Energy-Based Transfer for Reinforcement Learning

Zeyun Deng[♠] Jasorsi Ghosh[♠] Fiona Xie[♠]
Yuzhe Lu[♢] Katia Sycara[♣] Joseph Campbell[♠]
[♠] Purdue University [♢] AWS AI [♣] Carnegie Mellon University

***Abstract*—Reinforcement learning algorithms often suffer from poor sample efficiency, making them challenging to apply in multi-task or continual learning settings. Efficiency can be improved by transferring knowledge from a previously trained 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 show that energy scores reflect the teacher's state-visitation density and empirically demonstrate improved sample efficiency and performance across both single-task and multi-task settings.**

I. Introduction

Reinforcement learning (RL) is widely used for sequential decision-making in robotics, enabling agents to acquire complex motor skills [9, 1, 7, 30]. However, many RL algorithms suffer from poor sample efficiency, due to challenges related to credit assignment, modeling errors, and sparse rewards. While sample inefficiency may be tolerable when learning a single task, it becomes increasingly problematic in multi-task [32] or continual learning [28] settings, where agents must repeatedly learn to solve multiple, often related tasks. A natural question arises: *can we transfer knowledge from previously solved tasks to accelerate learning in new ones*?

One common approach to transfer is to reuse a previously trained teacher policy to guide a student policy's exploration in a new task, either directly (by suggesting actions [25, 26, 15]) or indirectly (by shaping rewards [3, 10]). This form of transfer learning can be highly effective: early in learning, even partial guidance can steer the student toward high-reward behaviors and minimize the need for random exploration. However, when tasks are sufficiently different this approach can impair the student's ability to learn; the teacher may issue sub-optimal guidance that biases exploration towards low-reward regions of the state-action space [4, 24].

In this paper, we introduce an introspective transfer learning method that selectively guides exploration only when the teacher's knowledge is likely to be helpful. 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 [13, 8] and out-of-distribution detection [14, 29, 20], 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, leading to more efficient training – not by issuing *more* guidance but by issuing *correct* guidance. Our contributions are as follows:

- 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 provide theoretical justification for our approach, showing that the energy score is proportional to the state visitation density induced by the teacher policy.
- We empirically demonstrate that our method yields more sample efficient learning and higher returns than standard reinforcement learning and transfer learning baselines, across both single-task and multi-task settings.

II. Related Work

Reinforcement learning is a general framework for sequential decision-making, where an agent learns a policy to maximize long-term reward [23]. However, RL often suffers from poor sample efficiency, especially in sparse-reward or high-dimensional environments [2, 19].

To improve sample efficiency, transfer learning reuses knowledge from prior tasks to accelerate learning in new ones [27]. In RL, this often follows a teacher-student paradigm, where a teacher policy trained on a source task guides a student on a related target task [24, 33]. Guidance may take the form of action suggestions [25], reward shaping [16], or policy initialization [12]. Parameter-based methods like fine-tuning are simplest, initializing the student from the teacher and adapting via further training [31, 18]. However, this can overly bias the student toward the teacher's behavior, limiting exploration and harming performance when tasks differ. Alternatively, behavior-based methods transfer knowledge by encouraging the student to mimic the teacher during training. *Policy distillation* adds an auxiliary loss to minimize divergence between student and teacher policies [21, 22]. In *action advising*, the teacher suggests actions to guide exploration, but poorly timed advice can impede learning [25]. To address this, recent methods adopt dynamic advising: *JumpStart RL* limits guidance to early steps of each episode [26], while *introspective action advising* uses shifts in the teacher's expected reward to decide when to intervene [4].

However, prior methods rely on pre-defined heuristics, hyperparameters, or brittle fine-tuning strategies which limit their generalizability. Our method addresses this gap by applying theoretically-grounded out-of-distribution detection to reliably

