Empowering Generalization for Deep Reinforcement Learning via Symbolic Planning

Extended Abstract

Tianpei Yang
Nanjing University
Suzhou, China
tianpei.yang@nju.edu.cn

Srijita Das
University of Michigan - Dearborn
Dearborn, United States
sridas@umich.edu

Christabel Wayllace
New Mexico State University
Las Cruces, United States
cwayllac@nmsu.edu

Matthew E. Taylor
University of Alberta and Amii
Edmonton, Canada
matthew.e.taylor@ualberta.ca

KEYWORDS

Symbolic Planning, Reinforcement Learning, Generalization

ACM Reference Format:
Tianpei Yang, Srijita Das, Christabel Wayllace, and Matthew E. Taylor. 2025. Empowering Generalization for Deep Reinforcement Learning via Symbolic Planning: Extended Abstract. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 3 pages.

1 INTRODUCTION

Deep Reinforcement Learning (RL) has achieved notable success across a diverse range of fields [1, 2, 16]. However, applying RL to complex domains faces a significant challenge due to sample inefficiency [11]. This issue is further aggravated in domains with sparse rewards. Another challenge is poor generalization: minor variations in object properties within a similar target domain can cause the trained policy to fail. [19, 22].

Symbolic models have long been used to aid RL generalization. Prior work includes Relational RL [6–8]. More recent work has integrated symbolic models with Deep RL, including Inductive Logic Programming [12], high-level programs [4, 18, 21, 23], and symbolic planning [9, 13, 20]. The drawback of most prior work integrating planning and RL is that they combine high-level meta-controllers and symbolic planners to select appropriate subgoals, adding complexity to the high-level process and increasing the training time. Additionally, the mapping between a symbolic state and a low-level state must be manually defined, requiring additional human effort. Moreover, prior work [13, 20] focuses on sample efficiency without accounting for knowledge transfer between different tasks, which we consider a critical capability.

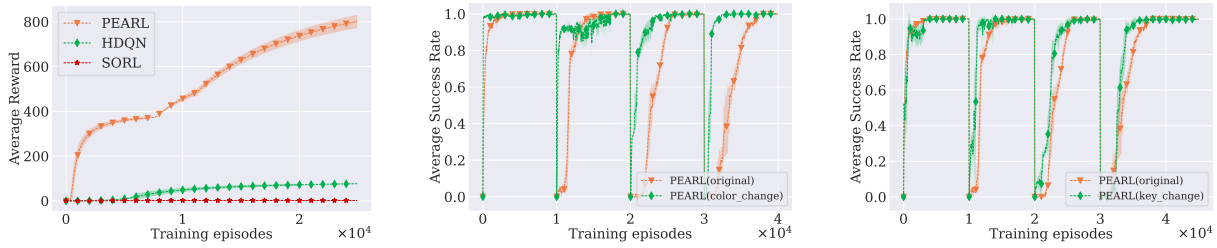
To mitigate these gaps in literature, we introduce Plan-guided Exploration and generAlization for Reinforcement Learning (PEARL). PEARL is a two-level structure incorporating a symbolic planner as the meta-controller to guide the low-level Deep RL agent to learn long-horizon tasks. Since the subgoals are directly derived from the symbolic plan, they are ordered and there is no need to use a separate meta-controller to choose the subgoals (in contrast to prior work). The low-level RL agent learns to achieve the subgoals sequentially, as proposed by the planner, using a single network. As a result, our proposed method is more efficient than prior work. Furthermore, we automatically learn the mapping between the state and action representations in the two levels using an autoencoder, reducing human effort. The symbolic representation also allows us to do zero-shot transfer and knowledge reuse in different tasks with small task variations. Our evaluation of PEARL on well known sparse reward Montezuma's Revenge domain outperforms both hierarchical and symbolic-planning-based baselines in terms of sample efficiency and generalization. Overall, our proposed framework is general and can be used with any off-the-shelf RL algorithm and symbolic planner for improved training and generalization in complex sparse reward environments.

Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Y. Vorobeychik, S. Das, A. Nowé (eds.), May 19 – 23, 2025, Detroit, Michigan, USA. © 2025 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

2 PEARL

We define PEARL with a tuple containing the elements required to represent 1) symbolic states and actions (as defined in a STRIPS [9] planning problem), 2) an RL agent as defined using an MDP, 3) an intrinsic reward to guide the RL agent achieve a subtask, and 4) a set of subgoals as identified by the planner. The final goal of the RL agent is to learn a policy that maximizes the expected discounted accumulated reward. Prior work uses human experts to define the state-action space mapping between the symbolic planner and low-level controller [5, 13, 20]. We propose a technique to automatically learn the state-action mapping between the planner and RL agents by using an autoencoder (i.e., an encoder and a decoder). The two steps for training PEARL are:

