TGRL: An Algorithm for Teacher Guided Reinforcement Learning

Idan Shenfeld [1] **Zhang-Wei Hong** [1] **Aviv Tamar** [2] **Pulkit Agrawal** [1]

Abstract

Learning from rewards (i.e., reinforcement learning or RL) and learning to imitate a teacher (i.e., teacher-student learning) are two established approaches for solving sequential decision-making problems. To combine the benefits of these different forms of learning, it is common to train a policy to maximize a combination of reinforcement and teacher-student learning objectives. However, without a principled method to balance these objectives, prior work used heuristics and problem-specific hyperparameter searches to balance the two objectives. We present a *principled* approach, along with an approximate implementation for *dynamically* and *automatically* balancing when to follow the teacher and when to use rewards. The main idea is to adjust the importance of teacher supervision by comparing the agent's performance to the counterfactual scenario of the agent learning without teacher supervision and only from rewards. If using teacher supervision improves performance, the importance of teacher supervision is increased and otherwise it is decreased. Our method, *Teacher Guided Reinforcement Learning* (TGRL), outperforms strong baselines across diverse domains without hyper-parameter tuning.

1. Introduction

In Reinforcement Learning (RL), an agent learns decision-making strategies by executing actions, receiving feedback in the form of rewards, and optimizing its behavior to maximize cumulative rewards. Such learning by trial-and-error can be challenging, particularly when rewards are sparse, or under partial observability (Madani et al., 1999; Papadimitriou and Tsitsiklis, 1987). A more data-efficient learning method is to directly supervise the agent with correct actions obtained by querying a *teacher*, as exemplified by the imitation learning algorithm called DAgger (Ross et al., 2011). Learning to mimic a teacher is significantly more data-efficient than reinforcement learning because it avoids the need to explore the consequences of different actions.

However, learning from a teacher can be problematic when the teacher is sub-optimal or when it's impossible to perfectly mimic the teacher. In the first problematic case of a sub-optimal teacher, because the agent attempts to mimic the teacher's actions perfectly, its performance is inherently limited by the teacher's performance. Developing methods for training agents that surpass their sub-optimal teachers is an active research area (Agarwal et al., 2022a; Kurenkov et al., 2019; Rajeswaran et al., 2017). The second problem occurs when the agent is unable to mimic the teacher. It can happen in the common scenario when the teacher chooses actions based on *privileged information* unavailable to the agent. For example, the teacher may have access to additional sensors when training in simulation (Lee et al., 2020; Chen et al., 2021; Margolis et al., 2021), external knowledge bases (Zhang et al., 2020), or accurate state estimates during training (Levine et al., 2015).

In some scenarios, the agent can make up for the information gap with respect to the teacher by accumulating information from a history of observations (Kumor et al., 2021; Swamy et al., 2022). However, in the most general scenario, just using the history is insufficient, and the agent must take information-gathering actions (i.e. explore) to acquire the information being used by the teacher before it can mimic it. However, since the teacher never performs information-gathering actions, the agent cannot learn such actions by mimicking the teacher. As an example, consider the "Tiger Door" environment illustrated in Figure 1 (Littman et al., 1995; Warrington et al., 2021). The agent is placed in a maze with a goal cell (green), a trap cell (blue), and a button (pink). Reaching the goal and trap cells provide positive and negative rewards, respectively. The location of the goal and trap cells randomly switch locations every episode. The teacher is aware of the location of all cells, whereas the agent (or the student) cannot observe the goal/trap cell locations. Instead, the student can go to the pink button, an action that reveals the goal location. In this setup, the goal-aware teacher takes action to directly reach the goal. However, the student must deviate from the teacher's actions to reach the pink button – a behavior that cannot be learned by imitation.

[1]Improbable AI Lab, Massachusetts Institute of Technology, Cambridge, USA [2]Technion - Israel Institute of Technology, Haifa, Israel. Correspondence to: Idan Shenfled <idanshen@mit.edu>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

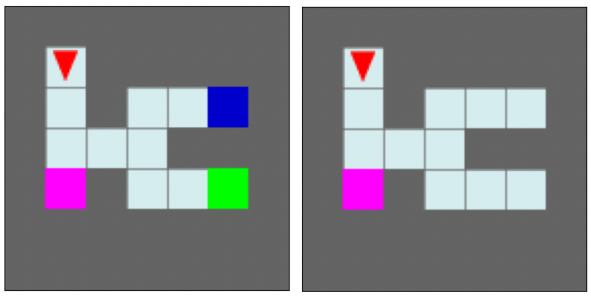

Figure 1: The Tiger Door environment. On the left is the teacher's observation, where the goal cell (in green) and the trap cell (in blue) are perceptible. On the right is the student's observation, where these cells are not visible, but there is a pink button; touching which reveals the other cells.

Consider the general scenario where the agent's optimality is measured by the rewards it accumulates. Both when the teacher is sub-optimal and when it cannot be mimicked, trying to imitate the teacher will result in sub-optimal policies. In these scenarios, a student with access to a reward function can benefit by jointly learning from both the reward and the teacher's supervision. Learning from rewards provides an incentive for the agent to deviate from a sub-optimal teacher to outperform it or carry out information gathering when learning from a privileged teacher. Thus, by combining both forms of learning, the agent can leverage the teacher's expertise to learn quickly but also try different actions to check if a better policy can be found. The balance between when to follow the teacher and use rewards is delicate and can substantially affect the performance (i.e., total accumulated rewards) of the learned policy. In the absence of a principled method to balance the two objectives, prior work resorted to task-specific hyperparameter tuning (Weihs et al., 2021; Nguyen et al., 2022; Agarwal et al., 2022b).

In this work, we present a principled solution to automatically balance learning from rewards and a teacher. Our main insight is that supervision from the teacher should only be used when it improves performance compared to learning solely from reward. To realize this, in addition to training the *main* policy that learns from both rewards and the teacher, we also train a second *auxiliary* policy that learns the task by only optimizing rewards using reinforcement learning. At every training step, our algorithm compares the two policies. If the *main* policy performs better, it indicates that utilizing the teacher is beneficial and the importance of learning from the teacher is increased. However, if the auxiliary policy performs better, the importance of the teacher's supervision in the main policy's objective is decreased. We call this algorithm for automatically adjusting the balance of imitation and RL objectives as *Teacher Guided Reinforcement Learning (TGRL)*.

We empirically evaluate TGRL on a range of tasks where learning solely from a teacher is inadequate and focus primarily on scenarios with a privileged teacher. The results show that TGRL is either comparable or outperforms existing approaches without the need for manual hyperparameter tuning. The most challenging task we consider is robotic in-hand re-orientation of objects using only touch sensing. The superior performance of TGRL demonstrates its applicability to practical problems. Finally, we also present experiments showing the effectiveness of TGRL in learning from sub-optimal teachers.

2. Preliminaries

Reinforcement learning (RL). We consider the interaction between the agent and the environment as a discrete-time Partially Observable Markov Decision Process (POMDP) (Kaelbling et al., 1998) consisting of state space $\mathcal{S}$, observation space Ω, action space $\mathcal{A}$, state transition function $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$, reward function $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, observation function $\mathcal{O} : \mathcal{S} \to \Delta(\Omega)$, and initial state distribution $\rho_0 : \Delta(\mathcal{S})$. The environment is initialized at an initial state $s_0 \sim \rho_0$. At each timestep t, the agent observes the observation $o_t \sim O(\cdot|s_t), o_t \in \Omega$, takes action a_t determined by the policy π, receives reward $r_t = R(s_t, a_t)$, transitions to the next state $s_{t+1} \sim \mathcal{T}(\cdot|s_t, a_t)$, and observes the next observation $o_{t+1} \sim O(\cdot|s_{t+1})$. The goal of RL (Sutton and Barto, 2018) is to find the optimal policy π^* maximizing the expected cumulative rewards (i.e., expected return). Since the agent has access only to the observations and not to the underlying states, seminal work showed that the optimal policy may depend on the history of observations $\tau_t : \{o_0, a_0, o_1, a_1 ... o_t\}$, and not only on the current observation o_t (Kaelbling et al., 1998). Our aim is finding the optimal policy $\pi^* : \tau \to \Delta(\mathcal{A})$ that maximizes the following objective:

$$\pi^* = \arg\max_{\pi} J_R(\pi) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]. \tag{1}$$

Teacher-Student Learning (TSL). Suppose the agent (also referred to as the *student* in this paper) has access to a *teacher* that computes actions, $a_t^T \sim \bar{\pi}(\cdot|o_t^T)$, using an observation space that may be different from the student, $o_t^T \sim \tilde{O}(\cdot|s_t); o_t^T \in \Omega^T$. We are agnostic to how the teacher is constructed. In general, it's a black box that can be queried by the student for actions at any state the student encounters during training. Given such a $\bar{\pi}$, we aim to train the student policy, $\pi_\theta(\cdot|o_t)$, operating from observations, $o_t \in \Omega$.

