Increasing Legibility During Learning Through Policy Shaping

Anisha Bhaskar Torres
University of Texas at Austin
anishab@utexas.edu

Elaine Short
Tufts University
elaine.short@tufts.edu

***Abstract*—In this paper, we build upon previous research in increasing legibility of robot movement and enhancing the efficiency of Q-learning with policy shaping. We modified a traditional Q-learning algorithm implemented on a robot with the following objectives: to make it more apparent to observers whether the robot was exploring or exploiting during the learning process, and given that it was exploiting, to highlight how many goals the robot had knowledge of by the trajectory chosen. After evaluation, we found that in addition to dramatically increasing average rewards over time in a large state space, the modified algorithm achieved both objectives in tasks where possible goals were close to each other or far apart. In fact, the modified algorithm performed drastically better than the original for both objectives in the case of a large grid size. The success of the modified algorithm in the largest of our chosen grid sizes and the computationally complex nature of the algorithm suggests that its most appropriate use case is large state spaces. Future work will extend these objectives to explore how legibility can be considered in the design of learning algorithms to increase transparency to non-technical human observers during the learning process.**

I. Introduction

With greater interactions between humans and robots arises a need for greater transparency in the goal of robot movement, otherwise known as legibility. Despite the substantial body of research in the study of robot movement, there remains much to be discovered in the intersection of robot learning and legibility of movement. This project focused on using policy shaping to enhance the legibility of robot movement during the Q-learning process. To do so, we identified two primary goals for our algorithm: 1. To choose actions during learning such that observers would know whether the robot was exploring or exploiting based on the legibility of those actions. For example, exploiting would lead to actions with increased legibility, while exploring would result in actions with decreased legibility. 2. To choose actions when exploiting to showcase how many goals the robot had previously visited. In other words, knowledge of one goal would be significantly less legible than knowledge of both goals. After evaluation, we found that in addition to accelerating learning, the modified algorithm achieved the former target, with significant differences from the original algorithm. However, the modified algorithm was only able to meet these objectives with grid sizes 3 and 9, and only with the classic and between tasks. In summary, this work begins to explore how legibility and reinforcement learning, two major fields in human-robot interaction research, can be combined to make the learning process more transparent to observers.

II. Related Work

A number of studies have been done to improve upon both policy shaping and the legibility of robot movement. After the development of policy shaping with humans giving feedback as a technique to improve Q-learning, its effectiveness has been tested in multiple ways, and several modifications have been made to further increase the robustness of the learning algorithm in a variety of situations. For instance, researchers have evaluated and engineered policy shaping algorithms to optimize the effectiveness of feedback in many cases, including with infrequent feedback [3] [7], conflicting feedback [12] [7], meaningful silence [3], or noisy feedback [12]. It has also been extended for optimal use with multi-tasking human teachers as in [6] [5], adapting policy shaping to better fit real world contexts.

In the field of legibility, the importance of the transparency of robot movement in interactions with humans has been widely established, with researchers identifying the key distinction between legibility and predictability, and their opposing roles in human-robot interaction [1]. Studies have not only highlighted the importance of legible rather than predictable motion in effective human-robot collaboration [4], but also in making robots more socially acceptable to humans [8]. Having recognized the value of legibility in human-robot interaction, various algorithms have been developed to increase the legibility of robot movements in different contexts. These situations include cluttered and uncluttered environments [11], and balancing trade-offs with optimal trajectories [1].

While extensive work exists within both fields, previous work combining Q-learning with legibility has been limited. There exist only a few studies on how to make the agent learn a more legible goal trajectory at the end of the learning process, whether it be by regularizing the learned policy of a Q-network [10], or by implementing reward shaping in a Q-learning algorithm [2] [9]. However, at the time of writing, no studies exist on increasing legibility during the learning process.

