00]	No@500k
UnlockPickup	Ours	**166.4 [86.2, 192.7]**	**0.588 [0.353, 0.731]**	**0.90 [0.55, 1.00]**	**66 [32, 100]**
UnlockPickup	No teacher	288 [279.5, 288]	0.000 [0.000, 0.032]	0.00 [0.00, 0.05]	No@0.5--4.8M
UnlockPickup	LLM4Teach-style	288 [288, 288]	0.00 [0.00, 0.00]	0.00 [0.00, 0.00]	No@500k
MultiRoom-N4-S5	Ours	**28.2 [27.0, 37.3]**	**0.678 [0.570, 0.696]**	**0.95 [0.90, 1.00]**	**45 [41, 46]**
MultiRoom-N4-S5	No teacher	80 [80, 80]	0.00 [0.00, 0.00]	0.00 [0.00, 0.00]	No@5M
MultiRoom-N4-S5	LLM4Teach-style	80 [80, 80]	0.00 [0.00, 0.00]	0.00 [0.00, 0.00]	No@500k

Values are median $[\min, \max]$ across seeds unless individual runs are listed. Episode length and return use 20-episode stochastic rollouts. No@B means the threshold was not reached by budget B; † marks a training-window threshold. The two clocks share this column because a training-window threshold and the teacher-free evaluation are extremely close.

Experimental setup. We train the RL student with PPO Schulman et al. (2017) and use Qwen3-14B Qwen Team (2025) as the LLM teacher in the main experiments. Teacher actions are executed according to the intervention probability defined in Section 3, with the intervention probability capped at 0.1. Actions from actual interventions are retained for replay supervision. At periodic checkpoints, we evaluate the student over 20 episodes without teacher-action intervention. Our primary measure of sample efficiency is the number of environment steps required to first reach a specified teacher-free success rate. We report thresholds of 25%, 50%, 75%, 85%, and 95%, together with the final teacher-free success rate. Results are summarized across random seeds, and runs that do not reach a threshold within their task-specific budget are identified as such.

Environments. We evaluate our method on six MiniGrid Chevalier-Boisvert et al. (2018) configurations spanning three task families: DoorKey with grid sizes of 5×5, 6×6, 8×8, and 16×16; UnlockPickup; and MultiRoom-N4-S5. The DoorKey variants provide a controlled comparison across increasing grid sizes. We also consider Env2, a custom continuous-control attack–defense task formulated as a reach–avoid game,[1] in which a single RL policy jointly controls two attackers against two autonomous, high-speed defenders. We examine the effect of intermediate-reward scaling on selected MiniGrid configurations and on Env2, and compare different LLM teachers on Env2.

Comparison methods. We compare our method with teacher-free PPO and a supervision-only baseline inspired by LLM4Teach Zhou et al. (2023). The latter is implemented within our training framework: it uses teacher-action labels to supervise the student but does not execute teacher actions in the environment, and its supervision weight is gradually reduced to zero. We also evaluate variants that remove action intervention or replay supervision, or alter the intervention controller.

4.2 Overall effectiveness

We evaluate sample efficiency using the environment interactions required to reach a success threshold, and assess the final RL student without teacher assistance. Table 2 compares our method with RL without a teacher and a loss-only distillation baseline implemented on the same tasks. Episode length and return describe the saved policies, while threshold times measure learning speed.

[1] https://anonymous.4open.science/r/env2-simulator-C7FB/

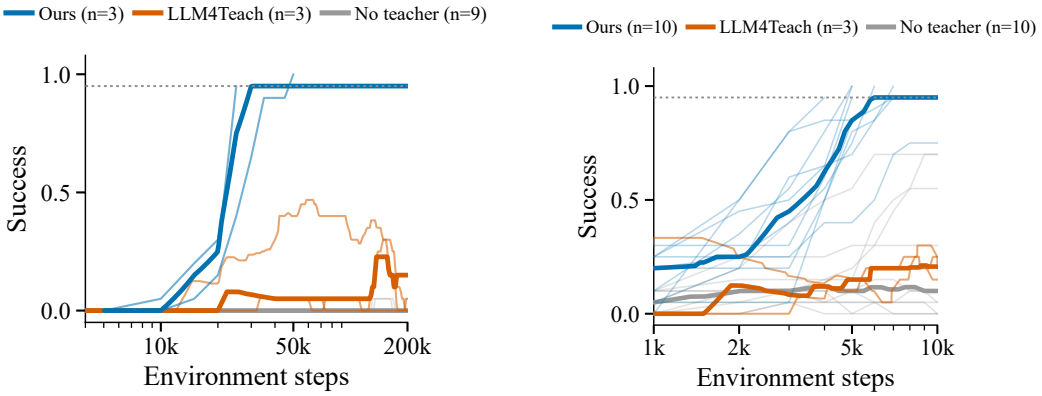


Figure 2: Teacher-free success during training on Env2 (left) and DoorKey 5×5 (right). Dark curves show smoothed median trends across runs, light curves show individual runs, and the dotted line marks 95% success. The curve labeled LLM4Teach represents our loss-only baseline on these tasks.

Across the six MiniGrid configurations, our method reaches 75% success after a median of 5k–75k steps and achieves final teacher-free success rates of 0.90–1.00. The improvement is visible at multiple stages of learning. On DoorKey 5×5 (Fig. 2, right), our student reaches 25%, 75%, and 95% success after 2k, 5k, and 6k steps, compared with 63k, 64k, and 64k steps for RL without a teacher. On this smallest grid the unassisted policy, which keeps training after the threshold, has a shorter median episode and a higher return at a similar final success; our runs stop at the first teacher-free evaluation that reaches 0.95. On DoorKey 8×8, DoorKey 16×16, and MultiRoom-N4-S5, our method reaches 75% after 20k, 75k, and 45k steps, respectively, whereas none of the no-teacher runs reaches this threshold within 5M steps. On UnlockPickup, our method reaches 75% after a median of 66k steps; none of the ten no-teacher runs reaches that threshold within its recorded budget.

The loss-only baseline examines whether guidance confined to policy updates yields comparable learning speed. We compare threshold times only where that run actually crosses the threshold, and we treat a recorded cutoff as censoring rather than as evidence that the baseline cannot learn the task. On DoorKey 5×5 and 6×6, both methods reach 75% inside the recorded window: 5k versus 24.1k steps on 5×5, and 10k versus 15k steps on 6×6. On DoorKey 8×8, all three of our runs reach 75% within 25k steps; one loss-only run reaches it at 20k steps, and the other two are still short of the threshold at the recorded 500k cutoff (Fig. 6). On DoorKey 16×16, UnlockPickup, and MultiRoom-N4-S5, every loss-only run is censored at 500k steps, so those entries do not support a claim that supervision alone fails. In Env2 (Fig. 2, left), our student reaches 75% and 95% after 25k and 30k steps, respectively, and achieves 0.90 final teacher-free success. The loss-only baseline has a higher median final success of 0.95 on this task, so our method does not lead on every final-policy metric. Moreover, its Env2 threshold times and those of the no-teacher baseline use a training-window statistic; we therefore do not calculate a direct speedup from those entries.

