ENERGY-BASED TRANSFER FOR REINFORCEMENT LEARNING

Zeyun Deng[1], **Jasorsi Ghosh**[1], **Fiona Xie**[1], **Kai Cheng**[1], **Yuzhe Lu**[2], **Katia Sycara**[3], **Joseph Campbell**[1]
[1] Purdue University [2] AWS AI [3] Carnegie Mellon University

ABSTRACT

Reinforcement learning remains sample inefficient, particularly when agents must adapt over time to related but distinct tasks, as in continual learning settings. Efficiency can be improved by transferring knowledge from a pretrained teacher policy to guide exploration in new but related tasks. However, if the new task sufficiently differs from the teacher's training task, the transferred guidance may be sub-optimal and bias exploration toward low-reward behaviors. We propose an energy-based transfer learning method that uses out-of-distribution detection to selectively issue guidance, enabling the teacher to intervene only in states within its training distribution. We theoretically link energy scores to state-visitation patterns under on-policy reinforcement learning. Empirically, our method improves sample efficiency and performance across both single-task and multi-task settings.

1 INTRODUCTION

Reinforcement learning (RL) excels at sequential decision-making (Sutton & Barto, 1998), but credit assignment, sparse rewards, and modeling errors makes it notoriously sample inefficient. This is limiting in multi-task or continual learning settings, where agents must repeatedly learn to solve new tasks, particularly when those tasks are related to ones they have seen before. A natural question arises: *can we transfer knowledge from previously solved tasks to accelerate learning in new ones*?

One common approach reuses a pretrained teacher to guide a student, either directly by suggesting actions (Uchendu et al., 2023) or indirectly by shaping rewards (Brys et al., 2015). Early guidance can steer the student toward high-reward behaviors and reduce the need for random exploration, making transfer learning an appealing strategy. However, when tasks differ sufficiently, teacher guidance can impair rather than accelerate learning. The teacher may issue suboptimal advice that biases exploration toward low-reward regions of the state-action space (Taylor & Stone, 2009), a phenomenon known as negative transfer.

In this paper, we introduce an introspective transfer learning method that selectively guides exploration only when the teacher's knowledge is likely to be beneficial. Our approach, *energy-based transfer learning* (EBTL), is based on the insight that guidance should only be issued when the student visits states that lie within the teacher's training distribution. Leveraging concepts from energy-based learning (LeCun et al., 2006) and out-of-distribution detection (Liu et al., 2020), the teacher computes energy scores over states visited by the student during training, treating high-energy states as in-distribution and therefore eligible for guidance. This mechanism enables the teacher to act only when it is sufficiently "familiar" with the current context, making training more efficient not by issuing *more* guidance but by issuing *correct* guidance. Our contributions are:

- We introduce an energy-based transfer learning method that selectively guides exploration only when the student's state lies within the teacher's training distribution.
- We theoretically show that energy scores tend to increase on visited states as training progresses toward convergence, supporting the use of energy as a familiarity signal.
- We empirically demonstrate that our method is more sample-efficient and has higher returns than other transfer baselines, across both single-task and multi-task settings.

2 RELATED WORK

Reinforcement learning provides a framework for agents to learn policies that maximize cumulative reward (Sutton & Barto, 1998), but remains notoriously sample-inefficient, particularly in sparse-reward environments (Andrychowicz et al., 2017). Transfer learning mitigates this by reusing knowledge from previously solved tasks to accelerate learning

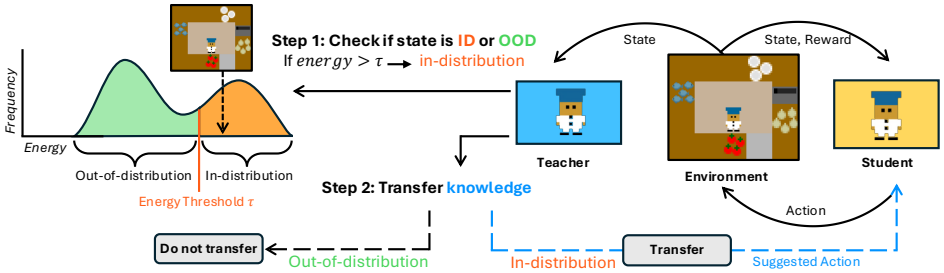

Figure 1: Overview of **energy-based transfer learning**. As the student interacts with the environment, the teacher: 1) checks if each state is in-distribution or out-of-distribution by comparing the state's energy score to a pre-defined *energy threshold*; 2) If the state exceeds the *energy threshold*, then it is considered in-distribution for the teacher and an expert action is suggested to the student.

on new ones (Weiss et al., 2016). A key distinction is whether the teacher interacts with the target task during transfer, yielding *offline* vs. *online* RL transfer.

Offline methods train teacher policies without target-task interaction, using only source data to learn representations (Bose et al., 2024) or task structure (Rosman & Ramamoorthy, 2012). Without target feedback, these methods must rely on broad generalization guarantees, which often yield conservative or mismatched advice, or prior assumptions about the target MDP, which require domain knowledge, making offline transfer brittle under covariate shift. Beyond fully offline methods, related strategies reuse source knowledge during target training. Pretraining initializes policies, value functions, or representations before online fine-tuning (Abel et al., 2018), but can misguide exploration when source and target distributions misalign. Hierarchical transfer uses high-level controllers with pre-learned options (Barreto et al., 2019), but degrades when the option library inadequately covers the target task.

By contrast, online transfer learning adapts during the student's training on the target task: a teacher pretrained on a source task monitors rollouts and provides guidance as the student learns. Guidance is delivered interactively through action suggestions (Torrey & Taylor, 2013) or reward shaping (Ng et al., 1999). Behavior-based approaches encourage the student to align with the teacher: policy distillation introduces an auxiliary divergence loss to promote imitation during training (Rusu et al., 2015; Schmitt et al., 2018), while action advising lets the teacher intervene with actions during exploration (Torrey & Taylor, 2013). A specific instantiation of action advising is probabilistic policy reuse (Fernández & Veloso, 2006), where a library of past policies biases exploration via a decaying follow probability. The usefulness of each past policy is tracked via a scalar reuse gain averaged over entire episodes, which conflates performance across states and tasks, making it unreliable in multi-task settings where a policy may be beneficial in some states but harmful in others. A persistent challenge is deciding when to advise, as poorly timed interventions can hinder learning (Torrey & Taylor, 2013). JumpStart RL restricts advice to a fixed episode prefix (Uchendu et al., 2023). Critic-based methods (Campbell et al., 2023) fine-tunes the teacher's critic on student exploration data to estimate transferability, but differences in predicted returns may come from inaccurate value estimates rather than true task mismatch, causing interventions to depend on value convergence. Beyond the transfer-RL literature, the question of when expert input should be received during learning has been studied extensively in imitation learning. DAgger (Ross et al., 2011) queries the expert unconditionally at every state visited by the learner, and AggreVaTe (Ross & Bagnell, 2014) instead queries the expert's cost-to-go at a single uniformly chosen time per trajectory. More recent active variants such as ASkDAgger (Luijkx et al., 2025) reduce expert load by querying only when the novice signals uncertainty about its own plan, a learner-side decision. EBTL addresses the same question from the opposite direction: the teacher introspects on its own familiarity with the current state and decides whether to advise, placing our work alongside introspective action advising (Campbell et al., 2023) within the teacher-side branch of this broader literature.

We adopt an online transfer setting that avoids requiring hand-crafted target-task knowledge and allows the teacher to decide, state by state, what knowledge is beneficial to transfer. Our method builds on this principle by applying theoretically grounded out-of-distribution detection to estimate teacher familiarity and selectively issue guidance. Importantly, we focus on an actor-based formulation of online transfer, where the decision of when to advise is derived from the teacher's policy rather than its value estimates, leading to a more stable and direct signal for intervention.

3 Background

Reinforcement Learning. We study a Markov decision process (MDP), defined by the tuple $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$: $\mathcal{S}$ is the state space; $\mathcal{A}$ the action space; $P(\cdot \mid s, a)$ the transition kernel on $\mathcal{S}$; $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ the reward; and $\gamma \in [0, 1)$

the discount factor. At time t, the agent in $s_t \in \mathcal{S}$ chooses $a_t \in \mathcal{A}$, transitions to $s_{t+1} \sim P(\cdot \mid s_t, a_t)$, and receives $r_t = R(s_t, a_t)$. The goal is to learn a policy $\pi(a \mid s)$ maximizing the discounted return $\mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t r_t]$.

Energy-Based Out-of-Distribution Detection. In this work, we determine whether a state is in-distribution (ID) or out-of-distribution (OOD) for a given policy. Common baselines include maximum softmax probability (MaxP) (Hendrycks & Gimpel, 2016), entropy (Sedlmeier et al., 2020), and temperature-scaled variants such as ODIN (Liang et al., 2017), all of which operate on the softmax output distribution. The root issue is that softmax normalization discards logit magnitude: for an OOD input with large diffuse logits, softmax can still produce a peaked distribution, yielding high MaxP, low entropy, and high ODIN confidence that incorrectly signals in-distribution (Nguyen et al., 2015; Hein et al., 2019). The *energy* score avoids this by operating directly on the raw logits without normalization. Given $\mathbf{x} \in \mathbb{R}^D$ and network logits $f(\mathbf{x}) \in \mathbb{R}^K$, the free energy is defined as

$$E(\mathbf{x}; f) \;=\; -T \log \sum_{i=1}^{K} e^{f_i(\mathbf{x})/T}, \tag{1}$$

where $T > 0$ controls logit sharpness. By preserving logit magnitude, energy correctly reflects the diffuseness of OOD inputs and has been shown to more reliably separate ID from OOD examples than softmax-based criteria (Liu et al., 2020). An input is classified as OOD if $E(\mathbf{x}; f) > \tau$ for a pre-computed threshold τ, and ID otherwise. Since policy networks are logit-based decision models subject to the same overconfidence failure mode, energy provides a natural uncertainty signal for RL state distributions, a connection we formalize in the following section.

4 Energy-Based Transfer Learning

Our goal is to improve sample efficiency in reinforcement learning, particularly when agents must solve related tasks. A common approach is to leverage a teacher policy trained on a source task to guide a student on a new target task. However, naïve teacher guidance can degrade performance when the student visits states outside the teacher's training distribution, driving uninformative exploration. We therefore propose a transfer framework where the teacher provides guidance only in states close to its training distribution, framed as *OOD detection for reinforcement learning*.

Problem Formulation. We adopt the MDP definition from Section 3. The transfer setting consists of a source task $\mathcal{M}_{\text{src}} = \langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \rho_{0,\text{src}}, \gamma \rangle$ and a target task $\mathcal{M}_{\text{tgt}} = \langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \rho_{0,\text{tgt}}, \gamma \rangle$ that differs only in the initial-state distribution. The shared state space, action space, transition kernel, reward function, and discount factor let π_{teacher} execute in the target task without architectural changes. In practice, the set of reachable states between source and target differs via the change in ρ_0, while any state reachable in both tasks shares the same transitions and rewards.

Let π_{teacher} and π_{student} denote the teacher and student policies, respectively. We denote a trajectory as $X = \{x_t\}_{t=1}^{n}$, where each transition $x_t = (s_t, a_t, s_{t+1}, r_t)$ consists of the state s_t, action a_t, next state s_{t+1}, and reward r_t. We define a score function $\phi(s; \pi)$, where a state s is considered ID with respect to a policy π if $\phi(s) \geq \tau$, for some threshold $\tau \in \mathbb{R}$, and OOD otherwise. Throughout this paper, ID is taken with respect to π_{teacher}; formally, ID states are those with substantial density under $d_{\text{src}}^{\pi_{\text{teacher}}}$, the discounted state-visitation distribution of π_{teacher} in $\mathcal{M}_{\text{src}}$. The action selection rule is then defined as:

$$a = \begin{cases} a_{\text{teacher}} \sim \pi_{\text{teacher}}(\cdot \mid s), & \text{if } \phi(s; \pi_{\text{teacher}}) \geq \tau, \\ a_{\text{student}} \sim \pi_{\text{student}}(\cdot \mid s), & \text{if } \phi(s; \pi_{\text{teacher}}) < \tau. \end{cases} \tag{2}$$

Assumptions. EBTL relies on two assumptions that together ensure the teacher's energy score correctly identifies states where its guidance is beneficial to the student:

(1) Teacher Reliability: The teacher policy π_{teacher} is trained online to near-optimality on the source task. Since on-policy training generates data by sampling from the teacher's own policy, the resulting state-visitation distribution concentrates on regions where the teacher has learned to act reliably. Consequently, guidance issued in these states is generally trustworthy. When the teacher is suboptimal, its advice may be unreliable even within its training distribution, placing an inherent ceiling on the achievable transfer gain.

(2) Observable Task Shift: Differences between source and target tasks are reflected in the observable state s. Equivalently, any shift in ρ_0 must induce a corresponding shift in the observable state distribution. For example, if tasks vary by goal location, the goal coordinates in s differ from those seen during teacher training, allowing the teacher to distinguish target from source states. In our benchmarks: Alternating-Goal shifts (goal location encoded in s); Unlocked-to-Locked shifts (locked doors with the goal reachable only via key collection, observable through door and key features in s); Overcooked shifts via different layouts and recipes encoded in the observation. In contrast, this

assumption does not hold if task differences depend on hidden variables not present in s, such as unobserved reward changes. Under these assumptions, energy-based filtering enables reliable knowledge transfer (Equation 2): frequently encountered states receive high energy scores and trigger teacher guidance, while rare states receive low scores and defer to the student policy.

4.1 Energy Scores and State Visitation

We seek a score function that reflects the teacher's familiarity with each state without explicitly tracking visitation counts. Drawing on energy-based out-of-distribution detection (Liu et al., 2020), we define the *energy score* $\boldsymbol{\phi(s;\pi_{\text{teacher}}) = -E(s;\pi_{\text{teacher}})}$, where $E(s;\pi_{\text{teacher}})$ is the free energy computed from the teacher's network logits (Equation 1) . We set ϕ to the *negative* free energy so that familiar (ID) states receive higher scores than unfamiliar (OOD) states. Note that Equation 1 is defined over a discrete set of logits, and is therefore most natural when the policy outputs a categorical distribution over a finite action space. For continuous action spaces, one practical approach is to discretize actions into bins or codebook entries, as widely adopted in robotics (Kim et al., 2024; Zitkovich et al., 2023) and autonomous driving (Zhou et al., 2025; Zhang et al., 2026), which allows Equation 1 to be applied without modification. When the policy is instead parameterized as a continuous distribution, such as a Gaussian with learned mean and variance, the energy score as defined does not directly apply. In this case, one must rederive an analogous score from the structure of the output distribution. We explore this direction in a MetaWorld robotics environment, where we derive and compute an energy score for a Gaussian policy directly, without resorting to discretization, and find that it yields strong transfer performance (Appendix E). Since the teacher policy is assumed to converge to a near-optimal policy prior to transfer, it is expected to gradually place higher probability mass on near-optimal actions during training. We formalize this key property below, which enables EBTL.

Proposition 1 (Asymptotic on-policy monotonicity of the energy score)**.** *Under on-policy training towards convergence, policy gradient updates tend to increase $\phi_\theta(s)$ for visited states.*

Proof. Assume actor-critic architecture for the on-policy update. Let $\pi_\theta(a \mid s) = \mathrm{softmax}(f_\theta(s)/T)_a$ with logits $f_\theta(s) \in \mathbb{R}^K$ for K actions, temperature $T > 0$, and write $\pi_j = \pi_\theta(j \mid s)$ for brevity. Let $A(s,a) = Q(s,a) - V(s)$ denote the advantage of action a in state s, where $Q(s,a)$ and $V(s)$ are the action-value and state-value functions, respectively.

