Proceedings of Machine Learning Research vol 242:1489–1501, 2024 6th Annual Conference on Learning for Dynamics and Control

Adaptive Teaching in Heterogeneous Agents: Balancing Surprise in Sparse Reward Scenarios

Emma Clark* EMMAC4@ILLINOIS.EDU
University of Illinois at Urbana-Champaign, Urbana, IL, USA
Kanghyun Ryu* KANGHYUN.RYU@BERKELEY.EDU
University of California Berkeley, Berkeley, CA, USA
Negar Mehr NEGAR@BERKELEY.EDU
University of California Berkeley, Berkeley, CA, USA
**These authors contributed equally to this work.*

Editors: A. Abate, M. Cannon, K. Margellos, A. Papachristodoulou

Abstract

Learning from Demonstration (LfD) can be an efficient way to train systems with analogous agents by enabling "Student" agents to learn from the demonstrations of the most experienced "Teacher" agent, instead of training their policy in parallel. However, when there are discrepancies in agent capabilities, such as divergent actuator power or joint angle constraints, naively replicating demonstrations that are out of bounds for the Student's capability can limit efficient learning. We present a Teacher-Student learning framework specifically tailored to address the challenge of heterogeneity between the Teacher and Student agents. Our framework is based on the concept of "surprise", inspired by its application in exploration incentivization in sparse-reward environments. Surprise is repurposed to enable the Teacher to detect and adapt to differences between itself and the Student. By focusing on maximizing its surprise in response to the environment while concurrently minimizing the Student's surprise in response to the demonstrations, the Teacher agent can effectively tailor its demonstrations to the Student's specific capabilities and constraints. We validate our method by demonstrating improvements in the Student's learning in control tasks within sparse-reward environments.

Keywords: Learning from Demonstration, Surprise, Heterogeneous Agents, Teaching Agents

1. Introduction

Learning from Demonstration (LfD) enables an agent to learn new tasks by imitating another agent. Compared to traditional Reinforcement Learning (RL), LfD is particularly beneficial for learning tasks that require numerous interactions. This benefit is further amplified in multi-agent scenarios, such as assembly (Knepper et al., 2013) or warehouse systems (Dusadeerungsikul et al., 2022). In these situations where agents often share common goals, training multiple agents from scratch for the same task is data-inefficient (Da Silva et al., 2017). Hence, utilizing a Teacher-Student framework, where one *Teacher* agent explores the environment and instructs others, can be a more effective approach (Ilhan et al., 2019). For example, consider different robot manipulators on an assembly line. As they share many elements, such as task objectives and working environments, having one robot learn the task first and then instruct others can be a more efficient strategy compared to having multiple robots learn in parallel.

However, agents in these scenarios cannot always be assumed to be entirely identical (Moreira et al., 2015). Even in systems with analogous agents, small variations in dynamics can occur between agents due to several reasons, such as under-performing components or different mechanical

. The code is available at https://github.com/labicon/Surprise_based_Teaching

parts (Da Silva et al., 2020). When Student agents learn from a Teacher agent's demonstrations, these discrepancies can lead to performance issues if the Student cannot replicate the Teacher's demonstrations (Ravichandar et al., 2020). For instance, in the example of manipulators on an assembly line, variations such as maximum joint angles in different robot models may exist. Then, the Teacher agent must provide demonstrations that are achievable for the Student during the teaching, even when those differences are not explicitly stated. These differences can be inferred through the trajectory of each agent. For example, when teaching a robot arm, the Teacher agent can infer the Student robot's capabilities by observing the robot's maneuvers and accordingly offer demonstrations that align with the Student's movements.

In this work, we address this challenge by presenting a Teacher-Student learning framework that adapts the Teacher's demonstration trajectories to the heterogeneity between the Student and Teacher agents. To quantify this heterogeneity, we introduce the concept of *surprise*, which measures the informational differences between agents or environments. Consequently, the surprise should be small for state-action pairs that have been encountered before and large for those that are unfamiliar. While existing work has focused on surprise between the agent and environment (Berseth et al., 2019), we use surprise to measure the differences between the Teacher's and Student's experiences, thereby quantifying their heterogeneity without explicit knowledge of these differences.

Following Achiam and Sastry (2017), we define surprise as the KL-divergence between two transition probability functions. First, we maximize the Teacher's surprise with respect to the environment to incentivize exploration, thus aiding the Teacher in learning its task in sparse-reward environments. Second, for the Student to effectively learn from the Teacher, we argue that the Teacher should also be able to reason about the Student's learning capabilities. Specifically, *we propose that the Teacher must consider the surprise as perceived from the perspective of the Student*. To achieve this, the Teacher learns its own policy with the objective of maximizing its own surprise while minimizing that of the Student. When the Teacher's demonstrations contain state-action sequences different from the Student's dynamics or constraints, the Student's surprise will be large. Consider the example of a manipulation task, where the Student agent has a smaller maximum joint angle than the Teacher. The Teacher may demonstrate trajectories that reach its maximum joint angle; however, the Student will never be able to directly match these trajectories due to physical limitations. Since the Student will never experience such a state, it will perceive a large surprise for such demonstrations. Consequently, the Teacher must adjust its policy to accommodate trajectories within the Student's maximum joint angle.