4.3 Contributions of the guidance mechanisms

We examine the roles of action intervention, demonstration supervision, and intervention regulation through ablations on Env2. Table 8 reports the environment steps required to reach teacher-free success thresholds and the final success of the RL student. The full method reaches 95% success after a median of 30k steps. Removing the supervised policy loss increases this time to 170k steps, while removing demonstration replay increases it to 70k steps. Without action intervention, none of the three runs reaches the threshold within 500k steps, and final teacher-free success is zero. Because the replay buffer stores samples only when teacher actions are executed, this ablation also prevents demonstration collection; it does not isolate the effect of action intervention alone. The same one-change rows on DoorKey are in Table 9. There the full method reaches 95% after a median of 11k steps. Always executing the student still reaches 95% (20k), and dropping the supervised loss reaches it at 12k. Setting $\alpha = 1$ leaves median teacher-free success at 0.40. On this DoorKey suite,

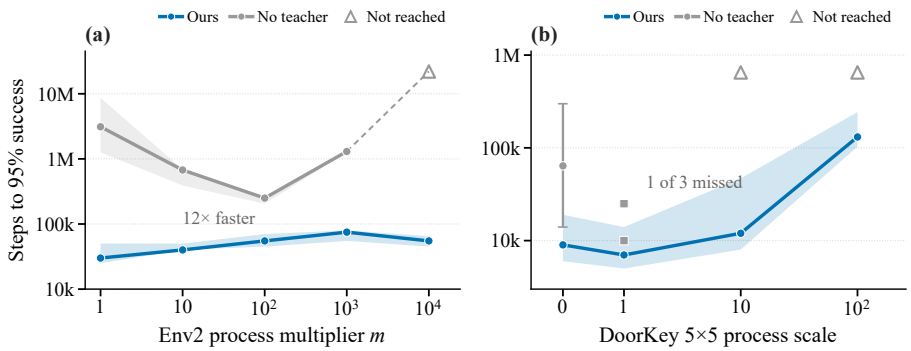


Figure 3: Effect of intermediate-reward scaling in Env2. Curves show median steps to 95% success, with shaded seed ranges. Our results use teacher-free evaluation; the no-teacher results use training-window success. The open triangle denotes runs that did not reach the threshold within 16.1–17.0 million steps.

Table 3: Effect of LLM teacher choice on Env2. Values are median $[\min, \max]$ over three runs; all student outcomes are evaluated without teacher assistance.

LLM teacher	T_{75} (k) ↓	T_{95} (k) ↓	Final success ↑
Qwen3-14B	25 [25, 30]	30 [25, 50]	0.90 [0.70, 1.00]
Qwen3-8B	30 [30, 35]	45 [30, 65]	0.90 [0.85, 0.95]
DeepSeek-V4.1-Flash	45 [40, 85]	70 [45, 85]	0.90 [0.85, 0.95]

intervention changes speed and full takeover still hurts, but neither channel is required for eventually crossing 95%.

The intervention schedule also affects independent learning. Reversing the adaptive rule increases the median time to 95% success from 30k to 80k steps. When the teacher executes every action ($\alpha = 1$), no run reaches 75% success within 100k steps, and final teacher-free success falls to 0.30. Figure 7 shows that the controller-set intervention probability generally decreases during training, although it may temporarily increase as the student's measured competence changes.

We first examine whether the result depends on the scale of the intermediate reward. In Env2, we vary its multiplier m over $\{1, 10, 10^2, 10^3, 10^4\}$ while keeping the guidance framework fixed. The median interactions required by our student to reach 95% teacher-free success are 30k, 40k, 55k, 75k, and 55k, respectively (Fig. 3). The student reaches the threshold under every tested scale, although reward magnitude affects learning speed. The no-teacher baseline also responds non-monotonically to reward scaling and does not reach 95% at $m = 10^4$ within its recorded budget. Its threshold times use a training-window success statistic, so we do not calculate a cross-method speedup from this comparison.

We next replace the LLM teacher while keeping the student-training framework fixed. As shown in Table 3, students trained with Qwen3-14B, Qwen3-8B, and DeepSeek-V4.1-Flash reach 75% teacher-free success after median times of 25k, 30k, and 45k steps, respectively. Their times to 95% are 30k, 45k, and 70k steps. All three attain a median final teacher-free success rate of 0.90. Thus, the framework works with each of the three tested teachers, while the speed of learning varies with teacher choice.

4.4 Learning under Increasing Task Difficulty

We use DoorKey grid size as a controlled proxy for task difficulty while retaining the same task objective. As the grid increases from 5×5 to 6×6, 8×8, and 16×16, the median interactions required by our method to reach 75% teacher-free success rise from 5k to 10k, 20k, and 75k steps, respectively (Fig. 4). The corresponding times to 95% are 6k, 15k, 30k, and 75k steps. Despite the increasing interaction cost, final median teacher-free success remains 0.95, 1.00, 1.00, and 0.90 across these sizes.

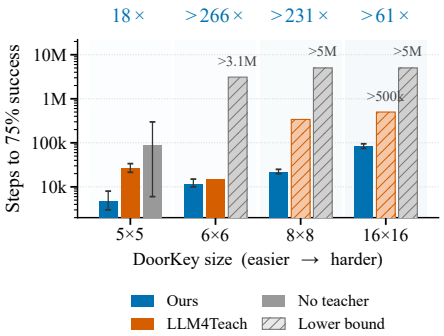


Figure 4: Interactions required to reach 75% teacher-free success as DoorKey grid size increases. Points show medians and whiskers show observed seed ranges. Hollow triangles and upward arrows mark no-teacher runs right-censored at 5M steps; their positions are training budgets, not observed threshold times.

Table 4: Env2 results under Gaussian observation noise. Training and teacher-free evaluation use the same noise level. T_{75} is median $[\min, \max]$ in thousands of steps over three runs; the last column reports runs reaching 95% within 500k steps.

Noise std. σ	T_{75} (k)	95% hits by 500k
0.05	55 [45, 60]	1/3 (at 400k)
0.10	40 [35, 60]	1/3 (at 430k)
0.30	75 [35, 100]	0/3
0.50	75 [40, 115]	0/3

RL without a teacher reaches 75% after a median of 64k steps on 5×5. On 6×6, two runs reach the threshold at 1.21M and 3.11M steps, while the third does not reach it within 5M steps. None of its runs reaches the threshold within 5M steps on 8×8 or 16×16. These results show that guidance remains effective as the grid becomes larger. Because several baseline runs are censored, they do not establish a precise or monotonically increasing multiplicative speedup.

We also train and evaluate students under Gaussian observation noise in Env2. Noise is added to the continuous geometric features of the observation; in these noisy settings, the LLM teacher constructs its guidance from the student's observation. As Table 4 shows, all three runs reach 75% teacher-free success at each tested nonzero noise level, with median threshold times between 40k and 75k steps. Reaching a stricter threshold is less reliable: only one of three runs reaches 95% within 500k steps at each of $\sigma = 0.05$ and 0.1, and none does so at $\sigma = 0.3$ or 0.5. The method therefore retains progress under noisy observations, but high-success performance remains sensitive to observation quality. The zero-noise main experiment uses a different, privileged teacher description, so its difference from these runs cannot be attributed to noise alone.

5 Conclusion

Our method reduces the steps needed for high teacher-free success. On DoorKey 5×5 it reaches 95% in 6k steps, against 64k for unassisted RL. On 8×8 and 16×16 it reaches 75% in 20k and 75k steps; unassisted RL does not within 5M. On Env2, executing every teacher action drops median teacher-free success from 0.90 to 0.30, and removing the supervised loss or the replay buffer raises the time to 95% from 30k to 170k or 70k. On DoorKey those same removals still cross 95%, so the two channels do not weigh the same on every task. The cap $\alpha_{\max} = 0.1$ keeps intervention sparse while stored demonstrations continue to supervise the student.

Limitations. The fuzzy rules and membership functions are fixed across tasks and are not learned. Uncertainty and agreement must be defined for the action space before training.