Energy change. Let $E_\theta(s) = -T \log \sum_{j=1}^K e^{f_j(s)/T}$ denote the free energy and $\phi_\theta(s) = -E_\theta(s)$ the energy score. A policy gradient step on $A(s,a) \log \pi_\theta(a \mid s)$ for a sampled action a updates each logit by $\Delta f_j = \frac{\eta\, A(s,a)}{T}\big[\mathbf{1}_{j=a} - \pi_j\big]$, where $\eta > 0$ is the step size and $\mathbf{1}_{j=a}$ is the indicator function. Using $\frac{\partial E_\theta(s)}{\partial f_j} = -\pi_j$, the first-order change in the score is (see Appendix A.1)

$$\Delta\phi_\theta(s) \;=\; \frac{\eta\, A(s,a)}{T}\left[\pi_a - \sum_{j=1}^K \pi_j^2\right].$$

The sign of $\Delta\phi_\theta(s)$ depends on two factors, both of which are non-negative towards convergence.

Non-negativity towards convergence. For the bracket term, conditioning on the high-probability event $\pi_a = \max_j \pi_j$ gives $\pi_a \geq \pi_j$ for all j, so $\sum_j \pi_j^2 \;\leq\; \pi_a \sum_j \pi_j \;=\; \pi_a$, since $\sum_j \pi_j = 1$, and hence $\pi_a - \sum_j \pi_j^2 \geq 0$.

For the advantage term, conditioning on $a = a^*$,

$$A(s,a^*) = Q(s,a^*) - V(s) = \sum_{a \neq a^*} \pi(a \mid s)\big[Q(s,a^*) - Q(s,a)\big] \geq 0,$$

since $a^* = \arg\max_a Q(s,a)$ implies $Q(s,a^*) \geq Q(s,a)$ for all a.

Together, near convergence the policy concentrates on a^*, so $\pi(a^* \mid s) \to 1$. On the event $a = a^*$, both factors are non-negative, implying $\Delta\phi_\theta(s,a^*) \geq 0$. For $a \neq a^*$, the contribution is weighted by $\pi(a \mid s)$, whose total mass vanishes as $\pi(a^* \mid s) \to 1$. Since $\Delta\phi_\theta(s,a)$ is bounded for fixed s, it follows that $\mathbb{E}_{a\sim\pi}[\Delta\phi_\theta(s)] = \pi(a^* \mid s)\Delta\phi_\theta(s,a^*) + \sum_{a\neq a^*} \pi(a \mid s)\Delta\phi_\theta(s,a) \to \Delta\phi_\theta(s,a^*) \geq 0$. Hence, policy gradient updates tend to increase the energy score on visited states in expectation near convergence. □

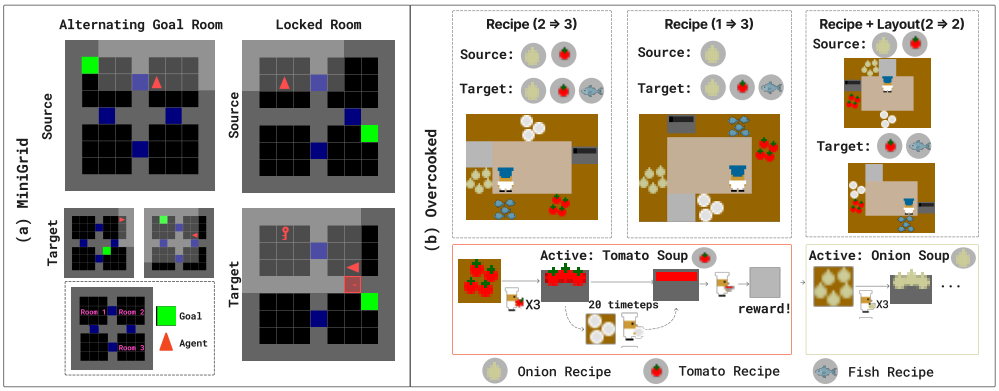

Figure 2: Environments used for empirical experiments. See Section 5 for detailed descriptions.

Algorithm 1 Energy-Based Transfer for Reinforcement Learning

Input: Teacher policy π_{teacher}, student policy π_{student}, energy threshold τ, decay function δ
while not done **do**
  Initialize empty batch $B \leftarrow \emptyset$
  for $t = 1 \rightarrow H$ **do**
    Sample $p \sim \mathcal{U}(0, 1)$ ▷ Sample probability of issuing guidance
    if $-E(s_t; \pi_{\text{teacher}}) \geq \tau$ and $p < \delta(t)$ **then**
      $a_t \leftarrow \pi_{\text{teacher}}(a \mid s_t)$ ▷ If s_t is ID
    else if $-E(s_t; \pi_{\text{teacher}}) < \tau$ **then**
      $a_t \leftarrow \pi_{\text{student}}(a \mid s_t)$ ▷ If s_t is OOD
    end if
    Take action a_t, observe r_t, s_{t+1}
    Compute importance ratio $\rho_t \leftarrow \frac{\pi_{\text{student}}(a_t|s_t)}{\pi_{\text{teacher}}(a_t|s_t)}$ if s_t is ID, else $\rho_t \leftarrow 1$ ▷ Correct for off-policy teacher actions
    $B \leftarrow B \cup (s_t, a_t, s_{t+1}, r_t, \rho_t)$
  end for
  Update π_{student} with batch B ▷ Any on-policy update
end while

4.2 Algorithm

Algorithm 1 summarizes EBTL. At each timestep, the teacher evaluates the current state via its energy-based OOD score to determine familiarity. If the state is classified as ID and a decaying schedule permits intervention, the teacher's action is adopted; otherwise, the student acts independently. The intervention threshold is set as $\tau = \text{Quantile}_q(\{\phi(s) \mid s \in \mathcal{S}_{\text{teacher}}\})$, where $\mathcal{S}_{\text{teacher}}$ denotes the states visited under the teacher policy during training and $q \in [0, 1)$ controls strictness. Smaller values of q admit a broader set of states as ID, whereas larger values impose stricter similarity to the teacher's training distribution. To control the rate of teacher intervention, we apply a linear decay schedule $\delta(t)$, following prior work (Schmitt et al., 2018; Uchendu et al., 2023; Campbell et al., 2023). This enables early reliance on the teacher while gradually transferring decision-making to the student. Appendix I establishes that some form of decay is necessary for the student to achieve autonomy, though the specific schedule is not critical provided it eventually reduces teacher influence to zero. Finally, since teacher actions are off-policy from the student's perspective, we apply importance-ratio correction to mitigate the resulting distributional mismatch (Campbell et al., 2023).

Energy Regularization. As discussed in Section 4.1, the score function $\phi(s)$ is related to the teacher's state-visitation frequency: frequently visited states tend to receive higher scores. This relation, however, provides no guarantee for states that lie outside the teacher's experience. OOD states may produce heterogeneous scores that overlap with ID values, weakening separability. Collapsing all OOD states to a uniformly low score would also remove useful structure by treating partially compatible states the same as completely unrelated ones. We therefore add an energy regularizer that enforces a margin between ID and truly OOD states, improving separability while preserving the relative ordering among familiar states. To improve separability, we augment teacher training with the energy-based loss of (Liu et al., 2020). Let $\mathcal{D}_{\text{in}}$ denote ID states collected during teacher training and $\mathcal{D}_{\text{out}}$ denote a set of OOD

states. For samples $s_{\text{in}} \sim \mathcal{D}_{\text{in}}$ and $s_{\text{out}} \sim \mathcal{D}_{\text{out}}$, we optimize the following loss:

$$\mathcal{L}_{\text{energy}} = \mathbb{E}_{\mathbf{s}_{\text{in}}}\left[\left(\max(0, m_{\text{in}} - \phi(\mathbf{s}_{\text{in}}))\right)^2\right] + \mathbb{E}_{\mathbf{s}_{\text{out}}}\left[\left(\max(0, \phi(\mathbf{s}_{\text{out}}) - m_{\text{out}})\right)^2\right],$$

$m_{\text{in}}, m_{\text{out}} \in \mathbb{R}$ are margins for ID and OOD energies, penalizing ID energies $< m_{\text{in}}$ and OOD energies $> m_{\text{out}}$. The teacher loss is $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{RL}} + \lambda\mathcal{L}_{\text{energy}}$, where $\lambda \in \mathbb{R}^+$ controls the regularization strength. In the main paper, ID samples are drawn from the teacher's most recent update batches to approximate the empirical state distribution induced by the teacher policy, and OOD samples are drawn from random rollouts in the student environment to approximate states outside the teacher's training distribution. The energy loss is robust to these choices (Appendix H): uniform random sampling of OOD states from the full observation space, without assuming target-task knowledge, yields comparable transfer performance; separability is largely insensitive to the margin values m_{in} and m_{out}; ID samples can be obtained by rolling out the teacher when training data are unavailable; and energy regularization accelerates convergence without degrading final teacher performance.

5 Experiments

We evaluate our method in two settings: **single-task** and **multi-task**. The single-task setting uses Minigrid (Chevalier-Boisvert et al., 2023) navigation tasks where the agent reaches a goal location. The multi-task setting uses Overcooked (Carroll et al., 2019), where the agent learns to cook different recipes. For each setting, we design scenarios with increasing covariate shift between teacher and student distributions to test robustness under progressively harder transfer. We examine: **whether our method improves sample efficiency** and **when the teacher provides guidance during learning**. We compare against baselines that do not require training or tuning additional networks beyond the teacher policy prior to transfer. This isolates the effect of the transfer mechanism itself, rather than gains from additional model capacity. In our main paper (Campbell et al., 2023), which assess transferability by fine-tuning a teacher's value function on student data. While this represents a valuable and complementary perspective on introspective action advising, it introduces an additional learned component into the pipeline that conflates two distinct sources of failure: true task mismatch and inaccurate critic learning. Because the method's reliability is contingent on value estimation quality, it is difficult to isolate whether a transfer failure reflects a genuine incompatibility between teacher and student tasks or simply an artifact of critic misfitting on limited student data. This entanglement makes direct comparison with policy-side introspection methods methodologically ambiguous, and we therefore treat it as an orthogonal line of work rather than a competing baseline.

We consider the following baselines: **(1) No Transfer.** The student is trained from scratch on the target task, providing the standard reference point for all transfer methods. **(2) Standard Action Advising (AA)** (Torrey & Taylor, 2013). The teacher provides actions throughout training with a decaying rate. This is equivalent to our method with $q = -1$, i.e., no OOD filtering. It serves as a key ablation, showing that indiscriminate guidance can lead to negative transfer when the teacher is applied in unfamiliar states. **(3) JumpStart RL (JSRL)** (Uchendu et al., 2023): A curriculum-based method that restricts guidance to early timesteps in each episode. It does not consider state-level distributional mismatch and may issue advice in irrelevant states due to its purely time-based schedule. **(4) Kickstarting RL (KSRL)** (Schmitt et al., 2018): A distillation-based method that transfers knowledge from a teacher policy via an auxiliary imitation loss whose weight is gradually annealed over training. While it encourages policy alignment, it does not account for distributional mismatch and applies supervision uniformly across states. **(5) Fine-Tuning**: The student is initialized from a pretrained teacher policy. Convolutional layers are frozen, and only the remaining parameters are updated during training. When comparing against baselines, all shared hyperparameters are kept the same across methods to ensure a fair comparison. For Minigrid, each baseline is tuned independently, with our method's configuration reported in this section and baseline details provided in Appendix D. Since our ablation studies on Minigrid already show that transfer performance is directly linked to the quality of filtered advice, we evaluate on Overcooked without environment-specific tuning to test generalization to a more complex setting. Due to long training times, hyperparameters for all methods are set to reasonable values based on their original implementations, with shared parameters such as decay rate kept consistent across methods.

5.1 Single-Task Setting: Minigrid

Our Minigrid environment consists of four interconnected rooms designed for single-task transfer, illustrated in Figure 2a. We construct two setups. **Alternating Goal Room.** The source task always places the goal in a random location in Room 1 (upper-left), while the target task randomly places it in either Room 1 (upper-left) or Room 3 (lower-right). The teacher should intervene only when the goal is in Room 1, where its prior experience applies; when the goal is in Room 3, the student must act independently. **Locked Room.** The source task contains no doors, while the target task features a locked door partitioning the lower area that requires collecting a key in the upper rooms to

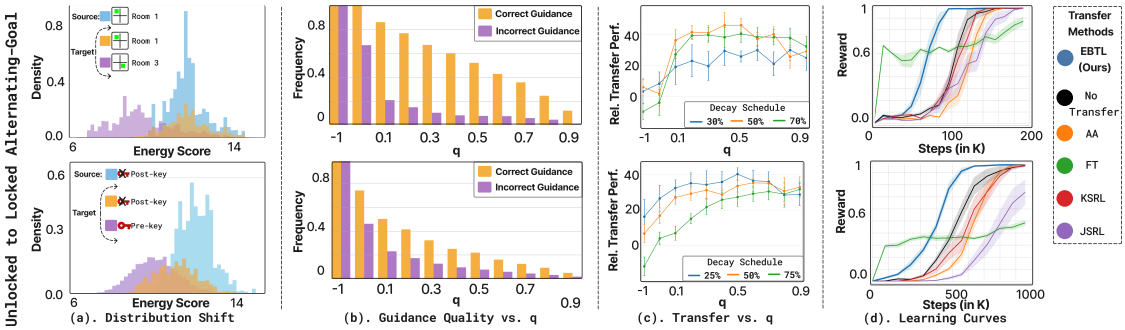

Figure 3: **Minigrid** (10 seeds). **(a)** Empirical energy score distributions with respect to the teacher policy. The source task (blue) shows the teacher's training distribution. The target task (orange + purple), measured during transfer, is bimodal: one mode overlaps with the source (shared sub-task, in-distribution), while the other does not (non-shared sub-task, out-of-distribution). **(b)** Guidance quality: fraction of correct (orange) vs. incorrect (purple) guidance at different q values with decay schedule excluded. **Correct guidance occurs when the teacher action matches the optimal action for the target-task state**. Orange: among all the target (shared sub-task with the source, in-distribution) states, the fraction at which the teacher recognizes the state as ID and issues correct advice (true positive rate). Purple: among all the target (non-shared sub-task with the source, out-of-distribution) states, the fraction at which the teacher incorrectly classifies the state as ID and issues advice anyway (false positive rate). An ideal gating signal yields an orange bar substantially higher than the purple bar. **(c)** Relative transfer (% vs. scratch) across q and decay schedules (50%: guidance reaches 0 at mid-training). $q = -1$ denotes always advising (AA baseline). **(d)** Learning curves: EBTL vs. baselines. Guidance ends at mid-training with $q = 0.5$.

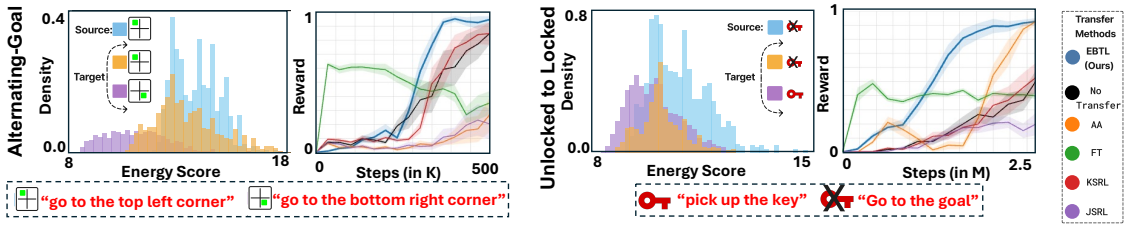

Figure 4: **Lang-Minigrid**. (10 seeds). Guidance ends at mid-training with $q = 0.5$. Task variation is specified through language instructions: in Alternating-Goal, the instruction indicates the target corner (e.g., *go to the top left corner*), while in Unlocked-to-Locked, it specifies sequential subgoals (e.g., *pick up the key, then go to the goal*).

open. This induces two distinct regimes: *pre-key states*, where the agent must solve a key-retrieval problem entirely absent from the source task and teacher guidance is unreliable, and *post-key states*, where the door is open and room connectivity is restored. Even in post-key states, the door remains visually present in the observation, introducing an observation shift the teacher has never encountered. Nevertheless, a reliable teacher should assign post-key states higher familiarity than pre-key states, as both involve the door's presence but post-key states are otherwise closer to the teacher's training distribution. Together, the novel pre-key phase, the persistent observation shift, and the altered state reachability make this a harder and more structured transfer problem than the Alternating Goal setting. We evaluate both setups under two state encodings. In the discrete setting, the observation is a grid matching the environment layout, where each cell records the object type, color, and state (e.g., whether a door is open or locked), where task variation simply appears through object locations such as the goal or key. Results are shown in Figure 3. In the language-conditioned setting, tasks are also specified via textual instructions, with results in Figure 4.