III. Background

A. Calculating Legibility

Previous work has defined legibility as the extent to which a human observer can discern the intention of a robot, given that the robot has a desired goal [1]. In our modified algorithm, we calculate legibility scores for each state in our path, as they represent points within a trajectory. The formula from [1] was simplified to fit a discrete trajectory and was not weighted based on whether it was early or late in the trajectory. The

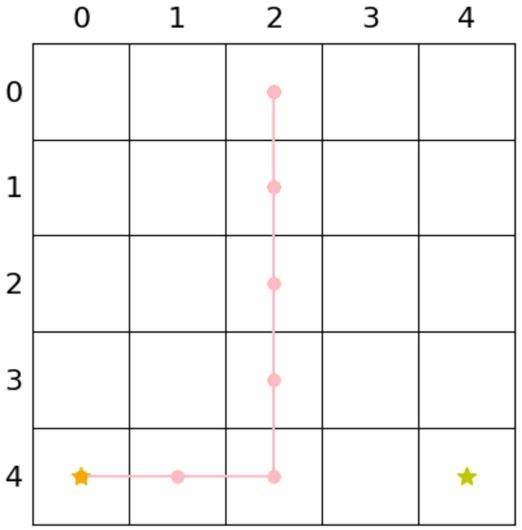

Fig. 1. The least legible optimal path for Classic 5

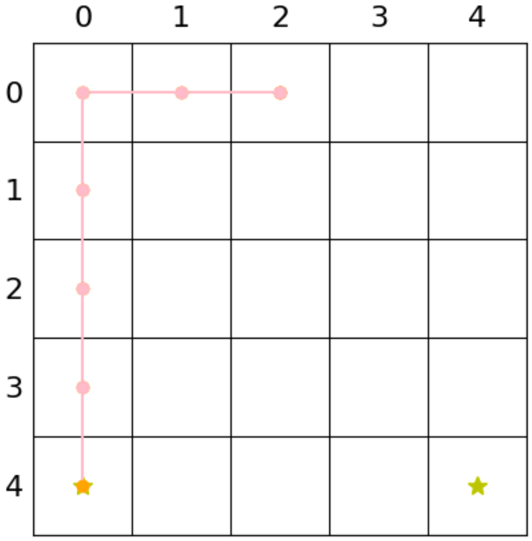

Fig. 2. The most legible optimal path for Classic 5

modified formula uses costs of trajectories to calculate the legibility score for a state. The cost of the trajectory, ξ, from the current state, *X*, to the end state, *E*, can be written as $C(\xi_{X\to E})$. *S'* in the formula below stands for the start state in the trajectory. With these variables defined, the legibility score for a particular state is calculated as follows:

$$\frac{\exp(-C(\xi_{S'\to X}) - C(\xi_{X\to E})}{\exp(-C(\xi_{S'\to E})} P(G)$$

To demonstrate this idea of legibility visually, Figure 1 shows the result of a normal Q-learning algorithm, in which either goal is likely, and the intended goal becomes clear only later in the trajectory. On the other hand, Figure 2 displays a more legible trajectory, with the intended goal being apparent early on.

B. MDP and Policy Shaping

A Markov Decision Process (MDP) can be described with the following tuple: (*S, A, T, R,* γ). The learning environment is composed of states, *S*, where each state-action pair maps to a new state in the transition function $T : S \times A \to$ *P(S)*, and *P(S)* is the probability distribution of the next state. The reward function, *R*, describes how each state-action pair also maps to a reward, where $R : S \times A \to R$. In the context of Q-learning, the discount factor, γ, where $0 \leq \gamma \leq 1$, represents the rate at which previous Q-values are discounted from each updated Q-value calculation. The Q-value for a particular state-action pair, *Q(s, a)*, when $s \in S$ and $a \in A$, is the expected total future reward that can be achieved by following the optimal policy from the current state-action pair to the end of the episode.

During traditional Q-learning, exploitation involves choosing the state-action pair with the highest expected returns for the current state. Policy shaping modifies this such that during exploitation, feedback from an external source is combined with the current Q-value, and the most probable action is chosen from there. The external feedback has no effect on the Q-table, instead influencing the policy chosen by the robot.