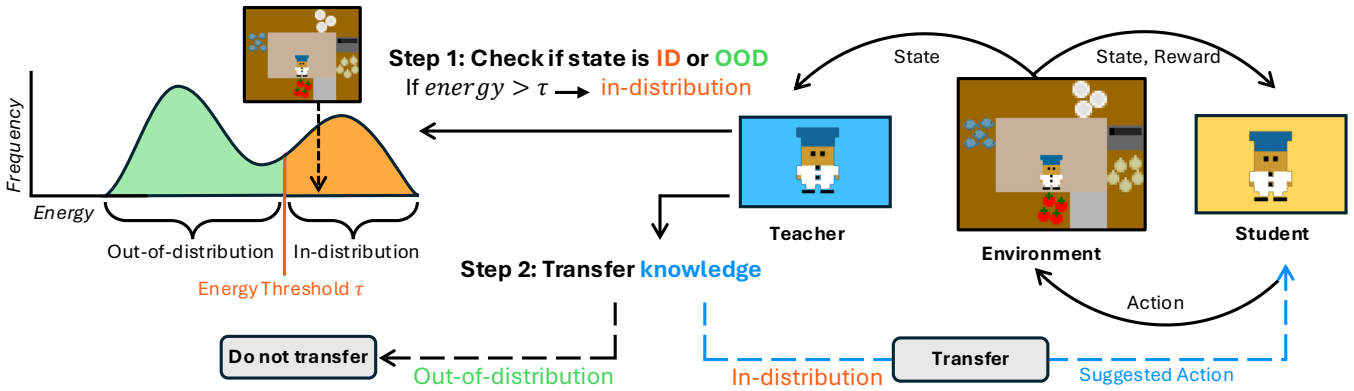


Fig. 1: Overview of **energy-based transfer learning**. During interaction, the teacher: 1) determines whether a state is in- or out-of-distribution based on a predefined energy threshold; 2) if the state's energy exceeds the threshold, it is treated as in-distribution and an expert action is suggested.

estimate teacher familiarity with a given state, enabling positive transfer performance even between tasks with high degrees of covariate shift.

III. Background

a) Reinforcement Learning.: We model our setting as a Markov Decision Process (MDP), defined by the tuple $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$, where $\mathcal{S}$ is the set of states, $\mathcal{A}$ is the set of actions, $P(s' \mid s, a)$ denotes the transition probability from state s to state s' given action a, $R(s, a)$ is the reward function, and $\gamma \in [0, 1)$ is the discount factor. At each timestep t, the agent observes a state $s_t \in \mathcal{S}$, selects an action $a_t \in \mathcal{A}$, transitions to a new state $s_{t+1} \sim P(\cdot \mid s_t, a_t)$, and receives a reward $r_t = R(s_t, a_t)$. We consider the infinite-horizon setting, where our objective is to learn a policy $\pi(a \mid s)$ that maximizes the expected discounted return: $\mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r_t\right]$.

b) Energy-Based Out-of-Distribution Detection.: In this paper, we are interested in determining whether a state lies within the training data support of a given policy. In supervised learning, this is broadly referred to as out-of-distribution (OOD) detection. A widely-used baseline for OOD detection uses the maximum softmax probability assigned to a predicted label [11]. However, softmax scores are not always reliable as neural networks can produce overconfident predictions for out-of-distribution inputs [17]. An alternative approach is to use the *energy score* of an input, which is computed from the raw logits of a network and has been shown to better separate in- and out-of-distribution examples [14].

Formally, given an input $\mathbf{x} \in \mathbb{R}^D$ and a neural network $f(\mathbf{x}) \in \mathbb{R}^K$ with logits $f_1(\mathbf{x}), \ldots, f_K(\mathbf{x})$, we define the *free energy* for $\mathbf{x}$ as follows, where $T > 0$ is a temperature parameter controlling the sharpness of the scaled logits:

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

An input is considered to be OOD if $E(\mathbf{x}; f) > \tau$ for an *energy threshold* τ and in-distribution (ID) otherwise. The energy threshold is pre-computed over a set of ID data.

IV. Energy-Based Transfer Learning

Our goal is to improve reinforcement learning sample efficiency, especially in multi-task settings. One way is to leverage a teacher policy trained on a related source task to guide the student in a new target task. However, naive guidance can hurt efficiency when the student visits states outside the teacher's experience, biasing exploration toward uninformative or low-reward regions. We address this by only allowing the teacher to suggest actions in states sufficiently close to its training distribution. We formalize the problem of *when* to issue guidance as *out-of-distribution detection for reinforcement learning*.

Problem Formulation. Let π_T and π_S 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. The action selection rule is then defined as:

$$a = \begin{cases} a_T \sim \pi_T(\cdot \mid s), & \text{if } \phi(s; \pi_T) \geq \tau, \\ a_S \sim \pi_S(\cdot \mid s), & \text{if } \phi(s; \pi_T) < \tau. \end{cases} \tag{2}$$

Equation 2 restricts teacher intervention to states where it has prior experience, deferring to the student in all other cases.

A. Energy Scores and State Visitation

We draw inspiration from recent work on energy-based out-of-distribution detection [14] and define our score function as the negative free energy of a state s under the teacher policy:

$$\phi(s; \pi_T) = -E(s; \pi_T),$$

where $E(s; \pi_T)$ is the free energy computed from the teacher's network. We refer to $\phi(s; \pi_T)$ as the energy score, which serves as a proxy for how likely the state is to belong to the teacher's training distribution $p(x)$. In on-policy reinforcement learning, training data is generated by rolling out the teacher policy π_T to collect experience. As a result, $p(x)$ is implicitly