EBTL consistently outperforms all baselines and is robust to hyperparameters. Across both Minigrid transfer setups, EBTL achieves the highest sample efficiency under discrete observations (Fig. 3d) and language-conditioned observations (Fig. 4). Notably, most baselines fail to surpass training from scratch, as indiscriminate guidance introduces incorrect advice that biases the student toward suboptimal behaviors and ultimately hurts performance. This failure is not due to poor hyperparameter choices but rather a fundamental issue: these methods issue guidance regardless of whether the teacher is familiar with the current state, making them susceptible to harmful advice in out-of-distribution states. Across our hyperparameter sweep (Appendix D), JSRL exhibits large variance across different settings and never surpasses EBTL, rarely exceeding training from scratch. KSRL shows slightly lower variance than JSRL but consistently underperforms, never reaching EBTL's performance level. Fine-tuning illustrates a different failure mode:

its reward curve starts high because the pretrained policy already handles the shared sub-task, but then plateaus. On the unseen sub-task, the policy reuses source-task behaviors that no longer apply, and unlearning them inadvertently degrades the shared sub-task as well, forcing a slow balancing act between the two. In contrast, EBTL has two key hyperparameters: the energy quantile q and the decay schedule. Across varying decay schedules (Fig. 3c), performance remains stable across a wide range: $q \in [0.2, 0.8]$ achieves nearly identical results, confirming that EBTL's advantage stems from its selective guidance mechanism rather than hyperparameter tuning.

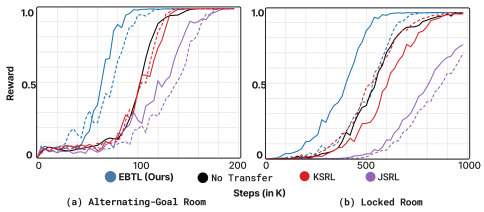

Figure 5: (10 seeds). Learning curves **with (solid)** and **without (dashed)** energy regularization in Minigrid environments.

Energy-based scores provide superior ID/OOD separability, driving EBTL's improvement over baselines. As the energy score is theoretically connected to state visitation under on-policy training, it serves as a natural proxy for teacher familiarity. While our primary contribution is not a new OOD detection method, the quality of ID/OOD separation directly impacts transfer performance. We therefore verify that the energy score provides more reliable separation than established baselines. As shown in Figure 6, MaxP (Hendrycks & Gimpel, 2016) and entropy (Sedlmeier et al., 2020) fail to separate ID and OOD distributions meaningfully. The unregularized energy score, however, achieves separation slightly better than ODIN (Liang et al., 2017) without requiring any OOD samples for hyperparameter calibration. This is consistent with Proposition 1: on-policy training naturally induces a useful energy structure over visited states. Nevertheless, it provides no guarantee on entirely unseen states, whose scores may overlap with ID values, as discussed in Section 4.2. Energy regularization addresses this gap, sharpening the ID/OOD boundary beyond what either ODIN or the unregularized score achieves. This translates to accelerated convergence as shown in Figure 5, particularly in the Locked Room setting where covariate shift is greater. Notably, other baselines show negligible change with or without energy regularization.

Generalization beyond exact state visits. This separability advantage translates directly into a generalization capability that exact state visitation counting cannot provide. Rather than rejecting any unvisited state as OOD, the threshold q admits states that structurally resemble the teacher's experience even if never visited exactly. The mechanism behind this generalization is the energy regularizer of Section 4.2: by pulling the energy of ID states up and the energy of OOD states down during teacher training, the regularizer induces a repulsive gradient that propagates through shared features. States that share more structure with the teacher's training distribution settle at higher energies than states that share less, so the OOD region is not collapsed into a single uniformly low score but spread according to its resemblance to ID experience. Figure 3a illustrates this under varying covariate shift: in Alternating Goal, ID and OOD states separate clearly, while in Locked Room, pre-key states receive lower scores than post-key states, preserving meaningful discrimination even under a strong shift. This is precisely the behavior that exact visitation cannot produce: a teacher trained without doors has never seen any door-present state, yet the regularizer's gradient generalizes the energy ordering to these novel-feature states based on how much they resemble familiar configurations. Under exact counting, a teacher trained without doors would reject all door-present states as OOD. With moderate $q \geq 0.2$, EBTL instead generalizes to these states and provides useful guidance after key collection, when navigation resembles the teacher's training distribution.

EBTL improves monotonically as teacher proficiency increases. We train teachers at different proficiency levels (Table 1) and observe positive transfer gains even with a suboptimal teacher, despite assuming a converged teacher. We conjecture that partially trained teachers retain meaningful energy-visitation structure; as training converges, energy scores on familiar states increase (Proposition 4.1), with actions there becoming increasingly optimal, explaining the monotonic improvement in transfer gain. In contrast, indiscriminate imitation baselines often fail to benefit from stronger teachers and may even deteriorate due to source–target misalignment, as copying teacher actions in unseen states performs no better than random. Interestingly, KSRL performs better at low teacher optimality, likely because an underfit teacher retains general behaviors that transfer better under covariate shift, which the student captures directly through a cross-entropy objective.

5.2 Multi-Task Setting: Overcooked

We create a single-agent variant of the popular Overcooked (Carroll et al., 2019) environment designed to evaluate multi-task learning. An overview is shown in Figure 2b. This environment is both *long-horizon* and *high-dimensional*. **Long-horizon.** Each timestep has one active recipe (onion, tomato, or fish soup). Completing it requires fetching and placing three matching ingredients into a pot, waiting 20 steps, retrieving a dish, and delivering the soup to the serving station. A new recipe is sampled after each delivery. **High-dimensional.** The state space is combinatorial, due to randomized placement of ingredient dispensers, pots, and serving stations; pot contents and cook-timer values;

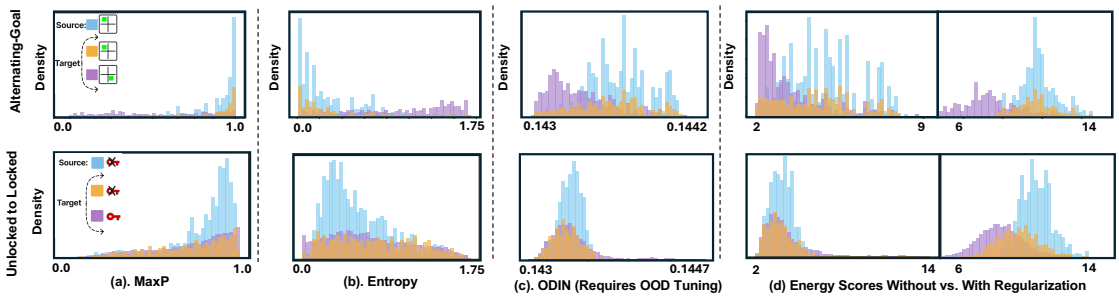

Figure 6: **OOD score distributions across detection methods in Minigrid.** Each panel shows the score distributions of the source task (blue) and target task (orange: shared sub-task, purple: non-shared sub-task) under the teacher policy. A good OOD detector should separate blue and purple while keeping blue and orange overlapping. (a) MaxP, (b) Entropy, and (c) ODIN, requiring OOD samples for hyperparameter calibration. (d) (Left) The energy score without regularization (no OOD samples) already matches ODIN, and (Right) with regularization (OOD samples used during teacher training) achieves substantially sharper separation.

Table 1: **Robustness to suboptimal teachers**. Relative transfer improvement (%) over training from scratch (mean ± 95% CI, 10 seeds) for Alternating Goal. Guidance ends at mid-training.

Optimality	EBTL (0.3)	EBTL (0.5)	EBTL (0.7)	AA	JSRL	KSRL	Fine-tuning
96%	41.2±4.9	**46.1±4.5**	40.3±4.7	6.1±5.6	−3.2±10.3	17.5±6.7	−32.2±12.9
70%	26.8±4.6	**31.2±5.7**	27.0±5.4	17.0±4.3	24.8±5.4	14.2±6.1	−104.3±8.6
45%	16.7±4.6	14.3±4.3	18.9±9.2	13.9±6.2	15.9±3.7	**21.5±5.7**	−80.6±3.0
15%	9.2±5.9	12.8±4.7	13.7±4.9	12.5±7.1	5.0±6.4	**22.9±4.3**	−110.5±4.4

counter inventories; held objects; agent orientations; and so on. Together these factors yield over 10^{12} states even in the simplest layout, making explicit visitation counting infeasible. (Appendix B.2). We evaluate two rooms of increasing complexity, a simple and a ring-shaped room. Observations encode the active recipe, allowing the agent to distinguish tasks. Covariate shift arises from recipe changes, which alter the required soup, and layout changes, reflected in differences in available ingredient dispensers between source and target.

For each room, we construct three transfer setups with progressively increasing shift between teacher and student environments, quantified by the KL divergence between their energy score distributions (Figure 7). **(1) Recipe Shift (2 ⇒ 3):** Both environments have all three ingredients. Source requires onion + tomato soup; target requires onion + tomato + fish soup. This is a pure recipe shift (one additional ingredient). **(2) Recipe Shift (1 ⇒ 3):** Both environments have all three ingredients. Source requires onion soup; target requires onion + tomato + fish soup. This is a larger recipe shift (two additional ingredients). **(3) Recipe + Layout Shift (2 ⇒ 2):** Source has onions and tomatoes and requires onion + tomato soup. Target has tomatoes and fish and requires tomato + fish soup. This combines both recipe shift (different recipe) and layout shift (different available ingredients).

EBTL maintains positive transfer under increasing covariate shift, consistently outperforming all baselines. As covariate shift between the source and target environments increases, transfer becomes more challenging. This is evident in the slower convergence from Recipe (2 ⇒ 3) to Recipe (1 ⇒ 3) in the Ring Room (Figure 7), where the teacher is only familiar with 1 rather than 2 recipes (out of 3 total). Despite this, EBTL yields positive transfer performance by restricting guidance to states associated with recipes that the teacher has encountered during training.

Shared layouts simplify OOD detection. When source and target share the same spatial layout, *i.e.*, Recipe (2 ⇒ 3) and Recipe (1 ⇒ 3), covariate shift arises solely from recipe encoding, producing a clearly bimodal energy distribution that separates ID and OOD states (Figure 7, rows 1 and 2). In contrast, under Recipe + Layout (2 ⇒ 2), the set of available ingredient dispensers differs between source and target, changing the objects present in the state. As a result, even states associated with familiar recipes may contain features rarely observed during teacher training, lowering ID energy scores and blurring the ID/OOD boundary.

Robustness to room complexity. EBTL achieves stable positive transfer across both Overcooked environments with high returns and low variance. In contrast, baselines without selective guidance degrade as layout complexity increases.

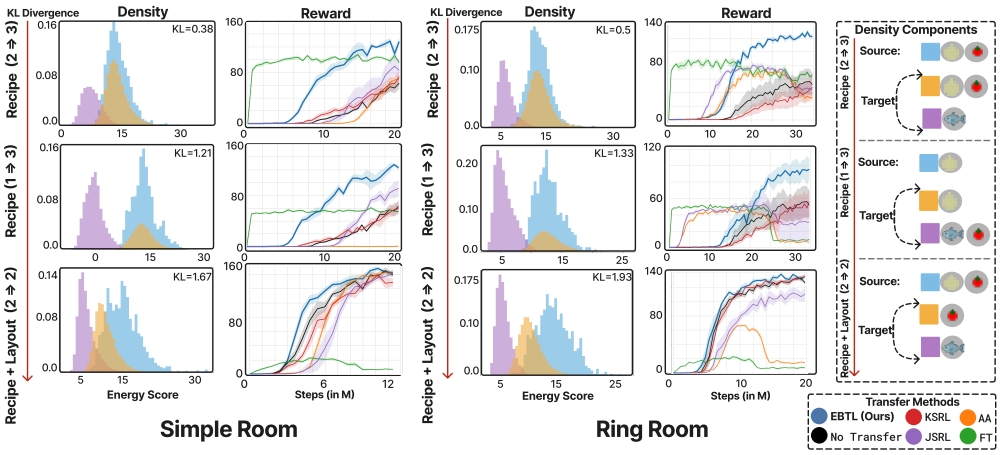

Figure 7: **Overcooked** (5 seeds). **(Cols 1 & 3)** Energy scores: source (blue) vs. target (orange + purple). Target is bimodal: one mode overlaps source (ID), one does not (OOD). **(Cols 2 & 4)** Learning curves: EBTL vs. baselines ($q = 0.5$ for Recipe Shift, $q = 0.2$ for Recipe + Layout Shift).

Action advising (AA) becomes unstable over training (Figure 7), indicating that over-reliance on suboptimal advice hinders learning. Fine-tuning often stalls in local minima under covariate shift: misaligned initialization biases early exploration toward low-reward regions, reinforcing bad value estimates and impeding recovery.

Challenge mode. Under very large covariate shifts, EBTL's gating offers less leverage and benefits from a more permissive threshold. EBTL's gating is effective when two conditions hold: (i) the teacher assigns higher energy to transferable target states than to non-transferable ones, and (ii) the transferable mass lies inside the teacher's high-energy region (above threshold τ). When both hold, q can be set anywhere in a broad range and useful guidance is admitted. The $2 \to 2$ Overcooked transfer is the boundary case: condition (i) still holds visibly, but condition (ii) breaks because both the recipe and the kitchen layout change between source and target. The energy-score histograms in Figure 7 make the cause visible: the $2 \to 2$ target distribution (orange and purple) sits substantially to the left of the teacher's in-distribution mass (blue), so nearly every target state is classified as out-of-distribution. We therefore configure the $2 \to 2$ transfer with $q = 0.2$ rather than the $q = 0.5$ used in all other Overcooked settings: at $q = 0.5$ the threshold is calibrated against the in-distribution mass and almost no states clear it, leaving the teacher essentially silent; relaxing to $q = 0.2$ admits a useful fraction of guidance. This pattern generalizes: as the covariate shift between source and target becomes large, q must be lowered to admit any guidance at all. EBTL's selective gating is therefore effective at handling covariate shift, but only up to a point. Once the target distribution barely overlaps with the source, the separation between transferable and non-transferable states collapses, and the selectivity that benefits EBTL in moderate-shift regimes provides less leverage. We view this as a known boundary of the method.

6 Conclusion and Future Directions

We introduced energy-based transfer learning (EBTL), which improves sample efficiency by selectively issuing teacher guidance based on an energy-based familiarity signal, without requiring additional network training or handcrafted OOD detectors. Empirically, EBTL consistently outperforms prior baselines across both single-task and multi-task transfer settings. More broadly, this work provides an initial step toward understanding how OOD detection can support transfer in reinforcement learning by linking energy to state-visitation patterns under on-policy training. A key limitation is that this connection may weaken in off-policy settings, where visitation no longer aligns with policy updates. Future work may explore more effective OOD detection methods for improved distributional separation, as well as extensions to multi-teacher settings where guidance is selected dynamically based on estimated familiarity.

REFERENCES

David Abel, Yuu Jinnai, Sophie Yue Guo, George Konidaris, and Michael Littman. Policy and value transfer in lifelong reinforcement learning. In *International conference on machine learning*, pp. 20–29. PMLR, 2018.

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.

André Barreto, Diana Borsa, Shaobo Hou, Gheorghe Comanici, Eser Aygün, Philippe Hamel, Daniel Toyama, Shibl Mourad, David Silver, Doina Precup, et al. The option keyboard: Combining skills in reinforcement learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Avinandan Bose, Simon Shaolei Du, and Maryam Fazel. Offline multi-task transfer rl with representational penalization. *arXiv preprint arXiv:2402.12570*, 2024.