IV. Domain

To test the modified algorithm in a variety of settings, we chose to use 3 different tasks, and 3 different sizes of square grids. Each of these tasks had a certain square on the grid where the robot began its learning each episode, and each grid had 2 ”goals”, or squares with rewards of 10 points. For the ”classic” task, the start square was placed in the top middle square of the grid, and the goals were placed on opposite sides of the bottom end, as shown in Figure 3. The ”behind” task, as shown in Figure 4, had the start square at the top left of the grid, and the 2 rewards in the bottom right of the grid. Lastly, the ”between” task setup was the start state in the middle of the grid, with one goal in the top left and the other in the bottom right, as in Figure 5.

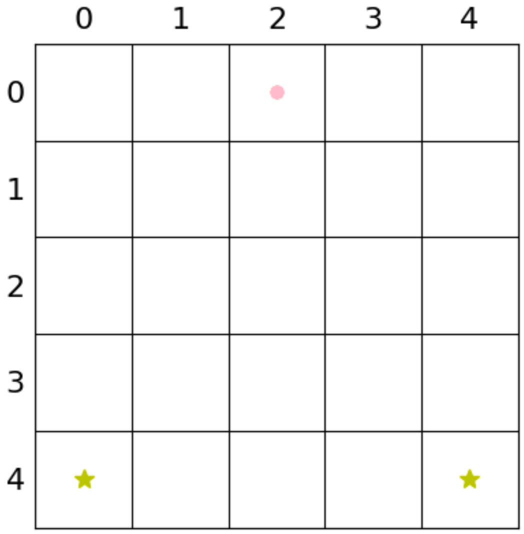

Fig. 3. Classic 5

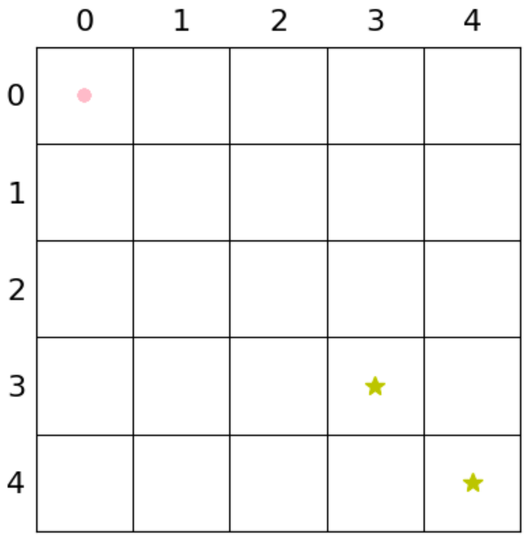

Fig. 4. Behind 5

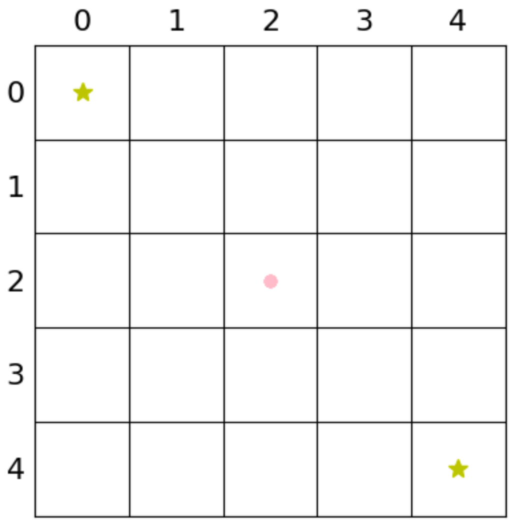

Fig. 5. Between 5

V. Implementation

The main modifications to the algorithm came from the use of legibility scores. One of the main assumptions of legibility is that the robot has a goal it is attempting to reach, which contrasts fundamentally with the random nature of Q-learning. To mitigate this, the learning process begins with the robot knowing nothing about the goals, as in a regular Q-learning algorithm. When the robot visits a new goal, the algorithm adds it to a list of goals the robot has been to. Each goal has separate Q-tables for each one so that learned trajectories for one goal do not interfere with learned trajectories for another, but all Q-tables are updated each time the robot takes a step. Once the robot knows of at least one goal, the robot will choose one randomly from its list of goals for the episode. After the robot has visited 2 or more goals, legibility scores become relevant for the robot's next movement.