A straightforward way to train the student is to use supervised learning to match the teacher's actions (Argall et al., 2009), $\max_\theta \mathbb{E}_{\bar{\pi}} \log \pi_\theta(a_t^T|o_t)$, where o_t is the student's observation and a_t^T is the teacher's action computed from its

observation, o_t^T, corresponding to the state s_t obtained by rolling out the teacher. However, recent work found that better student policies can be learned by using reinforcement learning to maximize the sum of rewards, where the reward is computed as the cross-entropy between the teacher and student's action distributions (Czarnecki et al., 2019). This leads to the following optimization problem:

$$\max_{\pi} J_I(\pi) := \max_{\pi} \mathbb{E}\left[-\sum_{t=0}^{H} \gamma^t H_t^X(\pi|\bar{\pi})\right] \tag{2}$$

where $H_t^X(\pi|\bar{\pi}) = -\mathbb{E}_{a\sim\pi(\cdot|\tau_t)}[\log \bar{\pi}(a|o_t^T)]$ is the Shannon cross-entropy, and for convenience in notation, π_θ is denoted as π. This objective is similar to DAgger (Ross et al., 2011) in optimizing the learning objective using the data collected by the student.

Problems in Teacher-Student Learning. To understand the problems in TSL, consider the recent result that implies that a student trained with TSL learns the statistical average of the teacher's actions for each observable state $o \in \Omega$:

Proposition 2.1. *In the setting described above, denote* $\pi^{TSL} = \arg\max_\pi J_I(\pi)$ *and* $f(o^T) : \Omega_T \to \Omega$ *as the function that maps the teacher's observations to the student's observations. Then, for any* $o^T \in \Omega_T$ *with* $o = f(o^T)$*, we have that* $\pi^{TSL}(o) = \mathbb{E}[\bar{\pi}(s)|o = f(o^T)]$.

Proof. See (Weihs et al., 2021) proposition 1 or (Warrington et al., 2021) theorem 1. □

The two problems due to Proposition 2.1: (i) Since the student's actions are the statistical average of the teacher's actions, it cannot outperform a sub-optimal teacher as there is no incentive to explore actions other than the teacher's. (ii) If the difference in observation spaces between the teacher and student is large, learning the statistical average can lead to sub-optimal performance. This is because the student cannot distinguish two different teachers' observations that appear identical in the student's observation space. As a result, the student policy does not mimic the teacher, but instead learns the *average action*, which can lead to sub-optimal performance (Eq. 1) (Kumor et al., 2021; Swamy et al., 2022). For example, in the Tiger Door environment, the student will follow the teacher until the second intersection (where the corridor splits into two paths for the two possible goal locations). The teacher policy takes a left or right action depending on where the goal is. Because the student does not observe the goal, it will learn to mimic the teacher's policy by assigning equal probability to actions leading to either of the sides. This policy is sub-optimal since the student will reach the goal only in 50% of trials.

3. Method

As Teacher-Student Learning can lead to a sub-optimal student, to outperform the teacher, the student needs to explore actions different from the teacher to find a better policy. We assume that the student has access to task rewards in addition to a teacher. This reward function can guide the exploratory process by determining when deviating from the teacher is fruitful. Following prior work (Czarnecki et al., 2019; Nguyen et al., 2022; Agarwal et al., 2022b), we consider the scenario of the student learning from a combination of reinforcement (Equation 1) and teacher-student (Equation 2) learning objectives:

$$\max_{\pi} J_{R+I}(\pi, \alpha) = \max_{\pi} \mathbb{E}\left[\sum_{t=0}^{H} \gamma^t (r_t - \alpha H_t^X(\pi|\bar{\pi}))\right] \tag{3}$$

where α is the balancing coefficient between the RL and imitation learning objectives. The joint objective can also be expressed as: $J_{R+I}(\pi, \alpha) = J_R(\pi) + \alpha J_I(\pi)$. Here, $J_I(\pi)$, can also be interpreted as a form of reward shaping (Ng et al., 1999), where the agent is negatively rewarded for taking actions that differ from the teacher's action.

As the balancing coefficient between the task reward and the teacher guidance, the value of α greatly impacts the algorithm's performance. A low α limits the guidance the student gets from the teacher, resulting in the usual challenges of learning solely from rewards. A high value of α can lead to excessive reliance on a sub-optimal teacher resulting in sub-optimal performance. Without a principled way to choose α, a common practice is to find the best value of α by conducting a separate and extensive hyperparameter search for every task (Schmitt et al., 2018; Nguyen et al., 2022). Besides the inefficiency of such search, as the agent progresses on a task, the amount of guidance it needs from the teacher can vary. Therefore, a constant α may not be optimal throughout training. Usually, the amount of guidance the student needs diminishes along the training process, but the exact dynamics of this trade-off are task-dependent, and per-task tuning is tedious, undesirable, and often computationally infeasible.

3.1. Teacher Guided Reinforcement Learning (TGRL)

Our notion of an optimal policy is one that achieves maximum cumulative task reward, and reinforcement learning optimizes this objective directly. Therefore, the teacher's supervision should only be used when it helps achieve better performance than just using task rewards. This idea is implemented by adding the following constraint: the performance of the policy learning from both rewards and teacher (i.e., the *main* policy) must be at par or outperform a policy trained using only task rewards (i.e., the *auxiliary* policy).

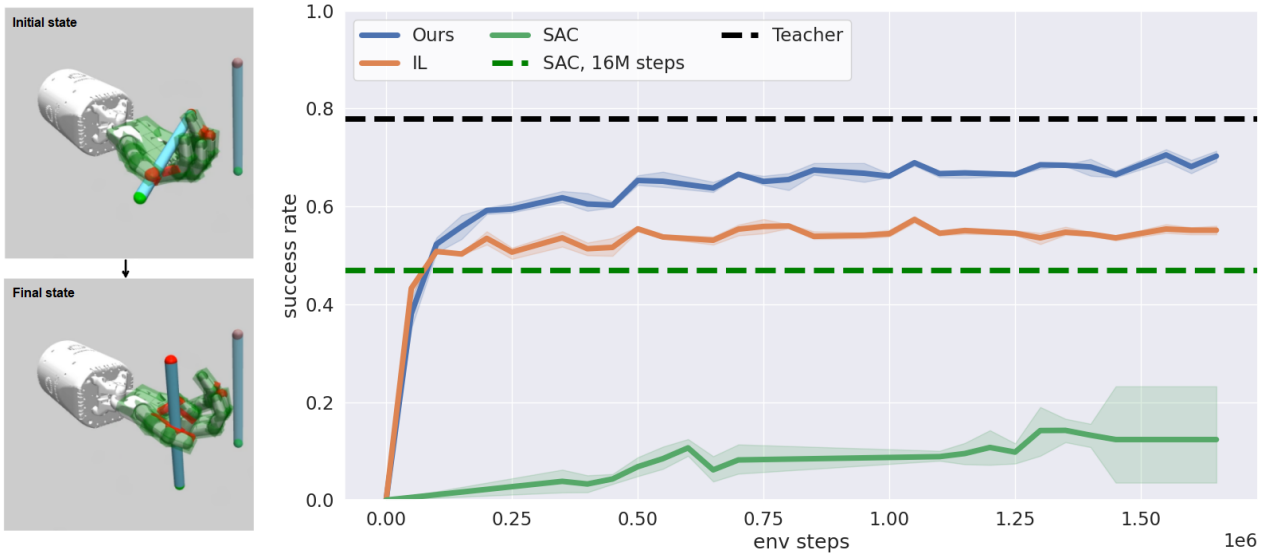

Figure 2: Success rate of a pen reorientation task by Shadow Hand robot, using tactile sensing only. While vanilla reinforcement learning takes a long time to converge, and Teacher-Student methods lead to a major drop in performance compared to the teacher, our algorithm is able to solve the task with reasonable sample efficiency.

Hence, our optimization problem becomes:

$$\max_{\pi} J_{R+I}(\pi, \alpha) \quad \text{s.t.} \quad J_R(\pi) \geq J_R(\pi_R) \tag{4}$$

where π_R is the *auxiliary* policy trained only using task reward (Eq. 1). Overall, our algorithm iterates between improving the auxiliary policy by solving $\max_{\pi_R} J_R(\pi_R)$ and solving the constrained problem in Equation 4. A recent paper (Chen et al., 2022) used a similar constraint in another context, to balance between exploration and exploitation in conventional RL. More formally, for $i = 1, 2, \ldots$ we iterate between two stages:

1. Partially solving $\pi_R^i = \arg\max_{\pi_R} J_R(\pi_R)$ to get an updated estimate $J_R(\pi_R^i)$.
2. Solving the i^{th} optimization problem:

$$\max_{\pi} J_{R+I}(\pi, \alpha) \text{ subject to } J_R(\pi) \geq J_R(\pi_R^i) \tag{5}$$

The constrained optimization problem in Equation 5 is solved using the dual lagrangian method, which has worked well in the reinforcement learning (Tessler et al., 2018; Bhatnagar and Lakshmanan, 2012). Using the Lagrange duality, we transform the constrained problem into an unconstrained min-max optimization problem. The dual problem corresponding to the primal problem in Equation 5 is:

$$\min_{\lambda \geq 0} \max_{\pi} \left[J_{R+I}(\pi, \alpha) + \lambda \left(J_R(\pi) - J_R(\pi_R) \right) \right] =$$

$$\min_{\lambda \geq 0} \max_{\pi} \left[(1+\lambda) J_{R+I}(\pi, \frac{\alpha}{1+\lambda}) - \lambda J_R(\pi_R) \right] \tag{6}$$

Where λ is the Lagrange multiplier. Full derivation can be found in appendix A. The resulting unconstrained optimization problem is comprised of two optimization problems. The first optimization problem (i.e., the inner loop) solves for π. Since $J_R(\pi_R)$ is independent of π, this optimization is akin to solving the combined objective of Equation 3 but with the importance of the imitation learning reward set to $\frac{\alpha}{1+\lambda}$. Further, for $\lambda \geq 0$, we have $\alpha \geq \frac{\alpha}{1+\lambda} \geq 0$, which means that α is the upper bound on the importance of imitation rewards. We also refer to the importance of imitation rewards, $\frac{\alpha}{1+\lambda}$, as the *balancing coefficient*.