Albert Bou, Matteo Bettini, Sebastian Dittert, Vikash Kumar, Shagun Sodhani, Xiaomeng Yang, Gianni De Fabritiis, and Vincent Moens. Torchrl: A data-driven decision-making library for pytorch, 2023.

Tim Brys, Anna Harutyunyan, Matthew E Taylor, and Ann Nowé. Policy transfer using reward shaping. In *AAMAS*, pp. 181–188, 2015.

Joseph Campbell, Yue Guo, Fiona Xie, Simon Stepputtis, and Katia Sycara. Introspective action advising for interpretable transfer learning. In *Conference on Lifelong Learning Agents*, pp. 1072–1090. PMLR, 2023.

Micah Carroll, Rohin Shah, Mark K Ho, Tom Griffiths, Sanjit Seshia, Pieter Abbeel, and Anca Dragan. On the utility of learning about humans for human-ai coordination. *Advances in neural information processing systems*, 32, 2019.

Maxime Chevalier-Boisvert, Bolun Dai, Mark Towers, Rodrigo Perez-Vicente, Lucas Willems, Salem Lahlou, Suman Pal, Pablo Samuel Castro, and Jordan Terry. Minigrid & miniworld: Modular & customizable reinforcement learning environments for goal-oriented tasks. In *Advances in Neural Information Processing Systems 36, New Orleans, LA, USA*, December 2023.

Fernando Fernández and Manuela Veloso. Probabilistic policy reuse in a reinforcement learning agent. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pp. 720–727, 2006.

Matthias Hein, Maksym Andriushchenko, and Julian Bitterwolf. Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 41–50, 2019.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, Fujie Huang, et al. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*, 2017.

Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. *Advances in neural information processing systems*, 33:21464–21475, 2020.

Jelle Luijkx, Zlatan Ajanović, Laura Ferranti, and Jens Kober. Askdagger: active skill-level data aggregation for interactive imitation learning. *arXiv preprint arXiv:2508.05310*, 2025.

Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Icml*, volume 99, pp. 278–287. Citeseer, 1999.

Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 427–436, 2015.

Benjamin Rosman and Subramanian Ramamoorthy. What good are actions? accelerating learning using learned action priors. In *2012 IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL)*, pp. 1–6, 2012. doi: 10.1109/DevLrn.2012.6400810.

Stephane Ross and J Andrew Bagnell. Reinforcement and imitation learning via interactive no-regret learning. *arXiv preprint arXiv:1406.5979*, 2014.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

Andrei A Rusu, Sergio Gomez Colmenarejo, Caglar Gulcehre, Guillaume Desjardins, James Kirkpatrick, Razvan Pascanu, Volodymyr Mnih, Koray Kavukcuoglu, and Raia Hadsell. Policy distillation. *arXiv preprint arXiv:1511.06295*, 2015.

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.

Andreas Sedlmeier, Robert Müller, Steffen Illium, and Claudia Linnhoff-Popien. Policy entropy for out-of-distribution classification. In *International Conference on Artificial Neural Networks*, pp. 420–431. Springer, 2020.

Richard S. Sutton and Andrew G. Barto. *Reinforcement learning - an introduction*. Adaptive computation and machine learning. MIT Press, 1998. ISBN 978-0-262-19398-6. URL http://www.incompleteideas.net/book/first/the-book.html.

Matthew E Taylor and Peter Stone. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10(7), 2009.

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.

Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big data*, 3:1–40, 2016.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pp. 1094–1100. PMLR, 2020.

Jiaru Zhang, Manav Gagvani, Can Cui, Juntong Peng, Ruqi Zhang, and Ziran Wang. Efficient and explainable end-to-end autonomous driving via masked vision-language-action diffusion. *arXiv preprint arXiv:2602.20577*, 2026.

Zewei Zhou, Tianhui Cai, Seth Z Zhao, Yun Zhang, Zhiyu Huang, Bolei Zhou, and Jiaqi Ma. Autovla: A vision-language-action model for end-to-end autonomous driving with adaptive reasoning and reinforcement fine-tuning. *arXiv preprint arXiv:2506.13757*, 2025.

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 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 Jie Tan, Marc Toussaint, and Kourosh Darvish (eds.), *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 2165–2183. PMLR, 06–09 Nov 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html.

APPENDIX

A SUPPORTING PROOF

A.1 PROOF OF PROPOSITION 1

Proof. Let $\pi_\theta(a \mid s) = \mathrm{softmax}(f_\theta(s)/T)_a$ with logits $f_\theta(s) \in \mathbb{R}^K$ for K actions and temperature $T > 0$, and write $\pi_j = \pi_\theta(j \mid s)$ for brevity. Let $A(s,a) = Q(s,a) - V(s)$ denote the advantage of action a in state s, where $Q(s,a)$ and $V(s)$ are the action-value and state-value functions, respectively.

Logit update. A policy gradient step on $A(s,a) \log \pi_\theta(a \mid s)$ for a sampled action a updates each logit by

$$\Delta f_j = \frac{\eta\, A(s,a)}{T} \big[\mathbf{1}_{j=a} - \pi_j\big],$$

where $\eta > 0$ is the step size and $\mathbf{1}_{j=a}$ is the indicator function, increasing the logit of a relative to all others when $A(s,a) > 0$.

Energy change. The free energy and score are defined as

$$E_\theta(s) = -T \log \sum_{j=1}^{K} e^{f_\theta(s)_j/T}, \qquad \phi_\theta(s) = -E_\theta(s).$$

Since $\frac{\partial E_\theta(s)}{\partial f_j} = -\pi_j$, the first-order change in the score is

$$\begin{aligned}
\Delta\phi_\theta(s) &= -\Delta E_\theta(s) \\
&= -\sum_{j=1}^{K} \frac{\partial E_\theta(s)}{\partial f_j} \cdot \Delta f_j \\
&= \frac{\eta\, A(s,a)}{T} \sum_{j=1}^{K} \pi_j \big[\mathbf{1}_{j=a} - \pi_j\big] \\
&= \frac{\eta\, A(s,a)}{T} \left[\pi_a - \sum_{j=1}^{K} \pi_j^2\right].
\end{aligned}$$

The sign of $\Delta\phi_\theta(s)$ depends on $A(s,a)$ and the bracket term $\big[\pi_a - \sum_j \pi_j^2\big]$.

Non-negativity towards convergence. Let $a^* = \arg\max_a Q(s,a)$ denote the optimal action. Towards convergence, the policy concentrates mass on a^*, so we condition on the high-probability event $a = a^*$, under which $\pi_{a^*} = \max_j \pi_j$.

For the bracket term, $\pi_{a^*} \geq \pi_j$ for all j implies

$$\sum_{j=1}^{K} \pi_j^2 \;\leq\; \pi_{a^*} \sum_{j=1}^{K} \pi_j \;=\; \pi_{a^*},$$

where the last equality uses $\sum_j \pi_j = 1$, and hence $\big[\pi_{a^*} - \sum_j \pi_j^2\big] \geq 0$.

For the advantage term, since $a^* = \arg\max_a Q(s,a)$,

$$A(s,a^*) \;=\; Q(s,a^*) - V(s) \;=\; \sum_{a \neq a^*} \pi(a \mid s) \big[Q(s,a^*) - Q(s,a)\big] \;\geq\; 0,$$

as $Q(s,a^*) \geq Q(s,a)$ for all a by definition of a^*.

Therefore, conditioning on $a = a^*$, both factors are non-negative and $\Delta\phi_\theta(s) \geq 0$. Since EBTL assumes the teacher has converged prior to transfer (Assumption 1), this condition is satisfied for the teacher at deployment time, and policy gradient updates reliably increase the energy score on visited states. □

B Training Details

B.1 GridWorld

Reward Structure and Action Masking. In the MiniGrid experiments, agents are trained under a sparse reward setting: a reward of 1 is given only when the agent successfully reaches the goal location. No shaped or intermediate rewards are provided, making the task highly exploration-dependent. To mitigate the resulting challenge and accelerate learning, we apply action masking to dynamically restrict the agent's action space based on its immediate environment. The action mask disables irrelevant or invalid actions at each timestep: (1) the *forward* action is masked out if the agent is facing a wall, preventing redundant collisions; (2) the *pickup* action is disabled unless the agent is directly facing a key; (3) the *toggle* action is masked out unless the agent is facing a door; (4) the *drop* action is always disabled, as object dropping is unnecessary in our tasks; and (5) the *done* action is permanently disabled, since it is not used in our environments. This selective pruning of the action space reduces the likelihood of unproductive behavior and enables the agent to focus on learning goal-directed policies more effectively.

Teacher Training. In both experimental setups, we train two variants of the teacher policy using standard Proximal Policy Optimization (PPO) in the source environment: one with the energy-based loss and one without. For the teacher trained with energy loss, the m_{in} and m_{out} are set to 10 and 15 respectively. These values are chosen arbitrarily, as the separation between energy distributions is insensitive to the exact threshold choice (see Section H.1). The training follows a consistent set of hyperparameters, as detailed in the next section. For the *unlocked-to-locked* environment, 800K-step checkpoints are selected from both training variants. For the *alternating-goal room* environment, 200K-step checkpoints are used. In the language-conditioned setting, 1.6M-step checkpoints are used for *unlocked-to-locked*, and 2M-step checkpoints for *alternating-goal room*.

Student Training. For each target task, we first train a student policy from scratch using standard PPO without any transfer to establish baseline performance. All experiments in the MiniGrid setups are conducted with 10 random seeds to ensure robustness. Within each domain, the student and teacher policies share the same model architecture.

B.2 Overcooked-AI

State Space Enumeration of Overcooked Consider the Simple Layout illustrated in Figure 2. The grid has dimensions 4×5. The interior region contains 6 traversable tiles where the agent can move. The exterior non-corner tiles are reserved for environment objects (ingredient dispensers, pot, dish dispenser, serving counter), and their placements can be randomized.

We explicitly count states under the *single-agent, lossless* encoding generated by `lossless_state_encoding_single_agent()` in `OvercookedGridWorld`: The resulting state space is a combinator of the following factors.

- **Kitchen-layout permutations.** Among the 14 exterior tiles, 8 are eligible for randomization. We must place 6 *distinct* stations (*onion*, *tomato*, *fish* dispensers; *server*; *pot*; *dish* stack), leaving the remaining 2 tiles as empty counters. The number of distinct assignments is
$$\binom{8}{6} 6! \;=\; {}_8P_6 \;=\; \frac{8!}{2!} \;=\; 20{,}160.$$
- **Agent position and orientation**: The agent may occupy any of the 6 interior tiles and face one of 4 directions (north, south, east, west), for a total of $6 \times 4 = 24$ possibilities.
- **Urgency flag**: Binary indicator with 2 possibilities, set to 1 if the remaining horizon is less than 40, and 0 otherwise.
- **Active recipe**: 3 possibilities, indicating the current recipe type (onion, tomato, or fish).
- **Pot state (mode + contents/recipe/timer)**: Idle with $k \in \{0, 1, 2\}$ ingredients (order irrelevant): $\sum_{k=0}^{2} \binom{k+2}{2} = 1 + 3 + 6 = 10$. With 3 ingredients, the pot is *cooking*: there are $\binom{5}{3} = 10$ recipe multisets, each with a remaining time in $\{1, \ldots, 20\}$, giving $10 \times 20 = 200$ states. When cooking finishes, it is *done* with one of the same 10 recipes. Total $= 10 + 200 + 10 = 220$.
- **Agent hand**: The agent may hold (i) nothing, (ii) an empty dish, (iii) a finished soup (all 3 slots filled with a combination of onion, tomato, and fish), or a single raw ingredient (onion, tomato, or fish). The number of distinct soup types is $\binom{3+3-1}{3} = \binom{5}{3} = 10$. Hence the total possibilities are
$$1 \text{ (nothing)} \;+\; 1 \text{ (empty dish)} \;+\; 10 \text{ (soup types)} \;+\; 3 \text{ (single ingredient)} = 15.$$

- **Counter items (2 exterior counters)**: Each counter has 15 options (empty; three ingredients; empty dish; ten soup types), giving 15^2 overall.

Multiplying the independent factors above gives:

$$|\mathcal{S}| = \underbrace{20{,}160}_{\text{layout}} \times \underbrace{24}_{\text{agent pos/orient}} \times \underbrace{2}_{\text{urgency}} \times \underbrace{3}_{\text{active recipe}} \times \underbrace{220}_{\text{pot state}} \times \underbrace{15}_{\text{agent hand}} \times \underbrace{15^2}_{\text{two counters}}$$
$$= 2{,}155{,}507{,}200{,}000 \approx 2.16 \times 10^{12}.$$

This already conservative count highlights why explicit state-visitation is infeasible for the teacher model; more complex layouts such as the *Ring* further enlarge the state space.

Reward Structure. In all Overcooked setups, no action masking is applied. Instead, shaped rewards are introduced to facilitate the training process. A shaped reward of 3 is given when the correct ingredient is added to a pot. An additional reward of 3 is awarded when a dish is picked up—provided there are no dishes already on the counter and the soup is either cooking or completed. A reward of 5 is granted when the soup is picked up. Furthermore, a shaped reward of 3 is given upon delivering the soup, regardless of whether it matches the currently active recipe. All shaped rewards follow a predefined linear decay schedule. In contrast, a sparse reward of 20 is awarded when the delivered soup matches the active recipe; this reward does not decay over time.

Teacher Training. In all Overcooked setups, teacher policies are trained in the source environment using standard Proximal Policy Optimization (PPO) with hyperparameters described in the following section. For each setup and source-target configuration, a specific checkpoint is selected to serve as the teacher for transfer. The table below lists the selected training step (in environment steps) corresponding to each teacher checkpoint.

Table 2: Selected teacher checkpoints (in environment steps) for each Overcooked setup and source-target configuration.

Setup	**Recipe (2 → 3)**	**Recipe (1 → 3)**	**Recipe + Layout (2 → 2)**
Simple Room	19,008,000	9,004,800	12,000,000
Ring Room	2,400,000	10,003,200	18,000,000

Student Training. In all Overcooked setups, student policies are trained in the target environment using PPO under a fixed transfer horizon. For the teacher trained with energy loss, the m_{in} and m_{out} are set to 12 and 14 respectively. The training is conducted using consistent hyperparameters, as detailed in the next section. All experiments are repeated with 5 random seeds to ensure stability and reproducibility. The transfer horizon varies depending on the setup and source-target configuration. The table below summarizes the number of environment steps used during student training for each case:

Table 3: Transfer horizons (in millions of environment steps) used for student training in each Overcooked setup and configuration. Each experiment is run with 5 random seeds.

Setup	**Recipe (2 → 3)**	**Recipe (1 → 3)**	**Recipe + Layout (2 → 2)**
Simple Room	20M	20M	12M
Ring Room	35M	35M	20M

C Hyperparameters

All experiments use teacher and student policies trained with TorchRL's proximal policy optimization (PPO) (Bou et al., 2023).

C.1 GridWorld

All experimental setups in GridWorld are trained using a fixed set of PPO hyperparameters, summarized in Table 4. These settings remain consistent across all teacher and student training runs within the domain. For the language-conditioned variant, we use a learning rate of 0.0001, while keeping all other hyperparameters identical.

Table 4: Hyperparameters used for all GridWorld experiments.

Hyperparameter	Value
Learning rate	0.0005
Discount factor (γ)	0.9
GAE lambda (λ)	0.8
Policy clip parameter	0.2
Value function clip parameter	10.0
Value loss coefficient	0.5
Entropy coefficient	0.01
Train batch size	256
SGD minibatch size	128
Number of SGD iterations	4
Number of parallel environments	8
Normalize advantage	False

C.2 Overcooked-AI

All Overcooked experiments use a shared set of core PPO hyperparameters, listed in table 5. These settings are consistent across teacher and student training. However, the learning rate and reward shaping horizon vary depending on the layout and recipe configuration, summarized in table 6. We use the following notation: O = Onion, T = Tomato, F = Fish, OT = Onion + Tomato, TF = Tomato + Fish, OTF = Onion + Tomato + Fish.

