Latent Behavior Cloning for Deep Reinforcement Learning Robotic Tasks

Romanos Voulgarakis[3], Dimitrios Katsikas[1], Nikolaos Passalis[2], and Anastasios Tefas[1]
[1]*Dept. of Informatics, Faculty of Sciences, Aristotle University of Thessaloniki*, Greece
[2]*Dept. of Chemical Engineering, Faculty of Engineering, Aristotle University of Thessaloniki*, Greece
[3]*Dept. of Electrical and Compt Engineering, Faculty of Engineering, Aristotle University of Thessaloniki*, Greece

***Abstract*—Deep Reinforcement Learning (DRL) has achieved remarkable success across various domains, yet its real-world applicability remains challenging due to training instability, particularly in complex environments with sparse rewards. Imitation learning mitigates this issue by leveraging expert demonstrations, allowing learning in scenarios where standard RL struggles. Experts are typically humans or teacher agents, often trained with privileged information (PI)—data available during training but not at inference. In this work, we extend beyond standard action replication from teacher to student by also transferring knowledge of latent representations. First, a privileged teacher is trained to use ground-truth information to accelerate learning. Then, a student agent, which lacks access to PI, is trained to align both its actions and the geometry of its intermediate representations with those of the teacher—effectively cloning both decision-making and latent behavior. We evaluate our method on two robotic manipulation tasks and demonstrate that latent behavior cloning significantly improves reward performance and convergence time compared to standard behavior cloning.**

I. Introduction

Deep Reinforcement Learning (DRL) has been established as an effective learning paradigm, demonstrating great success in scenarios where direct supervision is unavailable—either due to the lack of labels, their high cost, or the inability of humans to provide correct labels in complex environments. In recent years, DRL has been successfully employed in various domains, including, but not limited to, game-playing [5], natural language processing [11], and robotics [9]

Despite its successes, Deep Reinforcement Learning (DRL) remains challenging to apply in real-world scenarios due to issues in reward design, exploration, and efficiency. When rewards are sparse or delayed, DRL agents struggle to associate actions with long-term outcomes, making learning slow and unreliable [14]. Designing effective reward functions to guide learning is difficult and often results in handcrafted solutions that do not generalize well across different tasks. Even with a well-designed reward, complex tasks often require specific action sequences that agents struggle to discover on their own. Additionally, DRL algorithms can be unstable, requiring extensive trial and error to function properly.

To address these challenges, researchers have explored Imitation Learning (IL) [26] as an alternative to reinforcement learning, especially in environments where defining a reward function is difficult or exploration is impractical. IL allows an agent to learn from expert demonstrations instead of relying solely on trial and error. By using expert trajectories, IL reduces the need for manually designed rewards and extensive environment interactions. A common approach within IL is Behavior Cloning (BC) [24], which treats imitation as a supervised learning problem by training a model to map states to expert actions. While BC is straightforward and effective, it suffers from covariate shift, where the states encountered by the student during deployment differ from those in the expert's demonstrations.

Various lines of work have emerged to solve this issue, such as Inverse Reinforcement Learning (IRL) [22], which involves an apprentice agent that aims to infer the reward function underlying the observed demonstrations and Interactive Imitation Learning (IIL) [1], that assumes that the agent has access to an online expert who can be consulted during training e.g. to relabel data [21] or provide corrective interventions [12]. Finally, significant interest has been in merging behavior cloning and reinforcement learning into a unified framework. To this end, [4] proposed the Cycle of Learning (CoL), which employs an actor-critic architecture with a loss function that combines behavior cloning and one-step Q-learning losses within an off-policy algorithm, enabling a DRL agent to learn from human demonstrations.

The expert demonstrator is not limited to a human; it can also be a neural network that has already mastered the task. One way to obtain such a teacher is by training it with privileged information (PI)—ground-truth data that would normally be unavailable to the agent during deployment [2]. Unlike human demonstrators, who only provide final actions, a teacher agent grants access to its latent behavior, revealing the internal processes that lead to its decisions.

