Unveiling the Role of Expert Guidance: A Comparative Analysis of User-centered Imitation Learning and Traditional Reinforcement Learning

Amr Gomaa[1,2], Bilal Mahdy[1]

[1]German Research Center for Artificial Intelligence (DFKI), Saarbrücken, Germany
[2]Saarland Informatics Campus, Saarland University, Saarbrücken, Germany

Abstract
Integration of human feedback plays a key role in improving the learning capabilities of intelligent systems. This comparative study delves into the performance, robustness, and limitations of imitation learning compared to traditional reinforcement learning methods within these systems. Recognizing the value of human-in-the-loop feedback, we investigate the influence of expert guidance and suboptimal demonstrations on the learning process. Through extensive experimentation and evaluations conducted in a pre-existing simulation environment using the Unity platform, we meticulously analyze the effectiveness and limitations of these learning approaches. The insights gained from this study contribute to the advancement of human-centered artificial intelligence by highlighting the benefits and challenges associated with the incorporation of human feedback into the learning process. Ultimately, this research promotes the development of models that can effectively address complex real-world problems.

Keywords
Human-in-the-loop Learning, Learning From Demonstrations, Reinforcement Learning, Imitation Learning, Personalization

1. Introduction and Related Work

Human-centered artificial intelligence (HCAI) is an exciting new area of research that is attracting increasing attention from researchers of both artificial intelligence (AI) and human-computer interaction (HCI) [1, 2, 3, 4]. Despite the significant progress made in developing autonomous systems, these systems still rely heavily on human operators, local or remote, to intervene and help or take control in situations where the system cannot proceed, highlighting the need for HCAI techniques to promote trust, control, and reliability between users and machines [4]. However, developing and implementing these concepts remains a challenging and complex task [2]. As a result, there is still much room for improvement and further research in this field [3]. Several approaches have proposed ways to incorporate human knowledge into neural networks as a way of initialization, to guide network refinement, and to extract symbolic information from the network [5, 6]. More recent attempts have tried to combine deep learning

HITLAML'23: Human-in-the-Loop Applied Machine Learning, September 04–06, 2023, Belval, Luxembourg

amr.gomaa@dfki.de (A. Gomaa); bilal.mahdy@dfki.de (B. Mahdy)
https://amrgomaaelhady.github.io/ (A. Gomaa)
0000-0003-0955-3181 (A. Gomaa)

with knowledge bases in joint models (e.g., for construction and population) [7, 8]. Some work has focused on integrating neural networks with classical planning by mapping subsymbolic input to symbolic one, which automatic planners can use [9].

Recently, reinforcement learning (RL) [10] has reemerged as a promising machine learning approach within the field of autonomous systems (e.g., ChatGPT). These methods have demonstrated increasing effectiveness in optimizing reward functions for complex environments. However, shaping appropriate reward functions for intricate tasks and encompassing their aspects remains a challenge [11]. In contrast, humans excel at rapidly acquiring complex skills by observing and imitating others. Similarly, autonomous agents can take advantage of this concept, known as learning from demonstrations (LfD) [12], to address the challenges mentioned above using imitation learning (IL) methods using expert demonstrations [13]. Behavioral cloning (BC) [14] and Generative Adversarial Imitation Learning (GAIL) are the state-of-the-art and most prominent approaches employed to tackle imitation learning problems where the agent has access to state and action information from the demonstrations [15]. Significant progress has been made in Reinforcement Learning (RL) and Imitation Learning (IL) domains. Torabi et al. [16] introduced an advanced adaptation of behavioral cloning known as Behavioral Cloning from Observation, where the agent solely observes demonstration states without access to the corresponding demonstration actions. In a separate study by Taylor [17], several methods were proposed to facilitate the agent's optimal utilization of knowledge from suboptimal human demonstrations, including Learning from Human Demonstrations and Learning from Human Feedback. Fang et al. [18] compared reinforcement and imitation learning for indoor visual navigation. Unlike previous works, ours focuses solely on analyzing the efficacy of imitation learning techniques to assess the importance of learning from demonstrations as a human-in-the-loop learning paradigm in a highly complex environment, regardless of the application domain.