The second stage involves solving for λ. The dual function is always convex since it is the point-wise minimum of a linear function in λ (Boyd et al., 2004). Therefore it can be solved with gradient descent without worrying about local minimas. The gradient of Equation 6 with respect to the lagrange multiplier, λ, leads to the following update rule:

$$\lambda_{new} = \lambda_{old} - \mu[J_R(\pi) - J_R(\pi_R)] \tag{7}$$

Where μ is the step size for updating λ. See appendix A for full derivation. This update rule is intuitive: If the policy using the teacher (π) achieves more task reward than the auxiliary policy (π_R) trained without the teacher, then λ is decreased, which in turn increases $\frac{\alpha}{1+\lambda}$, making the optimization of π more reliant on the teacher in the next iteration. Otherwise, if π_R achieves a higher reward than π, then increase in λ decreases the importance of the teacher.

When utilizing Lagrange duality to solve a constrained optimization problem, it is necessary to consider the duality gap which is the difference between the optimal dual and primal values. A non-zero duality gap implies that the solution

Algorithm 1 Teacher Guided Reinforcement Learning (TGRL)

Input: λ_{init}, α, N_{collect}, N_{update}, μ
Initialize policies π and π_R, $\lambda_0 \leftarrow \lambda_{init}$
for $i = 1 \cdots$ **do**
  Collect N_{collect} new trajectories and add them to the replay buffer.
  for $j = 1 \cdots N_{\text{update}}$ **do**
    Sample a batch of trajectories from the replay buffer.
    Update Q_R and Q_I.
    Update π_R by maximizing Q_R
    Update π by maximizing $Q_R + \frac{\alpha}{1+\lambda} Q_I$
  end for
  Estimate $J_R(\pi) - J_R(\pi_R)$ using Eq. 8
  $\lambda_i \leftarrow \lambda_{i-1} + \mu[J_R(\pi) - J_R(\pi_R)]$
end for π =0

of the dual problem is only a lower bound to the primal problem and does not necessarily provide the exact solution (Boyd et al., 2004). Under certain assumptions listed in proposition 3.1, we show that for our optimization problem, there is *no duality gap* (proof in Appendix A). Thus, solving the dual problem also solves the primal problem.

Proposition 3.1. *Denote $\eta_i = J_R(\pi_R^i)$. Suppose that the rewards function $r(s,a)$ and the cross-entropy term $H^X(\pi|\bar{\pi})$ are bounded. Then for every $\eta_i \in \mathbb{R}$ the primal and dual problems described in Eq. 5 and Eq. 6 have no duality gap. Moreover, if the series $\{\eta_i\}_{i=1}^{\infty}$ converges, then there is no duality gap in the limit.*

Notice that in the general case, the cross-entropy term can reach infinity when the support of the policies does not completely overlap, violating the assumption of $H^X(\pi|\bar{\pi})$ being bounded. As a remedy, we clip the value of the cross-entropy term in our implementation of TGRL.

3.2. Implementation

Off-policy approach: We implemented our algorithm using an off-policy actor-critic approach. Off-policy learning allows data collected by both policies, π and π_R, to be stored in a common replay buffer used for training both policies. Our objective is to maximize the joint Q-value: $Q_{R+I} = Q_R + \frac{\alpha}{1+\lambda} Q_I$, where Q_R, Q_I denote the Q-value of actions with respect to the task (Equation 1) and imitation (Equation 2) rewards respectively. Instead of directly learning, Q_{R+I}, we train two critic networks, Q_R and Q_I, and combine their values to estimate Q_{R+I}. This choice enables us to estimate Q_{R+I} for different values of λ without any need for re-training the critics. We also represent π and π_R with separate actor networks optimized to maximize the corresponding Q-values. In the data collection step, half of the trajectories are collected using π and the other half using π_R. See Algorithm 1 for an outline of our method and Appendix B for further details.

Estimating the performance difference: As shown in Equation 7, the gradient of the dual problem with respect to λ is the performance difference between the two policies, $J_R(\pi) - J_R(\pi_R)$. To estimate the performance difference, one option is to perform Monte-Carlo estimation – i.e., roll-out trajectories using both policies and determine the empirical estimate of cumulative rewards. However, a good estimate of cumulative rewards requires collecting a large number of trajectories which would make our method data inefficient. Another data-efficient option is to reuse data in the replay buffer for estimating the performance difference. Because the data in the replay buffer was not collected using the current policies, we make the off-policy correction using approximations obtained by extending prior results from (Kakade and Langford, 2002; Schulman et al., 2015) known as the *objective difference lemma* to the off-policy case:

Proposition 3.2. *Let $\rho(s,a,t)$ be the distribution of states, actions, and timesteps currently in the replay buffer. Then the following is an unbiased approximation of the performance difference:*

$$J_R(\pi) - J_R(\pi_R) = \mathbb{E}_{(s,a,t)\sim\rho}[\gamma^t (A_{\pi_R}(s,a) - A_{\pi}(s,a))] \quad (8)$$

Another challenge in estimating the gradient of λ (i.e., the performance difference between the student and the teacher policies) is the variability in the scale of the policies' performance across different environments and during training. This makes it difficult to determine an appropriate learning rate for the weighting factor λ, which will work effectively in all settings. To address this issue, we found it necessary to normalize the performance difference value during the training process. This normalization allows us to use a fixed learning rate across all of our experiments.

4. Experiments

We perform four sets of experiments. In Sec. 4.1, we provide a comparison to previous work in cases where the teacher is too good to mimic. In Sec. 4.2 we solve an object re-orientation problem with tactile sensors, a difficult partial observable task that both RL and TSL struggle to solve. In Sec. 4.3 we look into the ability of the TGRL agent to surpass the teacher's performance. Finally, in Sec. 4.4 we do ablations of our own method to show the contribution of individual components.

4.1. TGRL performs well, without a need for hyperparameter tuning

We provide empirical evidence (1) showcasing the robustness of TGRL to choice of hyperparameters controlling the

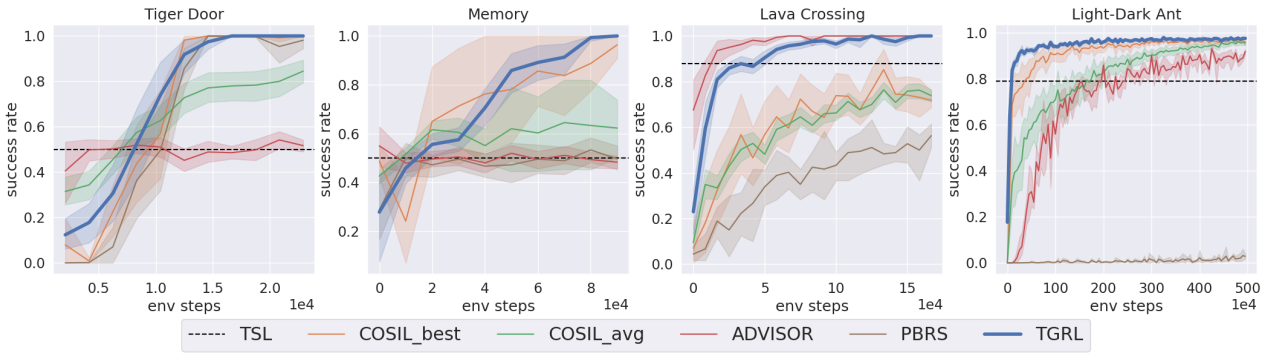

Figure 3: Comparing TGRL (blue) against algorithms proposed in prior work. TGRL is the only algorithm that performs consistently well across all environments.

update of the balancing coefficient and (2) comparison to prior work. We compare TGRL to the following:

TSL. A pure Teacher-Student Learning approach that optimizes only Equation 2.

COSIL (Nguyen et al., 2022). This algorithm also uses entropy-augmented RL (Eq. 3) to combine the task reward and the teacher's guidance. To adjust the balancing coefficient α, they propose an update rule for maintaining a fixed distance ($\bar{D}$) between the student's and teacher's policies by minimizing $\alpha(J_I(\pi) - \bar{D})$ using gradient descent. Choosing the right value of $\bar{D}$ is a challenge since its unknown apriori how similar the student and the teacher should be. Moreover, $\bar{D}$ can change drastically between environments, depending on the action space support. To tackle this issue, we run a hyperparameter search with $N = 8$ different values of $\bar{D}$ and report performance for the best hyperparameter per task ($COSIL_{best}$) and average performance across hyperparameters ($COSIL_{avg}$).