AI USE STATEMENT

We used generative AI tools to edit experimental code, iterate on design, interpret logs, suggest structure and title wording, revise text, and format references. We did not use them to fabricate results, citations, or proofs. All AI-assisted material was reviewed by the authors. Numbers come from recorded runs checked against the frozen tables. AI-generated code was kept only when it matched the intended protocol. The authors take responsibility for the final content.

REPRODUCIBILITY STATEMENT

The shared PPO student, $\alpha_{\max}$, λ_{base}, replay capacity, teacher-free evaluation schedule, and the 0.95 early-stop are frozen in Table 5 and Appendix A. No-teacher and language-teacher runs share that student; only the teacher-action source changes. The main teacher is Qwen3-14B. Membership functions, the eight Mamdani rules, and controller constants are in Section 3 and Appendix B. Dynamics and prompts are in Appendices C and J. Summaries are medians over seeds with the observed range; missed thresholds are marked. An anonymous code archive is included with the submission.

REFERENCES

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. *Advances in neural information processing systems*, 30, 2017.

Anthony Brohan, Yevgen Chebotar, Chelsea Finn, Karol Hausman, Alexander Herzog, Daniel Ho, Julian Ibarz, Alex Irpan, Eric Jang, Ryan Julian, et al. Do as i can, not as i say: Grounding language in robotic affordances. In *Conference on robot learning*, pp. 287–318. Pmlr, 2023.

Yuji Cao, Huan Zhao, Yuheng Cheng, Ting Shu, Yue Chen, Guolong Liu, Gaoqi Liang, Junhua Zhao, Jinyue Yan, and Yun Li. Survey on large language model-enhanced reinforcement learning: Concept, taxonomy, and methods. *IEEE Transactions on Neural Networks and Learning Systems*, 36(6):9737–9757, 2024.

Maxime Chevalier-Boisvert, Lucas Willems, and Suman Pal. Minimalistic gridworld environment for OpenAI gym. GitHub repository, 2018.

Yuqing Du, Olivia Watkins, Zihan Wang, Cédric Colas, Trevor Darrell, Pieter Abbeel, Abhishek Gupta, and Jacob Andreas. Guiding pretraining in reinforcement learning with large language models. In *International Conference on Machine Learning*, pp. 8657–8677. PMLR, 2023.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International conference on machine learning*, pp. 9118–9147. PMLR, 2022.

Ercument Ilhan, Jeremy Gow, and Diego Perez-Liebana. Action advising with advice imitation in deep reinforcement learning. *arXiv preprint arXiv:2104.08441*, 2021.

Martin Klissarov, Pierluca D'Oro, Shagun Sodhani, Roberta Raileanu, Pierre-Luc Bacon, Pascal Vincent, Amy Zhang, and Mikael Henaff. Motif: Intrinsic motivation from artificial intelligence feedback. In *International Conference on Learning Representations*, volume 2024, pp. 38888–38921, 2024.

Tejas D Kulkarni, Karthik Narasimhan, Ardavan Saeedi, and Josh Tenenbaum. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. *Advances in neural information processing systems*, 29, 2016.

Minae Kwon, Sang Michael Xie, Kalesha Bullard, and Dorsa Sadigh. Reward design with language models. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023.

Chin-Teng Lin and CS George Lee. Reinforcement structure/parameter learning for neural-network-based fuzzy logic control systems. *IEEE Transactions on Fuzzy Systems*, 2(1):46–63, 1994.

Yecheng Jason Ma, William Liang, Guanzhi Wang, De-An Huang, Osbert Bastani, Dinesh Jayaraman, Yuke Zhu, Jim Fan, et al. Eureka: Human-level reward design via coding large language models. In *International conference on learning Representations*, volume 2024, pp. 26516–26560, 2024.

Ebrahim H. Mamdani and Sedrak Assilian. An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies*, 7(1):1–13, 1975.

Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. *arXiv preprint arXiv:2108.03298*, 2021.

Andrew Y. Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the Sixteenth International Conference on Machine Learning (ICML)*, pp. 278–287, 1999.

Eduardo Pignatelli, Johan Ferret, Matthieu Geist, Thomas Mesnard, Hado van Hasselt, Olivier Pietquin, and Laura Toni. A survey of temporal credit assignment in deep reinforcement learning. *arXiv preprint arXiv:2312.01072*, 2023.

Qwen Team. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021.

Simon Schmitt, Jonathan J. Hudson, Augustin Zidek, Simon Osindero, Carl Doersch, Wojciech M. Czarnecki, Joel Z. Leibo, Heinrich Kuttler, Andrew Zisserman, Karen Simonyan, et al. Kickstarting deep reinforcement learning. *arXiv preprint arXiv:1803.03835*, 2018.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. In *arXiv preprint arXiv:1707.06347*, 2017.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. In *International Conference on Learning Representations*, volume 2024, pp. 25055–25083, 2024.

Joar Skalse, Nikolaus Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward gaming. *Advances in neural information processing systems*, 35:9460–9471, 2022.

Richard S Sutton and Andrew G Barto. The reinforcement learning problem. *Reinforcement learning: An introduction*, pp. 51–85, 1998.

Lisa Torrey and Matthew Taylor. Teaching on a budget: Agents advising agents in reinforcement learning. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pp. 1053–1060, 2013.

Ikechukwu Uchendu, Ted Xiao, Yao Lu, Banghua Zhu, Mengyuan Yan, Joséphine Simon, Matthew Bennice, Chuyuan Fu, Cong Ma, Jiantao Jiao, et al. Jump-start reinforcement learning. In *International Conference on Machine Learning*, pp. 34556–34583. PMLR, 2023.

Karthik Valmeekam, Matthew Marquez, Sarath Sreedharan, and Subbarao Kambhampati. On the planning abilities of large language models-a critical investigation. *Advances in neural information processing systems*, 36:75993–76005, 2023.

Zhenghai Xue, Zhenghao Peng, Quanyi Li, Zhihan Liu, and Bolei Zhou. Guarded policy optimization with imperfect online demonstrations. *arXiv preprint arXiv:2303.01728*, 2023.

Zihao Zhou, Bin Hu, Chenyang Zhao, Pu Zhang, and Bin Liu. Large language model as a policy teacher for training reinforcement learning agents. *arXiv preprint arXiv:2311.13373*, 2023.

A FROZEN STUDENT

The student is Stable-Baselines3 PPO Raffin et al. (2021); Schulman et al. (2017). Table 5 freezes the network, optimizer, rollout length, replay buffer, and the controller constants shared with Appendix B. On a domain, the no-teacher run and the language-teacher run use the same student. The language run changes only the source of a_t^T. Width and learning rate are not retuned per arm.

Vectorization uses two parallel environments. Language runs stop when a teacher-free evaluation first reaches 0.95. The same evaluation also records the first hit of 0.25, 0.50, 0.75, and 0.85 (Appendix D).

Table 5: Student and controller settings. No-teacher and language-teacher runs use the same student.