Our contributions can be summarized in three ways. First, we introduce a Teacher-Student framework in which the Teacher learns its own task in a sparse-reward environment while simultaneously teaching a Student with differing dynamics or constraints. Second, we leverage the notion of surprise, defined as the KL-divergence between the base transition probability function and the learned transition probability function, to enable the Teacher to manage the differing dynamics or constraints of Students without explicit knowledge of these factors. Finally, we empirically demonstrate that our Teacher can adapt its demonstrations to align with the Student's capabilities, resulting in the Student achieving higher rewards.

2. Related Works

Teaching Algorithms. The transfer of knowledge or skills from one agent to another, and the successful utilization of the learned policy by the latter, can be highly beneficial. Algorithms within the family of imitation learning (Hussein et al., 2017) and behavior cloning (Ly and Akhloufi, 2021)

enable an agent to learn a policy from the demonstrations of an expert agent or a human. However, these algorithms typically rely on the expert providing an optimal trajectory and are often limited to scenarios where the teaching and learning agents have similar kinematic systems (Ravichandar et al., 2020).

On the other hand, teaching mechanisms places greater emphasis on the Teacher agent's ability to suggest beneficial actions for the Student (Zimmer et al., 2014), rather than just demonstrating optimal trajectory. Importance advising (Torrey and Taylor, 2013) enables the Teacher to calculate a metric to offer advice when beneficial and to correct mistakes. This method allows for advice to be exchanged between simultaneously learning agents even without an expert Teacher; however, its application is primarily limited to discrete action spaces (Da Silva et al., 2017).

In real-world settings, teaching strategies need to adapt to challenges such as differences in kinematics and capabilities between agents, as well as restrictions on agent actions. While some adaptive control models (Petersen et al., 2000; Nishimura et al., 2021) can handle differences in noise distributions, they fall short when it comes to differences in dynamics parameters or state constraints. Liu et al. (2019) accounts for different dynamics using state-alignment while neglecting demonstration action. However, they neglect the non-feasible problem where the demonstration trajectory have infeasible state for the learning agent. While Cao et al. (2021) suggested a feasibility metric for an imitation learning agent focusing on informative teaching data that demonstrates feasible trajectories, we concentrate on the Teacher's side to provide beneficial demonstration data to meet the needs of the Student.

Surprise in Reinforcement Learning. Surprise is a concept originating from information theory and derived from Shannon entropy (Shannon, 2001). Surprise serves as a valuable tool for stabilization or exploration in RL. Minimizing surprise drives agents toward familiar state-action pairs (Berseth et al., 2019), while maximizing surprise encourages exploration toward unfamiliar state-action pairs (Mazzaglia et al., 2022). Bayesian surprise (Itti and Baldi, 2009) has been used for exploration incentive, using latent dynamics models to maximize information gain (Mazzaglia et al., 2021; Sun et al., 2011) or using Bayesian neural networks to model dynamics and maximize surprise (Houthooft et al., 2016). However, these methods can be computationally intensive, may not generalize well to continuous action spaces, or may struggle to scale in higher dimensional environments. To overcome these challenges, Achiam and Sastry (2017) defined surprise as the KL-divergence between the learned and true transition probability functions, aiming to motivate exploration with a reduced computational burden.

We use the notion of surprise defined in Achiam and Sastry (2017) to enable the Teacher to demonstrate state-action trajectories that are admissible for the Student. Our goal is to design a teaching method that allows the Teacher to provide effective demonstrations for a Student agent, even in the presence of differing state-space constraints and dynamics.

3. Utilizing Surprise to Instruct Heterogeneous Students

We introduce a Teacher-Student framework where the Teacher and Student do not necessarily share the same dynamics or constraints. Adapting to these differences is crucial for the Student to learn effectively. Both agents are assumed to lack prior knowledge of the environment, thereby learning simultaneously. We consider sparse reward environments for both the Teacher and the Student. As the Teacher must provide expert demonstrations for the Student to follow, it is necessary that the Teacher first explores the environment to learn a policy that addresses their task. Following Achiam and Sastry (2017), we define the Teacher's surprise as the KL-divergence between the Teacher's

learned transition probability model and the true transition probability model of the Teacher's environment. Since an agent updates its learned model based on training experiences, the KL-divergence between the learned model and the true environment will be small in state-action pairs that have been frequently visited. Consequently, augmenting the Teacher's reward to be a function of surprise can encourage the exploration of unfamiliar state-action pairs.

Additionally, we propose a strategy for the Teacher that considers the surprise perceived by the Student during the learning process. We define the Student's surprise using *the KL-divergence between the transition probability functions learned by the Teacher and the Student*. If the Student cannot replicate the Teacher's transition probability functions due to differences in dynamics or constraints, this will result in a high value of surprise for the Student. Therefore, by penalizing the Teacher's reward based on the Student's *perceived* surprise, we enable the Teacher to account for differences in environments and guide the creation of trajectories that satisfy the constraints or dynamics of the Student.