ADVISOR-off. An off-policy version of the algorithm from (Weihs et al., 2021) that uses a state-dependent balancing coefficient. First, an imitation policy is trained using only teacher-student learning loss. Then, for every state, the action distribution of the teacher policy is compared against the imitation policy. The states in which the two policies disagree are deemed to be ones where there is an information gap. For such states, the teacher is trusted less and more importance is given to the task reward.

PBRS (Walsman et al., 2023). A potential-based reward shaping (PBRS) method based on (Ng et al., 1999) to mitigate issues with Teacher-Student Learning. PBRS uses a given value function $V(s)$ to assign higher rewards to more beneficial states, which can lead the agent to trajectories favored by the policy associated with that value function:

$$r_{new} = r_{task} + \gamma V(s_{t+1}) - V(s_t) \tag{9}$$

where r_{task} is the task reward. Since their algorithm is on-policy, for fair comparison, we created an off-policy version of this method. For this, first, we train an imitation policy by minimizing only the teacher-student learning loss (Eq. 2). Then, we train a neural network to represent the value function of this imitation policy. Using this value function, we obtain an augmented rewards function described in Equation 9, which is then used to train a policy using the SAC algorithm (Haarnoja et al., 2018).

Experimental Domains. We experiment on diverse problems taken from prior works studying issues in Teacher-Student Learning spanning discrete and continuous action spaces, and both proprioceptive and pixel observations. For a description of each environment, see appendix B. For a fair comparison, we used the same code and Q-learning hyperparameters for all algorithms, tuning only the hyperparameters involved in balancing the teacher supervision against the task reward. The Q-learning hyperparameters correspond to hyperparameters of the RL algorithm chosen from the best-performing SAC agent. For TGRL we only used a single value for the initial coefficient λ_{init} and coefficient learning rate μ for all tasks (more details in Appendix B).

Comparison to Baselines. Results in Figure 3 show that while each baseline method succeeds in some sub-set of tasks, no baseline method is effective on all tasks. In contrast, TGRL, solves all tasks successfully with data efficiency comparable to the best baseline in each task. Most importantly, TGRL requires no task-specific hyperparameter tuning. Notice that $COSIL$ demonstrates comparable performance on three out of four tasks when its hyperparameters are carefully tuned (i.e., $COSIL_{best}$). However, the average performance across all hyperparameters (i.e., $COSIL_{avg}$) is significantly lower. This highlights sensitivity of $COSIL$ to the choice of hyperparameters. While *PBRS* does not require hyperparameter tuning, consistent with results from another work (Cheng et al., 2021), it converges slower than other teacher-student methods and doesn't consistently perform well across tasks.

ADVISOR achieves good performance on *Lava Crossing* and *Light-Dark Ant* tasks but converged to a sub-optimal pol-

icy achieving comparable performance to Teacher-Student Learning (TSL) on *Tiger Door* and *Memory* environments. The sub-optimal performance is due to a fundamental limitation of the *ADVISOR* algorithm. As a reminder, *ADVISOR* works by identifying states where the student has insufficient information to follow the teacher. For such states, instead of imitating the teacher, task rewards are used to decide the action. In the *Tiger Door* environment (see Figure 1), the student has sufficient information to follow the teacher until the state at which the two arms of the environment split. However, this is too late for the student should deviate from the teacher – to achieve optimal performance, the student should have deviated from the teacher earlier to goto the pink button. This example illustrates a problem that *ADVISOR* encounters in environments where the information-gathering actions deviating from the teacher need to be performed before encountering the state at which the student cannot imitate the teacher.

Robustness to Hyperparameters. To demonstrate the robustness of the choice of λ_{init}, we experimented with different values on *Lava Crossing* environment. The results in Figure 5 (left) shows that irrespective of the choice of λ_{init}, TGRL achieves the same asymptotic performance. This indicates that TGRL can effectively adjust λ, regardless of its initial value, λ_{init}.

4.2. TGRL can solve difficult environments with significant partial observability.

To investigate the performance of our method on a more practical task with severe partial observability, we experimented with the Shadow hand test on task of re-orienting a pen to a target pose using only touch sensors and proprioceptive information (Melnik et al., 2021). Consider a Teacher-Student setup where the teacher policy observes the pen's pose and velocity. The student, however, only has access to an array of tactile sensors located on the palm of the hand and the phalanges of the fingers. To solve the task, the student needs to move his fingers along the pen and use the reading of these sensors to infer its pose. The teacher does not need to take these information-gathering actions. Thus, just mimicking the teacher will result in a sub-optimal student.

To train all agents, the reward was set to the negative of the distance between the current pen's pose and the goal. The pen has rotational symmetry around the z axis, so the distance was computed only over rotations around the x and y axes. A trajectory was considered successful if the pen reached the goal orientation within 0.1 radians of the goal pose. The performance is averaged over 1,000 randomly sampled initial and goal poses.

The results are in Figure 2. First, to assess the difficulty of the task, we report the results of an RL agent trained with

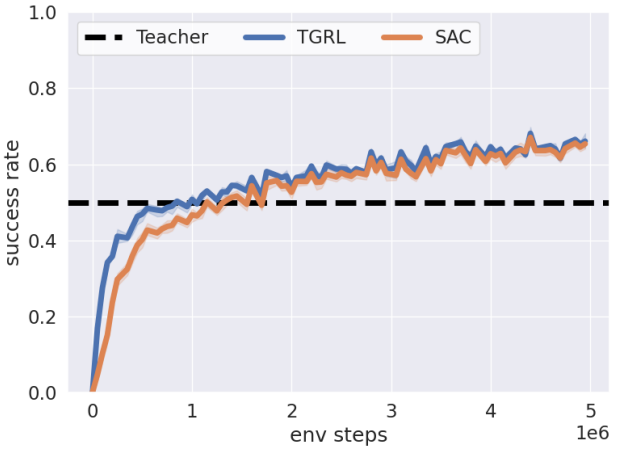

Figure 4: TGRL performance in the *Shadow Hand* environment with a sub-optimal teacher. TGRL is able to surpass the teacher and achieves asymptotic performance similar to that of RL.

Soft Actor-Critic and Hindsight Experience Replay (HER) (Andrychowicz et al., 2017) over the student's observation space. This RL agent has achieved a 47% success rate, demonstrating the difficulty of learning this task using RL alone. The teacher, trained also using SAC and HER but on the full state space, achieved a 78% success rate. Performing vanilla Teacher-Student learning using this teacher resulted in an agent with a 54% success rate. This performance gap shows that just imitating the teacher is not sufficient, and a deviation from the teacher's action is indeed required to learn a good policy. With TGRL, the agent achieved a significantly higher success rate of 73%. These results demonstrate the usefulness of our algorithm and its ability to use the teacher's guidance while learning from the reward at the same time. TGRL also outperforms baseline methods (see Appendix C for more details).

4.3. TGRL can surpass the Teacher's peformance

To evaluate the ability of TGRL to surpass a sub-optimal teacher, we conducted experiments in several domains. For the *Tiger Door* and *Lava Crossing* environments, we constructed teachers with different optimality levels ranging from 40% to 100% success rate. Results in Table 1 show that even with a sub-optimal teacher, TGRL learns the optimal policy in the *Tiger Door* environment. *Lava Crossing* is a more challenging task, where vanilla SAC achieves 0% success rate. Therefore, combining learning from task reward and teacher supervision allows TGRL to achieve better performance than the sub-optimal teacher, but still not 100% success rate.

In addition, experimented with a variant of the Shadow Hand environment, where both student and teacher have access

to the full state, but the teacher is sub-optimal. The results depicted in Figure 4 show that TGRL converges fast to the teacher's performance but than able to keep improving by utilizing task reward supervision, eventually exceeding the teacher's performance.

Table 1: TGRL Student's success rate for sub-optimal teachers. Mean and 95% CI over 10 random seeds. All agents were trained until convergence.

Teacher's Success Rate	100%	80%	40%
Student's success rate - Tiger Door	$100\pm0.0\%$	$100\pm0.0\%$	$100\pm0.0\%$
Student's success rate - Lava Crossing	$100\pm0.0\%$	$97 \pm 0.8\%$	$65 \pm 8.1\%$

4.4. Ablations

Joint versus separate replay buffer. We empirically found that having a joint replay buffer between the two policies, π and π_R, is necessary for good performance. In Figure 5, we compare the performance of our method with separate and joint replay buffers for the two policies on *Light-Dark Ant* environment. As a reminder, the auxiliary policy (π_R) limits the set of feasible policies in Equation 4. In tasks where it is hard to learn a good policy using only task rewards, the performance of π_R will be bad leading to a loose constraint which will be ineffective. Combining the replay buffer allows π_R to learn from trajectories collected by the main policy (π), thus enabling it to achieve better performance. This, in turn, leads to a stricter constraint on the main policy, pushing it to achieve better performance.