A. Legibility Calculations

From the formula described in the background, the legibility score at a certain state requires 2 pieces of information: the number of steps taken up to the current state, and the optimal steps needed from the current state to the goal state. Retrieving and storing the information could have been done multiple ways to the same end. Our project used a feedback table that stored the next state for each action for all states, and an optimal steps table that stored the optimal steps from each state to the end. Each time it was necessary to choose the next action from a certain state, the algorithm would loop through possible actions. From there, the feedback table would return the resulting state from each action given the current state. This state would be used by the optimal steps table to return the optimal steps, which was used along with the current number of steps taken to calculate the legibility score for the state. Using both tables rather than one resulted in the optimal steps table being easier to read and debug, but the runtime and memory use could have benefited by combining the feedback and optimal steps tables into one. In either manner, the optimal steps table would still be calculated and updated the same way.

B. Optimal Steps Table

One of the most interesting aspects of this research project was maximizing information acquired from learning in the shortest amount of time to have the most accurate calculations for legibility scores. The optimal steps managed the majority of this by propagating information forwards and backwards when needed. For example, as the robot would move through the grid, it would store its past states in a queue. After arriving at a goal, all of the states in the queue would be updated with the number of steps to the end in backwards propagation. Additionally, if the robot had visited a state with known optimal steps value and later visited a new state with an unknown value, the robot would use the known value to propagate information forward to the new state.

C. Exploration vs. Exploitation

During exploitation, the robot's choosing of an action is shaped by the legibility score for the next state through policy shaping, if the score is able to be calculated. The algorithm below represents the method used to choose a state-action pair when exploiting. A detail to note is that we chose to multiply the probability of each action by the sigmoid function $\frac{1}{1+e^x}$ rather than e^x for more balanced probabilities.

In addition to policy shaping with legibility scores during exploitation, the algorithm also has modifications to decrease the legibility in the exploration phase. During exploration, the robot handles its next action prioritizing exploring unknown feedback states, and then states with unknown optimal steps. If all current possible state-action pairs have known resulting states, and all resulting states have a known number of optimal steps to the end, then the robot explores based on minimum legibility. In other words, the robot chooses the state-action pair with the least legibility as the next point in its trajectory. In this way, based on the trajectory of the robot, it will be

Algorithm 1 Policy Shaping With Legibility Scores

```
default legib ← $P(G)$
$s \in S$
for $a \in (s \times a)$ do
    prob action ← default legib
    if $C(\xi_{s \to E})$ != none then
        prob action ←
            $\frac{\exp(-C(\xi_{S' \to s}) - C(\xi_{s \to E})}{\exp(-C(\xi_{S' \to E})} P(G)$
        prob action ← prob action$\times \frac{1}{1+\exp(qtablevalue)}$
    end if
end for
```

apparent to human observers whether the robot is exploring and exploiting, and what goals the robot has knowledge of.

D. Experimental Design

Each algorithm was run for 150 learning periods with each grid size-task combination. The following parameters were kept constant for all combinations: exploration rate = 0.7, discount factor = 0.85, learning rate = 0.7. Exponential decay rates were chosen for each algorithm in accordance with the best performance for the classic task, minimizing the average number of episodes to reach both goals, and maximizing the average reward over time at the end of the learning period. The following parameters were used to run the tests.

Grid Size	Episodes	Steps	Orig Exp Decay	Mod Exp Decay
3	50	25	0.025	0.05
5	100	50	0.05	0.05
9	200	100	0	0.01

VI. Results

The following results have a statistical confidence of 95%.