	DK 5×5	DK 6×6	DK 8×8	DK 16×16	Unlock	MultiRoom	Env2
Policy	CNN 16–32–64	same	same	same	same	same	MLP $[256, 256]$
Features / trunk	d=128, MLP $[128, 128]$	same	same	same	same	same	—
Obs / act	7×7 RGB / Disc.(7)	same	same	same	same	same	$\mathbb{R}^{77}$ / $[-1, 1]^2$
Horizon	250	360	640	2560	288	80	2000 (Δt=0.05)
n_{steps} / batch	128 / 64	same	same	same	same	same	2048 / 64
lr / γ / λ_{GAE}	$3{\times}10^{-4}$ / 0.99 / 0.95	same	same	same	same	same	same
clip / entropy / c_v	0.2 / 0.01 / 0.5	same	same	same	same	same	same
epochs / $\lvert\nabla\rvert$ clip	10 / 0.5	same	same	same	same	same	same
$\alpha_{\max}$	0.1	same	same	same	same	same	same
λ_{base} / $\lvert\mathcal{D}_T\rvert$	0.1 / 50k	same	same	same	same	same	0.5 / 100k
Teacher-free eval	every 1k, n=20	every 5k, n=20	every 5k, n=20	every 5k, n=20	every 1k, n=20	every 1k, n=20	every 5k, n=20
Stop on that eval	0.95	same	same	same	same	same	same
Process reward	—	—	—	—	—	—	ρ=0.3, m=1
Language model	qwen3:14b	same	same	same	same	same	same

B CONTROLLER CONSTANTS

Section 3 states the intervention rule, the supervision loss, and the eight Mamdani rules in Table 1 Mamdani & Assilian (1975); Lin & Lee (1994). This appendix freezes the membership functions, the defuzzification, and the numeric constants shared by every reported run on DoorKey, Unlock-Pickup, MultiRoom, and Env2. The student optimizer remains the per-domain PPO in Table 5.

Each controller input $x \in \{U_k, P_k, R_k\}$ is clipped to $[0, 1]$. A triangular membership with knots (a, b, c) is

$$\mu(x; a, b, c) = \begin{cases} 1, & a = b \text{ and } a \le x \le b, \\ 1, & b = c \text{ and } b \le x \le c, \\ \dfrac{x - a}{b - a}, & a < x < b, \\ \dfrac{c - x}{c - b}, & b < x < c, \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

The two unit cases are the flat shoulders: low is 1 on its left endpoint, and high is 1 on its right endpoint. Input knots are low $(0, 0, 1/2)$, medium $(0, 1/2, 1)$, and high $(1/2, 1, 1)$. Output knots on the centroid variable $z \in [0, 1]$ are zero $(0, 0, 1/3)$, low $(0, 1/3, 2/3)$, medium $(1/3, 2/3, 1)$, and high $(2/3, 1, 1)$.

Rule i fires at the minimum membership of the inputs it names. A dash in Table 1 omits that input from the minimum. Rules that share a consequent ℓ keep the strongest firing $\bar{w}_\ell$. On $N = 101$ uniform points $z_j = j/(N - 1)$,

$$\begin{aligned} \mu(z) &= \max_{\ell} \min\big(\bar{w}_\ell,\ \mu(z; a_\ell, b_\ell, c_\ell)\big), \\ q_k &= \frac{\sum_{j=0}^{N-1} z_j\, \mu(z_j)}{\sum_{j=0}^{N-1} \mu(z_j)}. \end{aligned} \tag{9}$$

If the denominator is zero, $q_k = 1/2$. The intervention probability is

$$\alpha_k = \alpha_{\max}\, q_k, \qquad \alpha_{\max} = 0.1. \tag{10}$$

Policy uncertainty is the normalized entropy $U_k = \mathcal{N}(\bar{h}_k)$. With $|\mathcal{A}|$ discrete actions,

$$\mathcal{N}(h) = \mathrm{clip}_{[0,1]}\left(\frac{h}{\log|\mathcal{A}|}\right). \tag{11}$$

On the Env2 action $a \in [-1,1]^2$ ($d = 2$), $\mathcal{N}$ rescales the differential entropy of an isotropic Gaussian between $\sigma_{\mathrm{low}} = 0.05$ and $\sigma_{\mathrm{high}} = 1$:

$$\begin{aligned} h_\sigma &= \tfrac{1}{2}d\big(1 + \log(2\pi)\big) + d\log\sigma, \\ \mathcal{N}(h) &= \mathrm{clip}_{[0,1]}\left(\frac{h - h_{\sigma_{\mathrm{low}}}}{h_{\sigma_{\mathrm{high}}} - h_{\sigma_{\mathrm{low}}}}\right). \end{aligned} \tag{12}$$

Agreement R_k averages $\mathrm{agr}(a_t^S, a_t^T)$ over the latest batch of steps at which a teacher action was executed:

$$\mathrm{agr}(a^S, a^T) = \begin{cases} \mathbf{1}[a^S = a^T], & \text{discrete,} \\ \mathrm{clip}_{[0,1]}\left(\dfrac{c+1}{2}\right), & \text{continuous,} \end{cases} \tag{13}$$

where $c = \langle a^S, a^T\rangle / (\|a^S\|\,\|a^T\|)$ when both norms are positive. A vanishing norm leaves the previous R_k unchanged.

The supervision weight is Eq. (5). With the frozen base weight it reads

$$\lambda_k = \lambda_{\mathrm{base}} + (1 - \lambda_{\mathrm{base}})\,\alpha_k(1 - R_k), \tag{14}$$

where $\lambda_{\mathrm{base}} = 0.1$ on the MiniGrid image tasks and $\lambda_{\mathrm{base}} = 0.5$ on Env2 (Table 5). The endpoints are $\lambda_k = \lambda_{\mathrm{base}}$ when $\alpha_k = 0$ or $R_k = 1$, and $\lambda_k = 1$ when $\alpha_k = 1$ and $R_k = 0$. The teacher is queried only when $z_t = 1$ in Eq. (2). An unparsable reply is replaced by a navigation fallback: a geometry heuristic on Env2, and a wall-avoiding rule with no action index on MiniGrid.

Action intervention stays on the fuzzy schedule for the whole run: α_k declines or rises with the measured signals, and stored pairs continue to supervise the student. Training stopped when teacher-free success first reached 0.95 (Appendix D).

C Environments

The student observation and the success event differ by domain. The controller in Appendix B is shared.

Env2. State $s \in \mathbb{R}^{77}$, action $a \in [-1,1]^2$, $\Delta t = 0.05$, horizon 2000 steps. Two missiles, two interceptors, one fixed target. A target hit pays $+300$. Interception of both missiles pays -220, a timeout pays -80, and a missile–missile collision pays -320.

The simulator also pays a per-step process term ϕ: closing speed on the target, time-to-go versus time-to-intercept, line-of-sight rate, and a penalty if the two missiles close on each other. In the reported runs that term is scaled by ρm and added to the terminal reward, with $\rho = 0.3$ and $m = 1$:

$$r_t = r_t^{\mathrm{term}} + \rho m\,\phi_t. \tag{15}$$

DoorKey. Official `MiniGrid-DoorKey-5x5-v0` Chevalier-Boisvert et al. (2018). The agent must pick up a key, open the door, and reach the green goal. Horizon 250. The student sees a 7×7 egocentric RGB image and uses the CNN in Table 5.

UnlockPickup. Official `MiniGrid-UnlockPickup-v0`. The agent must pick up the key, unlock the door, drop the key on empty floor beside the box, and pick up the box. Horizon $8 \cdot 6^2 = 288$. The student is the DoorKey CNN on the same 7×7 egocentric image.

MultiRoom. Official `MiniGrid-MultiRoom-N4-S5-v1`, four rooms. The legacy `v0` identifier is a six-room layout and is not used. The agent opens each closed door without a key, then walks to the green goal. Horizon $\text{maxNumRooms} \cdot 20 = 80$. The student is again the DoorKey CNN.

DoorKey 6×6, 8×8, and 16×16 use the DoorKey 5×5 student and the same controller. Horizons are 360, 640, and 2560. Teacher-free evaluation on those three maps is every 5k steps.

Seeds follow the frozen ledger: $n = 10$ on DoorKey 5×5 and UnlockPickup, and $n = 3$ on DoorKey 6×6, 8×8, 16×16, MultiRoom, and the Env2 language runs. Seeds that never reach a bar stay in the reported range.