Table 5: Shared PPO hyperparameters across all Overcooked experiments.

Hyperparameter	Value
Discount factor (γ)	0.99
GAE lambda (λ)	0.6
KL coeff	0.0
Reward clipping	False
Clip parameter	0.2
VF clip parameter	10.0
VF loss coeff	0.5
Entropy coeff	0.1
Train batch size	9600
SGD minibatch size	1600
SGD iterations	8
Parallel envs	24
Normalize advantage	False

Table 6: Setup-specific learning rates and reward shaping horizons.

Layout	Config	LR	Horizon
Simple	Recipe (O)	0.001	8M
	Recipe (OT)	0.001	15M
	Recipe (OTF)	0.001	25M
	Recipe + Layout (OT)	0.001	10M
	Recipe + Layout(TF)	0.001	10M
Ring	Recipe (O)	0.0006	10M
	Recipe (OT)	0.0006	20M
	Recipe (OTF)	0.0006	30M
	Recipe + Layout (OT)	0.0006	15M
	Recipe + Layout (TF)	0.0006	15M

D Baselines

D.1 Baseline Implementation

To ensure a fair comparison, we implement all baselines and our method within a unified training pipeline. All experiments use teacher and student policies trained with TorchRL's proximal policy optimization (PPO) (Bou et al., 2023). For methods where the teacher directly overwrites the student's action (e.g., AA and JSRL (Uchendu et al., 2023)), the resulting actions are off-policy with respect to the student. To ensure a valid comparison, we apply importance sampling correction to account for this mismatch. This allows us to isolate the effect of the teacher's guidance quality, rather than confounding factors introduced by off-policy data. For KSRL (Schmitt et al., 2018), which introduces an additional imitation objective, we incorporate the auxiliary loss on top of the standard PPO loss, following the formulation in the original work.

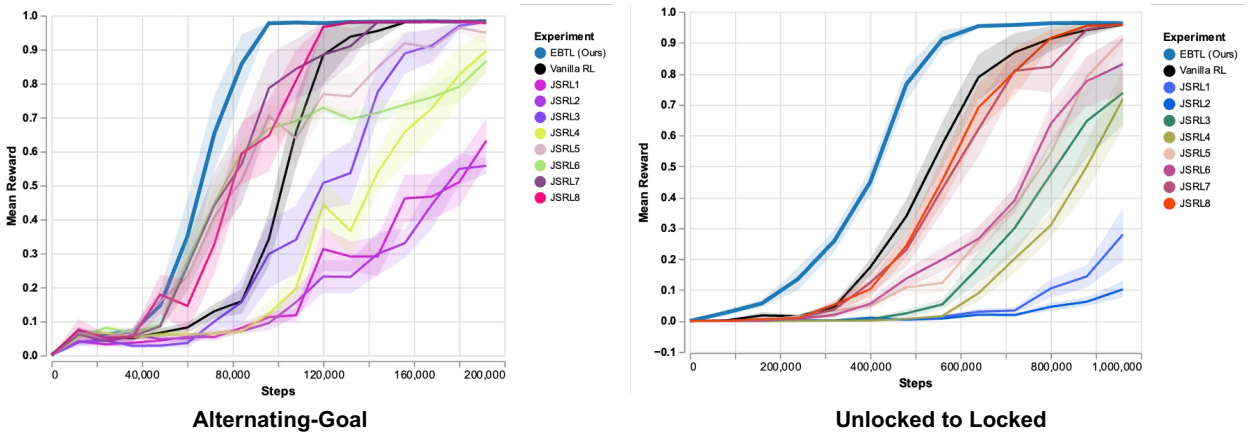

Figure 8: **10 seeds. JSRL hyperparameter tuning in MiniGrid.**

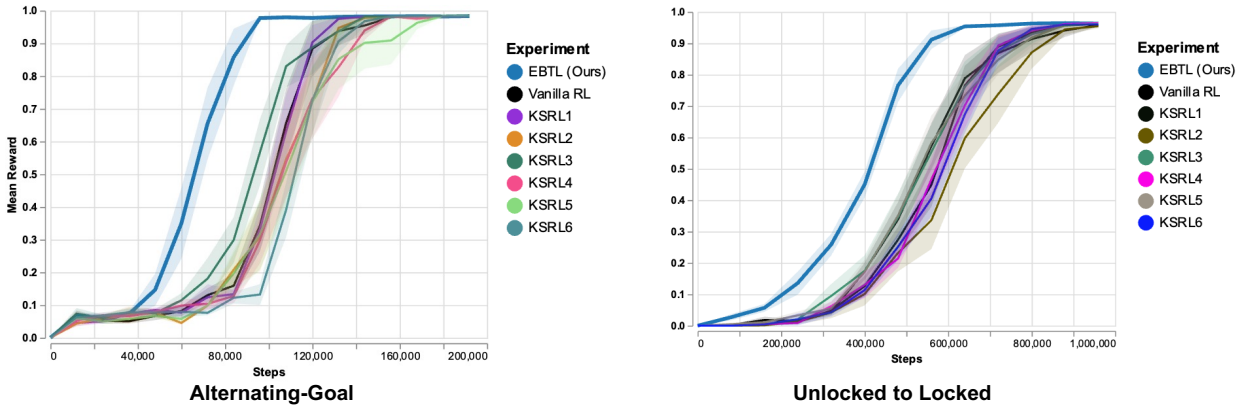

Figure 9: **10 seeds. KSRL hyperparameter tuning in MiniGrid.**

D.2 Hyperparameter Tuning for Baselines

JSRL Experiments Details. JumpStart RL (JSRL) (Uchendu et al., 2023) is a curriculum-based transfer method with three key hyperparameters: the number of stages, the tolerance for stage transition, and the initial number of advising steps. The curriculum progresses based on the student's evaluation performance rather than training steps alone: the policy is periodically evaluated, and a stage transition is triggered when the moving average of recent evaluations is within a specified tolerance of the previous best performance.

JSRL does not introduce additional trainable networks, but it requires repeated policy evaluation throughout training, increasing overall wall-clock time. In our experiments, we use 50 evaluation timepoints over the full training horizon and evaluate JSRL across a grid of its three key hyperparameters: the number of curriculum stages, the tolerance for stage transition, and the initial number of advising steps. Specifically, we sweep stages in $\{10, 30\}$, tolerance in $\{0, 0.05\}$, and initial advising steps in $\{50, 200\}$, resulting in eight configurations (JSRL1–JSRL8), as shown in Figure 8. We note that JSRL exhibits large variance across hyperparameter settings. We therefore report JSRL3 in the main paper as it represents the average performance case across configurations.

- **JSRL1**: stages = 30, initial advising steps = 200, tolerance = 0.05
- **JSRL2**: stages = 30, initial advising steps = 200, tolerance = 0.0
- **JSRL3**: stages = 10, initial advising steps = 200, tolerance = 0.05
- **JSRL4**: stages = 10, initial advising steps = 200, tolerance = 0.0
- **JSRL5**: stages = 30, initial advising steps = 50, tolerance = 0.05
- **JSRL6**: stages = 30, initial advising steps = 50, tolerance = 0.0

- **JSRL7**: stages = 10, initial advising steps = 50, tolerance = 0.05
- **JSRL8**: stages = 10, initial advising steps = 50, tolerance = 0.0

JSRL Results. Across all settings, JSRL performs poorly. This behavior is consistent with its design: since JSRL schedules guidance purely based on training progress and does not consider whether the current state lies within the teacher's training distribution, it may continue to provide guidance in mismatched states, leading to degraded transfer performance.

KSRL Hyperparameter Tuning. Kickstarting RL (KSRL) (Schmitt et al., 2018) has two key hyperparameters: the imitation weight λ, which controls the strength of the auxiliary distillation loss, and its decay schedule over training. We follow the original paper and evaluate $\lambda \in \{1, 2\}$.

KSRL Hyperparameter Configurations. Kickstarting RL (KSRL) (Schmitt et al., 2018) has two key hyperparameters: the imitation weight λ and its decay schedule. The original work considers both linear decay and Population-Based Training (PBT). As reported in Table 2 of (Schmitt et al., 2018), PBT achieves performance comparable to linear decay, while certain linear schedules that terminate imitation early can even outperform it. To ensure consistency across methods, we adopt linear decay in all experiments. For AlternatingGoal (200K training horizon), the imitation loss decays to zero at 30%, 50%, and 70% of training. For Unlocked-to-Locked (1M horizon), the decay ends at 25%, 50%, and 75% of training. These schedules match those used in our method's hyperparameter tuning (Figure 3). We evaluate $\lambda \in \{1, 2\}$, resulting in six configurations (KS1–KS6). As all configurations exhibit similar performance (Figure 9), we randomly choose the setting with $\lambda = 1.0$ in the main paper as the representative KSRL configuration.

- **KS1**: $\lambda = 1.0$, decay ends at 30% (AlternatingGoal) / 25% (Unlocked-to-Locked)
- **KS2**: $\lambda = 1.0$, decay ends at 50% (both settings)
- **KS3**: $\lambda = 1.0$, decay ends at 70% (AlternatingGoal) / 75% (Unlocked-to-Locked)
- **KS4**: $\lambda = 2.0$, decay ends at 30% (AlternatingGoal) / 25% (Unlocked-to-Locked)
- **KS5**: $\lambda = 2.0$, decay ends at 50% (both settings)
- **KS6**: $\lambda = 2.0$, decay ends at 70% (AlternatingGoal) / 75% (Unlocked-to-Locked)

KSRL Results. As shown in Figure 9, KSRL consistently underperforms EBTL across both transfer settings. On average, KSRL achieves performance comparable to training from scratch (Vanilla RL) and fails to provide meaningful transfer gains. Across different hyperparameter configurations (KS1–KS6), KSRL exhibits limited sensitivity to the choice of λ and decay schedule, and none of the settings reach the performance level of EBTL. This suggests that uniformly applying imitation, even with a decaying schedule, is insufficient under distribution shift, as it does not account for whether the teacher's behavior is reliable in the current state.

E Continuous Action Space

EBTL also applies to continuous control: the action space can be discretized into bins (as in OpenVLA (Kim et al., 2024)), or an energy score can be defined directly in the continuous domain. Appendix F provides the full continuous formulation (C-EBTL). In this case, the teacher policy outputs a mean and diagonal covariance for each state s, and the energy score is derived from the log-determinant of the covariance: $\phi(s) = -E(s) = \frac{1}{2} \log |\Sigma(s)| = \sum_{i=1}^{D} \log \sigma_i(s)$, where $\sigma_i(s)$ is the standard deviation of the i-th action dimension. We evaluate on Meta-World(Yu et al., 2020), a suite of continuous-control manipulation tasks for a simulated Sawyer arm. Both state and action spaces are continuous; the action space is four-dimensional, consisting of three Cartesian end-effector displacements plus a gripper command. We consider two settings: window manipulation (Window Open, Window Close) and button manipulation (Button Press, Button Press Down). In the window setting the teacher is trained on Window Open; in the button setting the teacher is trained on Button Press. In each setting, the *student must learn both tasks in that pair*.

Energy score separates ID from OOD. As shown in Figure 10, in both environments the distributions are clearly bimodal: states from the unseen target subtask cluster at a lower mode, while states from the teacher's training subtask occupy a higher one. Across both settings, the OOD mode concentrates near -2.8. This follows from the floor we impose on the policy standard deviations for numerical stability in the network, $\sigma_i \geq 0.5$, which implies $\phi(s) = \sum_{i=1}^{4} \log \sigma_i(s) \approx -2.8$. In contrast, the ID energies differ between settings, consistent with Proposition 3.1: when a state admits many compatible actions, on-policy updates raise $\phi_T(s)$. Window manipulation allows more compatible

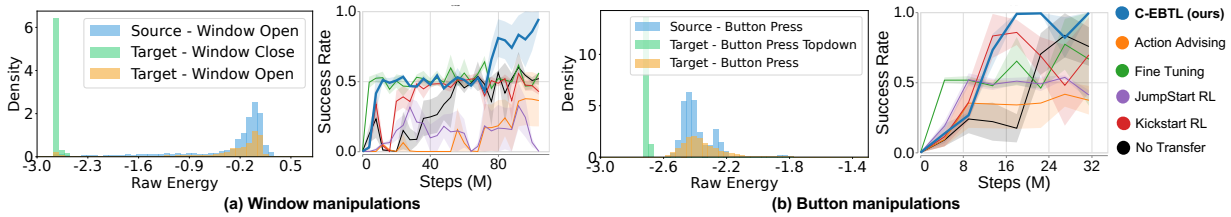

Figure 10: Results (3 seeds). In each environment, the left column shows the empirical density of the teacher's state energy ϕ. Blue traces the source rollouts used to train the teacher. Orange and green are target rollouts during transfer and typically form a bimodal pattern: one component overlaps the source shared task (in distribution), while the other occupies a separate region (non-shared task, out of distribution). q is set to 0.3 in both settings.

actions (approach the handle, then slide to the target), so its ID energies are higher. Button pressing requires precise alignment and actuation, so fewer actions are compatible and the ID energies remain lower than in window tasks, but still above the OOD mode.

F Energy formulation on Continuous Energy Score

F.1 Continuous Control

In robotic control, the action space $\mathcal{A} \subseteq \mathbb{R}^n$ is continuous and n-dimensional, with each component a_i corresponding to a separate control command. A common parameterization is a diagonal Gaussian policy, wherein the actor network outputs two vectors $\mu(s) = [\mu_1(s), \ldots, \mu_n(s)]^\top$ and $\sigma(s) = [\sigma_1(s), \ldots, \sigma_n(s)]^\top$, and the policy density is given by

$$p(a \mid s) = \frac{\exp\!\big(-\frac{1}{2}\,(a-\mu(s))^\top \Sigma(s)^{-1}(a-\mu(s))\big)}{(2\pi)^{n/2}\,|\Sigma(s)|^{1/2}} \tag{3}$$

where $\Sigma(s) = \mathrm{diag}(\sigma_1^2(s), \ldots, \sigma_n^2(s))$. Equivalently, $\pi_\theta(a \mid s) = \mathcal{N}\big(a;\, \mu(s),\, \Sigma(s)\big)$, so each action dimension is sampled as $a_i \sim \mathcal{N}(\mu_i(s), \sigma_i^2(s))$.

An energy-based model defines a joint energy function $E(s, y)\colon \mathcal{S} \times \mathcal{Y} \to \mathbb{R}$, which assigns a scalar energy to each state–output pair (s, y)(LeCun et al., 2006). From this one obtains the conditional Gibbs distribution

$$p(y \mid s) = \frac{\exp\big(-E(s,y)\big)}{Z(s)}, \quad Z(s) = \int_{\mathcal{Y}} \exp\big(-E(s,y')\big)\,\mathrm{d}y'. \tag{4}$$

where $Z(s)$ is the state-conditional partition function. Here, we derive a continuous-action energy formulation for diagonal Gaussian policies from Equation 3 and Equation 4. We define the joint energy as

$$E(s,a) = \tfrac{1}{2}\,(a-\mu(s))^\top\,\Sigma(s)^{-1}\,(a-\mu(s)). \tag{5}$$

The corresponding state-conditional partition function is

$$Z(s) = \int_{\mathbb{R}^D} \exp\big(-E(s,a')\big)\,\mathrm{d}a' = (2\pi)^{D/2}\,|\Sigma(s)|^{1/2}. \tag{6}$$

Connecting Equation 5, Equation 6, and Equation 4 and marginalizing out a yields the Helmholtz free energy $E(s) = -\log Z(s) = -\frac{D}{2}\log(2\pi) - \frac{1}{2}\log|\Sigma(s)|$. Since the additive constant $-\frac{D}{2}\log(2\pi)$ does not affect ranking, we drop it and define the simplified energy score as the negative of free energy:

$$\phi(s) = -E(s) = \tfrac{1}{2}\log|\Sigma(s)| = \sum_{i=1}^{D} \log \sigma_i(s). \tag{7}$$

F.2 Energy Score as OOD detector

Energy score in continuous action spaces. For a diagonal Gaussian policy $\pi_\theta(a \mid s) = \mathcal{N}(a;\, \mu_\theta(s), \Sigma_\theta(s))$ with $\Sigma_\theta(s) = \mathrm{diag}(\sigma_1^2(s), \ldots, \sigma_D^2(s)) \succ 0$, we define the state energy score as

$$\phi_\theta(s) = \tfrac{1}{2}\log|\Sigma_\theta(s)| = \sum_{i=1}^{D} \log \sigma_i(s). \tag{8}$$

Proposition 2 (Monotonicity under advantage-weighted updates)**.** *Fix a state s and a diagonal Gaussian policy $\pi_\theta(a \mid s)$. Consider a fixed batch $\mathcal{D}_s$ collected under the current policy and the surrogate objective*

$$\mathcal{L}_s(\theta) = \mathbb{E}_{a\sim\mathcal{D}_s}\big[\, w(s,a)\, \log \pi_\theta(a \mid s)\,\big], \tag{9}$$

where $w(s,a) \geq 0$ and $W = \mathbb{E}[w] > 0$.

