

Better Apprenticeship Learning with LLM Explanations

Ryanaa Grover, Aryan Vats, Nina Moorman, Aviral Agrawal, Matthew Gombolay

Georgia Institute of Technology
 rgrover30@gatech.edu, avats31@gatech.edu, nmoorman3@gatech.edu, aagrwal433@gatech.edu,
 matthew.gombolay@cc.gatech.edu

Abstract

As the population ages, care robots will play an increasing role in assisting caregiving by taking on repetitive or physically cumbersome activities. To effectively provide care, robotic agents must be able to meet the needs and preferences of care receivers. However, these needs and preferences may change over time, making it intractable to pre-define the way the care robot should act before deployment. Instead, the care robot should be able to learn directly from non-expert end-user demonstrations. However, prior work investigating the feasibility of learning a policy from older adult demonstrations finds that older adult demonstrators desire a better understanding of what the robot needs them to do, and how. To help demonstrators understand how to improve on suboptimal or heterogeneous demonstrations, we propose to utilize a Large Language Model to provide human-interpretable explanations of Shapley values of a policy. These explanations enable the demonstrator to understand how the policy is performing, and what changes are needed, informing their corrective demonstrations. We showcase our framework’s performance in deterministic and stochastic versions of Wumpus World.

Introduction

As the world’s population ages (WHOe 2023), care robots are increasingly being used to assist caregivers and care receivers with repetitive and physically cumbersome tasks (Thomaz 2023). For care robots to effectively assist in providing care, they must be able to meet the needs and preferences of their end users. As needs and preferences may change over time, as the care receiver ages or their environment is altered, it is intractable to pre-define prior to robot deployment what tasks should be executed and how.

Instead, the agent should have the capability to learn once it is deployed, for instance by learning from demonstrations given by the end users directly. For end users to provide effective demonstrations to the robot, they must demonstrate the behavior to a certain degree of optimality for accurate reward inference and behavior learning (Zhifei and Joo 2012). However, humans are not perfect demonstrators, they are often biased in systematic ways (Shah et al. 2019; Schrum et al. 2022), and require guidance when pro-

viding hierarchical demonstrations (reflective of the complexity of caregiving tasks) (Gopalan et al. 2022; Moorman et al. 2023b). Investigating the feasibility of learning a policy from older adult demonstrations, Hedlund-Botti et al. find that older adult demonstrators desire a better understanding of what the robot needs them to do, and how (Hedlund-Botti et al. 2024; Hedlund-Botti and Gombolay 2023). Improving model transparency can help guide the demonstrator in providing demonstrations that accurately represent the desired behavior.

One common method to provide users with an understanding of an agent’s policy is by leveraging feature importance analysis such as Shapley values, which explain individual model predictions by calculating the contribution of each feature to the prediction. As language explanations have been found to be preferred and to result in higher task performance compared to baselines such as feature importance maps and decision tree explanations (Silva et al. 2024), our work leverages an Large Language Model (LLM) to provide natural language explanations of Shapley values to demonstrators. Prior work has investigated leveraging LLMs with Explainable AI (XAI) modules to convey model rationale in a more human-understandable manner, such as in the context of anomaly detection (Ali and Kostakos 2023; Khediri et al. 2024). Huang et al. utilize policy summarization to explain a reward function (Huang et al. 2018). Similarly, Zhang et al. combine Shapley values and LLMs to increase transparency in MLC-based decision-making (Zhang and Chen 2024). However, to the author’s knowledge, prior work has yet to investigate leveraging LLM-based explanations of a model’s Shapley values to improve the demonstrator’s corrective demonstrations. By doing so, our system enables non-expert demonstrators to correct gaps in the robot’s learned policy.

In this work, we contribute the following:

1. We develop a framework that guides non-expert demonstrators in providing corrective demonstrations using LLM-generated explanations of Shapley values, contextualized in the environment and task.
2. We conduct a proof-of-concept pilot study to evaluate our pipeline and obtain initial results of LLM explanation validity, prior to deployment with the target population.