3.1. Preliminaries

In our problem, we consider the Teacher and Student as separate agents and denote them by subscript T and S, respectively. We model our environments as a Markov Decision Process (MDP) and define this MDP as the tuple $< \mathcal{S}, \mathcal{A}, r_e, P, \gamma >$ where the Teacher and Student have their own respective state spaces $\mathcal{S}_T, \mathcal{S}_S$, action spaces $\mathcal{A}_T, \mathcal{A}_S$, and extrinsic reward functions $r_{e_T} : \mathcal{S}_T \times \mathcal{A}_T \rightarrow \mathbb{R}, r_{e_S} : \mathcal{S}_S \times \mathcal{A}_S \rightarrow \mathbb{R}$. Each environment has a true transition probability function $P_T(s'|s,a)$ and $P_S(s'|s,a)$, which give the true probability of transition to state s' from the current state s given action a. While these transition probability functions depend only on the Teacher's and Student's state-action pairs, respectively, we omit their subscripts for easier notation. The agents have their learned transition probability functions denoted by a subscript ϕ: P_{ϕ_T} for the Teacher and P_{ϕ_S} for the Student. We define a stochastic policy for the Teacher $\pi_T(\cdot|s) : \mathcal{S}_T \rightarrow \mathcal{A}_T$ and Student $\pi_S(\cdot|s) : \mathcal{S}_S \rightarrow \mathcal{A}_S$ as a distribution over possible actions given a state s. We assume that the Teacher has full access to the Student's learned transition probability function P_{ϕ_S}.

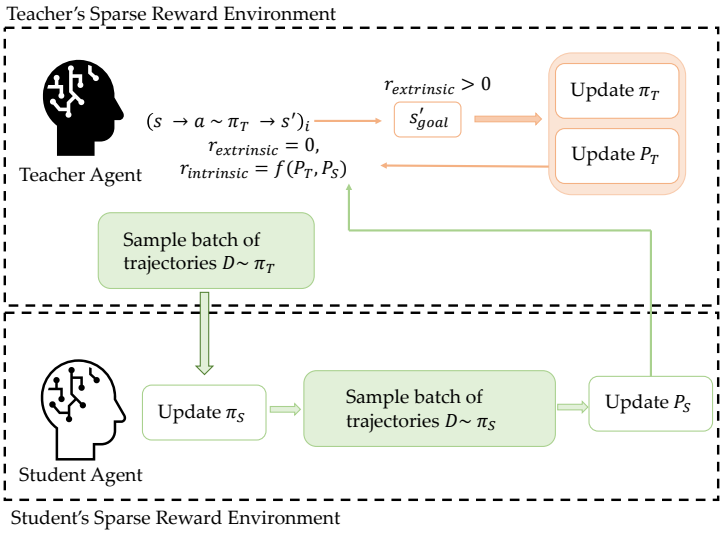


Figure 1: Overview of our Teacher-Student framework

3.2. Surprise for Exploration

In a sparse-reward setting, we motivate the Teacher to maximize its own surprise to explore the environment. We assume the Teacher does not have access to P_T. Instead, the Teacher utilizes its

own learned transition probability function, P_{ϕ_T}. As the Teacher agent explores the environment, this learned function should converge to the true transition function. Therefore, we can represent the Teacher agent's surprise by the KL-divergence between the learned transition probability function and the true transition probability function.

$$D_{KL}\big(P_T(\cdot|s,a)||P_{\phi_T}(\cdot|s,a)\big) = H\big(P_T(\cdot|s,a), P_{\phi_T}(\cdot|s,a)\big) - H\big(P_T(\cdot|s,a)\big),$$

where $H\big(P_T(\cdot|s,a)\big)$ is the entropy of $P_T(\cdot|s,a)$ and $H\big(P_T(\cdot|s,a), P_{\phi_T}(\cdot|s,a)\big)$ is the cross entropy of $P_T(\cdot|s,a)$ and $P_{\phi_T}(\cdot|s,a)$.

In deterministic environments with continuous state spaces, $\mathbb{E}_{s,a\sim\pi_T}[H\big(P_T(\cdot|s,a)\big)]$ is constant and can be dropped from the optimization problem (Achiam and Sastry, 2017). Therefore, the KL-divergence can be approximated as the cross entropy between the Teacher's learned transition probability function and the environment's true transition probability function. We compute the cross entropy between the two distributions, $P_T(\cdot|s,a)$ and $P_{\phi_T}(\cdot|s,a)$, as

$$\begin{aligned} H(P_T(\cdot|s,a), P_{\phi_T}(\cdot|s,a)) &= -\int_{\mathcal{S}} P_T(s'|s,a)\log\big(P_{\phi_T}(s'|s,a)\big)ds' \\ &= \mathbb{E}_{s'\sim P_T(\cdot|s,a)}\big[-\log P_{\phi_T}(s'|s,a)\big]. \end{aligned} \tag{1}$$