Algorithm 1 Energy-Based Transfer for Reinforcement Learning

Input: Teacher policy π_T, student policy π_S, energy threshold τ, decay function δ
while not done **do**
  Initialize empty batch $B \leftarrow \emptyset$
  for $t = 1 \rightarrow H$ **do**
    Sample $p \sim \mathcal{U}(0, 1)$
    $a_t \leftarrow \begin{cases} \pi_T(a \mid s_t) & \text{if } -E(s_t; \pi_T) \geq \tau \text{ and } p < \delta(t) \\ \pi_S(a \mid s_t) & \text{otherwise} \end{cases}$
    Take action a_t, observe r_t, s_{t+1}
    $B \leftarrow B \cup (s_t, a_t, s_{t+1}, r_t)$
  Update π_S with batch B

defined by the state-visitation distribution $d_\pi(s)$ of the teacher. Consequently, the free energy $E(s; \pi_T)$ is negatively related with the teacher's familiarity with a state – assigning lower values to frequently visited states and higher values to unfamiliar ones. Following convention [14], we set, the energy score ϕ to the *negative* free energy so that in-distribution states yield higher scores than out-of-distribution states.

Proposition 1. *Under on-policy training, let $d_\pi(s)$ denote the state-visitation distribution induced by policy π. Then the log of the visitation density is proportional to the score function $\phi(s) = -E(s)$:*

$$\log d_\pi(s) \propto \phi(s).$$

Proof: Given an energy-based model f, the density $p(s)$ is defined in terms of its energy $E(s)$ [13] as $p(s; f) = \frac{e^{-E(s;f)/T}}{Z}$, where $Z = \int_s e^{-E(s;f)/T}$ is the partition function and T is the temperature. Ignoring the normalizing constant Z and taking the logarithm of both sides, we obtain $\log p(s) \propto -E(s)$. In on-policy RL, training data is collected by sampling trajectories under the current policy π. Thus, the empirical distribution $p(s)$ corresponds to the marginal distribution over states visited by π, i.e., the state-visitation distribution $d_\pi(s)$. Substituting this into the previous expression, we obtain $\log d_\pi(s) \propto -E(s) = \phi(s)$. ∎

B. Algorithm

We summarize our approach in Algorithm 1. At a high-level, the student policy interacts with the environment to collect trajectories, while selectively receiving guidance from a teacher policy. At each timestep, EBTL evaluates whether the current state is familiar to the teacher using an energy-based OOD score. If the state is deemed in-distribution and a decaying probability schedule permits guidance, the action is sampled from the teacher policy; otherwise, the student policy acts. The resulting trajectories are stored in a batch and used to update the student policy.

To decide when to issue guidance, we compute a threshold $\tau \in \mathbb{R}$ as the empirical q-quantile of energy scores over teacher training states $\mathcal{S}_T$, i.e., $\tau = \text{Quantile}_q(\{\phi(s) \mid s \in \mathcal{S}_T\})$. Following prior work [22, 26, 4], we apply a linear decay schedule $\delta(t) = \max(0, \delta_0 - \kappa t)$ to control the probability of guidance. This enables early reliance on the teacher while gradually promoting student autonomy.

a) Energy Regularization.: As discussed in Section IV-A, the score function $\phi(s) = -E(s)$ correlates with the teacher's state-visitation frequency: frequently visited states tend to receive higher scores. However, this implicit signal may be insufficient to reliably distinguish in-distribution (ID) from out-of-distribution (OOD) states, as the teacher is trained solely on trajectories from its own environment and lacks exposure to OOD regions.

To improve separability, we adopt the energy-based loss from Liu et al. [14], augmenting the teacher's training with a fixed set of representative OOD states. Let $\mathcal{D}_{\text{in}}^{\text{train}}$ denote the set of ID states collected during teacher training and $\mathcal{D}_{\text{out}}^{\text{train}}$ a curated set of OOD states. Let $\mathbf{s}_{\text{in}} \sim \mathcal{D}_{\text{in}}^{\text{train}}$ and $\mathbf{s}_{\text{out}} \sim \mathcal{D}_{\text{out}}^{\text{train}}$ denote samples from each. Using the energy score $\phi(s) = -E(s)$, the loss is defined as:

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

where $m_{\text{in}} \in \mathbb{R}$ and $m_{\text{out}} \in \mathbb{R}$ are margin thresholds for ID and OOD energy scores, respectively. The first term penalizes ID states with energy scores below m_{in}; the second penalizes OOD states with energy scores above m_{out}. The overall teacher loss is $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{RL}} + \lambda \cdot \mathcal{L}_{\text{energy}}$ where $\lambda \in \mathbb{R}^+$ controls the weight of the energy regularization. In EBTL, OOD samples are drawn from random rollouts in the target environment. ID samples are drawn from the teacher's own training trajectories via random subsampling.

b) Off-Policy Correction: Actions originate from either the student policy (pure student sampling) or the teacher policy (teacher guidance), making trajectories off-policy relative to the student. We correct for this via importance sampling in both actor and critic updates. The importance ratio compares the current student policy to the behavior policy—either the previous student policy or the fixed teacher policy—ensuring stable training under mixed-policy rollouts.