Pre-training. In pre-training, the encoder takes a low-level state s as input and outputs a low-dimensional representation of the corresponding symbolic state ($\hat{s}$). The decoder takes this representation as input and outputs a reconstructed low-level state which is used to train the low-level RL policy. To pre-train the autoencoder, we generate a dataset ($500K$ images) from the environment and use a proposition set, which is a collection of necessary propositions in symbolic states to ensure that the learned embedding accurately represents the symbolic state. Beyond the standard reconstruction loss—measured as the distance between the input state s and the reconstructed state $\hat{s}$, we introduce a supervision signal derived



(a) Average return (b) Object color change (c) Object location change (shift left)

Figure 1: Results on Montezuma's Revenge: (a) average return trained on Room 2 and Room 1 (left); (b) average test success rate of achieving four subgoals in Room 2; (c) average test success rate of achieving four subgoals in Room 2.

from the proposition set. The autoencoder is then trained by jointly minimizing both the reconstruction loss and the distance between the learnt embedding and the supervision signal.

Online Learning. After the pre-training stage, PEARL moves to the online training stage. The initial low-level state s of the environment is fed into the encoder to produce a latent representation of the symbolic state. The planner generates a plan using the symbolic state representation. The effects of the first symbolic action in the plan (representing a symbolic state) is mapped to the corresponding low-level subgoal by passing it to the decoder. Lastly, the low-level agent learns the policy to reach the subgoal from the current low-level state and the encoded representation of the symbolic goal state. The agent receives an intrinsic reward if it achieves the subgoal. The autoencoder is also fine-tuned for stable training. The new initial state for the agent, defined by the state in which the current subgoal condition is met, is transformed into the symbolic state space and used by the symbolic planner to generate a new plan (if necessary) for the final goal.

3 EXPERIMENTAL EVALUATIONS

We present our results on the first two rooms in Montezuma's Revenge [3] to test the effectiveness of our method, compared with 2 baselines: (1) HDQN, a hierarchical deep RL method using human-designed subgoals [14]; and (2) SORL, a state-of-the-art algorithm that uses a three-level hierarchy consisting of planner, meta-controller, and controller [13]. We hand-coded Montezuma's revenge in PDDL, encoding each room as a different problem within the same domain and making sure to use general predicates whenever possible to facilitate generalization. We used the Fast Downward planning system [10] to generate the subgoals.

Figure 1 (a) shows the training performance of PEARL compared with two baselines on the first two rooms (Room 2, Room 1 (left), and Room 3 (right). We can see that PEARL receives an average reward of 800 in roughly 2×10^4 episodes, outperforming the baseline approaches. This validates our hypothesis that removing the meta-controller improves the training efficiency, as PEARL takes into account the ordered subgoals output by the planner. *In summary, PEARL effectively learns faster across multiple tasks.*

Figure 1 (b) shows the test results (green line) compared to the training results (orange line), demonstrating PEARL's policies succeed in the new environments. We observe that with small changes between the test environment and the training environment, PEARL immediately generalizes to the new environment without extra training. Even when the differences become larger, PEARL can still quickly achieve a success rate of 1(100%) with a few-shot adjustment. *In summary, PEARL can effectively generalize to new tasks with the same logic, requiring only zero- or few-shot learning.*

Finally, we test PEARL's knowledge reusability by fine-tuning its pre-trained policy, learned in the training environment, to adapt to new environments with semantic modifications, such as shifting an object's location to the left. When the object's location changes, the low-level policy for reaching the object might no longer work for the new task, but other knowledge from the original policy could still be reused. Figure 1 (c) shows that the fine-tuned policy (green line) achieves an average success rate of 1(100%) for each subgoal faster than PEARL learning from scratch (orange line). *In summary, PEARL can effectively reuse knowledge from pre-trained tasks to accelerate training on new tasks with different low-level subgoals.*

4 CONCLUSION AND FUTURE WORK

This paper presents a first step to show the advantages of combining symbolic planning with deep RL to improve learning and generalization. The experiments illustrate how PEARL requires less training than prior work, and how PEARL is also effective in generalization in domains that contain the same domain definitions as the training environment with slight variations in the environment. Additionally, as the Deep RL agent interacts with the environment to transition between pre-conditions and post-conditions of actions, PEARL could help identify incorrectly defined symbolic actions, complementing work that uses large language models to generate PDDL domains [15, 17].

ACKNOWLEDGMENTS

SD acknowledges support from MIDAS Propelling Original Data Science award. MET acknowledges support from the Alberta Machine Intelligence Institute (Amii); a Canada CIFAR AI Chair, Amii; Compute Canada; Mitacs; and NSERC.