In this work, we introduce Latent Behavior Cloning (LBC), which extends standard behavior cloning by transferring knowledge not only from the teacher's actions but also from its intermediate representations. Our framework enables a non-privileged student to learn from the hidden layers of a privileged-trained teacher, leveraging its structured decision-making process. We validate our approach on two challenging robotic manipulation tasks. First, we train a teacher using PI, including ground-truth states and object relationships relevant to the task. Then, we transfer its knowledge to a student agent trained only with visual input. Our results show that Latent Behavior Cloning significantly reduces the performance gap

ISBN: 978-9-46-459362-4

between the privileged teacher and the student, showing its effectiveness in realistic settings.

The rest of the paper is structured as follows. In Section II, we briefly discuss related works. The proposed method is then analytically presented in Section III, followed by its evaluation in Section IV. Finally, Section V concludes the paper.

II. Related Work

a) Behavioral Cloning and Reinforcement Learning: Combining Behavior Cloning (BC) with Deep Reinforcement Learning (DRL) naturally addresses the covariate shift problem and has been explored in both discrete action spaces [6] and continuous action spaces [8], [20], which are more commonly encountered in robotic manipulation tasks. DAPG [20] integrates imitation learning and reinforcement learning by first pretraining a policy with expert demonstrations using behavior cloning and then fine-tuning it with policy gradient updates, improving sample efficiency and performance in sparse-reward environments. Building on this line of work, [4] introduces the Cycle of Learning (CoL) framework, which employs an actor-critic architecture with a loss function that combines behavior cloning and one-step Q-learning losses within an off-policy algorithm, along with a pretraining step to learn from human demonstrations—resulting in faster training compared to DAPG. This paper builds upon CoL, as it provides a systematic approach to merging expert supervision with reinforcement learning.

b) Learning Using Privileged Information: Learning Using Privileged Information (LUPI) was introduced by Vapnik and Vashist for support vector machines [25] and has since been adopted in supervised learning settings [13], [19]. Privileged information (PI) refers to data that is unavailable during inference but accessible during training. The core idea is to transfer knowledge from an intelligent teacher, trained with PI, to a student model that performs inference without access to PI. This concept aligns well with many DRL scenarios, where privileged information about the ground-truth state of the environment may be available during training, while only partial observations from sensors can be used during inference. Several studies, such as [2], [15], have applied this teacher-student framework in DRL, where a teacher agent with access to privileged information (e.g., higher-quality sensors or additional modalities) provides guidance to a student agent that learns without such information.

c) Knowledge Distillation: To facilitate LUPI, knowledge from the privileged teacher agent must be transferred to the student agent. The process of knowledge distillation, originally proposed for model compression [7], has been widely explored in DRL settings to transfer decision-making knowledge from teacher to student [16], [23]. Probabilistic Knowledge Transfer (PKT) [17] offers a different approach by transferring the local geometry of the teacher's representations rather than matching exact predictions. This relaxation enables PKT to be applied across layers of different dimensions or earlier than the final decision-making layer, both in supervised [18] and DRL settings [10]. In this work, we apply PKT for latent behavior cloning in a LUPI scenario, transferring the underlying geometry of privileged teacher representations to the student agent.

III. Proposed Method

A. Preliminaries

We base our algorithm on the Twin Delayed Deep Deterministic Policy Gradient (TD3) [3] due to its proven effectiveness in continuous environments, making it well-suited for robotic manipulation tasks. However, our framework is not limited to TD3 and can be seamlessly applied to other actor-critic methods. TD3 employs a policy network to compute actions while incorporating techniques to reduce overestimation bias and improve training stability. TD3 uses a policy network π for computing actions

$$\pi(s \mid \theta_\pi), \tag{1}$$

and two Q-networks for evaluating state values

$$Q_i(s \mid \theta_{Q_i}), \quad i \in \{1, 2\}, \tag{2}$$

For each of these networks, a corresponding target network is maintained, denoted by

$$\pi'(s \mid \theta'_\pi) \quad \text{and} \quad Q'_i(s \mid \theta'_{Q_i}), \tag{3}$$