Thus, **this paper contributes to the field of imitation and reinforcement learning, evaluating its performance, robustness, and limitations**. We conduct a detailed investigation into the performance of these state-of-the-art imitation learning techniques in the context of a simulated Bird Hunter game using *Unity ml-agents*[1] and *Pytorch*[2] to evaluate and compare their effectiveness with traditional RL techniques; we investigate the impact of expert guidance and suboptimal demonstrations on imitation learning performance compared to traditional reinforcement learning in diverse environmental complexities. We utilize the Proximal Policy Approximation (PPO) [19] and the Soft-Actor Critic (SAC) [20] methods for our investigation as the most used reinforcement learning techniques, especially in simulation frameworks such as Unity. We provide valuable insights into the comparative efficacy of IL and traditional RL, contributing to the development of intelligent systems in various environmental contexts.

[1] https://github.com/Unity-Technologies/ml-agents

[2] https://pytorch.org/

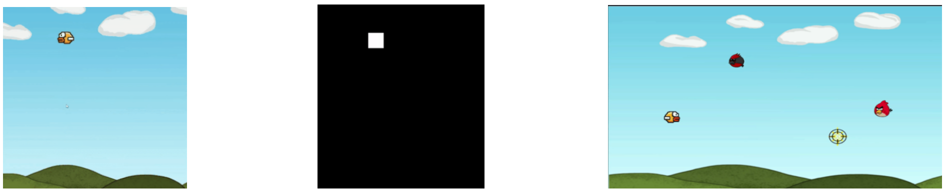

Figure 1: Screenshots (not to scale) of the Bird Hunter game showing the base environment (left), the grayscale backdrop camera view (middle), and the high-complexity environment (right).

2. Methodology

Our study adopts a systematic and progressive approach to comprehensively evaluate the effectiveness of imitation learning with suboptimal and expert demonstrations, as well as its comparison to reinforcement learning techniques such as PPO and SAC. Incremental complexities are introduced to the base environment, incorporating new parameters and analytical challenges at each stage, such as transitioning from grayscale to a colored environment and introducing various bird species with distinct reward schemes. In reinforcement learning, the agent interacts with an environment by selecting actions and receiving feedback through observations and rewards. The observations provide information about the current state of the environment, while the rewards serve as feedback signals that indicate the desirability of the agent's actions. Therefore, for each level of environment complexity, we establish the states of observation and action, as well as the corresponding reward structure (i.e., reward shaping).

Base Environment (Low-complexity Environment). We conducted our study using a preexisting 2D simulated Bird Hunter game to train an autonomous agent (see Figure 1). Initially, a grayscale backdrop was used, with the bird represented as a white box on a black background. The camera sensor captured grayscale images at a resolution of 50 pixels for each axis (x and y), resulting in an observation space of 2500 pixels (50 x 50 x 1), where the one represents a single channel image. The agent's actions involved discrete pixel coordinate pairs for movement, with shooting performed automatically and not treated as a separate action. The reward function (as seen in Equation 1) assigns a reward of (+1) for hitting a bird and a negative reward of (-0.01) for missing.

$$RewardFunction = \begin{cases} +1 & Bird\,hit \\ -0.01 & Bird\,missed \end{cases} \tag{1}$$

Colored Environment (Medium-complexity Environment). This setting builds on the initial environment by utilizing RGB color channels instead of grayscale for the background. The scaling and action space remain the same as in the base environment, while the observation space expands to 7500 pixels (50 x 50 x 3), with the three representing the color channels. The reward function remains consistent with that of the base environment.