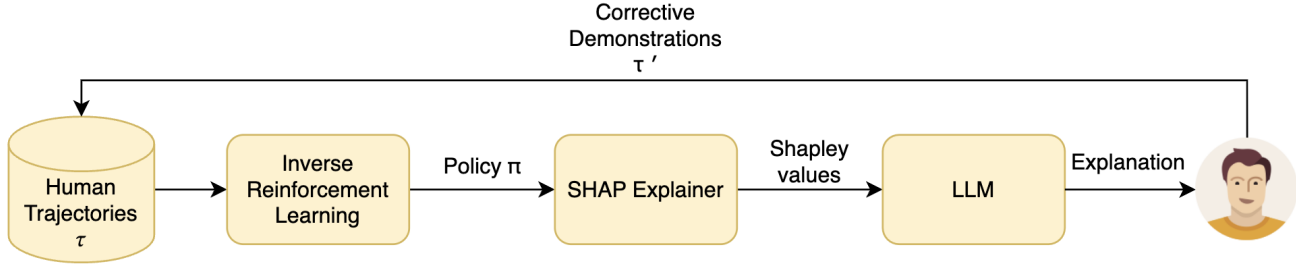


Figure 1: An overview of the proposed approach, where we leverage LLM-based explanations of the model’s feature importance to inform the demonstrator’s subsequent corrective demonstrations.

3. We propose a user study design to evaluate whether our system enables non-expert demonstrators to correct the robot’s learned policy.

Background

In this section, we discuss preliminaries related to feature importance analysis and inverse reinforcement learning.

SHapley Additive exPlanations

For a system to be interpretable, its decisions and actions must be human-understandable (Rodríguez-Lera et al. 2024). Machine learning models can either be inherently interpretable or can be post-hoc interpretable (Molnar 2018; Alvanpour et al. 2020). Some structured models, like decision trees, have self-explaining architectures where the weights associated with each feature indicate feature importance (Mahbooba et al. 2021). However, many other ML models have complex parameter spaces and are considered black-box models (Castelvecchi 2016). Post-hoc interpretability of black-box systems can be obtained with XAI techniques that use feature attribution to understand the relationship between inputs and outputs of a learned model.

SHapley Additive exPlanations (SHAP) is a model-agnostic method for explaining individual predictions by calculating the contribution of each feature to the prediction, denoted as Shapley values (Lundberg and Lee 2017). More formally, for a prediction $f(x)$ where x is the input feature vector, the Shapley value ϕ_i for feature i is calculated as the weighted average of the marginal contributions over all possible subsets of features, where the weight is given by the number of ways the subset can occur in the coalition. In robotics, SHAP has been used for interpretability to investigate robot failures and autonomous behaviors (Alvanpour et al. 2020; Gavriilidis et al. 2023). By computing Shapley values for each feature, we obtain an understanding of feature importance in model predictions, providing insights into the model’s decision-making.

Inverse Reinforcement Learning Formulation

Maximum Entropy Inverse Reinforcement Learning (MaxEnt IRL) as introduced by Ziebart (Ziebart et al. 2008) extends traditional Inverse Reinforcement Learning methods

by incorporating the principle of maximum entropy. Given a set of expert trajectories τ_{expert} , MaxEnt IRL seeks to find a reward function that not only explains the observed behavior but also maximizes the entropy of the agent’s policy. Building upon MaxEnt IRL, Maximum Entropy Deep Inverse Reinforcement Learning (Deep MaxEnt IRL) leverages deep neural networks to learn complex reward functions directly from demonstration (Wulfmeier, Ondruska, and Posner 2015).

Methodology

In this section, we introduce the framework’s core components and proposed user study procedure.

Framework Components

Our framework is composed of three modules.

Interpretability Module The interpretability module calculates Shapley values for the learned reward network, obtained via Deep MaxEnt IRL on an initial demonstration dataset. We use a feed-forward neural network to model the reward function, with three linear layers with Rectified Linear Unit (ReLU) activation. All the hyperparameters used are listed in the Appendix.

Contextualization Module The contextualization module takes in the Shapley values and outputs a natural language explanation that is more understandable for the demonstrator. This explanation enables demonstrators to evaluate whether the agent has learned the intended reward or is encountering issues related to specific components of the reward learning. This helps demonstrators identify actionable changes to their demonstration set, via corrective demonstrations. This component is key to enabling older adults and members of their care network to be actively involved in shaping the robot’s behavior, thus supporting aging in place.

In our experiments, we use Google DeepMind’s Gemini 1.5 Flash. The prompt to the LLM contains a manual describing the action space and observation space, and a detailed description of the features and their relevance to the problem, as depicted in Figure 2a. The mean and standard deviation of the Shapley values are additionally passed in

as part of the LLM’s prompt. The explanation returned describes (1) the meaning of each feature, (2) what the Shapley value of that feature indicates about the model, and (3) recommendations regarding how the demonstrator may wish to change their demonstrations, as shown in Figure 2b.