Actions used to form the Q-learning target are based on the target policy with added clipped noise on each dimension. After adding the noise, the target action is clipped to lie within the valid action range:

$$a(s') = \text{clip}\Big(\pi_t(s') + \text{clip}(\epsilon, -c, c),\ a_{\text{low}},\ a_{\text{high}}\Big), \tag{4}$$

where $\epsilon \sim \mathcal{N}(0, \sigma)$, $\pi_t(s')$ denotes the target policy (equivalent to $\pi'(s')$) and $\text{clip}(\cdot)$ is the clipping function. Target policy smoothing regularizes the algorithm by preventing the policy network from overfitting to the Q-networks. Both Q-networks learn from a single target computed as the minimum of the two target Q-values:

$$y(r, s', d) = r + \gamma(1 - d) \min_{i \in \{1,2\}} Q'_i\big(s', a'(s')\big). \tag{5}$$

The policy is updated by maximizing the output of the first Q-network:

$$\max_{\theta_\pi} \mathbb{E}_{s \sim \mathcal{D}}\Big[Q_{\theta_{Q_1}}\big(s, \pi(s \mid \theta_\pi)\big)\Big]. \tag{6}$$

Periodically, the target networks are updated using a soft update:

$$\theta'_\pi \leftarrow \rho\, \theta'_\pi + (1 - \rho)\, \theta_\pi, \tag{7}$$

$$\theta'_{Q_i} \leftarrow \rho\, \theta'_{Q_i} + (1 - \rho)\, \theta_{Q_i}. \tag{8}$$

From now on, the policy network (not the target) will be referred to as the actor, and Q_1 will be referred to as the critic.

B. Probabilistic Knowledge Transfer

Let us denote the internal representations of the teacher model as $f^{(T)}(\mathbf{s}) \in \mathbb{R}^M$ and those of the student model as $f^{(S)}(\mathbf{s}) \in \mathbb{R}^{M'}$, with dimensionalities M and M', respectively. To simplify the presentation of the proposed method, we define $\mathbf{x}_i$ as the internal representation of the models when presented with the i-th state sampled from the buffer, i.e.,

$$\mathbf{x}_i^{(T)} = f^{(T)}(\mathbf{s}_i) \text{ for the teacher model, and} \tag{9}$$

$$\mathbf{x}_i^{(S)} = f^{(S)}(\mathbf{s}_i) \text{ for the student.} \tag{10}$$

To estimate the conditional probability distributions of both the teacher and student representations, we employ kernel density estimation. Specifically, the conditional probability distribution for the teacher model is computed as:

$$p_{i|j}^{(T)} = \frac{K(\mathbf{x}_i^{(T)}, \mathbf{x}_j^{(T)}; 2\sigma^2)}{\sum_{k=1,k\neq j}^{N} K(\mathbf{x}_k^{(T)}, \mathbf{x}_j^{(T)}; 2\sigma^2)}, \tag{11}$$

and for the student model as:

$$p_{i|j}^{(S)} = \frac{K(\mathbf{x}_i^{(S)}, \mathbf{x}_j^{(S)}; 2\sigma^2)}{\sum_{k=1,k\neq j}^{N} K(\mathbf{x}_k^{(S)}, \mathbf{x}_j^{(S)}; 2\sigma^2)}. \tag{12}$$

Here, $K(\mathbf{x}, \mathbf{y}; \sigma^2)$ denotes a symmetric kernel function with bandwidth σ. In this study, we use a kernel function based on the cosine similarity metric, following observations from [17]. This kernel is defined as:

$$K_{\text{cosine}}(\mathbf{x}, \mathbf{y}) = \frac{1}{2}\left(\frac{\mathbf{x}^T\mathbf{y}}{\|\mathbf{x}\|_2\|\mathbf{y}\|_2} + 1\right). \tag{13}$$

To quantify the divergence between the probability distributions of the teacher and student models, PKT employs the Kullback-Leibler (KL) divergence, given by:

$$L_{PKT} = \sum_{i,j} p_{i|j}^{(T)} \log\left(\frac{p_{i|j}^{(T)}}{p_{i|j}^{(S)}}\right). \tag{14}$$

By minimizing L_{PKT}, we enforce the geometry of the representations of the student model to align with that of the teacher, ensuring that similar representations according to the teacher remain similar for the student.