Define the advantage-weighted mean and variance along coordinate i:

$$\bar{a}_{w,i} = \frac{\mathbb{E}[w a_i]}{W}, \qquad v_{w,i} = \frac{\mathbb{E}[w(a_i - \bar{a}_{w,i})^2]}{W}. \tag{10}$$

Let θ^+ denote the parameters after one gradient ascent step on $\mathcal{L}_s$ with step size $\eta > 0$. If

$$v_{w,i} \geq \sigma_i^2(s) \quad \forall i, \qquad v_{w,j} > \sigma_j^2(s) \text{ for some } j, \tag{11}$$

then

$$\phi_{\theta^+}(s) > \phi_\theta(s). \tag{12}$$

Proof. Parameterize $\theta_i = \log \sigma_i^2$. The gradients of $\mathcal{L}_s$ are

$$\frac{\partial \mathcal{L}_s}{\partial \mu_i} = \frac{W}{\sigma_i^2}(\bar{a}_{w,i} - \mu_i), \qquad \frac{\partial \mathcal{L}_s}{\partial \theta_i} = \frac{W}{2}\left(\frac{v_{w,i} + (\bar{a}_{w,i} - \mu_i)^2}{\sigma_i^2} - 1\right). \tag{13}$$

Let

$$\Delta_i = \frac{v_{w,i} + (\bar{a}_{w,i} - \mu_i)^2}{\sigma_i^2} - 1. \tag{14}$$

After one gradient ascent step,

$$\theta_i^+ = \theta_i + \eta \frac{W}{2}\Delta_i \quad \Longrightarrow \quad \sigma_i^{2+} = \sigma_i^2 \exp\left(\eta \frac{W}{2}\Delta_i\right). \tag{15}$$

Since $(\bar{a}_{w,i} - \mu_i)^2 \geq 0$, the condition $v_{w,i} \geq \sigma_i^2$ for all i and strict inequality for some j implies

$$\Delta_i \geq 0 \; \forall i, \qquad \Delta_j > 0 \text{ for some } j. \tag{16}$$

Thus $\sigma_i^{2+} \geq \sigma_i^2$ for all i, with strict increase for at least one coordinate. Therefore,

$$|\Sigma_{\theta^+}(s)| > |\Sigma_\theta(s)| \quad \Longrightarrow \quad \phi_{\theta^+}(s) = \tfrac{1}{2}\log|\Sigma_{\theta^+}(s)| > \tfrac{1}{2}\log|\Sigma_\theta(s)| = \phi_\theta(s). \tag{17}$$

□

Frequently visited states receive higher scores. We initialize the policy with log-standard-deviation parameters set to 0 (i.e., $\sigma_i = 1$ for all coordinates), a standard practice in continuous-control to stabilize early exploration (Yu et al., 2020). At the beginning of training, the predicted mean $\mu(s)$ is typically inaccurate, leading to large residuals $(a - \mu(s))^2$ when actions are sampled. Since the variance update is driven by the squared deviation between sampled actions and the current mean, each update at state s tends to increase the estimated variance when the observed spread of compatible actions is large. Under on-policy training, states that are visited more frequently are updated more often, and thus tend to accumulate larger variance estimates over time. As the energy score $\phi(s) = \frac{1}{2}\log|\Sigma(s)|$ increases monotonically with the variance, this leads to higher energy scores for frequently visited states. Furthermore, in continuous action spaces many tasks exhibit aleatoric uncertainty, where multiple actions can yield similar outcomes, resulting in a non-zero variance even after convergence. The final energy score therefore reflects the spread of compatible actions at state s, providing a practical proxy for state familiarity.

G Model Architecture

All MiniGrid experiments share the same model architecture shown in Figure 11a. Similarly, all Overcooked experiments use the architecture in Figure 11b. Due to layout size differences in Overcooked, the dense layer input size is set to 182 for *Simple* layouts and 257 for *Ring* layouts.

For the language-conditioned MiniGrid experiments (Figure 7(a)), the visual backbone is slightly modified to incorporate task embeddings. In the *unlocked-to-locked* setting, we use three convolutional layers with 32–64–96 channels and ReLU activations, producing an 864-dimensional flattened feature, followed by actor and critic heads of sizes 864→128→7 and 864→128→1, respectively. In the *alternating-goal room* setting, a lighter backbone with 16–32–64 channels produces a 576-dimensional feature, followed by single-layer heads 576→7 (actor) and 576→1 (critic). Language conditioning is implemented using fixed 128-dimensional sentence embeddings extracted from `google/embeddinggemma-300m`, which are fused with the visual features via FiLM modulation prior to the final linear layer.

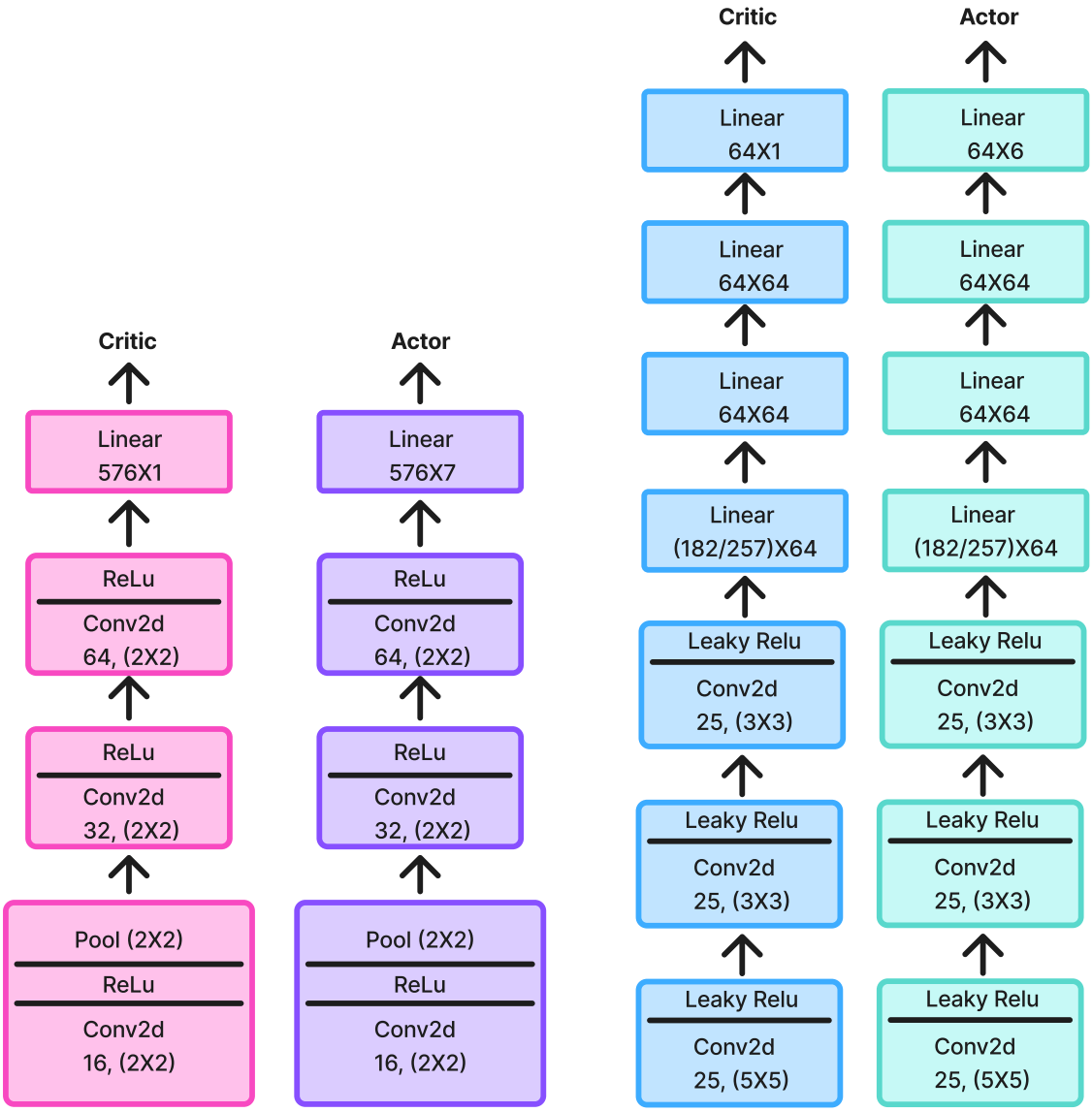

Figure 11: Actor-Critic architectures used in our experiments. (a) MiniGrid. (b) Overcooked.

H Energy-Based Loss

H.1 Effect of Margin Hyperparameters on Separation

We evaluate whether varying the energy thresholds m_{in} and m_{out} affects the teacher's ability to distinguish between false and true out-of-distribution (OOD) states. The energy loss used during training is defined over the energy score $\phi(s) = -E(s)$ as:

$$\mathcal{L}_{\text{energy}} = \mathbb{E}_{\mathbf{s}_{\text{in}} \sim \mathcal{D}_{\text{in}}^{\text{train}}} \left[\left(\max\left(0, m_{\text{in}} - \phi(\mathbf{s}_{\text{in}})\right) \right)^2 \right] + \mathbb{E}_{\mathbf{s}_{\text{out}} \sim \mathcal{D}_{\text{out}}^{\text{train}}} \left[\left(\max\left(0, \phi(\mathbf{s}_{\text{out}}) - m_{\text{out}}\right) \right)^2 \right].$$

Experimental Setup Experiments are conducted in the *GridWorld (unlocked-to-locked)* environment. During training, the in-distribution (ID) set consists of the most recent 3,000 frames collected from the agent's own trajectory. The out-of-distribution (OOD) set is fixed and sampled from 100 episodes of a random policy in the target environment, where the agent is randomly initialized in any room at the start of each episode to ensure unbiased state coverage (rather than being constrained to the upper room). We evaluate six combinations of $(m_{\text{in}}, m_{\text{out}})$ used in the energy

regularization loss (defined over energy scores $\phi(s) = -E(s)$): (10, 15), (5, 10), (15, 20), (10, 10), (15, 15), (12, 14). Each configuration is trained with 5 random seeds using a shared PPO setup and evaluated at the 800,000-step checkpoint.

Sensitivity Evaluation Protocol. We assess whether the teacher consistently distinguishes between *false OOD* states – those similar to ID states and where guidance should be issued – and *true OOD* states – those clearly out-of-distribution and where guidance should be withheld. Both sets are drawn from a fixed OOD dataset collected via a random policy in the target environment. For each $(m_{\text{in}}, m_{\text{out}})$ configuration, we compute the divergence between the energy score distributions of false and true OOD states across three training seeds using Jensen-Shannon divergence, total variation distance, Hellinger distance, and Kullback-Leibler (KL) divergence. To evaluate sensitivity, we apply one-way ANOVA and Kruskal-Wallis tests to determine whether this separation remains consistent across different regularization settings. A high p-value indicates that the teacher's ability to determine when to issue guidance is robust to the choice of $(m_{\text{in}}, m_{\text{out}})$.

Metric	ANOVA p-value	Kruskal–Wallis p-value
Jensen–Shannon	0.1138	0.1592
Kullback–Leibler	0.2457	0.1799
Total Variation	0.1728	0.2322
Hellinger Distance	0.1247	0.1592

Table 7: Statistical test results (p-values) for divergence between False OOD and True OOD energy distributions across different $(m_{\text{in}}, m_{\text{out}})$ settings.

Results. As shown in Table 7, we observe no statistically significant variation in the separation between false and true OOD states across different $(m_{\text{in}}, m_{\text{out}})$ configurations. The ANOVA and Kruskal-Wallis tests yield p-values above 0.1 for all four divergence metrics, indicating that the teacher's ability to distinguish between states where guidance should or should not be issued is stable across regularization settings.

H.2 Choice of ID States

When available, we set the threshold quantile q from the empirical distribution of *in-train* states $\mathcal{D}_{\text{in}}^{\text{train}}$ collected during teacher on-policy learning (with exploration). When the teacher's training distribution is unavailable, we approximate this *post-train* by rolling out the converged teacher (no exploration) and computing q from those states. Because exploration noise is negligible at convergence, these rollouts serve as a reliable proxy for the high-density regions of the teacher's visitation distribution. We validated both choices by running 100 on-policy evaluation episodes and setting q from the resulting state samples across 10 seeds in two GridWorld settings. Tables 8a–8b report student transfer performance (relative to training from scratch) across quantiles.

H.3 Choice of OOD Batches

The role of L_{energy} is to separate states that lie within the teacher's training distribution from those that do not. In the main paper, OOD states are taken from random rollouts of the student environment. This choice was made because they provide a concrete and intuitive illustration of this boundary. To address the concern about future leakage, we conduct a new experiment in the *Unlocked-to-Locked* setting where OOD states are sampled uniformly from the full MiniGrid observation space. This sampling does not assume anything about the target task; it draws from all valid observations that could occur in the domain, and we match the sample size to the original setup for a fair comparison. This removes any possibility of information leakage. Figure 12 reports the results. Performance remains consistent with the main paper, and we continue to observe clear transfer improvements.

H.4 Effect of Energy Regularization on Teacher Convergence

We assess whether adding the energy regularizer to the *teacher* objective harms final performance or slows learning. Across two GridWorld source tasks and 10 seeds, final returns are unchanged, while convergence is faster with the energy term.

We conjecture that the acceleration arises because the energy term adds an inductive bias that highlights which states are in-distribution (high score) versus out-of-distribution (low score), guiding updates toward familiar regions of the state space more efficiently.

Table 8: Choice of ID states for setting the quantile q. "Post-train" derives q from evaluation rollouts of the converged teacher (no exploration); "In-train" derives q from exploration-time states during teacher training. Values are student transfer performance relative to training from scratch (mean $\pm$ 95% CI)

(a) Alternating-Goal Environment

Threshold	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Post-train	-9.8 ± 6.0	21.2 ± 8.8	22.5 ± 9.0	28.9 ± 10.6	33.1 ± 8.0	37.8 ± 6.5	46.9 ± 4.2	37.8 ± 8.7	25.9 ± 12.0	34.7 ± 7.1
In-train	1.9 ± 8.0	36.4 ± 6.9	41.6 ± 4.3	41.2 ± 4.9	46.1 ± 8.2	46.1 ± 4.5	37.3 ± 4.2	40.3 ± 4.7	25.8 ± 7.4	29.2 ± 5.9

(b) UnlockedToLocked Environment

Threshold	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Post-train	11.2 ± 4.6	21.5 ± 4.5	31.1 ± 1.6	29.2 ± 4.8	31.7 ± 5.2	36.8 ± 4.6	40.0 ± 2.8	38.0 ± 3.3	35.8 ± 5.3	24.3 ± 3.9
In-train	16.8 ± 4.4	27.1 ± 3.8	29.3 ± 3.7	31.2 ± 4.8	28.8 ± 3.1	33.6 ± 3.4	35.3 ± 2.5	35.1 ± 2.7	30.7 ± 7.6	32.9 ± 5.8

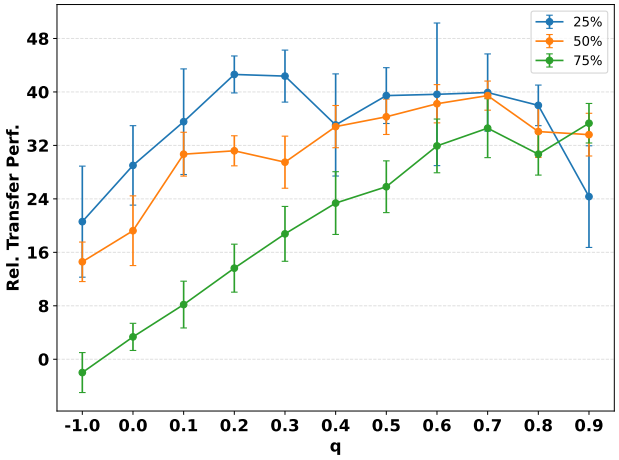

Figure 12: 10 seeds. Effect of replacing rollout-based OOD batches with randomly sampled OOD batches. $q = -1$ advise in all states (AA Baseline).