REFERENCES

[1] Marc G Bellemare, Salvatore Candido, Pablo Samuel Castro, Jun Gong, Marlos C Machado, Subhodeep Moitra, Sameera S Ponda, and Ziyu Wang. 2020. Autonomous navigation of stratospheric balloons using reinforcement learning. *Nature* 588, 7836 (2020), 77–82.

[2] Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. 2019. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680* (2019).

[3] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 2016. Openai gym. *arXiv preprint arXiv:1606.01540* (2016).
[4] Yushi Cao, Zhiming Li, Tianpei Yang, Hao Zhang, Yan Zheng, Yi Li, Jianye Hao, and Yang Liu. 2022. GALOIS: Boosting Deep Reinforcement Learning via Generalizable Logic Synthesis. In *NeurIPS*. 19930–19943.
[5] Andrew Chester, Michael Dann, Fabio Zambetta, and John Thangarajah. 2022. SAGE: Generating Symbolic Goals for Myopic Models in Deep Reinforcement Learning. *CoRR* abs/2203.05079 (2022).
[6] Kurt Driessens and Jan Ramon. 2003. Relational instance based regression for relational reinforcement learning. In *ICML*.
[7] Kurt Driessens, Jan Ramon, and Hendrik Blockeel. 2001. Speeding up relational reinforcement learning through the use of an incremental first order decision tree learner. In *ECML*.
[8] Sašo Džeroski, Luc De Raedt, and Kurt Driessens. 2001. Relational reinforcement learning. *MLJ* (2001).
[9] Richard E Fikes and Nils J Nilsson. 1971. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial intelligence* 2, 3-4 (1971), 189–208.
[10] Malte Helmert. 2006. The fast downward planning system. *Journal of Artificial Intelligence Research* 26 (2006), 191–246.
[11] Julian Ibarz, Jie Tan, Chelsea Finn, Mrinal Kalakrishnan, Peter Pastor, and Sergey Levine. 2021. How to train your robot with deep reinforcement learning: lessons we have learned. *The International Journal of Robotics Research* (2021).
[12] Zhengyao Jiang and Shan Luo. 2019. Neural logic reinforcement learning. In *International conference on machine learning*. PMLR, 3110–3119.
[13] Mu Jin, Zhihao Ma, Kebing Jin, Hankz Hankui Zhuo, Chen Chen, and Chao Yu. 2022. Creativity of AI: Automatic Symbolic Option Discovery for Facilitating Deep Reinforcement Learning. In *Proceedings of the Thirty-Sixth AAAI Conference on Artificial Intelligence*. AAAI Press, 7042–7050.
[14] Tejas D Kulkarni, Karthik Narasimhan, Ardavan Saeedi, and Josh Tenenbaum. 2016. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. *Advances in neural information processing systems* 29 (2016).
[15] Bo Liu, Yuqian Jiang, Xiaohan Zhang, Qiang Liu, Shiqi Zhang, Joydeep Biswas, and Peter Stone. 2023. Llm+ p: Empowering large language models with optimal planning proficiency. *arXiv preprint arXiv:2304.11477* (2023).
[16] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. 2016. Mastering the game of Go with deep neural networks and tree search. *Nature* (2016).
[17] Pavel Smirnov, Frank Joublin, Antonello Ceravola, and Michael Gienger. 2024. Generating consistent PDDL domains with Large Language Models. *arXiv preprint arXiv:2404.07751* (2024).
[18] Shao-Hua Sun, Te-Lin Wu, and Joseph J Lim. 2019. Program guided agent. In *International Conference on Learning Representations*.
[19] Markus Wulfmeier, Ingmar Posner, and Pieter Abbeel. 2017. Mutual alignment transfer learning. In *Conference on Robot Learning*. PMLR, 281–290.
[20] Fangkai Yang, Daoming Lyu, Bo Liu, and Steven Gustafson. 2018. PEORL: Integrating Symbolic Planning and Hierarchical Reinforcement Learning for Robust Decision-Making. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, Jérôme Lang (Ed.). ijcai.org, 4860–4866.
[21] Yichen Yang, Jeevana Priya Inala, Osbert Bastani, Yewen Pu, Armando Solar-Lezama, and Martin Rinard. 2021. Program synthesis guided reinforcement learning for partially observed environments. *Advances in neural information processing systems* 34 (2021), 29669–29683.
[22] Amy Zhang, Nicolas Ballas, and Joelle Pineau. 2018. A dissection of overfitting and generalization in continuous reinforcement learning. *arXiv preprint arXiv:1806.07937* (2018).
[23] Hao Zhang, Tianpei Yang, Yan Zheng, Jianye Hao, and Matthew E. Taylor. 2024. PADDLE: Logic Program Guided Policy Reuse in Deep Reinforcement Learning. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems* (Auckland, New Zealand) *(AAMAS '24)*. International Foundation for Autonomous Agents and Multiagent Systems, 2585–2587.