Limited Ammunition with Multiple Bird Environment (High Complexity Environment). In this environment, we enhance the complexity by assigning meaning to the colors in the agent's observation, rather than simply introducing a color channel to the environment. In addition to the existing yellow bird as primary target, two new types of birds are introduced. The red bird serves as a bonus, appearing when the agent successfully hits two yellow birds, while the black bird acts as a bomb, exploding upon contact (see Figure 1). Consequently, the reward function is updated to include additional rewards for the red bird (+2) and the black bird (-0.5) as seen in Equation 2.

$$RewardFunction = \begin{cases} +1 & Yellow\ Bird\ hit \\ +2 & Red\ Bird\ hit \\ -0.01 & Bird\ missed \\ -0.5 & Black\ Bird\ hit \end{cases} \tag{2}$$

Furthermore, we introduce new parameters to enhance the agent's convergence towards pinpoint accuracy. The parameter $Clip_{Size}$ is introduced to determine a preset amount of ammunition available for shooting. Another virtual-dependent parameter, $Ammo_t$, specifies the ammunition available to the player at time t. Furthermore, the duration of reload T_{reload} is incorporated to determine the time steps required to complete a reload action. At each time step t, if ammunition is available ($Ammo_t > 0$), the agent is compelled to shoot. Otherwise, a reload action is enforced, resetting the ammunition available to $Clip_{Size}$ after T_{reload} time steps as seen in Equation 3.

$$Ammo_t = \begin{cases} Ammo_{t-1} - 1 & Ammo_{t-1} > 0 \\ Clip\_Size & t \pmod{Clip\_Size + T_{reload}} = 0 \\ 0 & otherwise \end{cases} \tag{3}$$

3. Discussion and Results

In this section, we present the results obtained from different environment settings using various RL and IL approaches. The comparison between approaches in each respective environment is based on the evaluation metrics traditionally used to assess RL agents, as outlined below:

- *Cumulative Reward Function*: The mean reward obtained by the agent in a specified number of steps. Higher value indicates better performance.
- *Episode length*: The time taken for the agent to complete an episode, where episodes end when any bird is shot. Lower value indicates better performance.
- *Entropy*: A measure of the agent's uncertainty in choosing an action given the observed state. Lower value indicates better performance.

3.1. Low and Medium Complexity Setting

First, we compare the performance of both SAC and PPO RL algorithms for the grayscale environment and the RGB environment (i.e., low vs medium complex environments), then choose the superior RL algorithm to compare RL to IL approaches. Figure 2 shows the comparison of the RL algorithms in terms of the metrics mentioned above. While PPO's entropy is lower than that

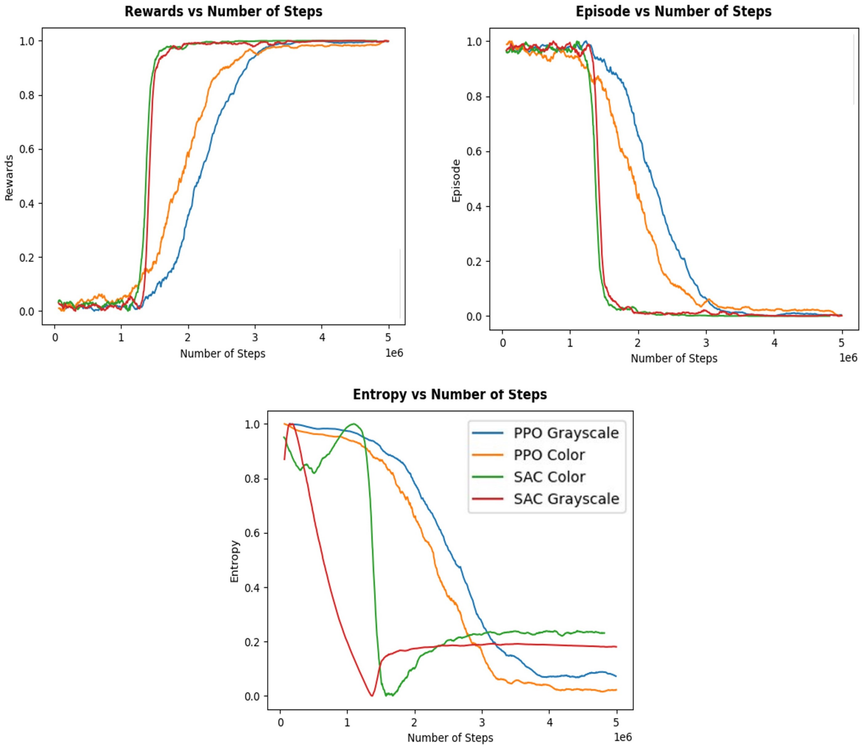

Figure 2: Comparing base environment with the RGB environment for different RL algorithms.