V. Experiments

We evaluate our method in two settings: **single-task transfer** and **multi-task transfer**. In the single-task setting, we use GridWorld [6], a navigation environment where the agent's objective is simply to reach a goal location. In the multi-task setting, we use Overcooked [5], where the agent must learn to solve multiple task variants, such as how to cook different recipes. For each environment, we construct multiple experimental settings that introduce increasing covariate shift between the teacher's training distribution and the student's target distribution. This allows us to evaluate the robustness of our method under progressively harder transfer scenarios.

In each domain, we examine learning performance with the goal of understanding: (1) whether our method leads to improved sample efficiency, and (2) when the teacher chooses

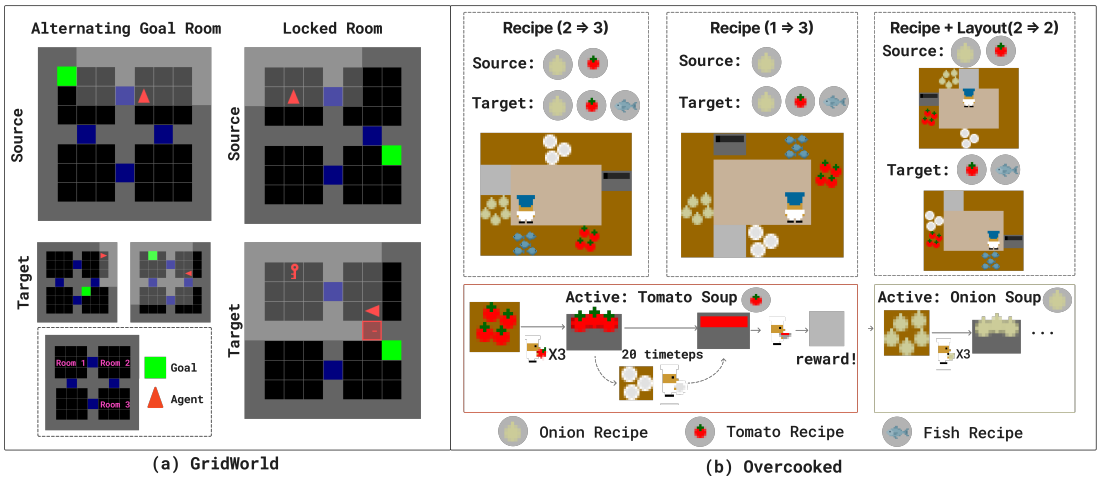


Fig. 2: Environments used in our empirical experiments. See Section V for details.

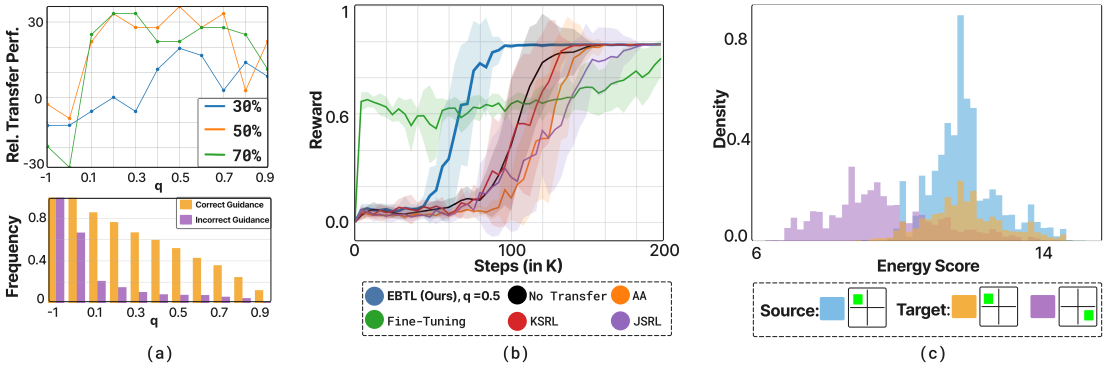


Fig. 3: **Alternating-Goal** results (10 seeds). **(a-Top)** Transfer performance across energy thresholds and decay schedules (e.g., 50% means guidance probability decays to 0 when training is 50% done). A threshold of -1 indicates that *all* guidance is given. **(a-Bottom)** Correct vs. incorrect guidance rates; guidance is considered correct if it is issued for in-distribution states (see Section V). **(b)** Evaluation returns for EBTL and baselines. **(c)** Energy score distributions: source (blue) vs. target (orange + purple), showing a bimodal split between ID (orange) and OOD (purple) states.

to provide guidance during the student's learning process. We compare our approach, energy-based transfer learning, against the following baselines:

- **No Transfer:** An agent trained from scratch with standard RL.
- **Action Advising (AA):** A teacher provides advice at every timestep. Advice issue rate decays over time using a predefined schedule.
- **Fine-Tuning:** The student is initialized from a pretrained teacher policy. Convolutional layers are frozen, and only the remaining parameters are updated during training.
- **Kickstarting RL (KSRL)** [22]: A policy distillation method that adds a cross-entropy loss between the student and teacher policies to encourage imitation.
- **JumpStart RL (JSRL)** [26]: A time-based advising method where the teacher provides guidance during the early part of each episode, with a decaying timestep threshold.