3.3. Computing the Surprise Perceived by the Student

In this section, we discuss how the Teacher can tailor trajectories to accommodate a Student with different dynamics or environmental constraints. We define the Student's surprise as a KL-divergence between the Student's learned transition probability function and that of the Teacher. Therefore, the Student surprise for state-action pair (s,a) is

$$D_{KL}\big(P_{\phi_T}(\cdot|s,a)||P_{\phi_S}(\cdot|s,a)\big).$$

Because the Teacher has access to both the Student's and Teacher's learned transition probability functions, we can directly calculate the KL-divergence between the two:

$$\begin{aligned} D_{KL}\big(P_{\phi_T}(\cdot|s,a)||P_{\phi_S}(\cdot|s,a)\big) &= \int_{\mathcal{S}} P_{\phi_T}(s'|s,a)\log\left(\frac{P_{\phi_T}(s'|s,a)}{P_{\phi_S}(s'|s,a)}\right)ds' \\ &= \mathbb{E}_{s'\sim P_{\phi_T}(\cdot|s,a)}\big[\log P_{\phi_T}(s'|s,a) - \log P_{\phi_S}(s'|s,a)\big]. \end{aligned} \tag{2}$$

When the Student and Teacher exhibit similar learned transition dynamics for specific state-action pairs, the KL divergence in (2) will be small, resulting in a lower surprise value for the Student. Conversely, if there are significant differences in their learned transition dynamics, stemming from different environment constraints or dynamics, the surprise value perceived by the Student in (2) will be large. In conclusion, a large surprise is likely to occur if the Teacher demonstrates a trajectory that is incompatible with the Student's dynamics or constraints. Therefore, our objective is to minimize this surprise during the Teacher's demonstrations.

3.4. Shaping the Teacher's reward for Adaptive Demonstrations

We define the intrinsic reward r_i for the Teacher as a weighted sum of the surprise terms for both the Teacher and the Student. The inclusion of the Teacher's surprise encourages exploration in a

sparse-reward environment, facilitating the exploration of novel state-action pairs. Simultaneously, incorporating the Student's surprise enables the Teacher to make informed inferences about the Student's dynamics and constraints. Consequently, this dual consideration allows the Teacher to tailor its demonstrations to align with the Student's capabilities. Each surprise incentive can be computed using (1) and (2) as follows

$$\begin{aligned} r_i(s,a) &= \eta_T D_{KL}\big(P(\cdot|s,a)||P_{\phi_T}(\cdot|s,a)\big) - \eta_S D_{KL}\big(P_{\phi_T}(\cdot|s,a)||P_{\phi_S}(\cdot|s,a)\big) \\ &\approx \eta_T \mathbb{E}_{s'\sim P(\cdot|s,a)}\big[-\log P_{\phi_T}(s'|s,a)\big] - \eta_S \mathbb{E}_{s'\sim P\phi_T(\cdot|s,a)}\big[\log P_{\phi_T}(s'|s,a) - \log P_{\phi_S}(s'|s,a)\big], \end{aligned} \tag{3}$$

where η_T and η_S are the weights of each surprise term. Following Achiam and Sastry (2017), η_T is given by

$$\eta_T = \frac{\eta_{0_T}}{\max\big(1, \frac{1}{|\mathcal{D}|}\sum_{(s,a)\in\mathcal{D}} r_{e_T}(s,a)\big)}, \tag{4}$$

where η_{0_T} is a predefined constant, r_{e_T} represents the extrinsic return from a trajectory rollout $\mathcal{D}$ of size $|\mathcal{D}|$ following the Teacher's policy. This factor scales the exploration bonus magnitude according to the extrinsic rewards of the environment. The Student's coefficient η_S is defined analogously, with extrinsic rewards calculated from trajectories sampled from the Student's policy, and the Student's predefined η_{0_S} constant.

The reshaped reward for the Teacher is designed to incentivize exploration of new states while simultaneously disincentivizing visits to states that are excessively unfamiliar to the Student. If the Teacher demonstrates trajectories that either violate the Student's constraints or differ significantly from the Student's dynamics, the Student's surprise will be high. This is due to the Student's learned transition model differing substantially from these trajectories. As a result, in an effort to maximize its reward as outlined in (3), the Teacher is encouraged to avoid such trajectories and instead demonstrate paths that are aligned with the dynamics and constraints of both the Teacher and the Student.