C. Latent Behavior Cloning

Following CoL, the student agent learns from a mixture of expert demonstrations and acquired transitions sampled from a replay buffer with a specified ratio. Unlike CoL which uses human demonstrations, here the expert is a teacher agent trained with PI. This allows the student to learn from new samples acquired using its policy while staying grounded on samples acquired using the privileged teacher.

Our proposed behavior cloning loss J_{BC} is a linear combination of losses applied at different layers. An L2 loss, $J_{BC_{L2}}$, is applied to the final action layer, performing standard behavior cloning. Simultaneously, individual PKT losses, $J_{BC_{PKT}}$, can be applied at multiple intermediate layers of the architecture.

Algorithm 1 Latent Behavior Cloning Algorithm

```
Input: Privileged trained teacher policy $\pi^T$, teacher replay buffer $D_T$, student replay buffer $D_S$ and hyperparameters.
Initialize: Actor $\pi(s \mid \theta_\pi)$, critics $Q_i(s \mid \theta_{Q_i})$, for $i \in \{1, 2\}$, and target networks $\pi'(s \mid \theta'_\pi)$, $Q'_i(s \mid \theta'_{Q_i})$.
Collect Teacher Rollouts: Run $\pi^T$ in the environment and collect rollouts to populate $D_T$.
for $i = 1, ..., T$ do
  Collect Student Rollouts: Run $\pi$ in the environment and collect rollouts to populate $D_S$.
  for $j = 1, ..., N$ do
    Sample batch $B$ from $D_S$ and $D_T$ with a fixed ratio (e.g., 75% from $D_S$, 25% from $D_T$).
    Compute $J_c$ and $J_\pi$ according to (Eq. 18, Eq. 17).
    Update critics: $\theta_Q \leftarrow \theta_Q - \nabla_{\theta_Q} J_{\mathrm{Q}}$
    if $j \bmod d = 0$ then
      Update policy:
          $\theta_\pi \leftarrow \theta_\pi - \nabla_{\theta_\pi} J_\pi$
      Update target networks:
          $\theta'_\pi \leftarrow \rho\,\theta'_\pi + (1-\rho)\,\theta_\pi$
          $\theta'_{Q_i} \leftarrow \rho\,\theta'_{Q_i} + (1-\rho)\,\theta_{Q_i}$
    end if
  end for
end for
```

The final loss function is given by:

$$J_{BC} = \beta_1 \cdot J_{BC_{L2}} + \beta_2 \cdot \sum_{i=1}^{N} J^i_{BC_{PKT}}, \tag{15}$$

where PKT is applied at N intermediate layers indexed by i, with β_1 and β_2 being weighting coefficients for L2 and PKT, respectively.

To mitigate training instabilities that we encountered, we also adopted the following modifications. Following [4], we L2 penalize the weights of both actor and critic, resulting in additional regularization losses J^{π}_{L2} and J^{c}_{L2} for the actor and the critic.

We also normalize the actor's Q-loss (Eq. 6), with a term a, defined as:

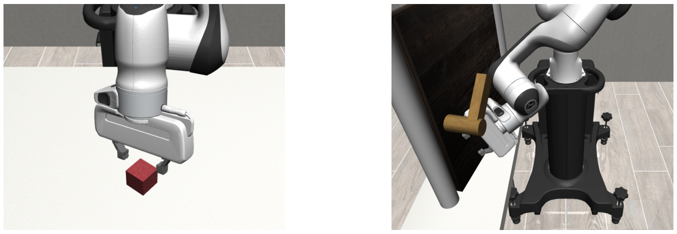
$$a = \frac{\lambda_{\text{norm}}}{\frac{1}{N}\sum_{(s_i, \pi(s_i|\theta_\pi))} |Q(s_i, \pi(s|\theta_\pi))|} \tag{16}$$

where λ_{norm} is a fixed hyperparameter, in order to balance the influence of the reward scaling. In practice, this normalization is carried out on a per batch basis, instead of the whole dataset. The total objective for the actor is defined as

$$J_\pi = J_A + \lambda_{\text{BC}} \cdot J_{BC} + \lambda_{\text{L2}} \cdot J^{\pi}_{\text{L2}}, \tag{17}$$

while for the critic as

$$J_c = J_Q + \lambda_{\text{L2}} \cdot J^{\pi}_{\text{L2}}, \tag{18}$$



(a) Block lifting env.