An anonymous code archive with training and evaluation entry points is included with the submission.

D EVALUATION PROTOCOL

Teacher-free success is the fraction of evaluation episodes in which the student selects every action ($z_t = 0$ in Eq. (7)). Each checkpoint rolls out $n_{\text{eval}} = 20$ stochastic episodes. Checkpoints are every 1k environment steps on DoorKey, UnlockPickup, and MultiRoom, and every 5k steps on Env2. On DoorKey 6×6, 8×8, and 16×16 the checkpoint spacing is 5k steps. The run stops when this fraction first reaches 0.95. The same curve records the first checkpoint at which success reaches 0.25, 0.50, 0.75, 0.85, and 0.95.

E EXPERIMENTAL RESULTS: OVERALL EFFECTIVENESS

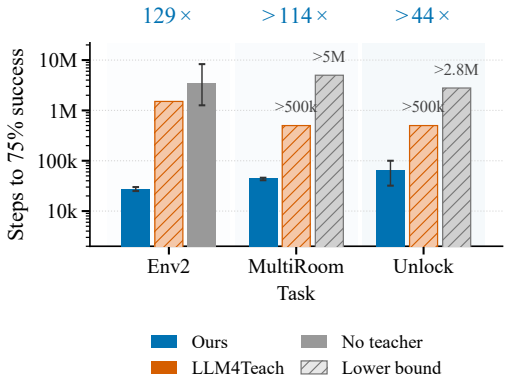
Table 2 in the main text compares episode length, return, final success, and time to 75%. The tables here add the full threshold record and a comparison against the teacher acting alone. Table 7 is that comparison on Env2. The student column is the trained policy evaluated with the teacher removed. The 14B column has no reinforcement-learning student: Qwen3-14B is queried at every simulator step and its action is executed immediately. The trained student reaches median teacher-free success 0.90 (range 0.70–1.00), while the same 14B model acting alone succeeds on 7 of 20 seeds (0.35). The 13 misses are timeouts, and none is an intercept. Sparse teacher intervention plus policy learning therefore beats handing the episode to the language model. The fuzzy-shell record uses $\alpha_{\max}$=0.1 and stops when teacher-free evaluation hits 0.95. Steps are thousands of environment steps. `no@`k means that threshold was not reached by budget k.

Table 6: Steps, in thousands, until teacher-free success reaches each rate (median $[\min, \max]$).

Environment	Arm	to25	to50	to75	to85	to95
Env2	Baseline	2328 [1080, 7409]	2883 [1188, 7637]	3113 [1263, 8310]	3114 [1265, 8382]	3114 [1265, 8665]
Env2	Ours	20 [20, 25]	25 [25, 30]	25 [25, 30]	30 [25, 30]	30 [25, 50]
DoorKey 5×5	Baseline	63 [3, 294]	63 [4, 297]	64 [6, 298]	64 [11, 299]	64 [14, 299]
DoorKey 5×5	Ours	2 [1, 4]	4 [2, 4]	5 [3, 8]	5 [4, 12]	6 [4, 13]
DoorKey 6×6	Baseline	850 [140, 2010]	2870 / 1180 / no@5M	3110 / 1210 / no@5M	3155 / 1210 / no@5M	3155 / 1225 / no@5M
DoorKey 6×6	Ours	10 [5, 10]	10 [5, 10]	10 [10, 15]	15 [10, 15]	15 [10, 15]
DoorKey 8×8	Baseline	2500 [1155, 3860]	no@5M / no@5M / no@5M	no@5M / no@5M / no@5M	no@5M / no@5M / no@5M	no@5M / no@5M / no@5M
DoorKey 8×8	Ours	15 [15, 20]	20 [20, 20]	20 [20, 25]	20 [20, 25]	30 [25, 30]
DoorKey 16×16	Baseline	no@5M	no@5M	no@5M	no@5M	no@5M
DoorKey 16×16	Ours	30 [30, 65]	70 [40, 95]	75 [75, 95]	75 [75, 115]	75 [75, 150]
UnlockPickup	Baseline	no@4700 / no@4800 / no@3M / no@3M / no@3M / no@2900 / no@3M / no@2800 / no@500 / no@500				
UnlockPickup	Ours	30 [21, 65]	46 [32, 66]	66 [32, 100]	73 [40, 100]	62 / no@100 / 45 / 68 / 84 / 72 / 75 / 92 / 77 / 80
MultiRoom-N4-S5	Baseline	no@5M	no@5M	no@5M	no@5M	no@5M
MultiRoom-N4-S5	Ours	32 [24, 33]	40 [31, 42]	45 [41, 46]	45 [41, 55]	55 [55, 55]

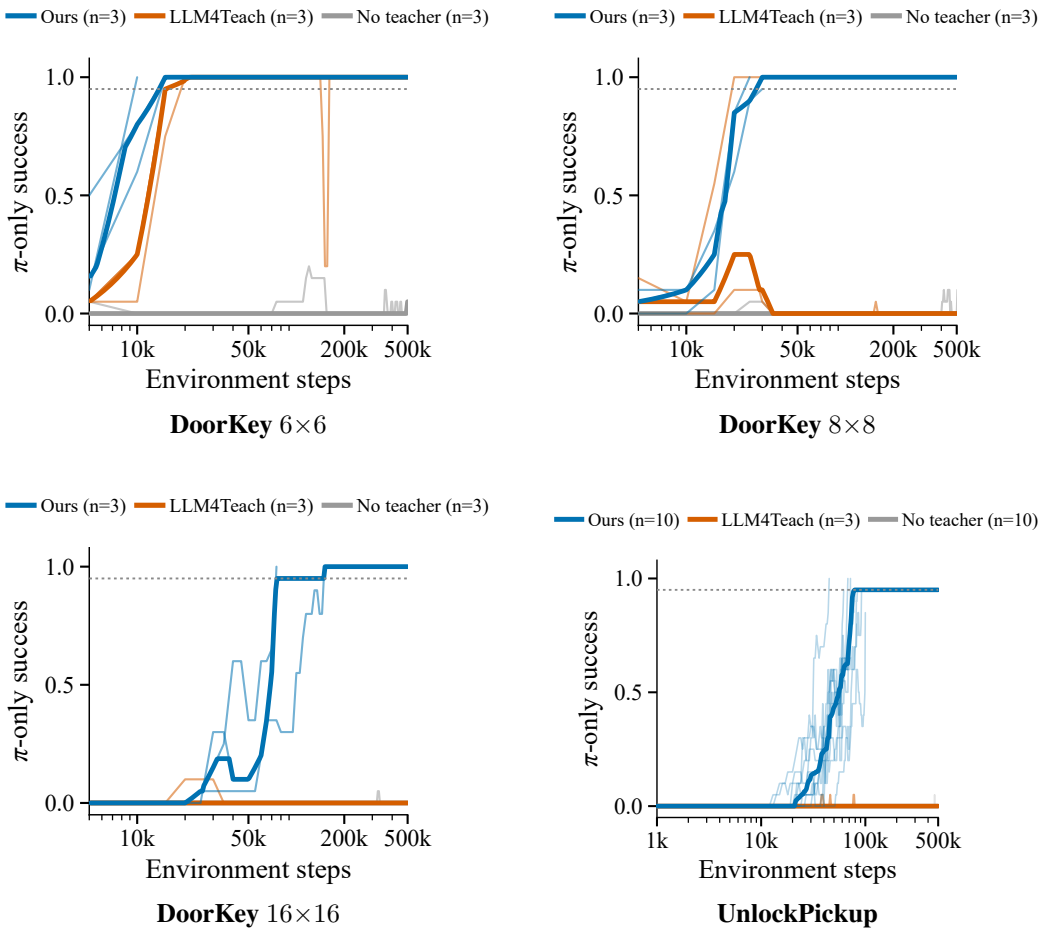
Table 7: Env2: trained student with the teacher removed, versus Qwen3-14B acting at every step with no reinforcement-learning student.

