Trajectory-Aware Human Feedback for Efficient Hierarchical Reinforcement Learning in Robotics

Naveen Prashanna Gurumruthy[1] Nikunj DineshKumar Gohil[2]

***Abstract*— Hierarchical Reinforcement Learning (HRL) offers a promising approach to tackle complex tasks in robot manipulation by decomposing them into manageable sub-tasks. However, current HRL methods often struggle with generating effective subgoals and ensuring efficient completion of these subgoals. This research proposes an extension to existing HRL frameworks by incorporating human feedback into the high-level policy (subgoal generation), utilizing trajectories from the low-level policy as additional input. This enhanced feedback mechanism aims to improve the efficiency and adaptability of HRL in complex robotic manipulation tasks, such as those involving fetch, pick-and-place, and other intricate manipulation scenarios.**

I. Introduction

Reinforcement Learning (RL) has emerged as a powerful paradigm for training intelligent agents to perform complex tasks. However, the problem of sparse rewards, common in such tasks, presents significant challenges. Sparse rewards make exploration difficult and can destabilize the learning process. HRL and Goal-Conditioned RL (GCRL) are two techniques that attempt to alleviate these issues [1], [2], [3].

In HRL, instead of using a single policy to learn a task, the task is divided into multiple subtasks. A meta-policy (high-level policy) chooses the subtask first, and then a low-level policy learns the specific subtask. For instance, in a "FetchPickAndPlace" task, the high-level policy might set a subgoal to "reach for the object," while the low-level policy would then be responsible for the precise movements and grasping actions required to achieve this subgoal. GCRL modifies the Markov Decision Process (MDP) to include a set of goals. The agent receives rewards for executing actions that achieve these goals.

While HRL and GCRL are effective for long-horizon tasks with sparse rewards, they still face difficulties in generating instructive subgoals and completing them efficiently. In the "FetchPickAndPlace" task, challenges might arise in determining the most effective sequence of subgoals (e.g., reaching, grasping, lifting, moving, placing) and ensuring that the low-level policy can successfully execute the necessary actions for each subgoal.

To address these challenges, this research proposes incorporating human feedback into the high-level policy of the HRL framework, utilizing trajectories from the low-level policy as additional input. This approach aims to leverage human expertise in evaluating both the appropriateness of the chosen subgoal and the feasibility of the low-level policy's actions in achieving that subgoal.

*This work was not supported by any organization

[1]Naveen Prashanna Gurumurthy is a Masters Student with the Department of Computer Science, University of Texas at Dallas, TX 75252, USA NaveenPrashanna.Gurumurthy@utdallas.edu

[2]Nikunj DineshKumar Gohil is also a Masters Student with the Department of Computer Science, University of Texas at Dallas, TX 75252, USA nikunj.gohil@utdallas.edu

II. Related Work

A. Hierarchical Reinforcement Learning

Extensive research in HRL focuses on identifying meaningful subgoals within long-horizon tasks. This includes studies on options, goals, and skills [4], [5], [6]. However, manually designing subgoals is costly and challenging for complex tasks, while automatic generation methods require extensive resources to search the entire state space. Recent work like CSD aims to discover skills through mutual information maximization, but combining these skills for task completion remains a challenge [7], [8]. HAC and HIRO address sparse rewards and non-stationarity at high levels but fall short in effective task decomposition [9], [10], [11]. For example, in a "FetchPush" task, difficulties might arise in automatically generating subgoals that effectively guide the robot arm to push a block around obstacles and towards a target position.

Previous works have also tried to allow collaboration between hierarchical levels in HRL to accelerate learning [12], [13], [14]. In our work, we believe our feedback mechanism (which is used to tune the meta-policy) being conditioned on the rollouts of the lower level policies might bring similar levels of inter-connections through the feedback signals instead of relying on explicit architecture modifications.

B. Goal-Conditioned Reinforcement Learning

GCRL has proven effective in addressing sparse reward problems by transforming traditional MDPs into goal-oriented MDPs [15]. This allows for the definition of reward functions based on the agent's proximity to the desired goal. Various approaches within GCRL, such as Hindsight Experience Replay (HER), have been proposed to enhance learning efficiency by relabeling failed trajectories as successful ones towards different goals. For instance, in a "FetchPush" task, even if the robot fails to push a block to the exact target position, HER can relabel this trajectory as successful for reaching the final block position. This creates a denser reward signal and facilitates learning.