All experiments use teacher and student policies trained with the TorchRL implementation of proximal policy optimization (PPO) [22]. Full hyperparameter details are provided in the Appendix.

A. *Single-Task Setting: GridWorld*

GridWorld consists of four interconnected rooms and serves as a controlled single-task setting. We design two transfer setups, As illustrated in Figure 2a:

(1) Alternating Goal Room. The source task always places the goal 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.

(2) Locked Room. The source task allows free movement between rooms, while the target task introduces a locked door between the upper and lower areas. To reach the goal, the agent must first retrieve a key – randomly placed in the upper rooms – and unlock the door. Since the teacher was not trained to find or use a key, it should only provide guidance after the key has been picked up, when the remaining navigation matches

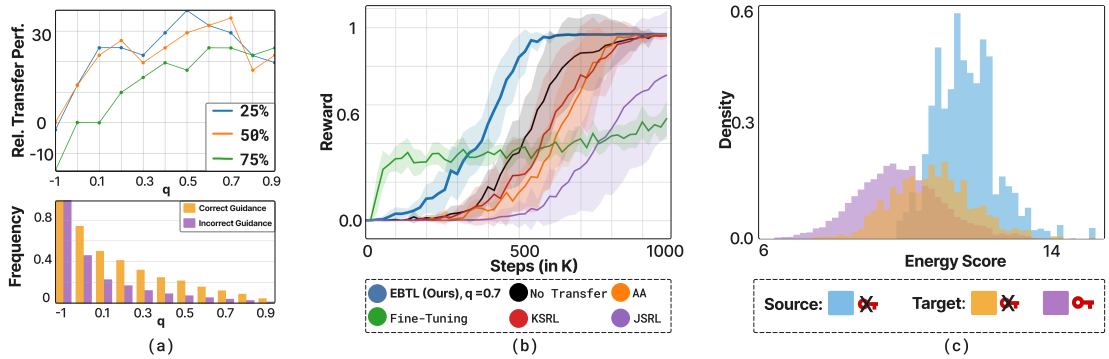
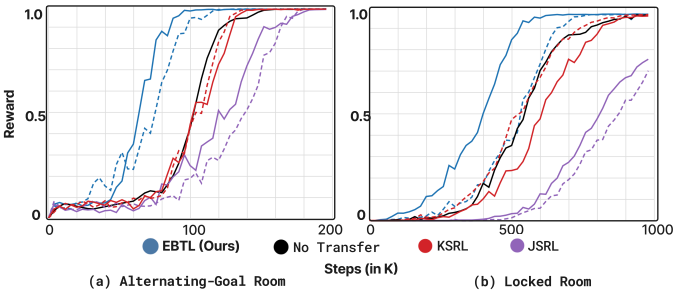
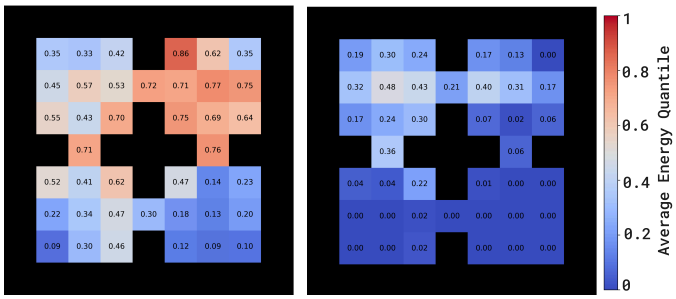


Fig. 4: **Locked Room** results (10 seeds). See Figure 3 for caption details.



(a) Transfer performance with (solid) vs. without (dashed) energy regularization.



(b) **Alternating-Goal**. Heatmaps showing the average energy quantile of each state under the teacher policy. Left: source task where the goal is always in Room 1 (upper-left). Right: target task states collected only when the goal is placed in Room 3 (lower-right). Higher quantiles indicate greater teacher familiarity.

Fig. 5: Comparison of energy-based transfer learning performance and corresponding state-wise energy visualization.

its prior experience.

The results for the Alternating Goal Room and Locked Room setups are illustrated in Figure 3 and Figure 4, respectively. We make the following observations.

EBTL consistently outperforms all baselines. In both transfer setups, EBTL achieves the highest sample-efficiency of all baselines. For the Alternating Goal Room and the Locked Room, when the energy threshold $q > 0.1$, EBTL rarely issues guidance in unfamiliar states, leading to significant improvements in transfer performance. As shown in Figure 5b, the teacher assigns higher energy scores to states encountered during training – when the goal is in Room 1 (upper-left) – compared to unseen states with the goal in Room 3 (lower-right). This illustrates the benefit of filtering guidance based on the teacher's familiarity with the state.