Environment	Student eval	14B every step
Env2	0.90 [0.70, 1.00]	0.35 (7/20)



Other environments

Figure 5: Steps until teacher-free success reaches 75%. The number above each group is no-teacher steps divided by ours. Hatched bars are lower bounds: at least one seed never reached 75%.



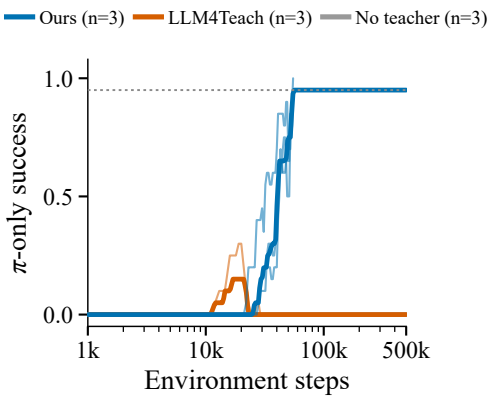


Figure 6: Success during training. Faint lines are seeds; the heavy line is the median. The dotted line is 0.95. On DoorKey 16×16, UnlockPickup, and MultiRoom, the orange LLM4Teach curve and the gray no-teacher curve both stay near zero, so they overlap. Env2 and DoorKey 5×5 are in Figure 2.

F Experimental results: guidance mechanisms

The main text discusses Env2 ablations in prose. Tables 8 and 9 each change one part of the same guidance loop. The first column names that change.

Full method leaves the loop as in Section 3: a fuzzy controller sets the intervention probability α_k, the teacher action replaces the student's action with that probability, executed teacher actions are stored, and a supervised loss on those stored pairs is weighted by λ_k.

Flip the fuzzy rules keeps the same inputs and the same membership functions, but swaps each rule's output level for its opposite, so a situation that would have raised intervention now lowers it. Teacher acts at every step fixes $\alpha = 1$, so the teacher's action is the action sent to the environment on every step. Hold α at its start keeps that probability at the value it takes on the first controller update, 0.089, and does not let later success, uncertainty, or agreement move it. Student action is always executed turns the takeover off: the environment always receives the student's action. The replay buffer only stores a pair when a teacher action is actually executed, so this row also collects no demonstrations. Drop the supervised loss still lets the teacher take over and still stores those actions, but sets the weight on $\mathcal{L}_{\text{sup}}$ to zero, so the stored pairs do not enter the policy update. Do not store teacher actions still lets the teacher take over, but does not keep (o_t, a_t^T) for later minibatches, so there is nothing to supervise with. Hold λ at its base value still stores demonstrations and still adapts α_k, but stops the formula in Eq. (5) from raising λ_k when agreement is low; the supervised loss stays at λ_{base}.

One further table uses an earlier shell: Ours on Env2, $\alpha_{\max}$=0.5, stop at 0.75. It compares letting the supervision weight decay all the way to 0 with holding a floor of 0.5, so demonstration replay does not disappear.

Table 8: Env2: each row changes one part of guidance. The fixed-opening row holds the intervention probability at its starting value 0.089. Steps are in thousands until teacher-free success reaches each rate.

What changes	to25	to50	to75	to85	to95	Eval
Full method	20 [20, 25]	25 [25, 30]	25 [25, 30]	30 [25, 30]	30 [25, 50]	0.90 [0.70, 1.00]
Flip the fuzzy rules	20 [10, 35]	25 [25, 40]	30 [30, 40]	30 [30, 70]	80 [35, 85]	0.90 [0.85, 0.90]
Teacher acts at every step	35 [25, 50]	95 / no@100 / 80	no@100 / no@100 / no@100	no@100 / no@100 / no@100	no@100 / no@100 / no@100	0.30 [0.25, 0.45]
Hold α at its start, 0.089	25 [15, 25]	30 [25, 30]	30 [30, 40]	35 [30, 40]	40 [35, 45]	0.95 [0.90, 1.00]
Student action is always executed	no@500	no@500	no@500	no@500	no@500	0.00 [0.00, 0.00]
Drop the supervised loss	50 [35, 110]	60 [55, 120]	140 [65, 155]	170 [65, 200]	170 [135, 210]	0.85 [0.85, 0.95]
Do not store teacher actions	15 [10, 25]	25 [25, 30]	40 [25, 65]	40 [35, 65]	70 [35, 120]	0.85 [0.80, 1.00]
Hold λ at its base value	20 [15, 25]	30 [25, 30]	30 [25, 45]	40 [30, 45]	40 [30, 50]	0.95 [0.95, 0.95]

Table 9: DoorKey: the same one-change rows as Table 8. Steps are in thousands.

What changes	to25	to50	to75	to85	to95	Eval
Full method	5 [2, 5]	8 [5, 8]	9 [8, 10]	9 [8, 10]	11 [9, 15]	0.95 [0.85, 1.00]
Flip the fuzzy rules	4 [3, 22]	10 [6, 23]	13 [13, 83]	15 [14, 83]	15 [15, 83]	1.00 [1.00, 1.00]
Teacher acts at every step	22 [15, 23]	31 [27, 44]	58 / no@100 / no@100	no@100	no@100	0.40 [0.40, 0.45]
Hold α at its start, 0.089	4 [4, 5]	6 [5, 7]	8 [7, 10]	9 [8, 10]	11 [10, 12]	0.95 [0.90, 0.95]
Student action is always executed	6 [2, 6]	18 [6, 25]	20 [13, 28]	20 [13, 28]	20 [15, 30]	1.00 [0.95, 1.00]
Drop the supervised loss	7 [4, 10]	10 [4, 11]	10 [4, 15]	12 [4, 15]	12 [6, 16]	0.95 [0.85, 1.00]
Do not store teacher actions	5 [3, 6]	6 [4, 6]	6 [6, 13]	9 [6, 15]	9 [8, 15]	0.95 [0.65, 1.00]
Hold λ at its base value	5 [4, 6]	6 [5, 7]	7 [5, 8]	7 [5, 8]	8 [5, 11]	0.95 [0.90, 1.00]

Table 10: Env2 supervision weight λ, decayed to 0 or kept at 0.5 so demonstration replay continues. This setting uses $\alpha_{\max}$=0.5 and stops at teacher-free success 0.75.

Schedule	to75	Eval
λ 1→0	65 [60, 90]	0.75 [0.65, 0.85]
λ 1→0.5	30 [30, 30]	0.90 [0.65, 1.00]

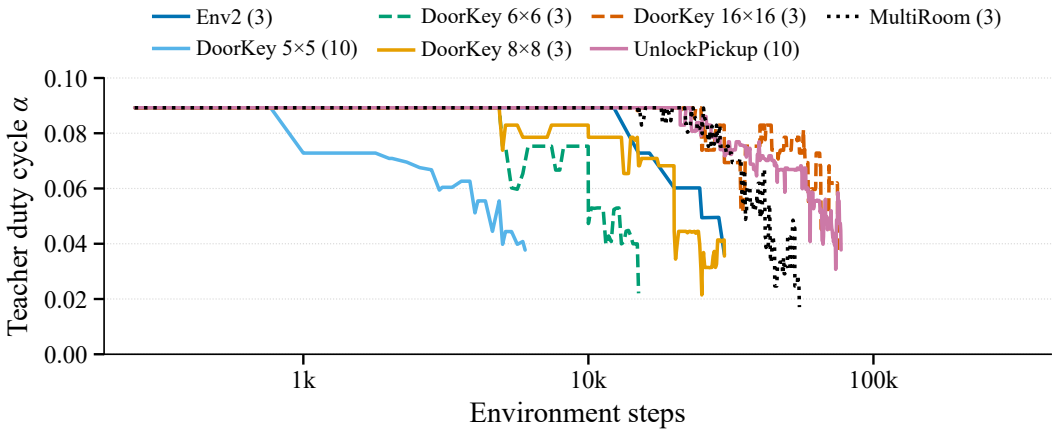


Figure 7: Intervention probability α during training (median across seeds). It starts at 0.089 and is capped at 0.1. A line ends once fewer than half the seeds are still training.