C. Reinforcement Learning from Human Feedback

Reinforcement Learning from Human Feedback (RLHF) has gained significant attention recently. RLHF can expedite

AI training and enhance capabilities by utilizing human feedback. It learns reward functions from pairwise comparisons and rankings based on human preferences. While human intuition can guide high-level decision-making in HRL, humans may struggle to offer immediate guidance that aligns with the agent's capabilities.

D. Guided Cooperation in Hierarchical Reinforcement Learning

Previous works have also tried to allow collaboration between hierarchical levels in HRL to accelerate learning. For example, the Guided Cooperation in Hierarchical Reinforcement Learning via Model-based Rollout (GCMR) framework, uses a forward dynamics prediction model to bridge inter-layer information synchronization and cooperation [14].

In our work, we posit that our feedback mechanism, used to tune the meta-policy, being conditioned on the rollouts of the lower level policies might bring similar levels of inter-connections through the feedback signals instead of relying on explicit architecture modifications. By incorporating human feedback and considering the lower-level policy's performance, we aim to achieve a more efficient and robust HRL framework.

III. Method

This research proposes a novel HRL framework that incorporates human feedback into the high-level policy, utilizing trajectories from the low-level policy as additional input. This framework, which we refer to as Human-Guided Hierarchical Reinforcement Learning (HG-HRL), addresses the limitations of traditional HRL methods by:

A. Trajectory-Informed Feedback

Humans provide feedback on the subgoal selection (high level) based not only on the current state and desired goal but also on the trajectory taken by the low-level policy in attempting to achieve the subgoal. This ensures that the subgoals are both instructive and achievable. For example, in a "FetchPickAndPlace" task, the human might provide feedback on whether the subgoal "reach for the object" is appropriate and also on the specific trajectory taken by the robot arm to reach the object, considering potential collisions or inefficiencies.

To facilitate this, we will develop an interface that presents the human with the following information:

- Current state of the robot and the environment (e.g., visual representation, joint angles).
- Desired goal of the task (e.g., final position of the object).
- Proposed subgoal generated by the high-level policy.
- Trajectory executed by the low-level policy in attempting to achieve the subgoal.

The human can then provide feedback on the subgoal's appropriateness and the trajectory's efficiency, using modalities such as:

- Binary feedback: Approve or reject the subgoal.
- Ranking: Rank a set of proposed subgoals.
- Natural language instructions: Provide specific instructions or corrections.

B. Preference Learning

A reward model is trained based on human preferences. This model captures human intuition about desirable subgoals and low-level trajectories, guiding the agent towards more efficient task completion. The reward model can be formalized as a function $r_{hf}(s, sg, \tau, g)$, where s is the current state, sg is the subgoal, τ is the trajectory of the low-level policy, and g is the desired goal. The reward model is trained using pairwise comparisons of (s, sg, τ, g) tuples, where human annotators provide feedback on which tuple they prefer, taking into account both the subgoal's relevance and the trajectory's efficiency.

We will explore different machine learning techniques for training the reward model, such as:

- Supervised learning: Train a classifier to predict human preferences based on labeled data.
- Reinforcement learning: Train a reward function using reinforcement learning algorithms, where the reward signal is derived from human feedback.

C. Dynamic Difficulty Adjustment

The difficulty of subgoals is dynamically adjusted based on the performance of the low-level policy. This ensures that the agent is constantly challenged while preventing it from getting stuck on overly difficult subgoals. For example, in a "FetchPush" task, if the low-level policy consistently fails to reach a subgoal that is too far from the current block position, the difficulty adjustment mechanism will reduce the distance of future subgoals, making them easier to achieve. This dynamic adjustment allows the agent to gradually learn more complex manipulation skills.

We will investigate different approaches for dynamic difficulty adjustment, such as:

- Curriculum learning: Gradually increase the difficulty of subgoals as the agent's performance improves.
- Adaptive subgoal generation: Adjust the subgoal generation process based on the low-level policy's success rate.

By combining these three key components, our HG-HRL framework aims to leverage human expertise to enhance the efficiency, adaptability, and overall performance of HRL in complex robotic manipulation tasks.

Figure 1 illustrates the overall framework of our HG-HRL approach. The high-level policy, informed by the current state (s), the desired goal (g), and the rollout of the low-level policy, generates a subgoal (sg). The low-level policy then takes actions (a) in the environment ("Env") to achieve this subgoal. The human provides feedback on the subgoal and the rollout, which is used to train the reward model and adjust the subgoal generation process.