Higher covariate shift makes OOD detection more challenging. In the Alternating Goal Room, the teacher clearly distinguishes when to give guidance, as shown by the well-separated energy distributions in Figure 3c. In contrast, the Locked Room introduces novel elements (e.g., door and key), causing greater covariate shift and reducing separation, as seen in Figure 4c. Still, the teacher assigns lower energy scores to pre-key states, showing it can differentiate familiar from unfamiliar regions.

There exists an optimal energy threshold q that balances filtering harmful and helpful guidance. The performance curves exhibit a mountain-shaped trend: increasing q initially boosts transfer performance by suppressing harmful advice in unfamiliar states. However, when q becomes too large, the teacher begins to withhold guidance even in familiar situations, limiting its usefulness. This trade-off is evident in both Figure 3a and Figure 4a, where performance declines once $q > 0.7$ due to overly conservative advising.

Energy regularization significantly improves EBTL but has little effect on other methods. As shown in Figure 5a, incorporating energy loss enables EBTL to converge faster, especially in the more challenging Locked Room environment where covariate shift is greater. In contrast, other baselines show no noticeable difference in performance regardless of whether the teacher was trained with or without energy regularization – their convergence times remain similar. Notably, even without energy loss, EBTL still matches or exceeds all baselines, highlighting the robustness of our approach.

B. Multi-Task Setting: Overcooked

We develop a single-agent variant of Overcooked [5] to evaluate multi-task learning. At each timestep, one recipe (onion, tomato, or fish soup) is active. The agent must place three matching ingredients into a pot; while the soup cooks (20 steps), it can retrieve a dish. Once cooked, the soup must be delivered to receive a reward. After each delivery, a new recipe is sampled uniformly, regardless of correctness. Rewards are sparse and given only for correct deliveries, with auxiliary shaping to accelerate training. Figure 2b illustrates the setup.

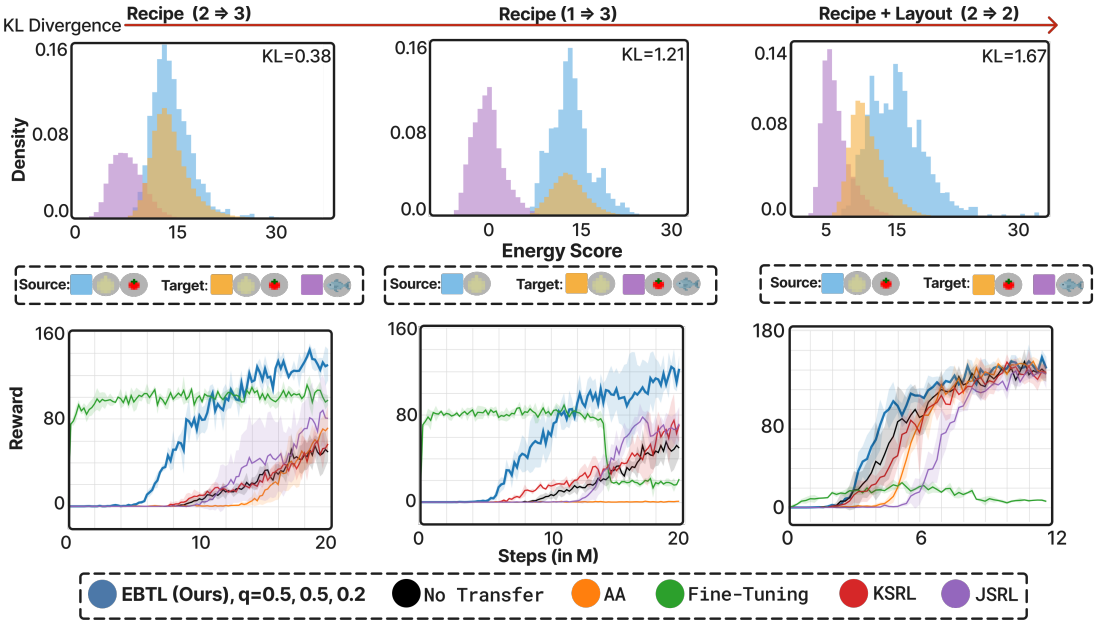


Fig. 6: **Overcooked** results (3 seeds). **(Top)** Energy score distributions under the teacher policy. The source task (blue) shows the training distribution. The target task (orange + purple) is bimodal, reflecting shared (orange, ID) and non-shared (purple, OOD) sub-tasks. **(Bottom)** Evaluation returns for EBTL and baselines. Quantiles: $q = 0.5$ (Recipe), $q = 0.2$ (Recipe + Layout).