I Decay Schedules

This section provides the full comparison between linear decay and several budget–based decay mechanisms. Although both approaches aim to reduce teacher influence over training, we found that linear decay is considerably easier to control and more stable across tasks.

All experiments use the *Unlocked-to-Locked* setting (1M steps). We evaluate five variants:

1. **No Decay**: the teacher issues advice whenever the state is classified as in-distribution.
2. **Single Budget (No Reset)**: a fixed budget equal to 10% of the total steps; once depleted, no further advice is allowed. This follows the *single budget* setting in Torrey & Taylor (2013), where the teacher is allocated a finite number of advice calls that are consumed over time. As the budget is exhausted, the frequency of advice implicitly decreases, effectively inducing a decay in teacher intervention without an explicit time-based schedule.
3. **Interval Budget (With Reset)**: every 10,240 steps, the teacher is allowed to spend 10% of that interval as advice.
4. **Interval Budget + Linear Decay**: variant (3) with a linear decay factor applied within each interval.
5. **Linear Decay (Ours)**: a single linear decay schedule over training, ending at the midpoint.

Figure 13 reports the performance of all five schedules. The results show that budget-based schemes, with or without reset, often lead to negative transfer when $q < 0.7$. The issue is not budget exhaustion but that a fixed budget allows

Table 9: Training steps to convergence (mean over 10 seeds; lower is better).

Teacher	Without energy loss	With energy loss
Alternating Room	100,000	60,000
Unlocked-to-Locked	260,000	220,000

too much advice early in training. This limits the student's opportunity to act on its own during the exploration phase, which is important for learning the target task. Because the number of early interventions depends on episode structure and exploration behavior, budget schedules are also difficult to tune.

Linear decay avoids this problem by reducing advice gradually and predictably over time. It prevents excessive early intervention while giving the student increasing autonomy as training progresses, without requiring a fixed allowance of advice.

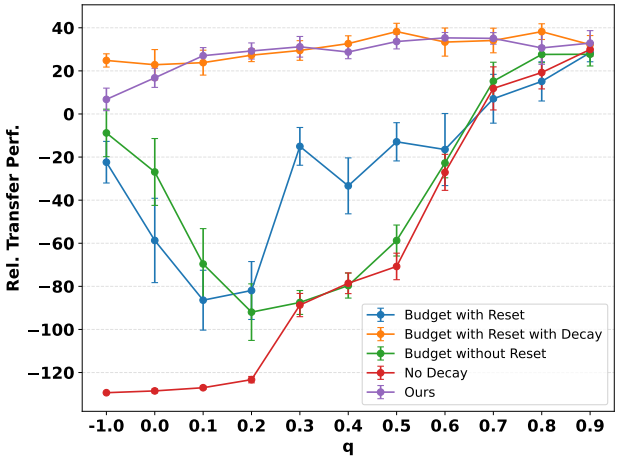

Figure 13: 10 seeds. Comparison of five decay schedules. $q = -1$ advise in all states (AA Baseline).

J Heatmaps of Average Energy Quantiles

We visualize the heatmaps of average energy quantiles under the teacher for Alternating Goal in Figure 14. The teacher assigns higher energy to states seen in training (goal in Room 1, upper-left) than to unseen states (goal in Room 3, lower-right).

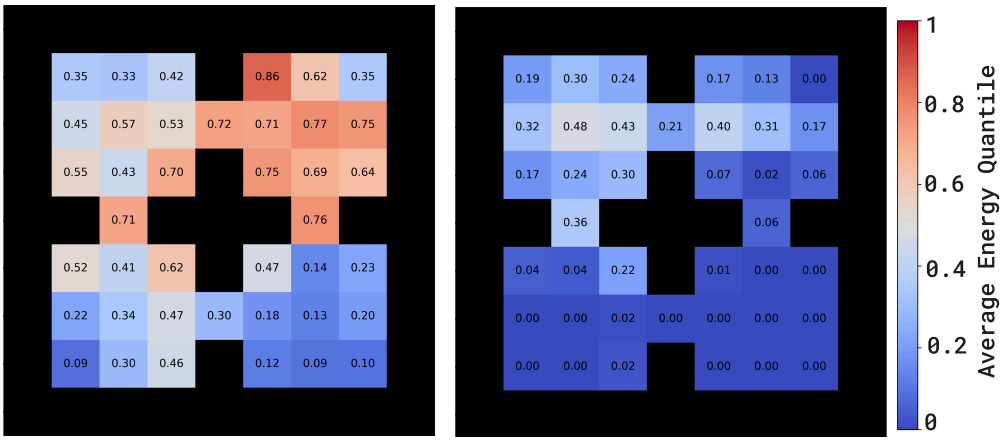

Figure 14: Heatmaps of average energy quantiles under the teacher for Alternating Goal. Left: source (goal in Room 1). Right: target (goal in Room 3). Higher quantiles indicate greater teacher familiarity.

K Advice Rates During Training

This section reports the rate at which the teacher issues advice and the rate at which the student follows that advice (after applying the decay schedule) in the *Unlocked-to-Locked* environment. We evaluate several values of q under multiple decay schedules (25%, 50%, 75% of the training horizon). The result is shown in Figure 15.

As training progresses, the issue-advice rate tends to increase. Once the student begins to solve more of the task on its own, it encounters states that fall within the teacher's training distribution more frequently, and the teacher identifies a larger fraction of states as familiar. The take-advice rate is the product of the issue-advice rate and the decay factor, so it decreases over time even as the teacher becomes more confident. These measurements describe how the influence of the teacher evolves throughout training and show that EBTL creates a controlled shift from teacher-guided actions to fully student-driven behavior.

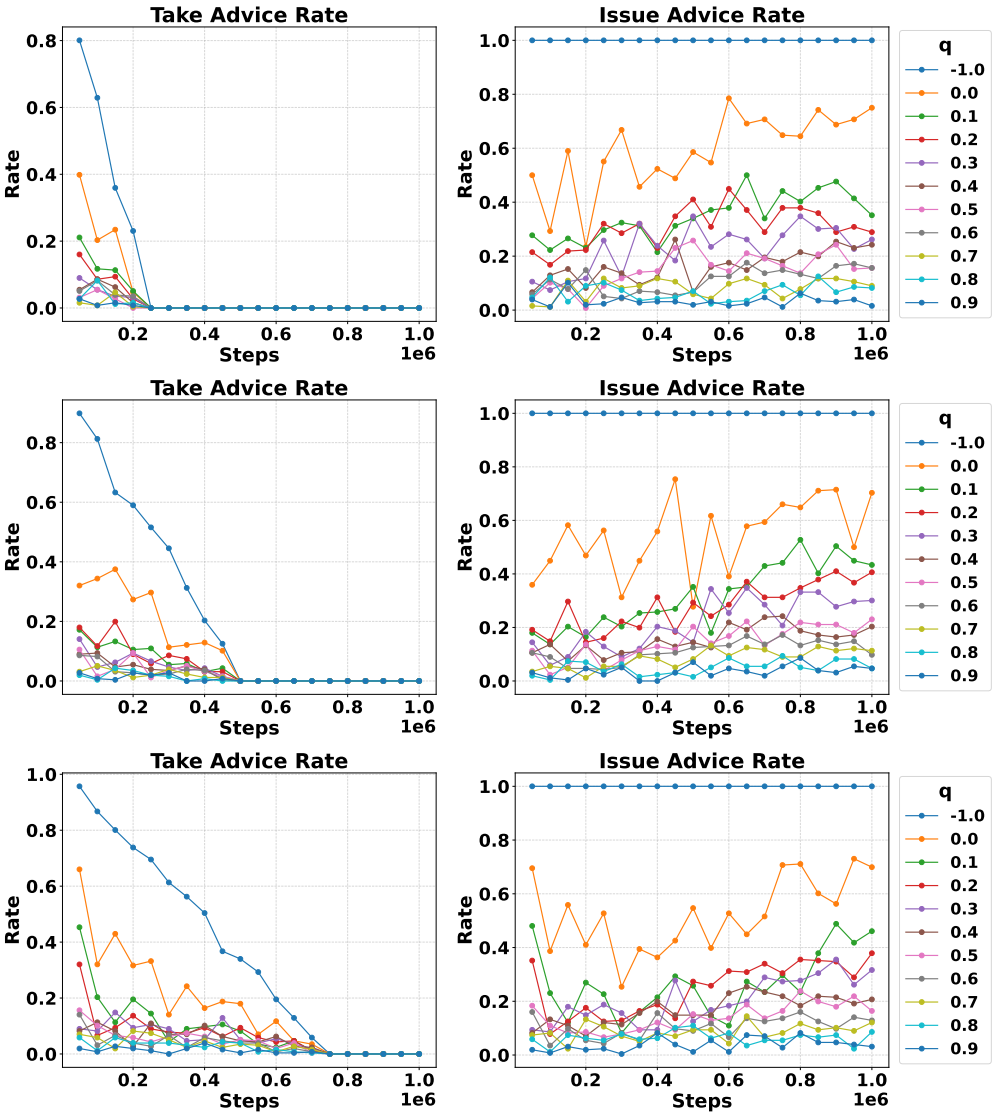

Figure 15: 10 seeds. Advice schedules where advice ends at (a) 25%, (b) 50%, and (c) 75% of training in the Unlocked-to-Locked environment. $q = -1$ advise in all states (AA Baseline).

L Additional Learning Curves Across q

Figure 3(c) reports relative transfer performance across varying q against the baselines, revealing a clear balance. When q is too small, the threshold is permissive and admits too much harmful advice in out-of-distribution states; when q

is too large, it is overly strict and the teacher provides too little guidance to be useful. The curve in panel (c) makes this visible: it climbs as q increases past the harmful-advice regime, flattens into a broad plateau across $q \in [0.2, 0.7]$, and declines again as guidance becomes too sparse. Beyond this plateau, relative transfer performance is limited. In the main paper, panel (d) plots learning curves only for the representative value $q = 0.5$, since showing all q values alongside the baselines would clutter the panel. To make explicit how the aggregate values in panel (c) map to the underlying training dynamics, Figure 16 provides the full set of curves for every $q \in \{0.2, 0.3, 0.4, 0.5, 0.6, 0.7\}$, all of which yield near-optimal sample efficiency and final return, remaining largely overlapping and well-separated from the baselines.

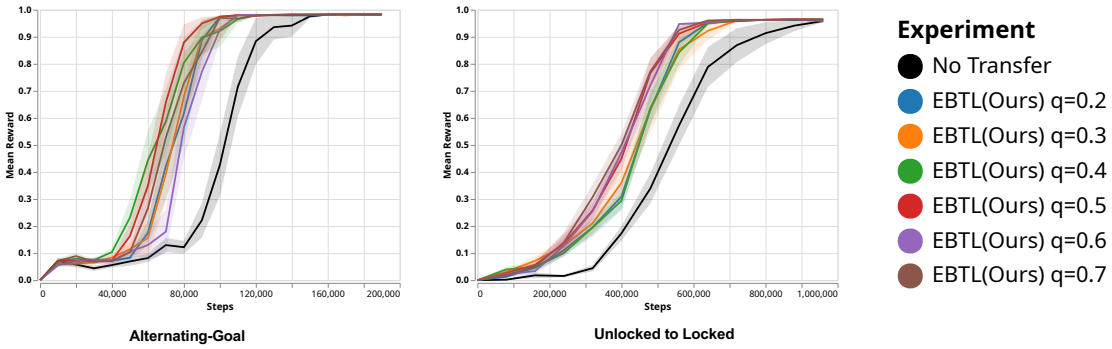

Figure 16: **Learning curves across** q (10 seeds). EBTL for $q \in \{0.2, \ldots, 0.7\}$ vs. baselines; near-identical across the range. Guidance ends at mid-training.

M Comparison to Critic-Based Introspection

Introspective Action Advising (IAA) (Campbell et al., 2023) is a closely related introspective action-advising method. Unlike EBTL, which derives its intervention signal from the teacher's *policy*, IAA operates on the *critic* side: it fine-tunes a copy of the teacher's value function on the student's data during transfer, and the quality of its advice depends heavily on this online estimation. Because this places IAA in a methodologically distinct family that hinges on a separately learned value estimate, we examine it here in the appendix rather than alongside the policy-side baselines in the main paper.

The mechanism proceeds in two phases. Let $V^{\text{Src}}_{\pi_T}$ denote the teacher's value function learned and frozen on the source task, and $V^{\text{new}}_{\pi_T}$ a copy of it that is fine-tuned during transfer. During an initial burn-in of δ steps, the teacher issues no advice; instead, $V^{\text{new}}_{\pi_T}$ is updated on the returns observed from the student's rollouts in the target task, adapting the source value estimate toward the target. After burn-in, for each state s visited by the student the teacher compares the two estimates and issues advice only when the value gap is below a threshold ϵ:

$$\left|V^{\text{new}}_{\pi_T}(s) - V^{\text{Src}}_{\pi_T}(s)\right| \leq \epsilon.$$

A small gap is taken to mean the teacher's behavior at s transfers to the target task, while a large gap signals task mismatch and advice is withheld.

Because the criterion compares a frozen source estimate against a critic that is still being fine-tuned on limited, non-stationary student data, the reliability of the resulting signal hinges on how the teacher's critic evolves over the course of the student's learning: early in transfer the value gap may reflect estimation error rather than genuine task mismatch. The evolution of this critic is therefore central to whether IAA can identify transferable states. To evaluate the baseline as favorably as possible, we examine the signal directly and measure how well the value-gap criterion separates transferable from non-transferable states throughout training, in both GridWorld settings (Alternating-Goal and Unlocked-to-Locked). For an equally fair comparison, when fine-tuning the teacher's critic we test two variants: one that freezes the convolutional backbone and adapts only the value head, and one that fine-tunes the full network. The results are shown in Figure 20 for Alternating-Goal and in Figure 19 for Unlocked-to-Locked.

As both figures show, the value-gap criterion only begins to separate transferable from non-transferable states once the student is more than halfway through training; before that point, the ID and Transferable-OOD curves are not reliably above the Non-transferable-OOD curve, so early advice is largely uninformative. This is a fundamental timing problem for transfer: as Figure 3(d) shows, EBTL has already converged by this stage, meaning IAA's signal

becomes trustworthy only after the window in which guidance is most valuable has passed. We further observe no clear difference between the full fine-tuning and frozen-backbone variants, indicating that the delay stems from the critic's adaptation to non-stationary student data rather than from the choice of which parameters to fine-tune.

To understand the original IAA configuration as thoroughly as possible, we sweep the decay schedule, testing variants that reach zero advising probability at 50% and at 75% of training. For an apples to apples comparison, we run each method under both the exponential decay used in the original IAA paper and the linear decay used by EBTL, and report the two schedules separately in Figure 17. We also include the no-transfer baseline and standard action advising (AA) without any filtering, so that the contribution of IAA's introspection signal can be isolated from the contribution of the schedule itself.