By combining these three key components, our HG-HRL framework aims to leverage human expertise to enhance the

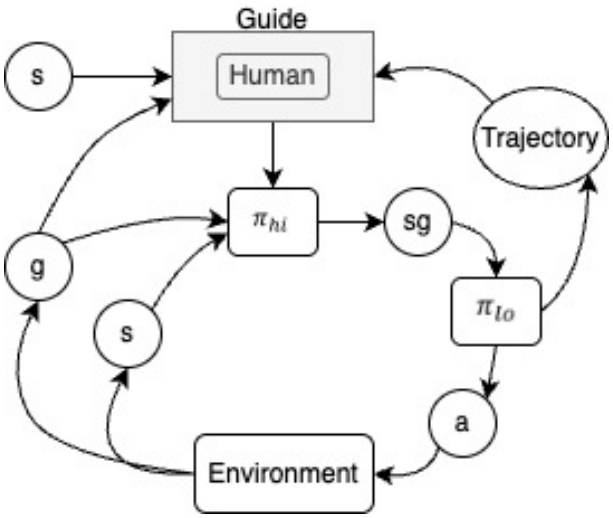

Fig. 1. Framework for Human-Guided Hierarchical Reinforcement Learning (HG-HRL)

efficiency, adaptability, and overall performance of HRL in complex robotic manipulation tasks.

IV. Experiment

In our experiments, we utilize the **FetchReach** environment, as shown in Figure 2, from the Gymnasium-Robotics suite as our testing ground. This environment serves as an ideal platform for evaluating the capabilities of our hierarchical reinforcement learning (HRL) framework, given its structured tasks and inherent complexity.

A. Hierarchical RL Framework

Our method adopts a **hierarchical RL structure** that consists of two interconnected levels of neural networks:

1) High-Level Neural Network (Subgoal Generator): The upper level employs a **feedforward neural network** to generate subgoals. The network takes as input the current state of the environment and the desired goal, and outputs an intermediate subgoal for the task. To enhance learning efficiency and adaptability, we introduce **human feedback** at this level.

- **Human Feedback Integration:** At certain intervals during training, the system presents two randomly selected action trajectories to a human operator. The operator provides feedback by choosing the preferred trajectory. This preference is incorporated into the network's reward function, allowing the model to adapt based on human intuition.
- **Reward Function:** The reward signal includes both traditional environmental feedback and the human preference score, which guides the high-level network towards generating more meaningful and achievable subgoals.

2) Low-Level Neural Network (Goal-Conditioned Actor-Critic): At the lower level, we use the **Deep Deterministic Policy Gradient (DDPG)** algorithm, which operates as an **actor-critic model**:

- The **actor network** takes the current state and the subgoal generated by the upper level as input and outputs the corresponding action.
- The **critic network** evaluates the actions generated by the actor, providing a score that influences the actor's policy updates.

To make the framework compatible with goal-conditioned learning, the DDPG algorithm has been modified to accept subgoals as part of its input. This enables the actor to execute actions that align with the overall task objective effectively.

B. Warm-Up Phase for Subgoal Generation

Given the complexity of generating meaningful subgoals in the initial stages of training, the high-level network undergoes a **warm-up phase**. During this phase, human feedback is collected continuously to help the model learn subgoal generation before passing control to the lower-level policy. This ensures that the initial subgoals are of sufficient quality to facilitate effective learning in the lower level.

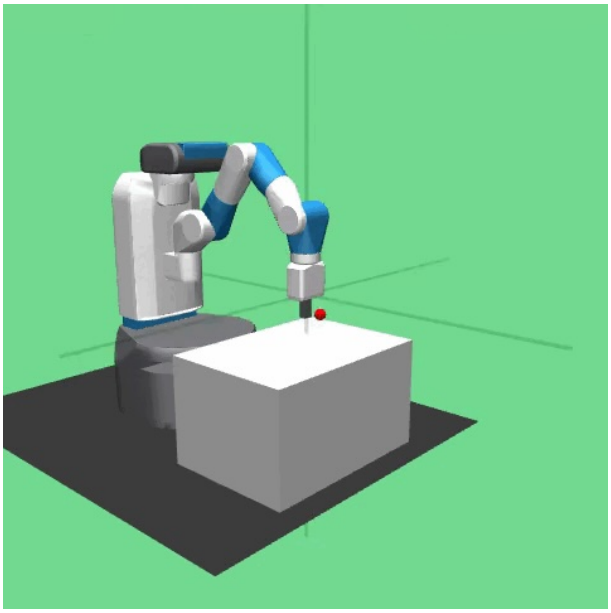

Fig. 2. Fetch-Reach Environment

V. Experimental Results

The FetchReach environment from Gymnasium-Robotics was selected as the testing ground to evaluate the hierarchical reinforcement learning (HRL) framework. This section outlines the progress achieved, challenges encountered, and the next steps required to complete the project.

