Student-Informed Teacher Training

Anonymous Author(s)
Affiliation
Address
email

Abstract: Imitation learning with a privileged teacher has proven effective for learning complex control behaviors from high-dimensional inputs, such as images. In this framework, a teacher is trained with privileged task information, while a student tries to predict the actions of the teacher with more limited observations, e.g., in a robot navigation task, the teacher might have access to distances to nearby obstacles, while the student only receives visual observations of the scene. However, privileged imitation learning faces a key challenge: the student might be unable to imitate the teacher's behavior due to partial observability. This problem arises because the teacher is trained without considering if the student is capable of imitating the learned behavior. To address this teacher-student asymmetry, we propose a framework for joint training of the teacher and student policies, encouraging the teacher to learn behaviors that can be imitated by the student despite the latters' limited access to information and its partial observability.

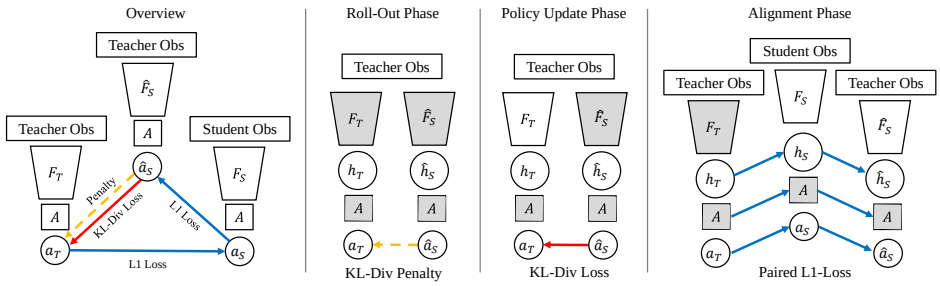
1 Introduction

State-of-the-art (SotA) policy learning approaches often rely on imitation learning to accelerate training [1, 2, 3, 4]. However, collecting expert demonstrations for imitation learning can be prohibitively expensive, which has led to the development of the teacher-student framework. In this framework, expert data is generated automatically by training a teacher policy using RL on privileged task information, benefiting from efficient simulation and a faster learning process. This approach eliminates the need for the student to extensively explore the environment, which can be a very challenging process when dealing with high-dimensional observations, such as images. However, privileged imitation learning can be hindered by information asymmetry between the teacher and student, where the student receives less informative observations and struggles to imitate the behavior of the teacher [5]. As a consequence of the information asymmetry, the teacher tends to over-rely on its full observability of the environment without considering the more limited observation space of the student. This causes the teacher to provide target actions that the student cannot infer from its observations, since the student lacks access to the same level of environmental information. To tackle these challenges, we propose a teacher-student knowledge distillation framework that encourages the teacher to learn behaviors that account for the capabilities of the student. Specifically, the objective function of the teacher is extended by adding the upper bound of the student performance within the imitation learning setting. This results in a reward term that penalizes the teacher for visiting states where there is a significant action mismatch between the student and teacher. Additionally, minimizing this upper bound leads to a second optimization term that directly supervises the weights of the teacher network.

2 Student-Informed Teacher Training

As shown in [6, 7, 8], the performance gap between student and teacher is upper-bounded by the action difference between both policies. Thus, minimizing the action difference under the state distribution of the expert also minimizes the performance gap $J(\pi_T) - J(\pi)$. Instead of trying to minimize the action difference by adjusting the student policy π_S, we propose to change the perspective and find a teacher policy π_T optimizing for the task reward while considering the alignment

Submitted to the 8th Conference on Robot Learning (CoRL 2024). Do not distribute.



(a) (b) (c) (d)

Figure 1: **Method Overview.** Our method leverages three networks (a), which are trained in three alternating phases: the roll-out phase (b), the policy update phase (c), and the alignment phase (d). The grey boxes represent networks frozen during the specific phase, i.e., the network parameters are not updated. (b) In the roll-out phase, the KL-Divergence between the proxy student $\hat{F}_S$ and teacher F_T is used as a penalty term. (c) In addition to the policy gradient, the teacher encoder is updated by backpropagating through the KL-Divergence between the action distribution of the teacher and the proxy student. (d) Using student observations, the proxy student is aligned to the student F_S while the student is aligned to the teacher network using consistency losses.