Fixed versus adaptive balancing coefficient. A benefit of TGRL is that the balancing coefficient in the combined objective (Equation 3) dynamically changes during the training process based on the value of λ. To investigate if an adaptive coefficient is indeed beneficial, we conducted an ablation study wherein we trained policies in the *Shadow Hand* environment with fixed coefficients. Figure 6 shows that the balancing coefficient of TGRL changes during training (left plot). At the start of training, the value is high, indicating that the teacher is given high importance. As the agent learns, the value decreases, indicating that learning from rewards is given more importance in the later stages of training. The results in the right plot of Figure 6 show that TGRL with a dynamically changing balancing coefficient outperforms ablated versions with fixed coefficients. This result indicates that TGRL goes beyond mitigating the need for searching the balancing coefficient – it also outperforms a fixed balancing coefficient found by rigorous hyperparameter search.

5. Discussion

While TGRL improved performance across all tasks, it has its limitations. If the agent needs to deviate substantially from the teacher, then intermediate policies during learning might have worse performance than the teacher before the agent is able to leverage rewards to improve performance. In such a case, the imitation learning policy is a local minimum, overcoming which may require additional exploration incentives (Pathak et al., 2017). Second, for the constraint in Equation 4 to be meaningful, π_R trained only with task rewards should achieve reasonable performance. While having a shared replay buffer with π may help π_R in some hard exploration problems, learning of π_R can fail which would make the constraint ineffective.

An interesting investigation that we leave to future work is to have a state-dependent balancing coefficient. As the difference between the teacher's and student's actions can be state-dependent, such flexibility can accelerate convergence and lead to better performance.

Acknowledgements

We thank the members of the Improbable AI lab for the helpful discussions and feedback on the paper. We are grateful to MIT Supercloud and the Lincoln Laboratory Supercomputing Center for providing HPC resources. The research was supported in part by the MIT-IBM Watson AI Lab, Hyundai Motor Company, DARPA Machine Common Sense Program, and ONR MURI under Grant Number N00014-22-1-2740.

Author Contributions

Idan Shenfeld Identified the current problem with teacher-student algorithms, developed the TGRL algorithm, derived the theoretical results, conducted the experiments, and wrote the paper. **Zhang-Wei Hong** helped in the debugging, implementation and the choice of necessary experiments and ablations. **Aviv Tamar** helped derive the theoretical results and provided feedback on the writing. **Pulkit Agrawal** oversaw the project. He was involved in the technical formulation, research discussions, paper writing, overall advising, and positioning of the work.

References

Omid Madani, Steve Hanks, and Anne Condon. On the undecidability of probabilistic planning and infinite-horizon partially observable markov decision problems. In *AAAI/IAAI*, pages 541–548, 1999.

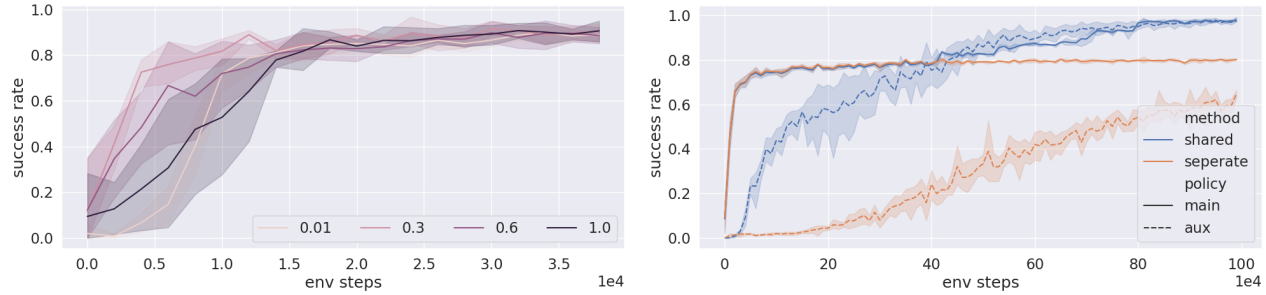

Figure 5: (Left) TGRL performance on *Lava Crossing* for different values of λ_{init}. (Right) Comparing the effect of separate and joint replay buffers between the main and auxiliary policies, π and π_R, evaluated on the *Light-Dark Ant* environment.

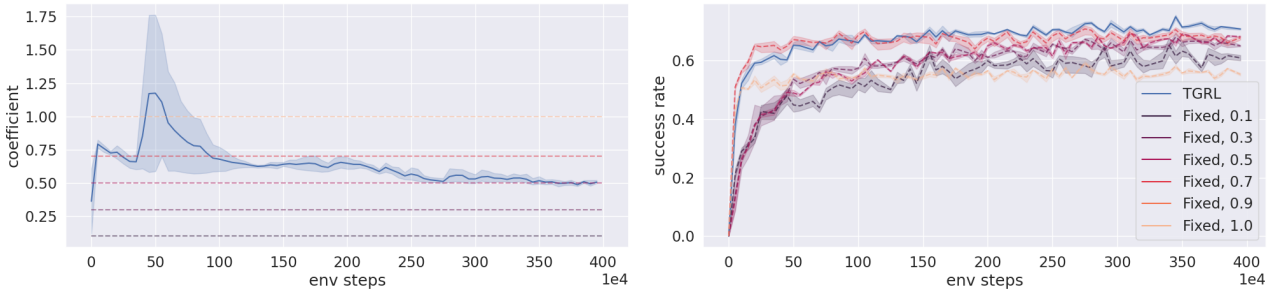

Figure 6: Adaptively balancing teacher guidance and rewards results in better asymptotic performance compared to fixing the balancing coefficient (λ). Experiment on the *Shadow Hand* environment. (Left) Dynamics of λ during training: At the start of training the agent relies more on the teacher and gradually the coefficient decreases, indicating more reliance on rewards. (Right) Performance of TGRL (blue) and ablated versions of TGRL using a fixed balancing coefficient (λ).

Christos H Papadimitriou and John N Tsitsiklis. The complexity of markov decision processes. *Mathematics of operations research*, 12(3):441–450, 1987.

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*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron Courville, and Marc G Bellemare. Beyond tabula rasa: Reincarnating reinforcement learning. *arXiv preprint arXiv:2206.01626*, 2022a.

Andrey Kurenkov, Ajay Mandlekar, Roberto Martin-Martin, Silvio Savarese, and Animesh Garg. Ac-teach: A bayesian actor-critic method for policy learning with an ensemble of suboptimal teachers. *arXiv preprint arXiv:1909.04121*, 2019.

Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *arXiv preprint arXiv:1709.10087*, 2017.

Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning quadrupedal locomotion over challenging terrain. *Science robotics*, 5(47): eabc5986, 2020.

Tao Chen, Jie Xu, and Pulkit Agrawal. A system for general in-hand object re-orientation. *Conference on Robot Learning*, 2021.

Gabriel Margolis, Tao Chen, Kartik Paigwar, Xiang Fu, Donghyun Kim, Sangbae Kim, and Pulkit Agrawal. Learning to jump from pixels. *Conference on Robot Learning*, 2021.

Weinan Zhang, Xiangyu Zhao, Li Zhao, Dawei Yin, Grace Hui Yang, and Alex Beutel. Deep reinforcement learning for information retrieval: Fundamentals and advances. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2468–2471, 2020.

Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *arXiv preprint arXiv:1504.00702*, 2015.

Daniel Kumor, Junzhe Zhang, and Elias Bareinboim. Sequential causal imitation learning with unobserved confounders. *Advances in Neural Information Processing Systems*, 34:14669–14680, 2021.

Gokul Swamy, Sanjiban Choudhury, J Andrew Bagnell, and Zhiwei Steven Wu. Sequence model imitation learning with unobserved contexts. *arXiv preprint arXiv:2208.02225*, 2022.

Michael L Littman, Anthony R Cassandra, and Leslie Pack Kaelbling. Learning policies for partially observable environments: Scaling up. In *Machine Learning Proceedings 1995*, pages 362–370. Elsevier, 1995.

Andrew Warrington, Jonathan W Lavington, Adam Scibior, Mark Schmidt, and Frank Wood. Robust asymmetric learning in pomdps. In *International Conference on Machine Learning*, pages 11013–11023. PMLR, 2021.

Luca Weihs, Unnat Jain, Iou-Jen Liu, Jordi Salvador, Svetlana Lazebnik, Aniruddha Kembhavi, and Alex Schwing. Bridging the imitation gap by adaptive insubordination. *Advances in Neural Information Processing Systems*, 34: 19134–19146, 2021.

Hai Nguyen, Andrea Baisero, Dian Wang, Christopher Amato, and Robert Platt. Leveraging fully observable policies for learning under partial observability. *arXiv preprint arXiv:2211.01991*, 2022.

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron Courville, and Marc G Bellemare. Reincarnating reinforcement learning: Reusing prior computation to accelerate progress. *arXiv preprint arXiv:2206.01626*, 2022b.

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

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. 2018.

Brenna D Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 2009.

Wojciech M Czarnecki, Razvan Pascanu, Simon Osindero, Siddhant Jayakumar, Grzegorz Swirszcz, and Max Jaderberg. Distilling policy distillation. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1331–1340. PMLR, 2019.