A. Legibility During Exploitation

This test measured the differences in the legibility score of the overall trajectory after having been to one goal subtracted from the score after having been to both goals. There was no statistically significant difference with any grid size for the behind task. This was most likely because of the location of the second reward behind the first reward made it very difficult for both algorithms to reach the second reward. The highest differences between the average legibility changes of the overall trajectories of the original and modified algorithms were seen with the classic task out of all tasks, and with the grid size of 9. All differences shown in the graph had a margin of error of at most 0.1. The greatest difference in legibility score was for classic 9, with a 0.4 difference between the original and the modified algorithm. This suggests that the modified algorithm can be more useful than the original Q-learning algorithm in exploring large state spaces in a more transparent manner, but may be less useful in smaller spaces.

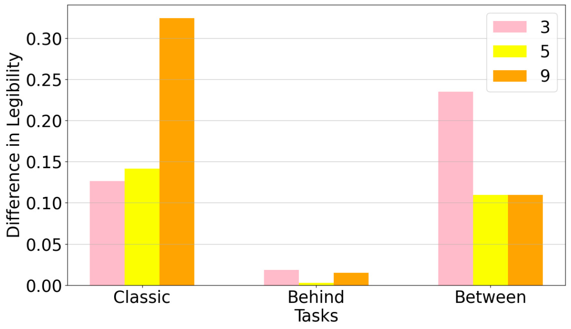

Fig. 6. Differences in legibility knowing 1 vs 2 goals for the original and modified algorithms

B. Legibility During Exploration vs Exploitation

To measure whether we satisfied our goal of increasing legibility during exploitation and decreasing legibility during exploration, we first calculated the legibility scores for each point in the learning agent's trajectory. We then divided this value by the total steps in all trajectories to calculate the average legibility score for the entire learning phase for each algorithm, and compared the results. The task where the modified algorithm performed the best was the classic task, with an increase of about .12 for classic 3, .13 for classic 5, and .33 for classic 9 compared to the original algorithm and a margin of error for all of .03 or less. The modified algorithm did well in the between task, but not as well as in the classic because the original task was already legible in the between task. Again, the behind task had no statistically significant difference, most likely because it was difficult for the modified algorithm to move legibly when one reward was behind the other. Surprisingly, the grid size with the worst performance was the 5 x 5 grid. This may be because the original algorithm performed the best with the medium sized grid.

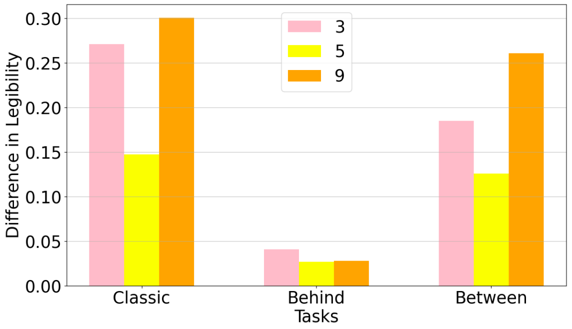

Fig. 7. Differences in exploration vs exploitation legibility of the original and modified algorithms

C. Learning Efficiency

In addition to comparing legibility scores, we wanted to investigate the effects of the modifications on the efficiency of learning.

1) Average Episode All Goals Found: The average episode in which both rewards were found was significantly better for classic 3 and classic 9. The modified algorithm took about 30% less time than the original algorithm did to find both tasks in classic 3, with a margin of error of 20%. The modified algorithm performed similarly with classic 5, taking about 30% less time with a margin of error of 10%. The modified algorithm actually took longer to find both rewards for between 5, and had no significant difference for classic 5. These results suggest that the modified algorithm allows for more efficient exploration of the learning environment in small and large environments.

2) Average Rewards Over Time: To quantify a measure of learning over time, we split each learning period into 5 intervals, and averaged the reward gained every 20% interval for 5 data points. Earlier data was more variable in many cases, while later data was more stable. Below are graphs displaying the average rewards over time for classic tasks of different grid sizes.