of SAC, indicating a relatively more stable choice of actions, SAC converged faster than PPO in terms of cumulative reward and step count. Thus, SAC is used in further comparisons. Next, we compare traditional RL algorithms (i.e., SAC) to IL techniques (i.e., BC and GAIL). Figure 3 and Table 1 show the results for the RL and IL comparison. It can be seen that while RL converges faster than both BC and GAIL, the latter IL techniques have a better entropy, indicating more stable learning and consistent action choices. It is also noticed that using the GAIL technique alone is not stable and hard to converge for this medium complexity environment, even for training for a very long number of steps (i.e., greater than million steps).

Lastly, the RGB environment was evaluated by comparing two types of demonstrations used to train imitation learning techniques (BC + GAIL): one from a proficient experienced user and the other from a suboptimal user. Both demonstrations came from the same user to ensure consistency, where the user attempted the shoting as best as he could for the expert demonstration and intentially missed few birds to record the demonstration of the suboptimal user. As a manipulation check, examination of the reward function showed that the competent expert performed the task with high accuracy, achieving a mean reward of 0.997 with no missed shots. On the other hand, the suboptimal demonstration had a mean reward of 0.81, indicating a higher frequency of missed shots. These demonstrations aimed to evaluate the performance of imitation learning under the same environment complexity and conditions. Figure 4 illustrates that the agent trained with the expert demonstration exhibited faster learning, greater consis-

Table 1
Comparison of Traditional RL and IL methods in the "Base" environment. The maximum reward achievable by the agent is one.

Metric	RL (SAC)	BC Only	GAIL Only	BC & GAIL
Step Count	**162k**	>500k	$\gg$1M	~500k
Cumulative Reward	**0.98**	0.89	No Convergence	0.95
Entropy	0.66	**0.45**	No Convergence	0.63

tency, and more confident action selection. In contrast, the agent trained with the suboptimal demonstration eventually converged, but it took twice as long as the expert demonstration.

3.2. High Complexity Setting

In order to further assess the performance of the GAIL, BC, and RL algorithms, we performed evaluations in a highly complex environment. This environment included multiple birds with different rewards and limited ammunition, as described in the Methods section. Building on the insights gained from the previously mentioned evaluations of imitation learning techniques, we modified the training approach for BC and GAIL. Instead of relying solely on demonstrations, these algorithms were trained with a combination of intrinsic and extrinsic rewards. This