Corrective Feedback Module Similar to prior works such as Reciprocal MIND MELD, we provide feedback to a demonstrator in order to improve their demonstrations over the long-term (Schrum, Hedlund-Botti, and Gombolay 2022). Specifically, if, upon receiving the explanation, the demonstrator wishes to improve the policy, they have the option to provide additional five corrective demonstrations. To emphasize the significance of the new demonstrations, we adopt a sampling approach to construct the new training set. All five corrective demonstrations are incorporated into the new training set, with the remaining ten demonstrations sampled from the existing pool. Subsequently, the model is trained on this new demonstration set.

In related prior work, Reward Verification and Learning using Explanations (REVEALE) (Mahmud et al. 2023), update and verify the learned reward function to ensure alignment with the demonstrator’s intent. However, in the verification phase of REVEALE, when the agent fails, it queries the demonstrator by presenting additional explanations from alternative candidate reward models, some of which may not accurately represent the correct reward model. Instead, this work communicates information about the current policy’s feature importance and queries additional human feedback.

Procedure

We propose the following user study procedure. First, we collect fifteen demonstrations (human trajectories for the given environment) from the demonstrator. Once complete, we train the reward network on these trajectories. Next, the Shapley values along with context grounding are passed to the LLM, which outputs an explanation that is then provided to the demonstrator. The demonstrator is additionally shown the performance of the agent through model evaluation metrics (such as mean reward over 1000 trials) and can choose to watch the agent play the game. The demonstrator can then choose to provide additional feedback to supplement the previous demonstrations or correct errors they have identified in the previous demonstration set. We then make a demonstration dataset that combines the five new demonstrations and obtains the remaining ten demonstrations by sampling from the previous demonstration set. We iterate over this process until the demonstrator communicates that they are satisfied with the agent’s performance. Our approach is depicted in Figure 1, and the process is more formally defined in Algorithm 1.

Assumptions Our system makes the following assumptions. First, we expect to have knowledge about the reward function features a priori. This is to say that, in order to yield actionable suggestions from the LLM, we assume that we know what the features of the reward function are. Furthermore, we expect that their natural language expression is intuitive enough for the LLM to make accurate inferences about how they impact the model.

Algorithm 1: Procedure

Require: τ : Human trajectories

Require: π : Policy

Require: ν : Reward network

1: **do**

2: $\pi \leftarrow \text{TrainPolicy}(\nu, \tau)$

3: $\text{explanation} \leftarrow \text{LLM}(\text{prompt})$

4: $\tau' \leftarrow \text{CorrectiveDemonstrations}$

5: $T' \leftarrow \text{SampleFromDemonstrationDataset}(\tau)$

6: $\tau \leftarrow T' \cup \tau'$

7: **while** Demonstrator chooses to go through another round of providing corrective demonstrations

Domains

In our pilot study, we employ both a deterministic and a stochastic Wumpus World environment to test our pipeline. The deterministic version of Wumpus World is a modified version of Frozen Lake where there are monsters along with holes, and the map is fully observable. The objective is for the robot to reach the goal while avoiding holes and a monster that can eat the robot. The map is a 4 x 4 grid world, and the robot can move one cell per time step, either up, down, left, or right. The monster chooses from a uniform random distribution which of the directions to proceed in. The monster is initialized to occupy one of the five grid positions shown in the positions indicated in red in Figure 3 and remains in this set of grid positions for the duration of the game. In the stochastic version of Wumpus World, the agent moves in the specified direction with some probability α (chosen to be either .8 or .9), and with some probability $1 - \alpha$ it “slips” and moves in a random direction. The ground truth reward used for evaluating the learned policy grants +50 for reaching the goal, -10 for getting eaten by the monster, -5 for falling into the hole, and -1 for each step.

While not directly representative of caregiving tasks, the Wumpus World environment serves as a suitable pilot study testbed for our proposed framework as it requires balancing various components of a reward function. Once the pipeline is evaluated in our user study, we can investigate more complicated environments such as in RCareWorld (Ye et al. 2022), the results of which will be more generalizable to practical applications in caregiving robots.

Metrics

In order to compare the performance of the framework in both the deterministic and stochastic Wumpus World environments, we employ two metrics in the pilot study. The first is the mean reward over 1000 trials. The second is the Win Rate, which is a percentage of the total trials that were successful in reaching the goal.