(b) Door opening env.

Fig. 1: Student agent's constrained viewpoint. The agent has access only to a single source visual information, with no privileged information.

IV. Experimental Evaluation

A. Robotic Environments

We evaluate the proposed method on two challenging robotic manipulation tasks provided by the Robosuite environment [27]. Specifically, we consider the following tasks:

1) Block Lifting: A robotic arm must lift a randomly placed cube above a specified height. The agent earns soft rewards for approaching, grasping, and lifting the cube.

2) Door Opening: A robotic arm must open a randomly placed door by operating its handle. Soft rewards are given for reaching and rotating the handle.

Rewards are normalized between 0 and 1, with a 200-step episode limit unless the maximum reward is achieved. The privileged teacher receives structured, low-dimensional observations (e.g., positions, distances, angles, velocities), while the student relies solely on visual input from a single viewpoint (Fig. 1), making the task significantly more challenging.

B. Networks Architecture

The teacher model, using a low-dimensional tabular input, is a three-layer MLP with 64 neurons per layer and ReLU activations. The actor's final activation is Tanh, while the critic output has no activation. The student receives a stack of three images to capture temporal information. Its architecture consists of two convolutional blocks, each with two convolutional layers with 3×3 filters with ReLU, followed by max-pooling. The output is flattened and processed by a three-layer MLP identical to the teacher's. In the critic network, the action vector skips the convolutional layers and is concatenated with the final dense layers.

C. Configuration

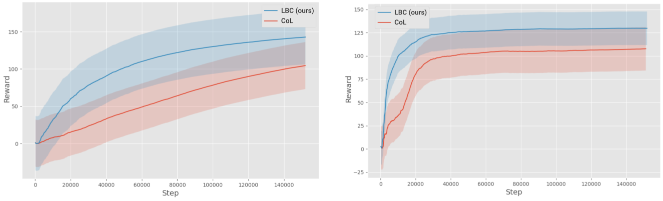
We train all agents using the Adam optimizer with a learning rate of 10^{-3} and a batch size of 256. The regularization weights are set to $\lambda_{L2} = 10^{-6}$, and the behavior cloning weight is set to $\lambda_{BC} = 1$. The weights for the L2 and PKT losses are defined as $\beta_1 = 1$ and $\beta_2 = 0.1$, respectively, with PKT applied between the teacher and student at the last two dense layers before the action layer. The normalization parameter λ_{norm} for the actor's Q-loss is fixed at 2.5. Finally, the sampling ratio between teacher and student samples, drawn from the corresponding replay buffers D_T and D_S, is set to $0.75/0.25$. We run all our experiments for 10 seeds.

D. Results

In Table I, we present the average achieved rewards for different methods. The first row shows the performance of the teacher agent, which is trained with privileged information. The student agent, restricted to visual information, fails to converge without behavior cloning. However, when PKT is used to distill the latent behavior of the teacher—i.e., the behavior within the intermediate dense layers—the performance of the student agent improves significantly in both environments, relative to CoL. In Figure 2, the average reward curves during training are illustrated, demonstrating not only convergence to a higher reward but also faster convergence.

TABLE I: Comparison of DRL agents at convergence. We report the mean value ± std over 10 runs. (–) indicates a lack of convergence.

Method	**Lift**	**Door**
TD3 (Teacher w. PI)	168	161
TD3 [3]	–	–
Behavior Cloning (CoL) [4]	107±22	105±30
Latent Behavior Cloning (ours)	**131±16**	**144±36**



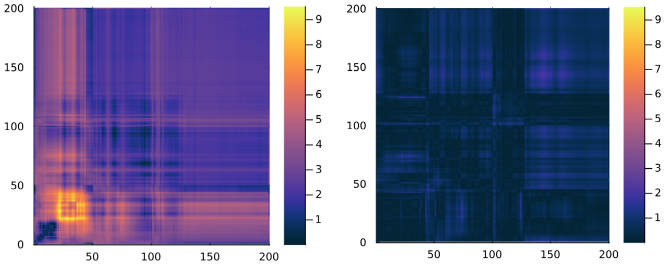
(a) Block lifting env.