We reproduce IAA, but its benefit is largely confined to the exponential decay schedule. Under the original IAA configuration of exponential decay with $\epsilon = 0.9$, we recover the result reported by Campbell et al. (2023), with IAA improving substantially over the no-transfer baseline. However, under this schedule even standard action advising without any filtering also exceeds the baseline, suggesting that part of IAA's reported gain stems from the aggressive early decay rather than from its introspection signal. The linear decay panel tells a different story: among the advising methods, only EBTL remains clearly above the baseline. Across both schedules, EBTL is the best performing method. We attribute this robustness to the fact that EBTL filters transferable from non-transferable states at the level of the policy itself and so does not need an aggressive schedule to mask poor advice. The exponential schedule benefits the other methods because it annuls advice quickly during the early exploration phase, limiting the influence of harmful guidance that they cannot otherwise filter.

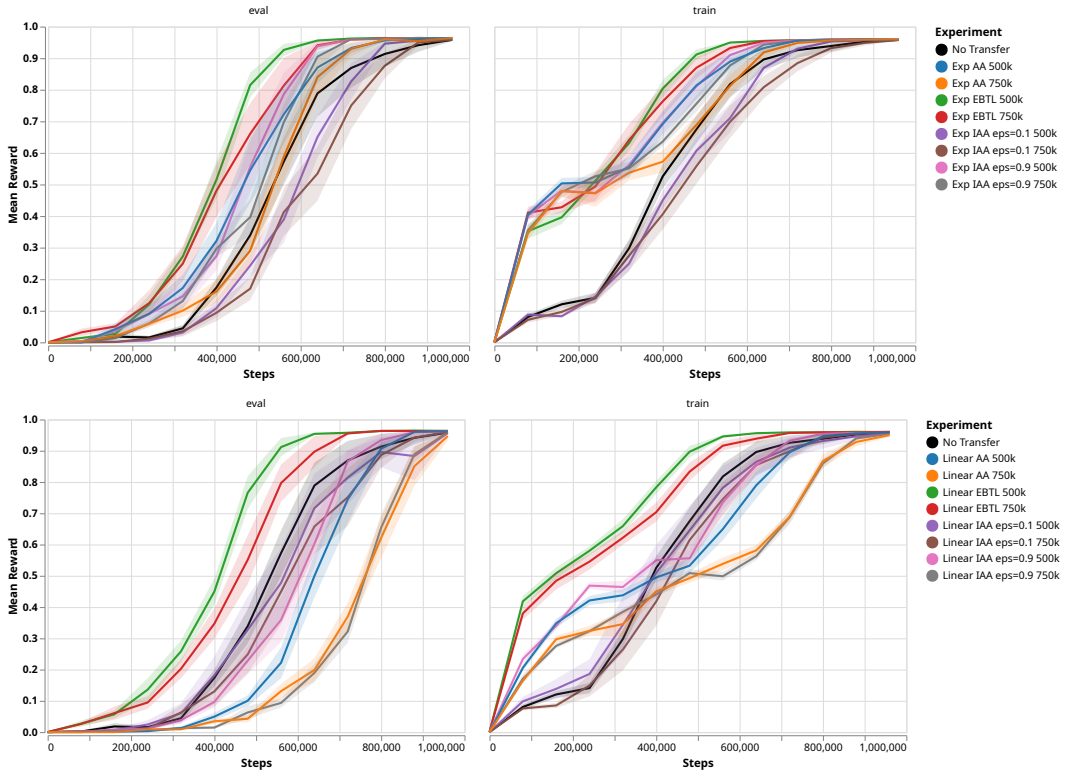

Figure 17: **IAA reproduction and decay-schedule comparison.** 10 seeds. Top: exponential decay (as in the original IAA paper). Bottom: linear decay (the EBTL default). Under exponential decay we reproduce the IAA result, but standard action advising without filtering also exceeds the baseline, suggesting the schedule itself contributes substantially to the gain. Under linear decay, only EBTL remains clearly above the baseline. EBTL is the best performing method under both schedules.

Figure 19 indicates that the teacher's critic is not informative at the start of student training and may require considerable adaptation before its signal becomes reliable. For a more complete evaluation, we therefore test IAA with substantially longer burn-in periods, so that the teacher's critic has more time to evolve on student data before any advice is issued. We sweep two burn-in windows, one beginning at 25% of training and one beginning at 50% of

training, paired with linear decay schedules that reach zero advising probability at the 50%, 75%, or 100% mark. We use $\epsilon = 0.1$ and $\epsilon = 0.15$, the two thresholds that Figure 19 shows yield the cleanest separation between transferable and non-transferable states. This sweep differs in intent from the previous one: rather than matching the configuration reported by Campbell et al. (2023), the goal is to wait until the critic has plausibly developed the ability to discriminate transferable from non-transferable advice before deploying it. Results are shown in Figure 18.

EBTL still outperforms IAA across the sweep, with roughly half of the IAA configurations clearing the no-transfer baseline. The best IAA configuration uses $\epsilon = 0.1$ with the burn-in ending at 250K steps and linear advice decay reaching zero at 750K. This is consistent with Figure 19: at $\epsilon = 0.1$ the teacher's critic only acquires meaningful separation between transferable and non-transferable states after roughly 40% of the horizon, so configurations that begin advising before then are largely working with an uninformative signal. The pattern reinforces our central claim: when the introspection signal genuinely separates transferable from non-transferable states, transfer improves. One challenge with IAA is that the timing of this separation depends on how the teacher's critic evolves under fine-tuning on student data, and that evolution is run-specific: it is not fully apparent until student training has progressed. Selecting an effective burn-in window in advance would in principle require knowing at what fraction of training the critic's value gap becomes informative, which is the trajectory captured in Figure 19, and that trajectory typically only emerges once student training is well underway. This makes it difficult to identify a strong IAA configuration ahead of time, and the configurations we report here should therefore be read as an upper bound on IAA's performance under favorable conditions rather than as a recommended deployment setting.

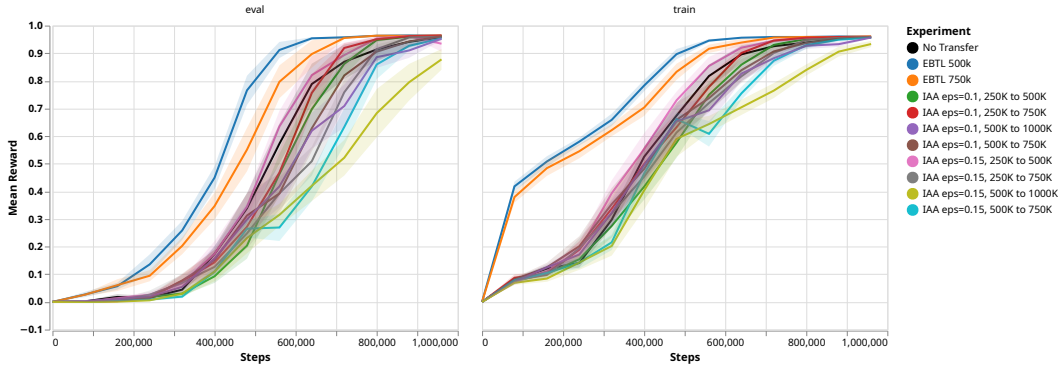

Figure 18: **IAA with extended burn-in periods.** 10 seeds. Linear Decay. Learning curves for IAA across burn-in start points (25% and 50% of training), advice decay endpoints (50%, 75%, and 100% of training), and thresholds ($\epsilon \in \{0.1, 0.15\}$).

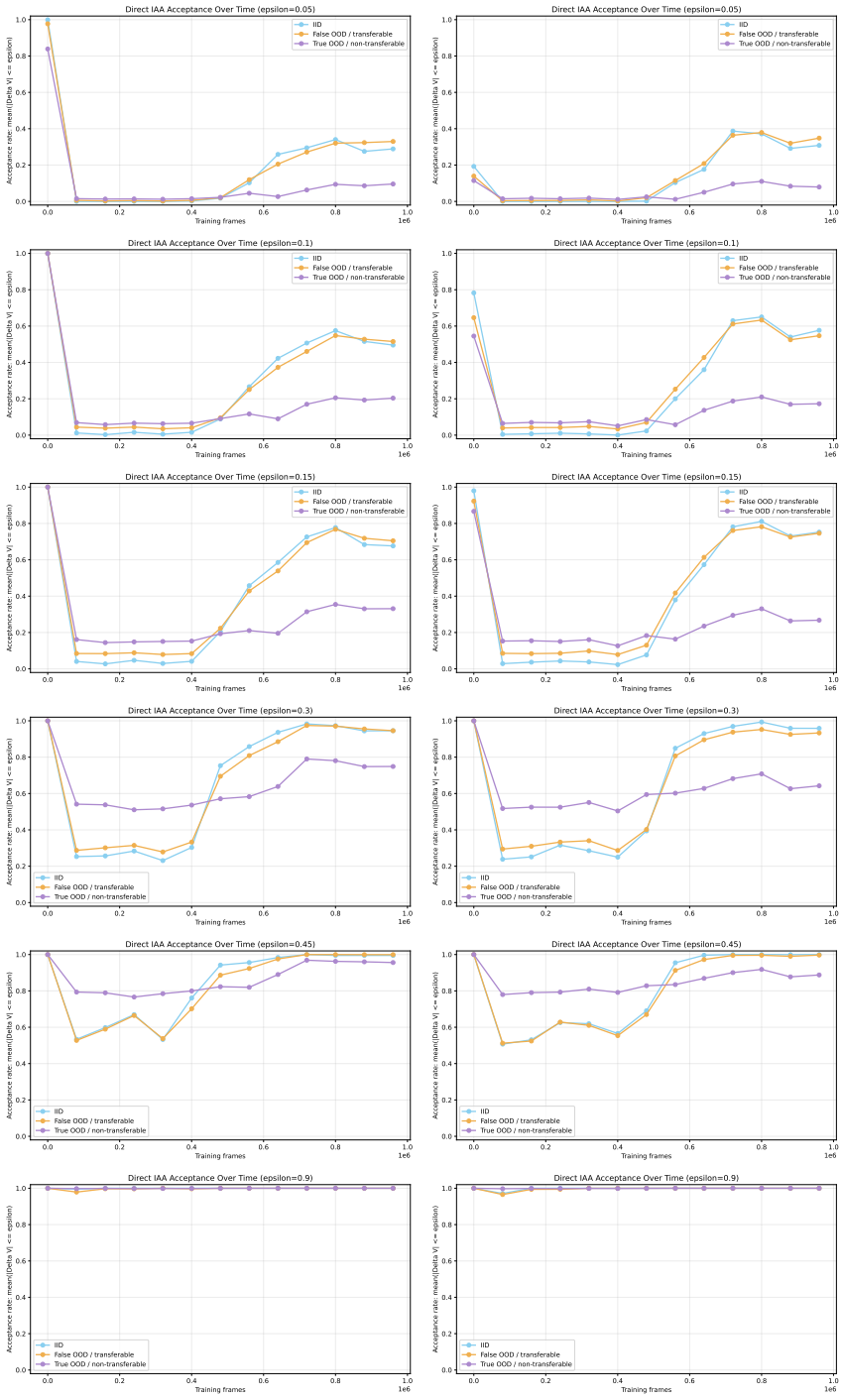

Figure 19: **IAA value-gap separation in the Unlocked-to-Locked setting**, shown across $\epsilon \in \{0.05, 0.1, 0.15, 0.3, 0.45, 0.9\}$ (one per row); left column freezes the convolutional backbone, right column fine-tunes the full network. Each panel tracks three groups of states over the course of the student's learning: *ID* (in-distribution, collected during teacher training), *Transferable OOD* (states encountered during the student's learning where the teacher *should* issue advice), and *Non-transferable OOD* (states where the teacher *should not*). A reliable signal should place the ID and Transferable-OOD curves above the Non-transferable-OOD curve, indicating that good advice is issued more often than bad advice.

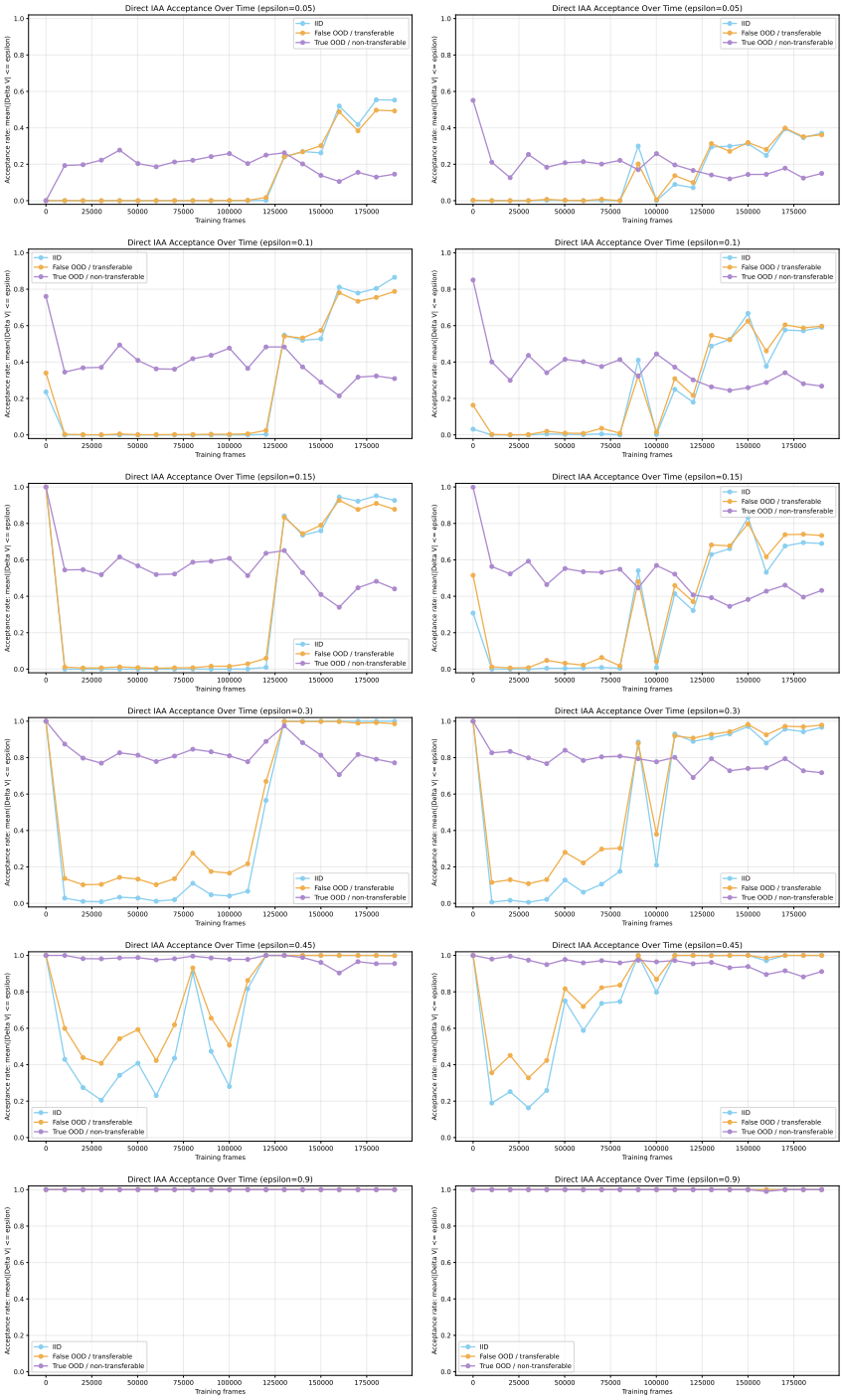

Figure 20: **IAA value-gap separation in the Alternating-Goal setting**, across $\epsilon \in \{0.05, 0.1, 0.15, 0.3, 0.45, 0.9\}$ (one per row); left column freezes the convolutional backbone, right column fine-tunes the full network. Curves and the expected ordering are as in Figure 19: a reliable signal keeps the ID and Transferable-OOD curves above the Non-transferable-OOD curve.