For the user study with our target population, we propose additional metrics to evaluate user perceptions of the agent’s behavior and of the LLM-generated explanations. Firstly, we ask participants prior to each corrective demonstration collection how successful they perceived the agent to be at completing the task. We choose to do so as prior work has found that perceived success plays a larger role in shaping trust

```

prompt = """We are training an inverse reinforcement learning model using Deep Max Entropy algorithm on the Wumpus World environment. The model uses a small feedforward neural network to learn the reward function from the given demonstrations. The training manual of the game given below has information about the environment.

* TRAINING MANUAL *

The game starts with the player at location [0,0] of the frozen lake grid world with the goal (treasure) located at far extent of the world e.g. [3,3] for the 4x4 environment. The player makes moves until they reach the goal or fall in a hole or gets eaten by the monster. There is always a path to the goal.

Holes in the ice are distributed in set locations of [(1,1), (1,3), (2,3), (3,0)]. The monster starts at location (2,2). It chooses uniformly randomly whether it wants to proceed in an upward direction [(2,2), 1,2), (0,2)] or towards the left [(2,2), (2,1), (2,0)].

## Action Space
The action shape is in the range {0, 3} indicating which direction to move the player.
- 0: Move left
- 1: Move down
- 2: Move right
- 3: Move up

The agent is in a stochastic environment where it can move in a random direction with a small probability.

## Observation Space
The observation is a value representing the player's current position as current_row * ncols + current_col (where both the row and col start at 0). For example, the goal position in the 4x4 map can be calculated as follows: 3 * 4 + 3 = 15. The number of possible observations is dependent on the size of the map.

## Starting State
The episode starts with the player in state [0] (location [0, 0]).

## Episode End
The episode ends if the following happens:
1. The player moves into a hole.
2. The player reaches the goal at max(nrow) * max(ncol) - 1 (location [max(nrow)-1, max(ncol)-1]).
3. The player gets eaten by the monster.

* END OF TRAINING MANUAL *

In order to understand the model better we use SHAP. The SHAPley values for the above model are given in the plot below. These are the features -
1. Distance to reach the goal.
2. Distance to the nearest hole.
3. Location of the monster
4. Distance to the monster
5. Location of the agent
Additional features cannot be provided and you should not recommend providing more features.

It is very important to remember that negative shap value for distance encourages the agent to move closer to the object and positive shap value for distance encourages it to move away from the object. We want to move closer to the goal and away from the monster.

Can you explain these values?
What features the robot is using to learn the reward function? Your response will help the demonstrator in understanding the robot learning better. How would you suggest these values should be modified to achieve the best reward function?
Be critical in your evaluation of how the demonstrator can improve the agent.
"""

```

(a) Example of prompt given to the LLM.

```

Do you wish to see the feedback from the agent for the following training run?(Y/N)
Y
-----
## Analyzing the SHAP Values and Model Behavior:
The SHAP values provide insightful information about the learned reward function. Let's break down each feature:

* **Agent Loc:** The near-zero mean SHAP value suggests the agent's location itself doesn't significantly influence the reward. This could imply the reward is more focused on actions and proximity to other objects rather than being in a specific location. The low standard deviation indicates this feature is relatively consistent across different situations.

* **Monster Loc:** The negative mean SHAP value suggests the agent is encouraged to move *closer* to the monster. This is **counterintuitive** and indicates a problem in the learned reward function. The monster should be avoided, not approached. The higher standard deviation hints at greater variability in how the monster's location impacts reward depending on the context.

* **Dist to Hole:** The positive mean SHAP value suggests the agent is encouraged to move *further away* from holes. This is **positive** and aligns with the goal of avoiding holes. The high standard deviation suggests the distance to holes might have a more significant impact on the reward in certain situations, perhaps when the agent is close to a hole.

* **Dist to Monster:** The negative mean SHAP value indicates the agent is encouraged to move *closer* to the monster. This is **another error** in the learned reward function, as the monster should be avoided. The significantly higher standard deviation than the previous feature highlights a potentially greater influence of distance to the monster on the reward compared to the monster's location itself.

* **Dist to Goal:** The negative mean SHAP value indicates the agent is encouraged to move *closer* to the goal. This is **correct** and aligns with the objective of reaching the goal. The relatively high standard deviation suggests that the distance to the goal might have varying influence depending on the situation. For example, when the agent is already close to the goal, the influence might be weaker.

## Interpreting the Model's Learning:
The model seems to have learned some aspects of the reward function correctly, such as moving away from holes and towards the goal. However, it is demonstrably **failing to learn the danger posed by the monster**. This is likely due to the following:

* **Insufficient Demonstrations:** The demonstrations might not explicitly show the negative consequences of getting eaten by the monster.
* **Reward Function Complexity:** The simple feedforward neural network might not be complex enough to accurately model the complex reward structure of the environment.
* **Stochasticity:** The environment's inherent randomness might be causing the model to misinterpret events, leading to the faulty monster-related behaviors.

## Recommendations for Improvement:
1. **Increase Demonstration Quality:** Provide more demonstrations where the agent explicitly avoids the monster and suffers negative consequences for getting eaten. This could include demonstrations where the agent takes a detour to avoid the monster, even if it means a longer path to the goal.
2. **Increase Model Complexity:** Experiment with more complex neural network architectures, such as deeper networks or networks with recurrent layers, which might better capture the nuances of the reward function.
3. **Adjust Reward Shaping:** Consider using reward shaping techniques to guide the model towards better behavior. This could involve providing small positive rewards for moving away from the monster or penalties for moving closer to it.
4. **Analyze Monster Behavior:** Carefully examine the monster's behavior in the environment. Is the monster's movement pattern consistent or unpredictable? Are there any biases in its movement that the model is learning incorrectly?
5. **Experiment with Regularization:** Incorporate regularization techniques into the training process to prevent overfitting and encourage the model to learn more robust representations.

## Conclusion:
The current model exhibits a fundamental misunderstanding of the monster's role in the environment. Addressing this issue through improved demonstrations, model complexity, and potential reward shaping techniques should significantly improve the agent's performance. By carefully analyzing the learned reward function through SHAP values and examining the environment's intricacies, the demonstrator can guide the agent towards achieving the desired behavior.

-----
Do you wish to provide corrective demonstrations?(Y/N)