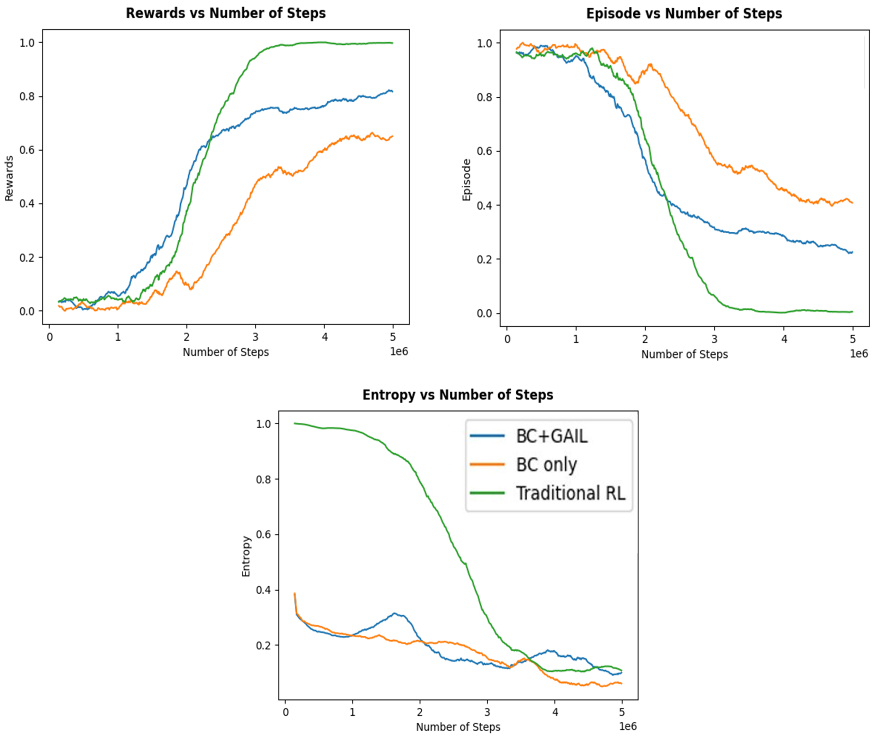

Figure 3: Comparing different IL algorithms in the RGB environment.

Table 2
Comparison between RL, BC and GAIL in the “Limited Ammo and Multiple Birds” environment. The maximum reward that the agent can achieve is two. The expert demo had an average reward of **1.5**.

Metric	RL	BC	GAIL
Convergence Step Count	2.5M	No Convergence	**1M**
Cumulative Reward	1.4	-0.77	**1.67**
Entropy	0.68	6.64	**0.23**

adjustment was made to address the tendency of BC and GAIL to deviate from an optimal policy when trained with demonstrations only. The RL and IL comparison results in this highly complex environment are presented in Figure 5 and Table 2, which provide a comparison of the RL, BC, and GAIL algorithms. These results offer insight into the performance and effectiveness of each algorithm in this challenging setting.

Traditional RL. In this highly complex environment, the traditional RL algorithm failed to capture an effective bird-shooting strategy. Instead, it resorted to “cheating” the environment by learning the average spawn locations of the red and yellow birds. The agent then focused solely on shooting at these specific spots, barely moving the cursor. Remarkably, the traditional RL agent achieved a score close to that of a human player using this method. This highlights the ability of RL algorithms to exploit loopholes given sufficient time.

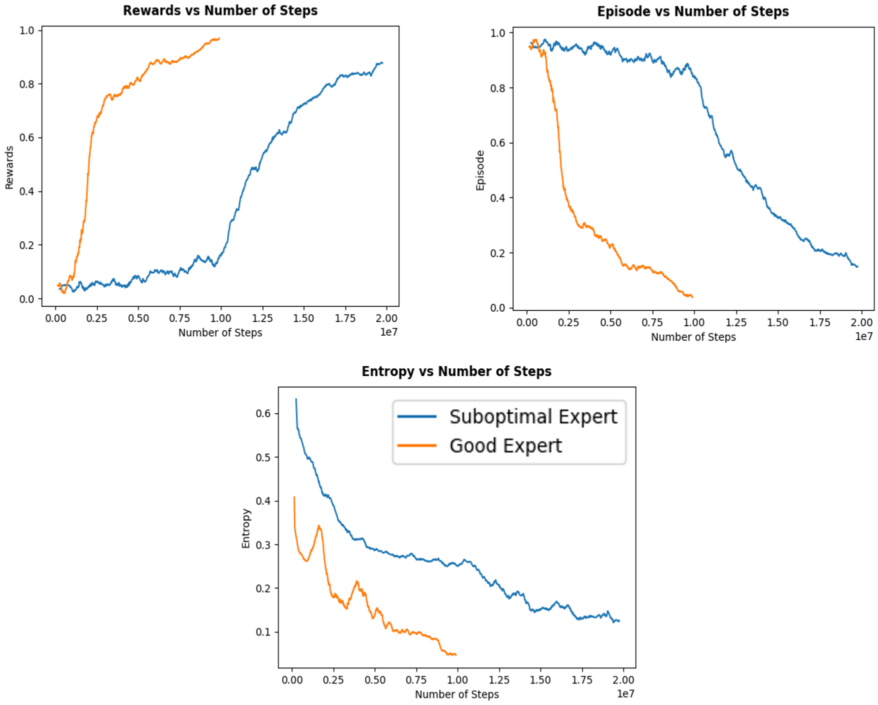

Figure 4: Comparing imitation learning (BC + GAIL) for a good demonstration (i.e., expert user) and suboptimal demonstration (i.e., novice user).