We construct three transfer tasks with increasing distribution shift between teacher and student environments:

1) **Recipe Shift (2 ⇒ 3):** Both the source and target environments include all three ingredients: onions, tomatoes, and fish. The source task requires onion and tomato soup, while the target task requires onion, tomato, and fish soup resulting in recipe shift.
2) **Recipe Shift (1 ⇒ 3):** Both environments again have all three ingredients. This time, the source task requires only onion soup while the target task requires onion, tomato, and fish soup, introducing a higher degree of recipe shift.
3) **Recipe + Layout Shift (2 ⇒ 2):** The source environment includes only onions and tomatoes and requires onion and tomato soup, while the target environment includes only tomatoes and fish and requires recipe and layout. This results in both recipe and layout shift.

The results are shown in Figure 6. The relative difficulty of each transfer scenario is reflected by the increasing KL divergence between the energy score distributions of the teacher's and student's states.

EBTL outperforms all baselines across all difficulty levels. In all three transfer setups, EBTL consistently achieves higher policy returns and sample-efficiency than baseline methods. As the divergence between teacher and student environments increases, transfer becomes more challenging. Notably, the Recipe (2 ⇒ 3) scenario shares the same layout as Recipe (1 ⇒ 3), but converges more quickly. This improvement stems from the teacher's ability to issue more useful guidance, i.e. when the student's current recipe is tomato soup – a task the teacher has seen during training.

Shared layouts simplify OOD detection. In scenarios where the source and target tasks share spatial layouts, i.e. Recipe (2 ⇒ 3) and Recipe (1 ⇒ 3), the covariate shift is due entirely to the recipe encoding in the observation. This results in a clearly bimodal energy distribution in the target task – one mode for ID states and another for OOD – simplifying the OOD detection problem (refer to the top row of Figure 6.

Layout shift makes OOD detection more challenging. In the Recipe + Layout (2 ⇒ 2) setting, the source and target tasks use different layouts, introducing a stronger covariate shift. This results in a systematic decrease in ID energy scores which blurs the ID/OOD boundary, as even states associated with familiar recipes appear slightly OOD due to the layout shift. Nevertheless, EBTL continues to avoid issuing guidance in states associated with unfamiliar recipes, resulting in positive transfer performance. In contrast, all baseline methods yield negative transfer and degrade learning performance compared to standard RL.

VI. Conclusion

We introduced energy-based transfer learning (EBTL), a method for improving sample efficiency in reinforcement learning through selective teacher guidance. EBTL uses energy scores as a proxy for familiarity, issuing advice only in states likely within the teacher's training distribution. Experiments across single-task and multi-task settings show that EBTL consistently outperforms baselines, especially under covariate shift. While effective, EBTL requires setting an energy threshold and is best suited for covariate rather than label shift. These limitations suggest future work on adaptive thresholding and broader transfer settings.

References

[1] Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, et al. Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019.

[2] 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.

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

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

[5] 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.

[6] 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.

[7] David B D'Ambrosio, Saminda Wishwajith Abeyruwan, Laura Graesser, Atil Iscen, Heni Ben Amor, Alex Bewley, Barney Reed, Krista Reymann, Leila Takayama, Yuval Tassa, et al. Achieving human level competitive robot table tennis. In *7th Robot Learning Workshop: Towards Robots with Human-Level Abilities*, 2024.

[8] Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In *International conference on machine learning*, pages 1352–1361. PMLR, 2017.

[9] Tuomas Haarnoja, Ben Moran, Guy Lever, Sandy H Huang, Dhruva Tirumala, Jan Humplik, Markus Wulfmeier, Saran Tunyasuvunakool, Noah Y Siegel, Roland Hafner, et al. Learning agile soccer skills for a bipedal robot with deep reinforcement learning. *Science Robotics*, 9(89):eadi8022, 2024.

[10] Anna Harutyunyan, Sam Devlin, Peter Vrancx, and Ann Nowé. Expressing arbitrary reward functions as potential-based advice. In *Proceedings of the AAAI conference on artificial intelligence*, volume 29, 2015.

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

[12] Hao Ju, Rongshun Juan, Randy Gomez, Keisuke Nakamura, and Guangliang Li. Transferring policy of deep reinforcement learning from simulation to reality for robotics. *Nature Machine Intelligence*, 4(12):1077–1087, 2022.

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

[14] 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.

[15] Richard Maclin and Jude W Shavlik. *Incorporating advice into agents that learn from reinforcements*. University of Wisconsin-Madison. Computer Sciences Department, 1994.

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

[17] 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*, pages 427–436, 2015.

[18] Celal Öztürk, Murat Taşyürek, and Mehmet Uğur Türkdamar. Transfer learning and fine-tuned transfer learning methods' effectiveness analyse in the cnn-based deep learning models. *Concurrency and computation: practice and experience*, 35(4):e7542, 2023.

[19] Rémy Portelas, Cédric Colas, Lilian Weng, Katja Hofmann, and Pierre-Yves Oudeyer. Automatic curriculum learning for deep rl: A short survey. *arXiv preprint arXiv:2003.04664*, 2020.