(b) Door opening env.

Fig. 2: Average reward curves (mean±std over 10 runs) during training for our proposed Latent Behavior Cloning (LBC) with blue, versus Cycle of Learning (CoL) with red.

We identify two key reasons for these results. LBC offers a softer objective than standard BC, by preserving batch-level geometry of the representations, rather than enforcing an exact match, easing student learning. Additionally, LBC provides direct supervision in intermediate layers, enabling a gradual learning process for the final action behavior. In contrast, standard behavior cloning directly regresses the teacher's actions, which is challenging when observation spaces differ.

Figure 3 compares the similarity matrices of the second-to-last layer representations for the student and teacher, derived from the same state despite different observation spaces. Large discrepancies indicate significant differences in how each model assigns similarities to sample pairs. As shown in Figure 3a, standard behavior cloning fails to reconstruct these pairwise similarities, losing the local geometry of the representations. In contrast, Figure 3b shows that LBC more accurately aligns the student's similarity matrix with the teacher's, resulting in smaller residuals. This suggests that our approach effectively transfers hidden knowledge from the teacher's intermediate layers, enhancing the student's ability to learn the teacher's actions.



(a) Behavior Cloning (CoL) (b) Latent Behavior Cloning

Fig. 3: Pairwise similarity differences between each student and the teacher for a random batch, where smaller values indicate a closer match in similarity assignments.

V. Conclusions

In this work, we present a novel method for transferring knowledge of latent representations from a teacher trained with privileged information to a student without access to the same data. To achieve this, we propose Latent Behavior Cloning (LBC), which uses Probabilistic Knowledge Transfer to align the hidden representations of the teacher and student agents. LBC enables the student to replicate the teacher's behavior in the hidden layers rather than limiting behavior cloning to the action layer, like in previous works. Experiments on robotic manipulation tasks demonstrate the superiority of our approach over standard behavior cloning.

Acknowledgments

The work presented here is supported by the RoboSAPIENS project funded by the European Commission's Horizon Europe programme under grant agreement number 101133807. This publication reflects the authors' views only. The European Commission is not responsible for any use that may be made of the information it contains.

References

[1] Carlos Celemin, Rodrigo Pérez-Dattari, Eugenio Chisari, Giovanni Franzese, Leandro de Souza Rosa, Ravi Prakash, Zlatan Ajanović, Marta Ferraz, Abhinav Valada, and Jens Kober. Interactive imitation learning in robotics: A survey, 2022.

[2] Dian Chen, Brady Zhou, Vladlen Koltun, and Philipp Krähenbühl. Learning by cheating. In Leslie Pack Kaelbling, Danica Kragic, and Komei Sugiura, editors, *Proceedings of the Conference on Robot Learning*, Proceedings of Machine Learning Research, pages 66–75. PMLR, 30 Oct–01 Nov 2020.

[3] Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods, 2018.

[4] Vinicius G. Goecks, Gregory M. Gremillion, Vernon J. Lawhern, John Valasek, and Nicholas R. Waytowich. Integrating behavior cloning and reinforcement learning for improved performance in sparse reward environments. In *Adaptive Agents and Multi-Agent Systems*, 2019.

[5] Danijar Hafner, J. Pasukonis, Jimmy Ba, and Timothy P. Lillicrap. Mastering diverse domains through world models. *ArXiv*, 2023.

[6] Todd Hester, Matej Vecerík, Olivier Pietquin, Marc Lanctot, Tom Schaul, Bilal Piot, Dan Horgan, John Quan, Andrew Sendonaris, Ian Osband, Gabriel Dulac-Arnold, John P. Agapiou, Joel Z. Leibo, and Audrunas Gruslys. Deep q-learning from demonstrations. In *AAAI Conference on Artificial Intelligence*, 2017.

[7] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *ArXiv*, 2015.