A. Progress Achieved

1) Upper-Level Policy Implementation:

- The upper-level policy, a feedforward neural network for subgoal generation, was successfully implemented. The network takes the current state and the desired goal as inputs and generates subgoals to guide the lower-level policy.

- A mechanism to collect human feedback was integrated, wherein the system renders two different trajectories every five steps. The human user selects the preferred trajectory, and this feedback is incorporated into the training process.
- A loss function based on the Bradley-Terry algorithm was designed to process the user feedback. This loss helps the subgoal generator learn to produce subgoals that are both achievable and aligned with human preferences.

2) Subgoal Generation and Integration:

- The trained upper-level policy was used to generate subgoals, which were integrated into the observation space of the environment.
- The observation space was modified to replace the original desired goal with the subgoal generated by the upper-level policy.

3) Lower-Level Policy Implementation:

- The Deep Deterministic Policy Gradient (DDPG) algorithm was implemented as the lower-level policy. The actor network takes the state and subgoal as inputs and outputs actions, while the critic network evaluates the actions.
- Modifications were made to ensure the DDPG implementation was compatible with the subgoal-based observation space.

B. Challenges Encountered

- **Integration Complexity:** Combining the upper-level and lower-level policies into a cohesive framework introduced significant debugging challenges. Ensuring that subgoals generated by the upper-level policy align with the expectations of the lower-level policy required iterative refinement.
- **Computational Constraints:** The dual-level framework requires substantial computational resources for training, particularly with the inclusion of human feedback. Limited computing power significantly slowed down the experimentation process.
- **Training Difficulty:** Training the hierarchical framework in the FetchReach environment proved challenging due to its sparse reward structure and the complexity of the problem statement.

C. Next Steps

- Finalize the integration of the subgoal generator with the lower-level policy.
- Conduct extended training sessions under sufficient computational resources to fully evaluate the proposed framework.
- Test the framework on more complex environments, such as FetchPush or FetchPickAndPlace, to validate its generalizability.

VI. Evaluation

The performance of our proposed HRL framework will be rigorously evaluated based on the following metrics:

- **Task Success Rate:** This metric will measure the percentage of successful task completions across a variety of robotic manipulation tasks. We will compare the performance of our human-guided HRL approach to baseline HRL methods, such as HAC [9] and HIRO [10], to demonstrate the improvement in task completion achieved through incorporating human feedback.
- **Learning Efficiency:** We will track the learning progress of the robot over time by monitoring reward curves and task completion times. This will allow us to assess the efficiency of our approach in learning complex manipulation tasks. Our aim is to demonstrate that incorporating human feedback leads to faster and more effective learning compared to baseline HRL methods.
- **Subgoal Quality:** We will analyze the quality of the subgoals generated by the high-level policy under the influence of human feedback. This will involve assessing the relevance and effectiveness of these subgoals in guiding the robot towards achieving the overall task goal. We will compare the quality of these human-influenced subgoals to those selected by traditional HRL methods, which often struggle with generating instructive and achievable subgoals.
- **Human Feedback Efficiency:** We will evaluate the efficiency of human feedback by measuring the amount of feedback required to achieve a certain level of performance. This will help us understand the trade-off between human effort and robot learning efficiency.

By analyzing these metrics, we aim to demonstrate the effectiveness of our proposed human-guided HRL framework in improving the learning efficiency, adaptability, and overall performance of robots in complex manipulation tasks.

VII. Conclusion

This project explores the integration of hierarchical reinforcement learning (HRL) with human feedback in the **FetchReach** environment, aiming to address the challenges associated with sparse rewards and complex task decomposition. By adopting a two-level neural network framework, we incorporate human preferences into the high-level policy and employ a goal-conditioned Deep Deterministic Policy Gradient (DDPG) algorithm for the low-level policy. This innovative approach seeks to improve both the efficiency and adaptability of HRL in robotic manipulation tasks.