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, 1999.

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.

Eric Chen, Zhang-Wei Hong, Joni Pajarinen, and Pulkit Agrawal. Redeeming intrinsic rewards via constrained optimization. *arXiv preprint arXiv:2211.07627*, 2022.

Chen Tessler, Daniel J Mankowitz, and Shie Mannor. Reward constrained policy optimization. *arXiv preprint arXiv:1805.11074*, 2018.

Shalabh Bhatnagar and K Lakshmanan. An online actor–critic algorithm with function approximation for constrained markov decision processes. *Journal of Optimization Theory and Applications*, 153(3):688–708, 2012.

Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *In Proc. 19th International Conference on Machine Learning*. Citeseer, 2002.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 1889–1897, 2015.

Aaron Walsman, Muru Zhang, Sanjiban Choudhury, Dieter Fox, and Ali Farhadi. Impossibly good experts and how to follow them. In *The Eleventh International Conference on Learning Representations*, 2023.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*, 2018.

Ching-An Cheng, Andrey Kolobov, and Adith Swaminathan. Heuristic-guided reinforcement learning. *Advances in Neural Information Processing Systems*, 34: 13550–13563, 2021.

Andrew Melnik, Luca Lach, Matthias Plappert, Timo Korthals, Robert Haschke, and Helge Ritter. Using tactile sensing to improve the sample efficiency and performance of deep deterministic policy gradients for simulated in-hand manipulation tasks. *Frontiers in Robotics and AI*, page 57, 2021.

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. In *Advances in Neural Information Processing Systems*, pages 5048–5058, 2017.

Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *Proceedings of the 34th International Conference on Machine Learning*, pages 2778–2787, 2017.

Santiago Paternain, Luiz Chamon, Miguel Calvo-Fullana, and Alejandro Ribeiro. Constrained reinforcement learning has zero duality gap. *Advances in Neural Information Processing Systems*, 32, 2019.

R Tyrrell Rockafellar. *Convex analysis*, volume 18. Princeton university press, 1970.

Robert Platt Jr, Russ Tedrake, Leslie Kaelbling, and Tomas Lozano-Perez. Belief space planning assuming maximum likelihood observations. 2010.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 2015.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Tianwei Ni, Benjamin Eysenbach, and Ruslan Salakhutdinov. Recurrent model-free rl can be a strong baseline for many pomdps. In *International Conference on Machine Learning*, pages 16691–16723. PMLR, 2022.

A. Derivations and Proofs

A.1. Derivation of the Dual Problem

Denote $\eta_i = J_R(\pi^i_{RL})$, and given the Primal Problem we derived in Eq. 5:

$$\max_{\pi} J_{R+I}(\pi, \alpha) \quad \text{subject to} \quad J_R(\pi) \geq \eta_i$$

The corresponding Lagrangian is:

$$\mathcal{L}(\pi, \lambda) = J_{R+I}(\pi, \alpha) + \lambda \left(J_R(\pi) - \eta_i\right) =$$

$$\mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t (r_t - \alpha H_t^X(\pi|\bar{\pi}))\right] + \lambda \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r_t\right] - \lambda\eta_i =$$

$$\mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t \left((1+\lambda) r_t - \alpha H_t^X(\pi|\bar{\pi})\right)\right] - \lambda\eta_i =$$

$$\mathbb{E}_\pi \left[(1+\lambda) \sum_{t=0}^{\infty} \gamma^t \left(r_t - \frac{\alpha}{1+\lambda} H_t^X(\pi|\bar{\pi})\right)\right] - \lambda\eta_i =$$

$$(1+\lambda) J_{R+I}(\pi, \frac{\alpha}{1+\lambda}) - \lambda\eta_i$$

And therefore out Dual problem is:

$$\min_{\lambda \geq 0} \max_{\pi} \left[(1+\lambda) J_{R+I}(\pi, \frac{\alpha}{1+\lambda}) - \lambda\eta_i\right]$$

A.2. Derivation of update rule for λ

The gradient of the dual problem with respect to λ is:

$$\nabla_\lambda \left[(1+\lambda) J_{R+I}(\pi, \frac{\alpha}{1+\lambda}) - \lambda\eta_i\right] =$$

$$\nabla_\lambda \left[\mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t \left((1+\lambda) r_t - \alpha H_t^X(\pi|\bar{\pi})\right)\right] - \lambda\eta_i\right] =$$

$$\mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r_t\right] - \eta_i =$$

$$J_R(\pi) - \eta_i$$

A.3. Duality Gap - Proof for Proposition 3.1

We start by restating our assumptions and discuss why they hold for our problem:

Assumption A.1. The rewards function $r(s, a)$ and the cross-entropy term $H^X(\pi|\bar{\pi})$ are bounded.

Justification for A.1. This is achieved by using a clipped version of the cross entropy term. We will add that we found the clipping helpful in practice since it stops this term from reaching infinity when the support of the teacher and the student action distributions are not the same.

Assumption A.2. The sequence $\{\eta_i\}_{i=1}^{\infty}$ is monotincally increasing and converging, i.e., there exist $\eta \in \mathbb{R}$ such that $\lim_{i\to\infty} \eta_i = \eta$.

Justification for A.2. We will remind that the sequence $\{\eta_i\}_{i=1}^{\infty}$ is the result of incrementally solving $\max_{\pi_R} J_R(\pi_R)$. Having this sequence be monotonically increasing is equivalent to a guarantee for policy improvement in each optimization step, an attribute of several RL algorithms such as Q-learning or policy gradient (Sutton and Barto, 2018). Regarding convergence, since the reward is upper bound from assumption A.1, then we have an upper bounded monotonically increasing sequence of real numbers, which is proved to converge.

Assumption A.3. There exist $\epsilon > 0$ such that for all i, the value of η_i is at most $J_R(\pi^*) - \epsilon$.

Justification for A.3. This assumption is equivelant to stating that $J_R(\pi^*) - J_R(\pi_R) > 0$, meaning that π_R is never optimal. Without further assumption on the algorithm used to optimize π_R, we can not guarantee that this will not happen. However, if it happens, it means that we were able to find the optimal policy, and therefore there is no need to continue with the optimization procedure. As a remedy, we will define a new sequence $\{\tilde{\eta}_i\}_{i=1}^{\infty}$ where $\tilde{\eta}_i = \eta_i - \epsilon$ and will use it instead of the original η_i. Since ϵ can be as small as we want, its effect on the algorithm is negligible and it served mainly for the completeness of our theory.

Before going into our proof, we will cite Theorem 1 of (Paternain et al., 2019), which is the basis of our results:

Theorem A.4. *Given the following optimization problem:*

$$P^* = \max_{\pi} \mathbb{E}_{\pi}\left[\sum_{i=0}^{H} \gamma^t r_0(s_t, a_t)\right] \quad \textit{subject to}$$

$$\mathbb{E}_{\pi}\left[\sum_{i=0}^{H} \gamma^t r_i(s_t, a_t)\right] \geq c_i, \quad i = 1...m,$$

And its Dual form:

$$D^* = \min_{\lambda \geq 0} \max_{\pi} \mathbb{E}_{\pi}\left[\sum_{i=0}^{H} \gamma^t r_0(s_t, a_t)\right] +$$

$$\lambda \sum_{i=1}^{m} \left[\mathbb{E}_{\pi}\left[\sum_{i=0}^{H} \gamma^t r_i(s_t, a_t)\right] - c_i\right]$$

suppose that r_i is bounded for all $i = 0, ..., m$ and that Slater's condition holds. Then, strong duality holds, i.e., $P^* = D^*$.

Having stated that, we will move to prove the two parts of our proposition:

Proposition A.5. *Given assumption A.1 and A.3, for every $\eta_i \in \mathbb{R}$, the constrained optimization problem Eq. 5 and its dual problem defined in Eq. 6 do not have a duality gap.*

Proof. We align our problem with Theorem A.4 notations by denote as follows:

$$r_0 : r_t - \alpha H_t^X, \quad r_1 : r_t, \quad c_1 : \eta_i$$

And we can see that our problem is a specific case of the optimization problem defined above. For every η_i, there is a set feasible solutions in the form of an ϵ-neighborhood of π^*. This holds since $J_R(\pi^*) > J_R(\pi) - \epsilon$ for every $\pi \notin \pi^*$. Therefore, Slater's condition holds as it required that the feasible solution set will have an interior point. Together with assumption A.1, we have all that we need to claim that Theorem A.4 applies to our problem. Therefore, there is no duality gap. □

Proposition A.6. *Given all our assumptions, the constrained optimization problem at the limit:*

$$\max_{\pi} J_{R+I}(\pi, \alpha) \quad \textit{subject to} \quad J_R(\pi) \geq \eta$$

has no duality gap.

Proof. Our proof will be based on the Fenchel-Moreau theorem (Rockafellar, 1970) that states:

If (i) Slater's condition holds for the primal problem and (ii) its perturbation function P(ξ) is concave, then strong duality holds for the primal and dual problems.