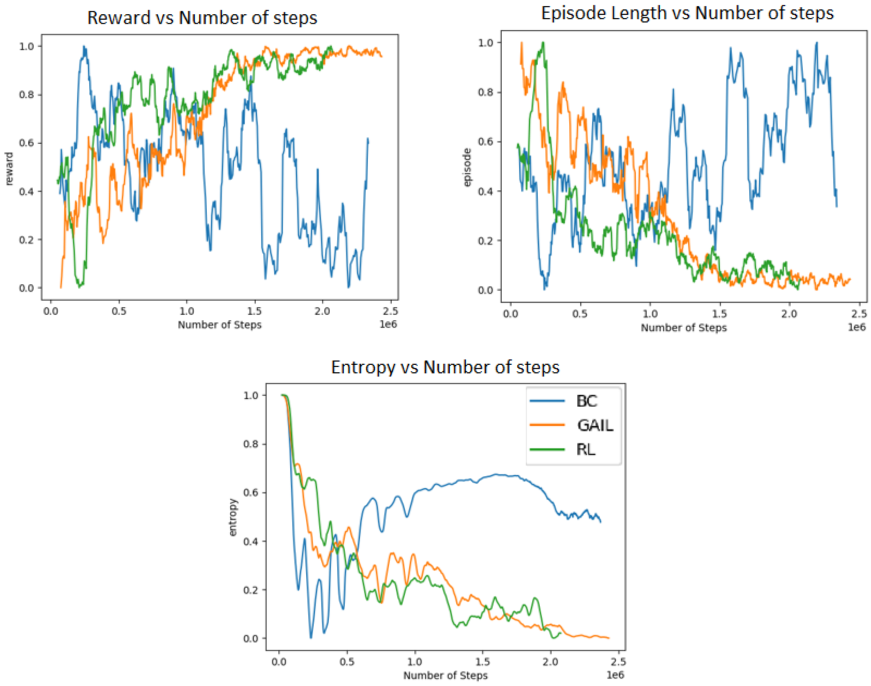

Figure 5: Comparison of average cumulative reward, episode length and model entropy between RL, BC and GAIL in the Limited Ammo environment

Behavioural Cloning. In contrast, the BC algorithm encountered significant difficulties in achieving the score of a human player. Since the recorded demonstration did not utilize the environment loophole but instead moved the cursor around and aimed at the red and yellow birds while avoiding the black ones, the agent struggled to replicate the demonstrated behavior and failed to converge or show improvement after a substantial number of iterations. This underscores the limitations of imitation learning algorithms relying solely on expert demonstrations and their reduced capacity for exploratory behavior compared to traditional RL.

GAIL. Initially, the GAIL algorithm faced similar challenges as the BC algorithm. However, due to its combined approach, GAIL was able to break free from recorded behavior and discover the same environment loophole exploited by the traditional RL algorithm. Ultimately, GAIL achieved the highest score among all algorithms, surpassing even the recorded human demonstrations, while achieving the lowest model entropy. This aligns with the notion that GAIL is particularly effective when dealing with environments of high complexity and dimensions.