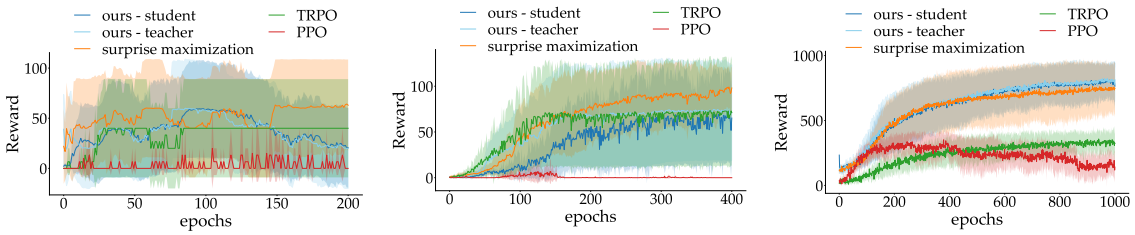
Finally, the Teacher's objective is to maximize the augmented sum of extrinsic and intrinsic rewards. Therefore, the objective function for the Teacher is as follows

$$L_T(\pi_T) = \mathbb{E}_{a_t\sim\pi_T(\cdot|s_t)}\big[\sum_{t=0}^{H}\gamma^t\big(r_{e_T}(s_t,a_t) + r_i(s_t,a_t)\big)\big]. \tag{5}$$

3.5. Implementation Details

We can use any RL algorithm to optimize the Teacher policy that aims to maximize (5). In our experiments, we implemented Trust Region Policy Optimization (TRPO) (Schulman et al., 2015) to train the Teacher policy, enabling the examination of both continuous and discrete action spaces. This choice also allows for a direct comparison with Achiam and Sastry (2017), which used TRPO as their base optimization method, thus providing a more conclusive analysis of performance differences when incorporating the Student's surprise. For the Student agent, we implement behavioral cloning (BC) (Bain and Sammut, 1995; Ross and Bagnell, 2010) to learn from the Teacher's demonstration.

We use probabilistic neural networks (Chua et al., 2018) to learn the transition probability functions of the Teacher and Student agents. We model the transition probability functions of the Teacher agent as a Gaussian distribution $P_{\phi_T}(\cdot|s,a) = \mathcal{N}\big(\mu_{\phi_T}(s,a), \Sigma_{\phi_T}(s,a)\big)$, where μ_{ϕ_T} is the learned



(*a*) Mountain Car (*b*) Cart Pole Swing Up (*c*) Half Cheetah

Figure 2: Mean and standard deviation of reward for the three environments are shown. Results are from 5 random seeds for Mountain Car and Half Cheetah and 8 random seeds for Cart Pole Swing Up. Our algorithm is deployed in a setting where the Teacher and Student are in the same environment. Baselines are trained in a single-agent setting where they are trained without the Teacher. Both Teacher and Student in our Teacher-Student framework can learn successful policy in sparse-reward environments.

mean and Σ_{ϕ_T} is the learned covariance of the distribution. We update the mean and covariance of the distributions by optimizing the negative log-likelihood loss function

$$\text{loss}_{NLL} = \sum_{i=0}^{N} (\mu_{\phi_T}(s_i, a_i) - s_{i+1}^T)\Sigma_{\phi_T}^{-1}(s_i, a_i)(\mu_{\phi_T}(s_i, a_i) - s_{i+1}) + \log\left(\det \Sigma_{\phi_T}(s_i, a_i)\right), \quad (6)$$

where N is the batch size and s_{i+1} is the sampled true next state of the environment given state and action pairs (s_i, a_i). The same procedures are applied to the Student's learned transition probability function $P_{\phi_S}(\cdot|s, a)$.

At each training epoch, we first update the Teacher's transition probability model with (6) and the Teacher's policy to maximize (5). Then, the Teacher gives demonstrations to the Students using the Teacher's policy. We update the Student policy with BC for the Teacher demonstration. Finally, the Student's transition probability model is updated with the trajectory rollout of its policy.

4. Experiments

We implement our algorithm in sparse reward environments introduced by Houthooft et al. (2016): Mountain Car, Cart Pole Swing Up, and sparse Half Cheetah. In Section 4.1, we initially test our method with both the Student and Teacher operating in the same environment to ensure that our approach doesn't obstruct the learning capabilities of either agent in homogeneous settings.. Next, in Section 4.2, we show that our method helps the learning of the Student agent which has a different environment from the Teacher.

4.1. Identical Teacher-Student Environments

Figure 2 presents the results of our Teacher-Student framework when the Teacher and Student have identical environments. Compared to our Teacher-Student framework, TRPO, PPO (Schulman et al., 2017), and surprise-maximization (Achiam and Sastry, 2017) algorithms tries to learn a policy without the Teacher. Both Teacher and Student in our method performs at a similar level to that of Achiam and Sastry (2017). Therefore, we can conclude that the Student-Teacher framework does not diminish the learning capabilities of the agents. In addition, observing the low returns of TRPO and PPO, we conclude that surprise motivation is necessary for exploration in sparse reward environments.

4.2. Heterogeneity Between Teacher and Student

4.2.1. EXPERIMENT SETUP

After confirming that our framework does not impact learning in the homogeneous setting, we test our learning framework in the heterogeneous setting where the Teacher and Student are in environments with different dynamics or constraints where we can examine the ability of the Teacher agent to adjust its trajectories according to the Student's environment. Especially, we focus on settings where the Student's ability is limited compared to the Teacher. This is because even if the Student has superior ability compared to the Teacher, its effectiveness would be limited to the Teacher's policy due to imitating the Teacher's policy.