between teacher and student. Thus, we want to find a teacher policy π_T that maximizes

$$\tilde{J}(\pi_T) = \mathbb{E}_{s\sim d_{\pi_T}, a\sim\pi_T(\cdot|s)}[r(s,a)] - \mathbb{E}_{s\sim d_{\pi_T}}[D_{KL}(\pi_T(\cdot|s), \pi_S(\cdot|s))] \tag{1}$$

$$= \mathbb{E}_{s\sim d_{\pi_T}, a\sim\pi_T(\cdot|s)}[r(s,a) - D_{KL}(\pi_T(\cdot|s), \pi_S(\cdot|s))] \tag{2}$$

$$\propto \mathbb{E}_{\tau\sim p_\theta}[R(\tau) - D_\theta(\tau)] = \int p_\theta(\tau)(R(\tau) - D_\theta(\tau))d\tau. \tag{3}$$

In the last step, the discounted state distribution is changed to the expectation over trajectories $\tau \sim p_\theta$, which are induced by the expert policy π_T and represent the state and corresponding actions $\tau = \{s_0, a_0, s_1, a_1, ...\}$. Additionally, we define the return $R(\tau) = \sum_{s_t,a_t\in\tau} \gamma^t r(s_t, a_t)$ and the sum of discounted KL-Divergences $D_\theta(\tau) = \sum_{s_t\in\tau} \gamma^t D_{KL}(\pi_T(\cdot|s_t), \pi_S(\cdot|s_t))$. We use subscript θ, to emphasize that the probability distribution over the trajectories p_θ and $D_\theta(\tau)$ is dependent on the parameter of the teacher network θ. Following the classical policy gradient to obtain the optimal policy, we take the gradient of Eq. 3 with respect to the teacher parameters θ

$$\nabla_\theta \tilde{J}(\pi_T) = \nabla_\theta \int p_\theta(\tau)(R(\tau) - D_\theta(\tau))d\tau \tag{4}$$

$$= \int \nabla_\theta p_\theta(\tau) R(\tau) d\tau - \int \nabla_\theta p_\theta(\tau) D_\theta(\tau) d\tau - \int p_\theta(\tau) \nabla_\theta D_\theta(\tau) d\tau \tag{5}$$

$$= \underbrace{\int \nabla_\theta p_\theta(\tau)(R(\tau) - D_\theta(\tau))d\tau}_{\text{Policy Gradient}} - \underbrace{\int p_\theta(\tau)\nabla_\theta D_\theta(\tau)d\tau}_{\text{KL-Div Gradient}}. \tag{6}$$

As can be observed, we end up with the standard policy gradient optimizing the task reward while also considering the teacher-student misalignment for each trajectory. This weighted KL-Divergence D_θ can be interpreted as a reward encouraging the teacher policy to visit states where the student and teacher are aligned and avoid states with a large misalignment. The second term contains the expectation of the gradient with respect to the teacher network over the expert states, which represents a direct supervision on the teacher weights by enforcing the prediction of the same action distribution as the student.

3 Joint Learning Framework

Building on the formulation in Sec. 2, we propose a practical framework to tackle the teacher-student asymmetry. Following Eq. 6, we adapt the widely-used PPO algorithm [9] to train the teacher to learn behaviors that can be imitated by the student. At the same time, we train the student network to imitate the teacher based on a subset of the collected environment interactions containing

student and teacher observations. By using a subset of paired teacher and student data, we avoid the (usually expensive) simulation of student observation for each time step the teacher interacts with the environment. An overview of the proposed method is shown in Figure 1 a).

To implement the objective in Eq. 3 inside the teacher training, we implement two key components: (i) a proxy student network taking as input teacher observations and (ii) a shared action decoder network. Excluding the critic, our method consists of three different networks: the teacher network F_T, the student network F_S, and the proxy student network $\hat{F}_S$, which all share the same action decoder network A.