[20] Jie Ren, Peter J Liu, Emily Fertig, Jasper Snoek, Ryan Poplin, Mark Depristo, Joshua Dillon, and Balaji Lakshminarayanan. Likelihood ratios for out-of-distribution detection. *Advances in neural information processing systems*, 32, 2019.

[21] 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.

[22] 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.

[23] Richard S Sutton, Andrew G Barto, et al. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

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

[25] 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, pages 1053–1060, 2013.

[26] 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*, pages 34556–34583. PMLR, 2023.

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

[28] Maciej Wołczyk, Michał Zając, Razvan Pascanu, Łukasz Kuciński, and Piotr Miłoś. Continual world: A robotic benchmark for continual reinforcement learning. *Advances in Neural Information Processing Systems*, 34: 28496–28510, 2021.

[29] Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. Generalized out-of-distribution detection: A survey. *International Journal of Computer Vision*, 132(12):5635–5662, 2024.

[30] Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Mastering visual continuous control: Improved data-augmented reinforcement learning. *arXiv preprint arXiv:2107.09645*, 2021.

[31] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *Advances in neural information processing systems*, 27, 2014.

[32] 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*, pages 1094–1100. PMLR, 2020.

[33] Zhuangdi Zhu, Kaixiang Lin, Anil K Jain, and Jiayu Zhou. Transfer learning in deep reinforcement learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):13344–13362, 2023.

APPENDIX

A. Training Details

1) GridWorld:

a) 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.

b) 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 D0b). 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.

c) Student Training: For each target task, we first train a student policy from scratch using standard PPO without any transfer to establish baseline performance. In the *unlocked-to-locked* environment, the total training horizon for transfer experiments is set to 1 million steps, while in the *alternating-goal room* environment, it is set to 200,000 steps. 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.

2) Overcooked-AI:

a) 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.

b) 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.

Subset (2→3)	Subset (1→3)	Inter. (2→2)
19M	9M	12M

TABLE I: Teacher checkpoints (in env steps)

c) 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 3 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:

Subset (2 → 3)	**Subset (1 → 3)**	**Intersection (2 → 2)**
20M	20M	12M

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

B. Hyperparameters

1) GridWorld: All experimental setups in GridWorld are trained using a fixed set of PPO hyperparameters, summarized in III. These settings remain consistent across all teacher and student training runs within the domain.

2) Overcooked-AI: All Overcooked experiments use a shared set of core PPO hyperparameters, listed in IV. 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 V. We use the following notation: O = Onion, T = Tomato, F = Fish, OT = Onion + Tomato, TF = Tomato + Fish, OTF = Onion + Tomato + Fish.

C. Model Architecture

All MiniGrid experiments share the same model architecture shown in Fig. 7a. Similarly, all Overcooked experiments use the architecture in Fig. 7b.

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

TABLE III: Hyperparameters used for all GridWorld 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 IV: Shared PPO hyperparameters across all Overcooked experiments.

Config	LR	Horizon
Subset (O)	0.001	8M
Subset (OT)	0.001	15M
Subset (OTF)	0.001	25M
Inter. (OT)	0.001	10M
Inter. (TF)	0.001	10M

TABLE V: Setup-specific learning rates and reward shaping horizons.

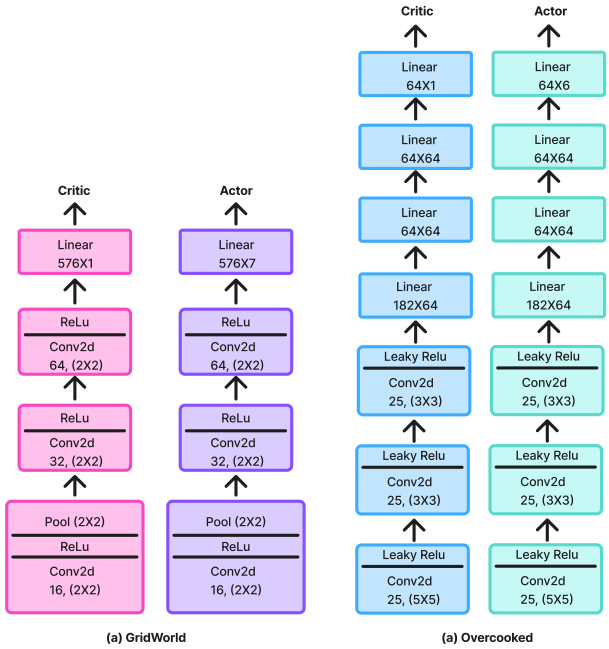


Fig. 7: Actor-Critic architectures used in our experiments. (a) MiniGrid. (b) Overcooked.

D. Sensitivity of Energy-Based 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:

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

a) 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 3 random seeds using a shared PPO setup and evaluated at the 800,000-step checkpoint.

b) 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 $(m_{\text{in}}, m_{\text{out}})$.

c) Results: As shown in Table VI, 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.

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