G Experimental results: reward scaling

Figure 3 in the main text shows Env2. The tables here vary the process-term multiplier on Env2, DoorKey, and MultiRoom. Steps are thousands of environment steps. `no@`k means that threshold was not reached by budget k.

Table 11: Env2 process-term multiplier m. Steps in thousands (median [min, max]). $\times$ is the speedup over the baseline.

m	Ours to25	Ours to50	Ours to75	Ours to85	Ours to95	Ours $\times$	Baseline to25	Baseline to50	Baseline to75	Baseline to85	Baseline to95
1	20 [20, 25]	25 [25, 30]	25 [25, 30]	30 [25, 30]	30 [25, 50]	104×	2328 [1080, 7409]	2883 [1188, 7637]	3113 [1263, 8310]	3114 [1265, 8382]	3114 [1265, 8665]
10	25 [15, 25]	25 [20, 30]	25 [25, 35]	30 [25, 40]	40 [40, 50]	—	300 [235, 325]	510 [340, 525]	540 [370, 570]	630 [385, 675]	675 [390, 690]
100	30 [20, 30]	35 [30, 40]	40 [30, 50]	50 [40, 65]	55 [45, 70]	—	110 [70, 110]	120 [95, 125]	165 [160, 165]	170 [165, 175]	250 [205, 255]
1000	25 [20, 35]	30 [25, 45]	40 [30, 60]	60 [40, 70]	75 [55, 80]	—	300 [245, 555]	590 [355, 755]	820 [615, 1075]	895 [845, 1230]	1300 [1265, 1350]
10000	30 [25, 35]	40 [40, 50]	45 [40, 55]	45 [40, 55]	55 [45, 65]	—	4345 [2785, 5890]	15000 / 13265 / no@16100	no@16500 / no@17M / no@16100	no@16500 / no@17M / no@16100	no@16500 / no@17M / no@16100

Table 12: Env2 when the process-term multiplier m changes during training. Steps in thousands.

Arm	Schedule	to25	to50	to75	to85	to95	Eval
m1	m=1 throughout	20 [20, 25]	25 [25, 30]	25 [25, 30]	30 [25, 30]	30 [25, 50]	0.90 [0.70, 1.00]
full50	m=50 throughout	25 [25, 25]	30 [25, 35]	40 [35, 40]	40 [40, 50]	50 [40, 50]	0.95 [0.90, 0.95]
at20k_1to50	first 20k at m=1, then m=50	25 [15, 25]	30 [25, 35]	40 [30, 45]	45 [40, 45]	50 [45, 50]	1.00 [0.95, 1.00]
at20k_50to1	first 20k at m=50, then m=1	25 [15, 25]	30 [25, 30]	35 [35, 35]	35 [35, 50]	60 [40, 65]	0.95 [0.90, 1.00]

Table 13: DoorKey 5×5: scale of the process term. Steps in thousands.

Scale	Ours to25	Ours to50	Ours to75	Ours to85	Ours to95	Baseline to25	Baseline to50	Baseline to75	Baseline to85	Baseline to95
0	5 [3, 9]	7 [4, 11]	9 [5, 17]	9 [6, 19]	9 [6, 19]	63 [3, 294]	63 [4, 297]	64 [6, 298]	64 [11, 299]	64 [14, 299]
1	3 [1, 6]	4 [3, 7]	5 [4, 13]	6 [4, 14]	7 [5, 14]	9 [4, 16]	10 [8, 20]	12 [10, 25]	12 [10, 25]	25 / no@10 / 10
10	5 [1, 8]	8 [5, 9]	10 [7, 15]	11 [7, 46]	12 [8, 47]	7 [4, 11]	16 [7, 33]	21 [15, 156]	15 / no@20 / no@160	no@10 / no@20 / no@160
100	14 [2, 46]	91 [35, 204]	101 [48, 213]	119 [74, 213]	131 [103, 243]	2 [1, 5]	34 [5, 52]	no@500 / 89 / no@500	no@500 / no@90 / no@500	no@500 / no@90 / no@500

Table 14: MultiRoom: scale of the process term, with our method. Steps in thousands.

Scale	to25	to50	to75	to85	to95
0	71 [67, 79]	120 [103, 120]	107 / no@200 / 195	107 / no@200 / 195	no@200
1	56 [52, 67]	79 [63, 86]	91 [65, 113]	91 [73, 114]	92 [80, 120]
10	125 [57, 196]	172 [97, 204]	202 [135, 261]	209 [208, 301]	230 [208, 303]
100	102 [68, 443]	234 / 192 / no@500	244 / 218 / no@500	272 / 221 / no@500	272 / no@500 / no@500

Table 15: MultiRoom: same scale, no teacher. Steps in thousands.

Scale	to25	to50	to75	to85	to95
0	no@5M	no@5M	no@5M	no@5M	no@5M
1	no@500	no@500	no@500	no@500	no@500
10	no@500	no@500	no@500	no@500	no@500
100	no@500	no@500	no@500	no@500	no@500

H Experimental results: observation noise

Table 4 in the main text reports time to 75% and whether 95% is reached by 500k steps. Table 16 is the same Env2 sweep at every threshold, including the no-teacher rows. The $\sigma = 0$ rows repeat the Env2 row of Table 6. Entries are median $[\min, \max]$ in thousands of environment steps, or one value per seed when the ledger does not summarize them. none@k means that seed had not reached the bar at k thousand steps.

Table 16: Env2 with Gaussian noise on the observation. Steps, in thousands, until teacher-free success reaches each rate.

σ	Method	25%	50%	75%	85%	95%
0	No teacher	2328 [1080, 7409]	2883 [1188, 7637]	3113 [1263, 8310]	3114 [1265, 8382]	3114 [1265, 8665]
0	Language	20 [20, 25]	25 [25, 30]	25 [25, 30]	30 [25, 30]	30 [25, 50]
0.05	No teacher	3075 [1020, 3285]	none@5000 / 1070 / 3330	none@5000 / 1250 / 3385	none@5000 / 1445 / 3560	none@5000 / 1855 / 3605
0.05	Language	15 [10, 15]	25 [15, 30]	55 [45, 60]	85 [50, 265]	none@500 / none@500 / 400
0.1	No teacher	2155 [1935, 2705]	2750 [2235, 2780]	2780 [2335, 3050]	2780 [2385, 3130]	3810 [3385, 4765]
0.1	Language	15 [10, 20]	25 [15, 40]	40 [35, 60]	50 [45, 200]	none@500 / 430 / none@500
0.3	No teacher	910 [500, 1780]	2550 [510, 2775]	3220 / 1430 / none@5000	3220 / 1760 / none@5000	4580 / 3610 / none@5000
0.3	Language	10 [10, 10]	25 [25, 40]	75 [35, 100]	none@500 / 75 / 115	none@500
0.5	No teacher	2710 / 315 / none@5000	none@5000	none@5000	none@5000	none@5000
0.5	Language	20 [10, 30]	45 [40, 50]	75 [40, 115]	485 / none@500 / none@500	none@500

I Teacher models

The student and the controller stay fixed. Table 17 swaps only the language model on Env2. Qwen3-14B, Qwen3-8B Qwen Team (2025), and DeepSeek-V4.1-Flash each reach median teacher-free success 0.90. Time to 95% is 30, 45, and 70 thousand environment steps. Qwen3-4B, Qwen3-1.7B, and Llama-3.1-8B Grattafiori et al. (2024) fail the open-loop check and are not trained.