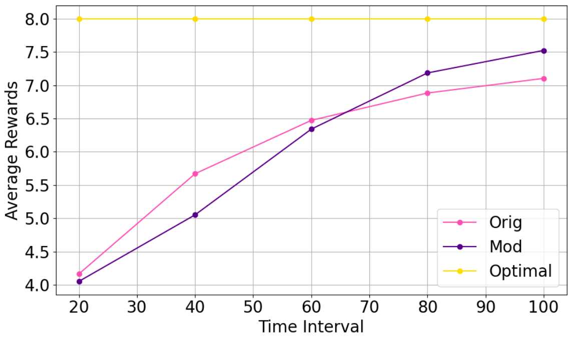

Fig. 8. Classic 3

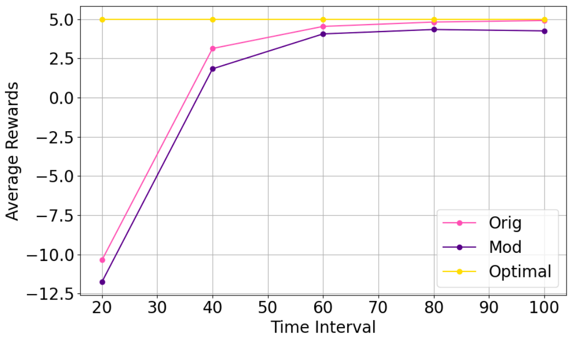

Fig. 9. Classic 5

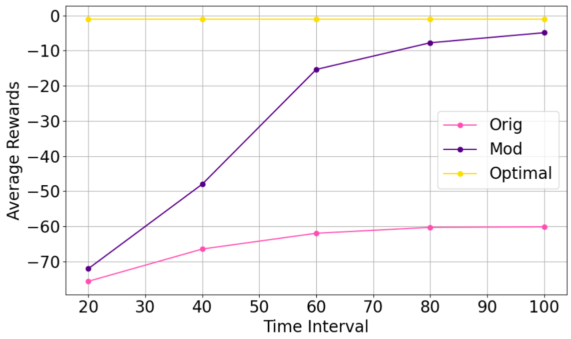

Fig. 10. Classic 9

As shown in the figures above, there was no significant difference in the average rewards over time between the original and modified algorithms for classic 3 and 5. This pattern continued for the behind and between tasks for grids 3 and 5. However, with a grid size of 9, the average reward over time increased significantly, with the last 80% of episodes for a 9 x 9 grid. Given that all goals were found earlier for every grid size and task type, a continuation of this pattern would have been that there would be differences in the average reward over time for all grid sizes and task types. That was not the case. This may be due to the legibility-focused exploitation resulting in less optimal trajectories in smaller state spaces.

VII. Discussion

Our modification of the Q-learning algorithm focused on increasing legibility during the learning process through policy shaping. As the learning agent explored, the algorithm gained more information about the environment. A natural consequence of this information was that the robot became more legible over time, which showcased the number of goals the learning agent had knowledge of, one of our primary goals. Additionally, the algorithm attempted to increase the difference in legibility between when the robot was exploring and exploiting to increase transparency about current learning phase to human observers. Both of these objectives were met only when grid sizes were 3 and 9, and only with classic and between tasks. Though in most cases the modified algorithm was able to find both goals faster, the modified algorithm was not much different from the original algorithm in the average rewards over time for most cases. Overall, the modified algorithm performed the best in comparison to the original algorithm with the grid size of 9, indicating that the algorithm is best suited to large environments.