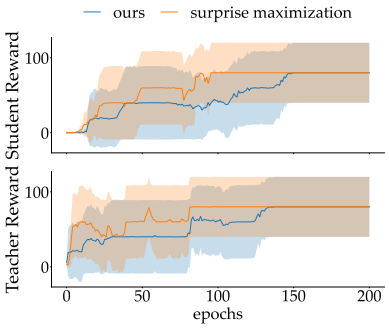
In the Mountain Car environment, we adjust the car power constant in the dynamics equation. We keep the baseline value of $p = 0.001$ for the Teacher and reduce that of the Student to $p = 0.0067$. In this lower-power version, an agent requires a greater force to achieve the desired velocity compared to the higher-power setup. Consequently, the Teacher should demonstrate trajectories utilizing larger force magnitudes than those used in its high-power environment. For the Cart Pole Swing Up environment, we have two experiments with different x-position constraints and different pole masses. In the different x-position constraints experiment, we have the Teacher's constraint to be $|x_{pos}| \leq 3.6$ while the Student is using $|x_{pos}| \leq 2.4$. For the different pole mass experiments, the Teacher agent has a pole mass of $m = 0.1$ while the Student has a heavier pole mass of $m = 0.12$. We note that the maximum episode length in the CartPole was increased to 500 for faster exploration. Finally, in the sparse Half Cheetah environment, the Teacher's head angle is unconstrained while the Student's is constrained to $|\theta_{head}| \leq 1$rad.

In Figure 2, we observed that baseline algorithms that lack an exploration motivation failed to learn successful policies in complex sparse-reward environments. Consequently, we used surprise maximization (Achiam and Sastry, 2017) as the sole baseline. In this baseline, the Teacher attempts to learn an optimal policy by exploring the environment through surprise-maximization while neglecting the Student's perceived surprise. This comparison highlights the efficacy of incorporating the Student's surprise into the Teacher's reward structure. Meanwhile, our method enables the Teacher to tailor demonstrations more effectively to the Student when their environments are dissimilar. In both scenarios, the Student performs BC on the Teacher's policy.

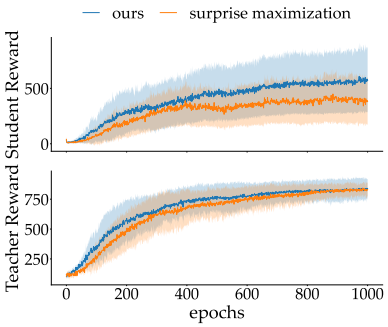
4.2.2. RESULTS

Figure 3 presents the training results of each experiment. In contrast to Figure 2, where the Student's performance closely mirrors that of the Teacher, Figure 3 exhibits a notable gap between the returns of the Teacher and the Student due to differences in their environments. Furthermore, the Student's average return is higher in our method compared to the baseline, with the exception of the Mountain Car environment. Despite minimal differences in the Teacher's performance across the two methods, there is a noticeable disparity in the Student's rewards. This observation challenges the expectation that the Student's behavior cloning performance should closely mirror that of the Teacher, indicating that our method produces trajectories more conducive to the Student's learning while maintaining effective learning of the Teacher. Therefore, we can conclude that the Teacher's objective of minimizing the Student's surprise can improve the Student's performance, even in the absence of explicit knowledge about these discrepancies.

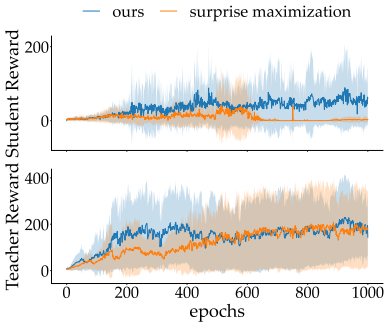
In Figure 4, we further analyze the Teacher's behavior by plotting its demonstration trajectories throughout training in the Mountain Car environment. Initially, the trajectories of each algorithm



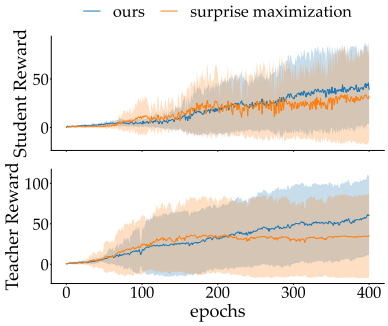
(*a*) Mountain Car - Different Power



(*b*) Half Cheetah - Constrained Tip Angle



(*c*) CartPole SwingUp - Different x_{pos} Constraint



(*d*) CartPole SwingUp - Different Pole Mass

Figure 3: Teacher and Student training results where each agent has different constraints or dynamics. While the average reward of the Teacher is similar for both methods, the Student learning from our Teacher achieves higher average rewards. These show that our method can provide better demonstrations for the Student with different constraints/dynamics.

appear similar. However, as training progresses, significant differences emerge in the actions taken at specific points in the state space. By the end of training, it is evident that the Teacher using our method has adapted to demonstrate larger forces. This adaptation is particularly beneficial for the Student agent's learning process, as it aligns better with the Student's need to apply greater force due to its lower power configuration.