Proxy Student Network To compute the action difference used in the penalty term and to obtain the KL-Div Gradient in Eq. 6, a forward pass through both the teacher and student networks is required for each collected sample. However, simulating high-dimensional student observations, such as images, is often computationally expensive, contradicting the initial goal of accelerating training. To avoid this simulation overhead, we introduce a separate neural network $\hat{F}_S$ that imitates the current student policy based on the teacher observations. This allows us to approximate the actions of the student at each expert state without additional simulation cost. The proxy student network is trained during the alignment phase, where both student and teacher observations are available for a subset of environment interactions. Our proposed framework consists of three alternating training phases: (i) the classical policy roll-out, (ii) the policy update, and (iii) the alignment phase, see also Figure 1 (b)-(c). The first two phases follow the standard on-policy training, while in the alignment phase (iii), the student F_S is aligned to the teacher F_T, and the proxy student $\hat{F}_S$ is aligned to the student F_S. For both network alignments, paired student and teacher observations are used. In the following, we provide more details about the specific training phases.

Roll-out Phase Our proposed framework introduces a minor modification to the roll-out phase of the standard teacher training, specifically in the reward computation. In addition to the task reward, we also add a penalty term computed based on the action difference between the teacher and the proxy student. This penalty encourages the teacher to only visit states in which the student can predict the same actions as the teacher, thereby improving alignment between the student and teacher. Furthermore, during each roll-out phase, we store a subset of expert states required in the alignment phase. Depending on the simulation envrionment, this subset can be randomly selected states or a fixed number of environments from which student observations are generated.

Policy Update Phase The gradient of the KL-Div term in Eq. 6 can be integrated into the policy update of the teacher, during which the network weights are updated using the clipped policy gradient of PPO. Since both the policy gradient and the KL-Divergence gradient are computed over the state distribution of the teacher, we can use the teacher states inside the roll-out buffer to compute the KL-Divergence between the action distributions of the teacher and the proxy student. This allows us to update the network parameters of the teacher in a single backward pass through the combined loss function.

Alignment Phase The alignment phase focuses on aligning the features across the encoders of the teacher, proxy student, and student. This phase is the only one that requires paired teacher and student observations, which are simulated from a subset of the teacher's experiences during the roll-out phase. We align the student encoder with the teacher encoder by computing the L1 loss between their corresponding features and between the activations of the frozen shared action decoder. To prevent the collapse of the model into predicting constant outputs, gradients are only backpropagated to the student encoder. Similarly, the proxy student is aligned with the student using the L1 loss on the encoded features, with gradients only backpropagated to the proxy student. The parameters of the teacher remain unaffected during this phase and are only updated during the policy update phase.

4 Experiments

We evaluate our student-informed teacher training framework on the task of vision-based drawer opening using a robot arm. We compare our method to multiple behavior cloning (BC) and DAgger and ablate the benefits of the alignment penalty and loss terms in all three tasks.

Setup We adapt the publicly available *Omniverse Isaac Gym Reinforcement Learning Environments*

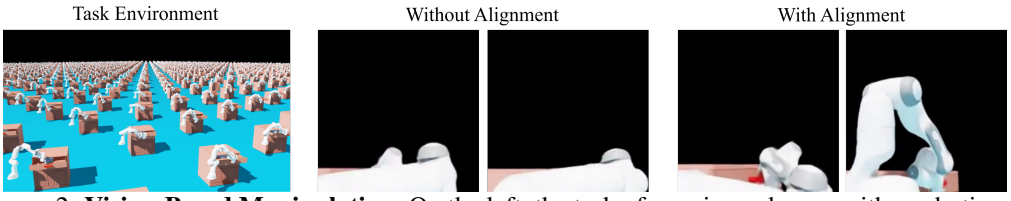


Figure 2: **Vision-Based Manipulation.** On the left, the task of opening a drawer with a robotic arm is visualized for all of the parallel environments. The two images in the center (Without Alignment) are sample images given to the student, which show the teacher behaviors trained without our alignment. Our approach with alignment leads to behaviors that the student can imitate more easily, i.e., the robot does not block the red drawer handle, as visualized in the two images on the right.

Methods	Success Rate
BC	0.27 ± 0.02
DAgger	0.31 ± 0.24
w/o Align (Ours)	0.60 ± 0.04
w Align (Ours)	0.77 ± 0.12

Table 1: **Manipulation Success Rates.** The mean and standard deviation of the success rate for the task of opening the drawer obtained from three different trainings runs.