4. Conclusion and Future Work

In conclusion, we compared policy optimization techniques and model architectures across various complexities of the environment, providing valuable information and avenues for future research. PPO demonstrated stable convergence and lower model entropy, indicating increased confidence in action selection. However, SAC exhibited superior sample efficiency and faster convergence, emphasizing the stability-efficiency trade-off, making it favorable when time is limited. The imitation learning algorithms converged slower but had a lower model entropy, relying heavily on expert demonstrations and limiting loophole exploitation. Traditional reinforcement learning algorithms discovered loopholes through reward-shaping complexity rather than learning intended behavior. GAIL performed well by effectively capturing expert demonstrations, achieving higher scores, and lower model entropy. This highlights the potential of imitation learning to overcome reinforcement learning limitations. On the other hand, reinforcement learning outperformed imitation learning in simple low-complexity environments where reward shaping is not challenging. Future research should explore performance in different domains, and develop hybrid approaches that take advantage of multiple algorithms to enhance convergence, stability, and exploration capabilities.

Acknowledgments

This work is partially funded by the German Ministry of Education and Research (BMBF) under the TeachTAM project (Grant Number: 01IS17043) and the CAMELOT project (Grant Number: 01IW20008).

References

[1] W. Xu, Toward human-centered ai: a perspective from human-computer interaction, interactions 26 (2019) 42–46.

[2] A. Nowak, P. Lukowicz, P. Horodecki, Assessing artificial intelligence for humanity: Will ai be the our biggest ever advance? or the biggest threat [opinion], IEEE Technology and Society Magazine 37 (2018) 26–34.

[3] J. J. Bryson, A. Theodorou, How Society Can Maintain Human-Centric Artificial Intelligence, Springer Singapore, Singapore, 2019, pp. 305–323.

[4] B. Shneiderman, Human-centered artificial intelligence: Reliable, safe & trustworthy, International Journal of Human–Computer Interaction 36 (2020) 495–504.

[5] J. W. Shavlik, Combining symbolic and neural learning, Machine Learning 14 (1994) 321–331. URL: http://link.springer.com/10.1007/BF00993982. doi:10.1007/BF00993982.

[6] L. Von Rueden, S. Mayer, J. Garcke, C. Bauckhage, J. Schuecker, Informed machine learning–towards a taxonomy of explicit integration of knowledge into machine learning, Learning 18 (2019) 19–20.

[7] A. Ratner, C. Ré, Knowledge base construction in the machine-learning era, Queue 16 (2018) 50:79–50:90. URL: http://doi.acm.org/10.1145/3236386.3243045. doi:10.1145/3236386.3243045.

[8] H. Adel, Deep learning methods for knowledge base population, Ph.D. thesis, LMU, 2018.

[9] M. Asai, A. Fukunaga, Classical planning in deep latent space: Bridging the subsymbolic-symbolic boundary, in: Proceedings of the Conference on Artificial Intelligence (AAAI'18), AAAI Press, 2018, pp. 6094–6101.

[10] R. S. Sutton, A. G. Barto, Reinforcement learning: An introduction, 2018.

[11] D. Hadfield-Menell, S. Milli, P. Abbeel, S. Russell, A. Dragan, Inverse reward design, 2017. arXiv:1711.02827.

[12] B. Argall, S. Chernova, M. Veloso, B. Browning, A survey of robot learning from demonstration, Robotics and Autonomous Systems 57 (2009) 469–483.

[13] C. Finn, S. Levine, P. Abbeel, Guided cost learning: Deep inverse optimal control via policy optimization, 2016. arXiv:1603.00448.

[14] S. Ross, N. Melik-Barkhudarov, K. S. Shankar, A. Wendel, D. Dey, J. A. Bagnell, M. Hebert, Learning monocular reactive uav control in cluttered natural environments, 2013.

[15] J. Ho, S. Ermon, Generative adversarial imitation learning, 2016. arXiv:1606.03476.

[16] F. Torabi, G. Warnell, P. Stone, Behavioral cloning from observation, 2018. arXiv:1805.01954.

[17] M. E. Taylor, Improving reinforcement learning with human input, 2018.

[18] Q. Fang, X. Xu, X. Wang, Y. Zeng, Target-driven visual navigation in indoor scenes using reinforcement learning and imitation learning, CAAI Transactions on Intelligence Technology 7 (2022) 167–176.

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

[20] T. Haarnoja, A. Zhou, P. Abbeel, S. Levine, Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor, in: International conference on machine learning, PMLR, 2018, pp. 1861–1870.