We further investigate the impact of the Student surprise term by varying its weight in the sparse Half Cheetah environment. Figure 5 reveals that a higher emphasis on the Student surprise results in an increased performance gap between Student agents learning from the two different Teachers. This suggests that prioritizing the Student's surprise in the Teacher's objective encourages the Teacher to develop a policy that is more closely aligned with the Student's capabilities, which may differ from those of the Teacher.

5. Conclusions and Future works

We proposed a Teacher-Student learning framework, where the Teacher adapts its policy to demonstrate pedagogically effective trajectories to a Student agent acting under different constraints or dynamics parameters. This is achieved by minimizing Student surprise with respect to the Teacher's demonstration, while simultaneously maximizing the Teacher's surprise to encourage exploration.

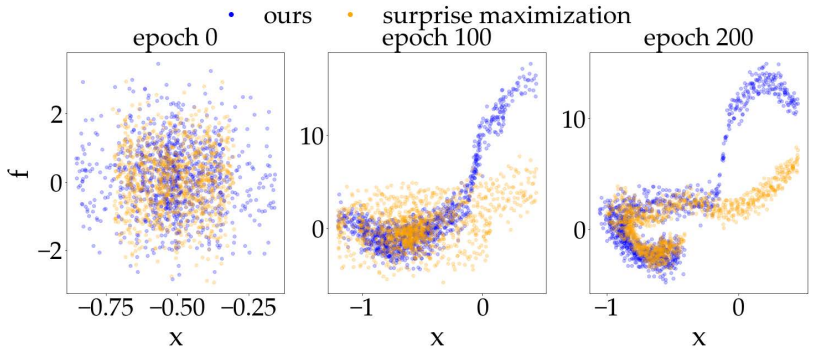
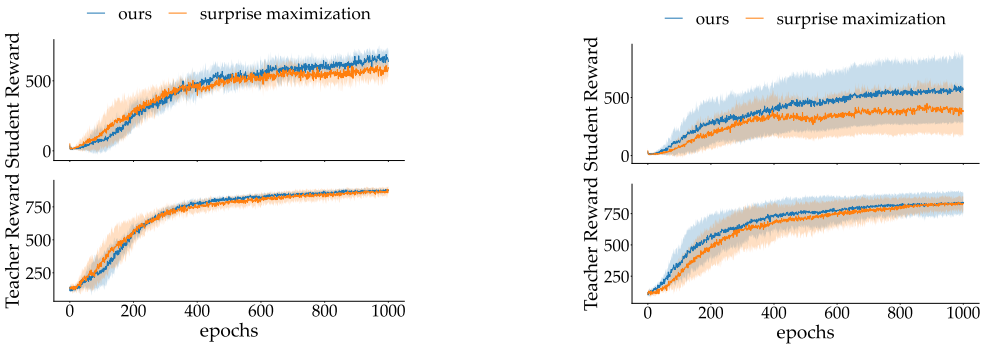


Figure 4: Teacher demonstration for Mountain Car environment where the Student has less power available than the Teacher. In training epoch 0, both methods appear to be similarly random. In the 100th epoch, our method begins to exhibit larger forces corresponding to the f-axis on the figures. At the end of training, we see there is a clear distinction between the forces exhibited by the two methods. Our method adapts to the low-power dynamics of the Student environment by demonstrating much larger forces compared to the surprise maximization algorithm.



(*a*) $\eta_{0_T} = 0.001, \eta_{0_S} = 0.001$ (*b*) $\eta_{0_T} = 0.001, \eta_{0_S} = 0.005$

Figure 5: Training results in a sparse Half Cheetah environment with varying weights on Student surprise. The performance gap of the Student widens with an increased weight on Student surprise. This suggests that placing greater emphasis on Student surprise leads the Teacher to provide demonstrations that are more easily followed by the Student.

We implemented this algorithm in sparse reward environments and demonstrated that a behavior cloning agent learns more effectively from a Teacher trained with our method. We showed that our method can adapt the teaching trajectories such that the Student learns more efficiently by examining the demonstrations of the Teacher agent in the Mountain Car environment. Moreover, our method achieved comparable performance to baseline algorithms when both the Teacher and the Student were learning within the same environment settings.

As future works, we plan to extend our algorithm to the offline-online settings where transition dynamics of agents or Teacher policy are pre-trained offline. Another interesting future work could allow the Teacher to learn latent strategies to predict the transition dynamics of the Student rather than having full access to the Student's learned transition probability functions.

Acknowledgments

This work is supported by the National Science Foundation, under grants CNS-2423130 and CCF-2423131. The authors would like to thank Dr. Jean-Baptiste Bouvier for his valuable feedback.

References

Joshua Achiam and Shankar Sastry. Surprise-based intrinsic motivation for deep reinforcement learning. *arXiv preprint arXiv:1703.01732*, 2017.

Michael Bain and Claude Sammut. A framework for behavioural cloning. In *Machine Intelligence 15*, pages 103–129, 1995.