Although significant progress has been made in conceptualizing and implementing the framework, completing the project fully has proven challenging due to its computational intensity and the time constraints of the course. The project's complexity, particularly the integration of human feedback and the two-level hierarchy, required iterative adjustments and meticulous experimentation. These demands extended beyond the typical scope of a course project.

Despite these limitations, the preliminary stages have demonstrated the potential of human-guided subgoal generation in enhancing task success rates and learning efficiency. The introduction of a warm-up phase for the high-

level policy, during which human feedback is continuously utilized, lays the foundation for more effective subgoal generation. This design ensures that the system begins with a strong capability to guide the low-level policy towards task completion.

Moving forward, the framework's full potential could be realized through additional computational resources, longer training durations, and more extensive testing. Future work may involve scaling the approach to more complex environments, refining the feedback mechanism, and exploring real-world applications of this framework in robotic systems. The project has highlighted the importance of combining human intuition with machine learning algorithms to tackle intricate problems, particularly in domains where direct supervision or task decomposition is critical.

In conclusion, while the project remains incomplete due to practical constraints, it provides a valuable foundation for future research in hierarchical reinforcement learning. The integration of human feedback represents a promising direction for advancing the field, particularly in addressing sparse rewards and improving the interpretability and efficiency of reinforcement learning models in robotics. This work serves as a stepping stone for further exploration, underscoring the potential of human-augmented machine learning in tackling challenging tasks.

REFERENCES

[1] A. S. Vezhnevets, S. Osindero, T. Schaul, N. Heess, M. Jaderberg, D. Silver, and K. Kavukcuoglu, "Feudal networks for hierarchical reinforcement learning," in *International conference on machine learning*. PMLR, 2017, pp. 3540–3549.

[2] O. Nachum, S. Gu, H. Lee, and S. Levine, "Data-efficient hierarchical reinforcement learning," in *Advances in Neural Information Processing Systems*, 2018, pp. 3303–3313.

[3] A. Levy, R. Platt, and K. Saenko, "Hierarchical reinforcement learning with hindsight," in *International Conference on Learning Representations*, 2019.

[4] R. S. Sutton, D. Precup, and S. Singh, "Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning," in *Artificial intelligence*, vol. 112, no. 1-2. Elsevier, 1999, pp. 181–211.

[5] D. Precup, "Temporal abstraction in reinforcement learning," in *icml*, 2000.

[6] T. D. Kulkarni, K. Narasimhan, A. Saeedi, and J. B. Tenenbaum, "Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation," *Advances in neural information processing systems*, vol. 29, 2016.

[7] A. Sharma, A. Suresh, R. Ramesh, A. Rawal, L. Pinto, D. Kalashnikov, A. Kumar, O. Rybkin, A. Trischler, Gülçehre *et al.*, "Dynamics-aware unsupervised discovery of skills," in *International Conference on Learning Representations*, 2019.

[8] B. Eysenbach, A. Gupta, J. Ibarz, and S. Levine, "Diversity is all you need: Learning skills without a reward function," in *International Conference on Learning Representations*, 2019.

[9] A. Levy, G. Konidaris, R. Platt, and K. Saenko, "Hierarchical actor-critic," in *International Conference on Learning Representations*, 2017.

[10] O. Nachum, S. Gu, H. Lee, and S. Levine, "Hiro: Hierarchical reinforcement learning with off-policy correction," in *International Conference on Learning Representations*, 2018.

[11] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke *et al.*, "Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation," *arXiv preprint arXiv:1806.10293*, 2018.

[12] W. Fedus, S. Gu, V. Zambaldi, K. Hofmann, H. Lee, and S. Levine, "Hyperbolic hierarchical reinforcement learning," in *Advances in Neural Information Processing Systems*, 2019, pp. 8998–9009.

[13] F. Christianos, L. Lechner, M. Geist, and J. Peters, "Shared experience actor-critic for multi-agent reinforcement learning," in *Advances in Neural Information Processing Systems*, 2020, pp. 10 661–10 672.

[14] H. Wang, Z. Tang, L. Yang, Y. Sun, F. Wang, S. Zhang, and Y. Chen, "Guided cooperation in hierarchical reinforcement learning via model-based rollout," *arXiv preprint arXiv:2309.13508*, 2023.

[15] T. Schaul, D. Horgan, K. Gregor, and D. Silver, "Universal value function approximators," *International conference on machine learning*, pp. 1312–1320, 2015.