As such, this work has several limitations and areas for exploration. One primary limitation of this work is the fact that it is much more memory intensive than a regular Q-learning algorithm in storing past states, optimal steps, multiple Q-tables, and more. This characteristic makes the algorithm unsuitable for use in small environments. Additionally, calculating legibility scores is not the most ideal way to evaluate whether the robot's intentions are obvious to human viewers. At times, especially with large grid sizes, the legibility of states on the furthest columns of the grid end were calculated to be very high. Even if the robot moved in the direction of the other reward, the legibility of the new state would stay high because the robot would still be in the area of high legibility. Realistically, the true legibility of this action would be much lower, so calculating legibility without asking human observers is not a completely realistic measure. Future work could incorporate direction into legibility measures, as well as human feedback during the policy shaping period and evaluation. Another limitation of this work is the small scope of the problem. We observed how the modified algorithm had the greatest change with the largest grid tested; it would be interesting to investigate whether this extends infinitely on to larger and larger state spaces, or if performance declines at a certain point. More generally, other areas of study could include applying legibility to different types of reinforcement learning, an area that has lots of potential for exploration.

VIII. Conclusion

As reinforcement learning techniques continue to be developed for use in environments with increasing human-robot interaction, there arises a need for more transparency to non-technical humans during learning processes. To address this, the purpose of our work was to design a modified Q-learning algorithm to increase transparency to humans during the learning process. This was achieved by attempting to increase the gap between the legibility scores of the robot's trajectory during exploration and exploitation, and when only one goal was known versus both of them. Our results show that our algorithm was most successful compared to the original in the cases of the grid size of 9, in both accelerating average rewards over time, as well as in achieving our two targets. All in all, our work represents a start in investigating how legibility and reinforcement learning can be used for robot learning in human-robot interactions.

Acknowledgments

This work is supported by the Directed Research Experiences for Undergraduates (DREU) of the Computing Research Association (CRA). Thank you to members of the AABL Lab at Tufts University for their assistance and feedback.

References

[1] K. Lee A. Dragan and S. Srinivasa. Legibility and predictability of robot motion. In *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 301–308, 2013.

[2] M. Bied and M. Chetouani. Integrating an observer in interactive reinforcement learning to learn legible trajectories. In *2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pages 760–767, Naples, Italy, August 2020. hal-02984877.

[3] T. Cederborg, I. Grover, C. Isbell, and A. Thomaz. Policy shaping with human teachers. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, page 3366–3372. AAAI Press, 2015.

[4] A. Dragan, S. Bauman, J. Forlizzi, and S. Srinivasa. Effects of robot motion on human-robot collaboration. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*, pages 51–58, 2015.

[5] T. Faulkner, R. Gutierrez, E. Short, G. Hoffman, and A. Thomaz. Active attention-modified policy shaping: socially interactive agents track. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pages 728–736, 2019.

[6] T. Faulkner, E. Short, and A. Thomaz. Policy shaping with supervisory attention driven exploration. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 842–847, 2018.

[7] S. Griffith, K. Subramanian, J. Scholz, C. Isbelle, and A. Thomaz. Policy shaping: integrating human feedback with reinforcement learning. In *NIPS'13: Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, pages 2625–2633, 2013.

[8] N. Hetherington, E. Croft, and M. Van der Loos. Hey robot, which way are you going? nonverbal motion legibility cues for human-robot spatial interaction. In *IEEE Robotics and Automation Letters, Vol. 6, No. 3*, pages 5010–5015, 2021.

[9] Y. Liu, Y. Zeng, B. Ma, Y. Pan, H. Gao, and X. Huang. Improvement and evaluation of the policy legibility in reinforcement learning. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, pages 3044–3046, 2023.

[10] M. Persiania and T. Hellström. Policy regularization for legible behavior. In *Neural Computing and Applications, Vol. 35 (2023)*, pages 16781–16790, 2022.

[11] M. Schmidt-Wolf, T. Becker, D. Oliva, M. Nicolescu, and D. Feil-Seifer. Through the clutter: Exploring the impact of complex environments on the legibility of robot motion. In *arXiv preprint arXiv:2406.00119*, 2024.

[12] T. Wei, T. Faulkner, and A. Thomaz. Policy shaping in continuous state spaces (student abstract). In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.