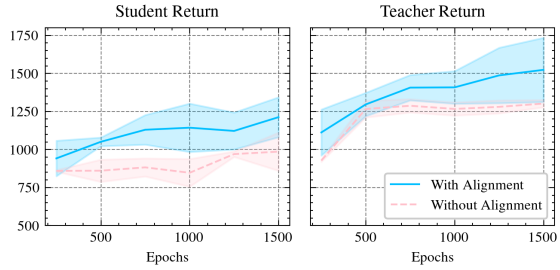


Figure 3: **Manipulation Returns.** The mean returns achieved by the student and teacher trained with and without alignment, averaged over three training runs.

for Isaac Sim repository, specifically modifying the cabinet-opening task. In this task, a Franka robot arm is trained to open a drawer that contains objects inside. The teacher receives the default observations, which include the robot's joint positions and velocities and the relative position between the gripper and the drawer handle, providing privileged information for effective manipulation. The student observations do not include relative distance information, and instead, an image is given together with the state of the robot arm as observation. Instead of an unobstructed top view, the viewpoint of the camera is selected closer to the robot arm, which enables self-occlusion. This camera setup is closer to real-world applications on mobile robots, such as humanoid robots, where certain arm configurations may obstruct cameras. We use a frozen DINOv2 [10] encoder to extract flattened image features, which are passed through a five-layer MLP before being fed into the shared action decoder, which comprises one layer. For the teacher and proxy student, we use a three-layer MLP to process the 1D observations.

Table 1 reports the success rates for our method (with and without alignment), a DAgger-trained student, and a BC student across three training runs. Our framework with and without alignment significantly outperforms students trained with DAgger and BC, with success rates of up to 0.77 compared to 0.47 (DAgger) and 0.27 (BC). This improvement can be explained by the shared task decoder, which is trained by leveraging the training samples of the teacher. Our method with alignment improves student success rates by 17% compared to the non-aligned framework while also consistently achieving higher returns (Figure 3). These results demonstrate that our framework helps the teacher learn better behaviors for the student. Student policies trained without alignment achieve non-zero success rates due to the small sampling interval of the cabinet position. This allows them to memorize behaviors without relying heavily on the images. All teachers, regardless of alignment, reach a 100% success rate. Interestingly, the return of the teacher trained with alignment is also constantly higher than without alignment. A possible explanation is that the teacher learns gripper movements optimized for robustness without trying multiple times to grab the handle, which is a difficult behavior for the student lacking relative pose information. As can be seen in Figure 2, our method leads to several different behaviors, which also explains the high variance of success rates and returns. With alignment, the teacher learns once to grab the handle from a top-down configuration while another teacher lowers its first two elements to make the red handle visible. In both cases, the red handle is visible right before the gripper touches it. This shows that our alignment leads to emerging teacher behaviors that consider the imitation difficulties of the student.

References

[1] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. *Robotics: Science and Systems*, 2023.

[2] J. Luo, C. Xu, X. Geng, G. Feng, K. Fang, L. Tan, S. Schaal, and S. Levine. Multi-stage cable routing through hierarchical imitation learning. *IEEE Transactions on Robotics*, 2024.

[3] O. M. Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, T. Kreiman, C. Xu, et al. Octo: An open-source generalist robot policy. *Robotics: Science and Systems*, 2024.

[4] R. Doshi, H. Walke, O. Mees, S. Dasari, and S. Levine. Scaling cross-embodied learning: One policy for manipulation, navigation, locomotion and aviation. *8th Conference on Robot Learning*, 2024.

[5] H. Nguyen, A. Baisero, D. Wang, C. Amato, and R. Platt. Leveraging fully observable policies for learning under partial observability. *arXiv preprint arXiv:2211.01991*, 2022.

[6] T. Xu, Z. Li, and Y. Yu. Error bounds of imitating policies and environments. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 15737–15749. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/b5c01503041b70d41d80e3dbe31bbd8c-Paper.pdf.

[7] U. Syed and R. E. Schapire. A reduction from apprenticeship learning to classification. *Advances in neural information processing systems*, 23, 2010.

[8] S. Ross, G. Gordon, and D. 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.

[9] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[10] M. Oquab, T. Darcet, T. Moutakanni, H. V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, R. Howes, P.-Y. Huang, H. Xu, V. Sharma, S.-W. Li, W. Galuba, M. Rabbat, M. Assran, N. Ballas, G. Synnaeve, I. Misra, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski. Dinov2: Learning robust visual features without supervision, 2023.