Glen Berseth, Daniel Geng, Coline Devin, Nicholas Rhinehart, Chelsea Finn, Dinesh Jayaraman, and Sergey Levine. SMiRL: Surprise minimizing reinforcement learning in unstable environments. *arXiv preprint arXiv:1912.05510*, 2019.

Zhangjie Cao, Yilun Hao, Mengxi Li, and Dorsa Sadigh. Learning feasibility to imitate demonstrators with different dynamics. *arXiv preprint arXiv:2110.15142*, 2021.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *Advances in Neural Information Processing Systems*, 31, 2018.

Felipe Leno Da Silva, Ruben Glatt, and Anna Helena Reali Costa. Simultaneously learning and advising in multiagent reinforcement learning. In *Proceedings of the 16th Conference on Autonomous Agents and Multiagent Systems*, pages 1100–1108, 2017.

Felipe Leno Da Silva, Garrett Warnell, Anna Helena Reali Costa, and Peter Stone. Agents teaching agents: A survey on inter-agent transfer learning. *Autonomous Agents and Multi-Agent Systems*, 34:1–17, 2020.

Puwadol Oak Dusadeerungsikul, Xiang He, Maitreya Sreeram, and Shimon Y Nof. Multi-agent system optimisation in factories of the future: cyber collaborative warehouse study. *International Journal of Production Research*, 60(20):6072–6086, 2022.

Rein Houthooft, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. Vime: Variational information maximizing exploration. *Advances in Neural Information Processing Systems*, 29, 2016.

Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. Imitation learning: A survey of learning methods. *ACM Computing Surveys (CSUR)*, 50(2):1–35, 2017.

Ercüment Ilhan, Jeremy Gow, and Diego Perez-Liebana. Teaching on a budget in multi-agent deep reinforcement learning. In *2019 IEEE Conference on Games (CoG)*, pages 1–8. IEEE, 2019.

Laurent Itti and Pierre Baldi. Bayesian surprise attracts human attention. *Vision Research*, 49(10): 1295–1306, 2009.

Ross A Knepper, Todd Layton, John Romanishin, and Daniela Rus. Ikeabot: An autonomous multi-robot coordinated furniture assembly system. In *2013 IEEE International Conference on Robotics and Automation*, pages 855–862. IEEE, 2013.

Fangchen Liu, Zhan Ling, Tongzhou Mu, and Hao Su. State alignment-based imitation learning. *arXiv preprint arXiv:1911.10947*, 2019.

Abdoulaye O. Ly and Moulay Akhloufi. Learning to drive by imitation: An overview of deep behavior cloning methods. *IEEE Transactions on Intelligent Vehicles*, 6(2):195–209, 2021. doi: 10.1109/TIV.2020.3002505.

Pietro Mazzaglia, Ozan Catal, Tim Verbelen, and Bart Dhoedt. Self-supervised exploration via latent bayesian surprise. In *ICLR2021, the 9th International Conference on Learning Representations*, 2021.

Pietro Mazzaglia, Ozan Catal, Tim Verbelen, and Bart Dhoedt. Curiosity-driven exploration via latent Bayesian surprise. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 7752–7760, 2022.

Mayron César O Moreira, Jean-François Cordeau, Alysson M Costa, and Gilbert Laporte. Robust assembly line balancing with heterogeneous workers. *Computers & Industrial Engineering*, 88: 254–263, 2015.

Haruki Nishimura, Negar Mehr, Adrien Gaidon, and Mac Schwager. Rat ilqr: A risk auto-tuning controller to optimally account for stochastic model mismatch. *IEEE Robotics and Automation Letters*, 6(2):763–770, 2021.

Ian R Petersen, Matthew R James, and Paul Dupuis. Minimax optimal control of stochastic uncertain systems with relative entropy constraints. *IEEE Transactions on Automatic Control*, 45(3):398–412, 2000.

Harish Ravichandar, Athanasios S Polydoros, Sonia Chernova, and Aude Billard. Recent advances in robot learning from demonstration. *Annual review of Control, Robotics, and Autonomous Systems*, 3:297–330, 2020.

Stéphane Ross and Drew Bagnell. Efficient reductions for imitation learning. In *Proceedings of the thirteenth International Conference on Artificial Intelligence and Statistics*, pages 661–668. JMLR Workshop and Conference Proceedings, 2010.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International Conference on Machine Learning*, pages 1889–1897. PMLR, 2015.

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

Claude Elwood Shannon. A mathematical theory of communication. *ACM SIGMOBILE mobile computing and communications review*, 5(1):3–55, 2001.

Yi Sun, Faustino Gomez, and Jürgen Schmidhuber. Planning to be surprised: Optimal Bayesian exploration in dynamic environments. In *Artificial General Intelligence: 4th International Conference, AGI 2011, Mountain View, CA, USA, August 3-6, 2011. Proceedings 4*, pages 41–51. Springer, 2011.

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.

Matthieu Zimmer, Paolo Viappiani, and Paul Weng. Teacher-student framework: A reinforcement learning approach. In *AAMAS Workshop Autonomous Robots and Multirobot Systems*, 2014.