Table 17: Env2 with a different language model as the teacher. Steps, in thousands, until teacher-free success reaches each rate. Open-loop is the teacher acting alone, before the student is trained.

Teacher	Open-loop	25%	50%	75%	85%	95%	Final
Qwen3-14B	8/8	20 [20, 25]	25 [25, 30]	25 [25, 30]	30 [25, 30]	30 [25, 50]	0.90 [0.70, 1.00]
Qwen3-8B	18/20	20 [15, 20]	25 [20, 30]	30 [30, 35]	30 [30, 40]	45 [30, 65]	0.90 [0.85, 0.95]
DeepSeek-V4.1-Flash	20/20	25 [25, 30]	30 [30, 45]	45 [40, 85]	45 [45, 85]	70 [45, 85]	0.90 [0.85, 0.95]
Qwen3-4B	0/20	—	—	—	—	—	—
Qwen3-1.7B	1/20	—	—	—	—	—	—
Llama-3.1-8B	0/20	—	—	—	—	—	—

J FROZEN PROMPTS

Each teacher call has a human instruction and a system description of the current step. The blocks below are that frozen wording. Table 18 keeps the coaching intent and changes only the words.

Env2. Human input. Every call starts with the same three lines, then one phase sentence, then one line for each missile, then the same ending.

```
PHASE / NOW ONLY (ignore any longer mission plan): coach this instant.
Axis: right = positive lateral (+a), left = negative lateral (-a).
No numeric throttle --- only the coach's plain English.
```

When the missiles are close to the target:

```
Close in: keep the coached sides, finish cleanly.
```

When a threat is ahead:

```
Threat ahead: open laterally NOW with the coached sides ---
do not wait until the last second. Do not second-guess with sin_herr.
```

When the path looks clearer:

```
Path looks clearer --- follow the coached sides and close range.
```

A dead missile is told:

```
Missile 1: dead --- zero its lateral.
```

A live missile that should barely move is told:

```
Missile 1: hold steady, almost no lateral this step.
```

Otherwise that missile gets one frozen sentence, with left or right filled in. Missile 2 uses the same list.

```
Missile 1: give only a feather touch toward the left.
Missile 1: barely kiss the stick toward the left.
Missile 1: the tiniest drift toward the left.
Missile 1: ease a bit toward the left.
Missile 1: nudge toward the left.
Missile 1: lean a little toward the left.
Missile 1: bank firmly toward the left.
Missile 1: cut in toward the left.
Missile 1: push with clear intent toward the left.
Missile 1: yank hard toward the left.
Missile 1: haul aggressively toward the left.
Missile 1: slam the stick toward the left.
```

Every call ends with:

```
Reply ONLY a continuous [a0,a1] in [-1,1].
Infer how hard to push from the wording; match left/right.
No canned equal-and-opposite template.
Mission cue: evade the interceptors and keep closing on the target.
```

System input. The task, which way is left and right, and a snapshot of the two missiles and the interceptors.

DoorKey. Human input. Before the key is picked up:

```
Get the yellow key first. Don't wander around the room.
```

Once the key is in hand and the door is still shut:

```
You have the key. Go open the yellow door.
```

Once the door is open:

```
The door is open. Go to the green goal.
```

System input. The room grid, where the agent is facing, and what it can do this step.

UnlockPickup. Human input. Before the key is picked up:

```
Get the key first. Don't wander around the room.
```

Once the key is in hand and the door is still locked:

```
You have the key. Go open the door.
```

Once the door is open:

```
The door is open. Put the key down, then pick up the box.
```

System input. Where the agent stands, what it faces, what it holds, and where the key, the door, and the box are.

MultiRoom. Human input. While a closed door is still ahead:

```
Go open the next closed door.
```

Once those doors are open:

```
The door is open. Go to the green goal.
```

System input. Where the agent stands, what it faces, where the closed doors are, and where the green goal is.

The frozen wording is stable when the coaching intent stays the same. Table 18 holds the freeze shell fixed (n=3, π-only). Paraphrase rewrites the system addendum and leaves each missile line as in the freeze. That arm reaches 75% in a median of 35 thousand steps, the same median as the freeze, with teacher-free success 0.85. Synonyms call left and right port and starboard, swap the words for how hard to push, and remap the axis line so the sign still matches the word. That arm reaches 75% in 40 thousand steps, with success 0.90. The freeze row is 35 thousand steps and success 0.80. Same intent, different words, same speedup. Dropping the axis line (right is positive lateral) keeps the coaching sentences and removes the sign convention. Steps to 75% rise to a median of 100 thousand, and success stays at 0.85.

K PROMPT-INGREDIENT ABLATION

On Env2 the language student needs the frozen coach's per-missile direction. Table 19 removes one piece of that coach and retrains. The full coach reaches median teacher-free success 0.90, in 30 thousand steps to 95%. Removing the phase sentence or the intensity word still reaches 0.85, in 50 and 100 thousand steps to 95%. Replacing the coach with the axis sentence alone, or dropping each missile's left/right line, leaves every seed at success 0.00 within 500 thousand steps. Labeling both missiles "Body", instead of "Missile 1" and "Missile 2", drops final success to 0.45. The full-coach and no-teacher rows repeat the Env2 row of Table 6.

Table 18: Env2 prompt wording, with the same coaching intent. Steps to 75% teacher-free success are in thousands.

Wording	to75 (k)	Eval hit
Freeze	35 [25, 55]	0.80 [0.50, 0.90]
Paraphrase	35 [35, 50]	0.85 [0.85, 0.90]
Synonyms	40 [20, 50]	0.90 [0.85, 0.95]
No axis legend	100 [30, 125]	0.85 [0.80, 0.90]

Table 19: Env2 with one piece of the prompt removed. Steps, in thousands, until teacher-free success reaches each rate.

Arm	25%	50%	75%	85%	95%	Final
Full coach	20 [20, 25]	25 [25, 30]	25 [25, 30]	30 [25, 30]	30 [25, 50]	0.90 [0.70, 1.00]
Drop phase	20 [20, 25]	25 [20, 25]	30 [30, 35]	40 [35, 45]	50 [40, 55]	0.85 [0.75, 1.00]
Drop magnitude	15 [15, 55]	30 [15, 60]	45 [45, 130]	50 [45, 130]	100 [90, 170]	0.85 [0.65, 0.90]
Axis only	none@500	none@500	none@500	none@500	none@500	0.00 [0.00, 0.00]
Drop side	none@500	none@500	none@500	none@500	none@500	0.00 [0.00, 0.00]
Drop identity	35 [30, 50]	55 [30, 60]	70 [60, 185]	none@500 / 80 / 95	none@500 / none@500 / 480	0.45 [0.45, 0.75]
No teacher	2328 [1080, 7409]	2883 [1188, 7637]	3113 [1263, 8310]	3114 [1265, 8382]	3114 [1265, 8665]	0.65 [0.05, 0.90]