Denote $\eta_{\lim}$ the limit of the sequence. Without loss of generality, we assume that $\eta_{\lim} = J_R(\pi^*) - \epsilon$. If not, we will just adjust ϵ accordingly. As in the last proof, Slater's condition holds since there is a set of feasible policies for the problem. Regarding the second requirement, the sequence of perturbation functions for our problem is:

$$P(\xi) = \lim_{i\to\infty} P_i(\xi)$$

$$\text{where} \quad P_i(\xi) = \max_{\pi} J_{R+I}(\pi, \alpha)$$

$$\text{subject to} \quad J_R(\pi) \geq \eta_i + \xi$$

Notice that this is a scalar function since $P_i(\xi)$ is the maximum objective itself, not the policy that induces it. We will now prove that this sequence of functions converges point-wise:

- For all $\xi > \epsilon$ we claim that $P(\xi) = \lim_{i\to\infty} P_i(\xi) = -\infty$. As a reminder η_i converged to $J_R(\pi^*) - \epsilon$. It means that there exists N such that for all $n > N$, we have $|\eta_n - J_R(\pi^*) + \epsilon| < \frac{\xi}{2} - \epsilon$. Moreover, since

$J_R(\pi^*) - \epsilon$ is also the upper bound on the series of η_i we can remove the absolute value and get:

$$0 \leq J_R(\pi^*) - \epsilon - \eta_n < \frac{\xi}{2} - \epsilon$$

This yields the following constraint:

$$J_R(\pi_\theta) \geq \eta_n + \xi > J_R(\pi^*) - \frac{\xi}{2} + \xi = J_R(\pi^*) + \frac{\xi}{2}$$

But since $\xi > \epsilon > 0$ and π^* is the optimal policy, no policies are feasible for this constraint, so from the definition of the perturbation function, we have $P_n(\xi) = -\infty$. This holds for all $n > N$ and, therefore also $\lim_{i\to\infty} P_i(\xi) = -\infty$.

- For all $\xi \leq \epsilon$ we will prove convergence to a fixed value. First, we claim that the perturbation function has a lower bound. This is true since the reward function and the cross-entropy are bounded, and the perturbation function value is a discounted sum of them.
  In addition, the sequence of $P_i(\xi)$ is monotonically decreasing. To see it, remember that the sequence $\{\eta_i\}_{i=1}^{\infty}$ is monotonically increasing. Since $J_R(\pi)$ is also upper bounded by $J_R(\pi^*)$, then the feasible set of the $(i+1)$ problem is a subset of the feasible set of the i problem, and all those which came before. Therefore if the solution to the i problem is still feasible it will also be the solution to the $i+1$ problem. If not, then it has a lower objective (since it was also feasible in the i problem), resulting in a monotonically decreasing sequence. Finally, for every η_i there is at least one feasible solution, $J_R(\pi^*)$, meaning the perturbation function has a real value. To conclude, $\{P_i(\xi)\}_{i=1}^{\infty}$ is a monotonically decreasing, lower-bounded sequence in $\mathbb{R}$ in therefore it converged.

After we established point-wise convergence to a function $P(\xi)$, all that remain is to proof that this function is concave. According to proposition A.5, each optimization problem doesn't have a duality gap, meaning its perturbation function is concave. Since every function in the sequence is concave, and there is pointwise convergence, $P(\xi)$ is also concave. To conclude, from the Fenchel-Moreau theorem, our optimization problem doesn't have a duality gap in the limit. □

A.4. Performance Difference Estimation - Proof for Proposition 3

Proposition: Let $\rho(s, a, t)$ be the distribution of states, actions, and timesteps currently in the replay buffer. Then the following is an unbiased approximation of the performance difference:

$$J_R(\pi) - J_R(\pi_R) =$$

$$\mathbb{E}_{(s,a,t)\sim\rho}[\gamma^t(A_{\pi_R}(s,a) - A_\pi(s,a))]$$

Proof: Let π_{RB} be the behavioral policy induced by the data currently in the replay buffer, meaning:

$$\forall s \in S \quad \pi_{RB}(a|s) = \frac{\sum_{a' \in RB(s)} I_{a'=a}}{\sum_{a' \in RB(s)} 1}$$

Using lemma 6.1 from (Kakade and Langford, 2002), for every two policies π and $\tilde{\pi}$ We can write:

$$\eta(\tilde{\pi}) - \eta(\pi) = \eta(\tilde{\pi}) - \eta(\pi_{RB}) + \eta(\pi_{RB}) - \eta(\pi) =$$

$$-[\eta(\pi_{RB}) - \eta(\tilde{\pi})] + \eta(\pi_{RB}) - \eta(\pi) =$$

$$-\sum_s \sum_{t=0}^{\infty} \gamma^t P(s_t = s|\pi_{RB}) \sum_a \pi_{RB}(a|s) A_{\tilde{\pi}}(s,a) +$$

$$\sum_s \sum_{t=0}^{\infty} \gamma^t P(s_t = s|\pi_{RB}) \sum_a \pi_{RB}(a|s) A_\pi(s,a) =$$

$$\sum_s \sum_{t=0}^{\infty} \gamma^t P(s_t = s|\pi_{RB}) \sum_a \pi_{RB}(a|s)[A_\pi(s,a) - A_{\tilde{\pi}}(s,a)]$$

Assuming we can sample tuples of (s, a, t) from our replay buffer and denote this distribution $\rho_{RB}(s, a, t)$ we can write the above equation as:

$$\eta(\tilde{\pi}) - \eta(\pi) = \sum_{s,a,t} \rho_{RB}(s,a,t)\gamma^t[A_\pi(s,a) - A_{\tilde{\pi}}(s,a)]$$

Which we can approximate by sampling such tuples from the replay buffer.

B. Experimental Details

In this section, we outline our environment, training process and hyperparameters.

Environment details. The following list contain details about all the environment used to test our algorithm and compare it to the baselines.

Tiger Door. A robot must navigate to the goal cell (green), without touching the failure cell (blue). The cells, however, randomly switch locations every episode, and their nature is not observed by the agent. The maze also includes a pink button that reveals the correct goal location. Pixel observations with discrete action space.

Lava Crossing. A minigrid environment where the agent starts in the top-left corner and needs to navigate through a maze of lava in order to get to the bottom-right corner. The episode ends in failure if the agent steps on the lava. The teacher has access to the whole map, while the student only

sees a patch of 5x5 cells in front of it. Pixel observations with discrete action space.

Memory. A minigrid environment. The agent starts in a corridor containing two objects. It then has to go to a nearby room containing a third object, similar to one of the previous two. The agent's goal is to go back and touch the object it saw in the room. The episode ends in success if the agent goes to the correct object and in failure otherwise. While the student has to go to the room to see which object is the current one, the teacher starts with that knowledge and can go to it directly. Pixel observations with discrete action space.

Light-Dark Ant. A Mujoco Ant environment with a fixed goal and a random starting position. The starting position and the goal are located at the "dark" side of the room, where the agent has access only to a noisy measurement of its current location. It has to take a detour through the "lighted" side of the room, where the noise is reduced significantly, enabling it to understand its location. On the other hand, the teacher has access to its precise location at all times, enabling it to go directly to the goal. This environment is inspired by a popular POMDP benchmark (Platt Jr et al., 2010). Proprioceptive observation with continuous action space.

Training process. Our algorithm optimizes two policies, π, and π_R, using off-policy Q-learning. The algorithm itself is orthogonal to the exact details of how to perform this optimization. For the discrete Gridworld domains (*Tiger Door*, *Memory* and *Lava Crossing*), we used DQN (Mnih et al., 2015) with soft target network updates, as proposed by (Lillicrap et al., 2015), which has shown to improve the stability of learning. For the rest of the continuous domains, we used SAC (Haarnoja et al., 2018) with the architectures of the actor and critic chosen similarly and with a fixed entropy coefficient. For both DQN and SAC, we set the soft target update parameter to 0.005. As was mentioned in the paper, we represent the Q function using to separate networks, one for estimating Q_R and another for estimating Q_E. When updating a Q function, it has to be done with respect to some policy. We found that doing so with respect to policy π yields stable performance across all environments.

For *Tiger Door*, *Memory*, and *Lava Crossing*, the teacher is a shortest-path algorithm executed over the grid map. For *Light-Dark Ant*, the teacher is a policy trained using RL over the teacher's observation space until achieving a success rate of 100%. In all of our experiments, we average performance over 5 random seeds and present the mean and 95% confidence interval.

For all proprioceptive domains, we used a similar architecture across all algorithms. The architecture includes two fully-connected (FC) layers for embedding the past observations and actions separately. These embeddings are then passed through a Long Short-Term Memory (LSTM) layer to aggregate the inputs across the temporal domain. Additionally, the current observation is embedded using an FC layer and concatenated with the output of the LSTM. The concatenated representation is then passed through another fully-connected network with two hidden layers, which outputs the action. The architecture for pixel-based observations are the same, with the observations encoded by a Convolutional Neural Network (CNN) instead of FC. The number of neurons in each layer is determined by the specific domain. The rest of the hyperparameters used for training the agents are summarized in 7.

Our implementation is based on the code released by (Ni et al., 2022).