[8] Tao Huang, Kai Chen, Bin Li, Yunhui Liu, and Qingxu Dou. Demonstration-guided reinforcement learning with efficient exploration for task automation of surgical robot. *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[9] Julian Ibarz, Jie Tan, Chelsea Finn, Mrinal Kalakrishnan, Peter Pastor, and Sergey Levine. How to train your robot with deep reinforcement learning: lessons we have learned. *The International Journal of Robotics Research*, pages 698 – 721, 2021.

[10] Dimitrios Katsikas, Nikolaos Passalis, and Anastasios Tefas. Bidirectional knowledge transfer for continual deep reinforcement learning in financial trading. In *2024 IEEE 34th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, 2024.

[11] Timo Kaufmann, Paul Weng, Viktor Bengs, and Eyke Hüllermeier. A survey of reinforcement learning from human feedback. *ArXiv*, 2023.

[12] Michael Kelly, Chelsea Sidrane, K. Driggs-Campbell, and Mykel J. Kochenderfer. Hg-dagger: Interactive imitation learning with human experts. *2019 International Conference on Robotics and Automation (ICRA)*, pages 8077–8083, 2018.

[13] John Lambert, Ozan Sener, and Silvio Savarese. Deep learning under privileged information using heteroscedastic dropout. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8886–8895, 2018.

[14] Haozhe Ma, Zhengding Luo, Thanh Vinh Vo, Kuankuan Sima, and Tze-Yun Leong. Highly efficient self-adaptive reward shaping for reinforcement learning. *ArXiv*, 2024.

[15] Yunpeng Pan, Ching-An Cheng, Kamil Saigol, Keuntaek Lee, Xinyan Yan, Evangelos A. Theodorou, and Byron Boots. Agile autonomous driving using end-to-end deep imitation learning. *Robotics: Science and Systems XIV*, 2017.

[16] Emilio Parisotto, Jimmy Ba, and Ruslan Salakhutdinov. Actor-mimic: Deep multitask and transfer reinforcement learning. *CoRR*, 2015.

[17] Nikolaos Passalis and Anastasios Tefas. Learning deep representations with probabilistic knowledge transfer. In *Computer Vision – ECCV 2018: 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XI*, page 283–299, Berlin, Heidelberg, 2018. Springer-Verlag.

[18] Nikolaos Passalis, Maria Tzelepi, and Anastasios Tefas. Heterogeneous knowledge distillation using information flow modeling. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2336–2345, 2020.

[19] Danil Provodin, Bram van den Akker, Christina Katsimerou, Maurits Kaptein, and Mykola Pechenizkiy. Rethinking knowledge transfer in learning using privileged information, 2024.

[20] Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *ArXiv*, 2017.

[21] Stéphane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *International Conference on Artificial Intelligence and Statistics*, 2010.

[22] Stuart Russell. Learning agents for uncertain environments (extended abstract). In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, COLT' 98, page 101–103, New York, NY, USA, 1998. Association for Computing Machinery.

[23] Andrei A. Rusu, Sergio Gomez Colmenarejo, Çaglar Gülçehre, Guillaume Desjardins, James Kirkpatrick, Razvan Pascanu, Volodymyr Mnih, Koray Kavukcuoglu, and Raia Hadsell. Policy distillation. *CoRR*, 2015.

[24] Faraz Torabi, Garrett Warnell, and Peter Stone. Behavioral cloning from observation. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, IJCAI'18. AAAI Press, 2018.

[25] Vladimir Vapnik and Rauf Izmailov. Learning using privileged information: similarity control and knowledge transfer. *J. Mach. Learn. Res.*, page 2023–2049, January 2015.

[26] Maryam Zare, Parham Mohsenzadeh Kebria, Abbas Khosravi, and Saeid Nahavandi. A survey of imitation learning: Algorithms, recent developments, and challenges. *IEEE Transactions on Cybernetics*, 2023.

[27] Yuke Zhu, Josiah Wong, Ajay Mandlekar, Roberto Martín-Martín, Abhishek Joshi, Kevin Lin, Abhiram Maddukuri, Soroush Nasiriany, and Yifeng Zhu. robosuite: A modular simulation framework and benchmark for robot learning. In *arXiv preprint arXiv:2009.12293*, 2020.