Fair Hyperparameter Tuning. We attempt to ensure that comparisons to baselines are fair. In particular, as part of our claim that our algorithm is more robust to the choice of its hyperparameters, we took the following steps. First, we re-implemented all baselines, and while conducting experiments, maintained consistent joint hyperparameters across the various algorithms. Second, all the experiments of our own algorithm, TGRL, used the same hyperparameters. We used $\alpha = 3$, initial λ equal to 9 (and so the effective coefficient $\frac{\alpha}{1+\lambda} = 0.3$) and coefficient learning rate of 3e$-$3. Finally, for every one of the baselines we performed for each environment a search over all the algorithm-specific hyperparameters with N=8 different values for each one and report the best results (besides for COSIL, where we also report the average performance across hyperparameters).

C. Additional Results

Here we record additional results that were summarized or deferred in Section 4. In particular:

Environments without information differences. Determining if the information difference between the teacher and the student in a given environment will lead to a sub-optimal student is a complex task, as it is dependent on the specific task and the observations available to the agent, which can vary significantly across different environments. As such, it can be challenging to know beforehand if this problem exists or not. In the following experiment, we demonstrate that even in scenarios where this problem does not exist, the use of our proposed TGRL algorithm yields results that are comparable to those obtained using traditional Teacher-Student Learning (TSL) methods, which are typically considered the best approach in such scenarios. This highlights the robustness and versatility of our proposed approach.

The experiment includes three classic POMDP environments from (Ni et al., 2022). These environments are a

version of the Mujoco *Hopper*, *Walker2D*, and *HalfCheetah* environments, where the agent only have access to the joint positions but not to their velocities. The teacher, however, has access to both positions and velocities. As can be seen in Figure 8, TGRL converges a bit slower than TSL but still manage to converge to the teacher's performance.

Full training curves for Shadow Hand experiments. In Figure 10, we provide the full version of the training curves that appears in Figure 2.

	Tiger Door	Lava Crossing	Memory	Light-Dark Ant	Shadow Hand
Max ep. length	100	225	121	100	100
Collected ep. per iter.	5			10	120
RL updates per iter.	500			1000	1000
Optimizer	Adam				
Learning rate	3e-4				
Discount factor (γ)	0.9				
Batch size	32			128	128
LSTM hidden size	128			256	128
Obs. embedding	16			32	128
Actions embedding	16			32	16
Hidden layers after LSTM	[128,128]			[512,256]	[512, 256, 128]

Figure 7: Hyperparameters table.

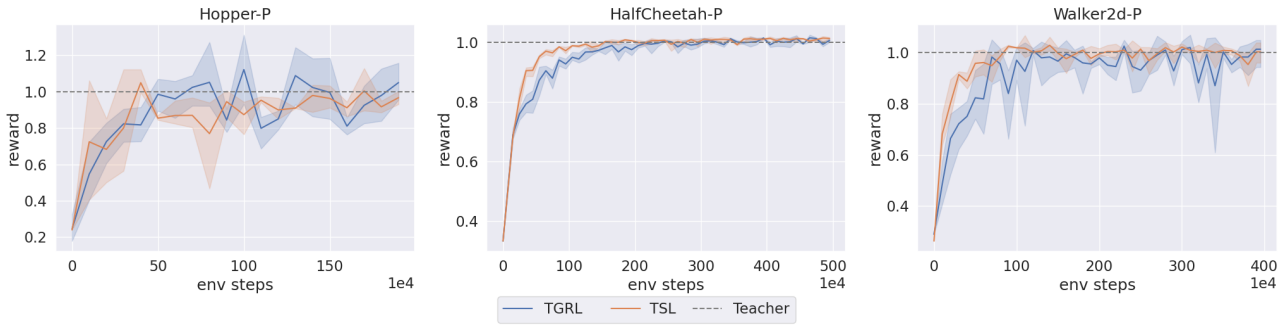

Figure 8: TGRL versus Teacher-Student Learning on domains without information difference. The rewards are normalized based on the teachers' performance.

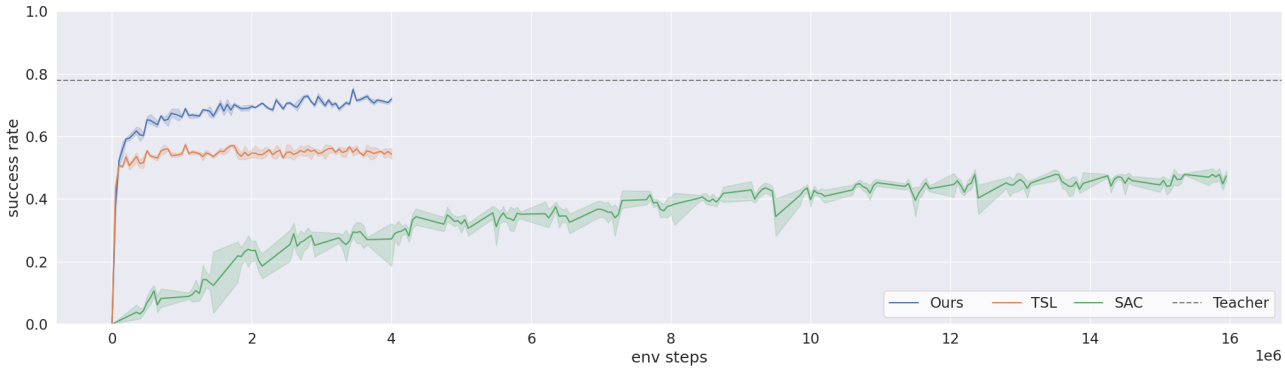

Figure 9: Full training curve of *Shadow Hand* pen reorientation with tactile sensors task

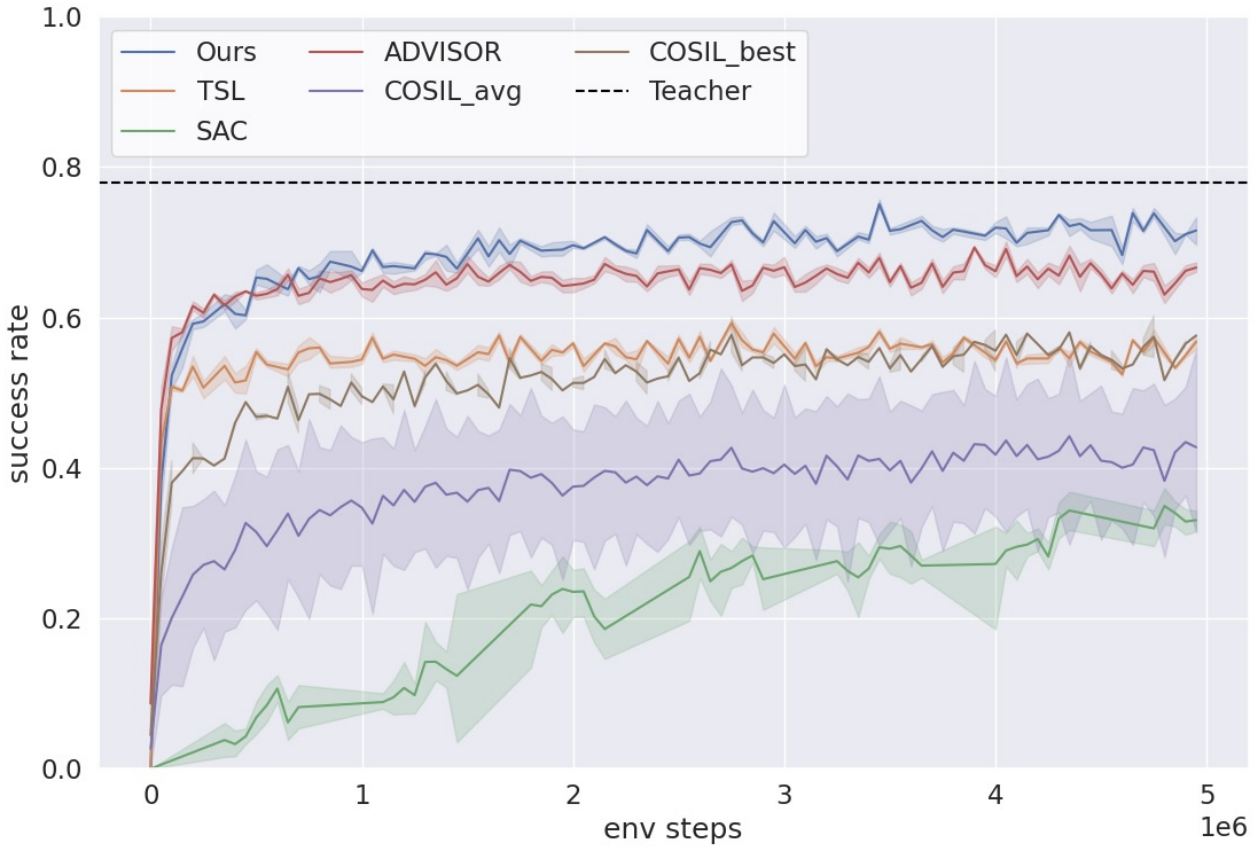

Figure 10: Comparison to baselines of *Shadow Hand* pen reorientation with tactile sensors task