```

(b) Example of a response given by the LLM. We notice in this response's recommendation section that the LLM encourages the demonstrator to provide demonstrations that span the demonstration space, by suggesting demonstrations that avoid the monster to the detriment of the path length, as well as demonstrations where the monster eats the player.

Figure 2: This figure depicts the prompt provided to the LLM, along with an example of a response given by the LLM.

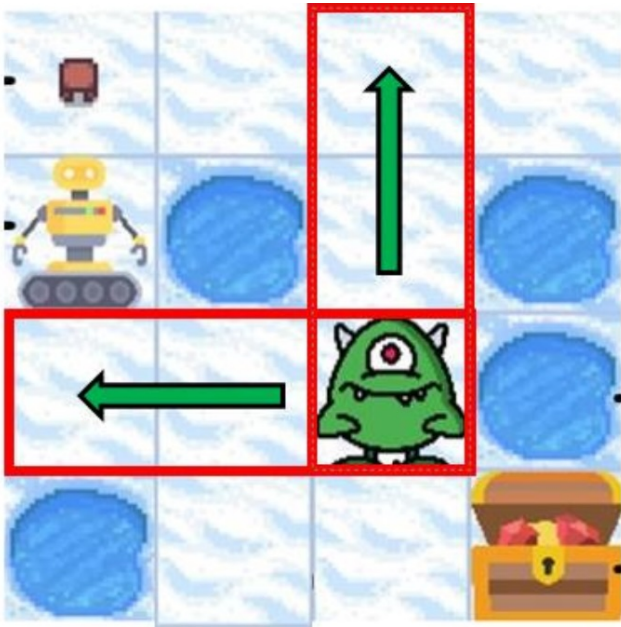


Figure 3: Wumpus World containing Monsters, Holes, and a Treasure Box.

and perceived usability (Moorman et al. 2023a). Additionally, we propose to conduct an open-ended interview where the participant will answer the following questions.

- How clear was the explanation?
- How understandable was the explanation ?
- Did the explanation make sense?
- Was the explanation sufficiently detailed?
- How relevant was the explanation?
- How can the explanation be improved?

These qualitative results will provide some insight into whether the feedback provided is perceived as practical, and understandable for non-expert users.

We additionally propose to evaluate the consistency of the LLMs responses. As the demonstrations provided by each participant will be different, we evaluate consistency with respect to each participant’s dataset. To do so, we query the LLM 10 times for each person’s demonstration set and determine how consistent the LLM’s explanations and suggestions are, similar Verma et al’s conviction perturbation test (Verma, Bhambri, and Kambhampati 2024). We propose to employ the following subjective checks of the LLM explanations.

- Consistency of the features suggested to change.
- Consistency of suggested changes to the features.
- Correctness of reasons for the suggested changes.

Population

So far, we have conducted a proof-of-concept pilot study to evaluate the pipeline, where the participants were the authors of this work for up to two rounds of corrective feedback, the

results of which can be found in the following section, in Table 1. In future work, the authors plan on running a user study with care providers and care receivers to evaluate the usability and perceptions of the framework.

Results and Discussion

In this section, we report the results of proof-of-concept pilot study of the framework.

In Table 1, we compare the changes in performance over multiple rounds of feedback in the deterministic and stochastic Wumpus World environments. We observe that in all environments, as the demonstrator goes through each round of corrective feedback using our system, the reward increases. In the deterministic environment, one round of feedback is sufficient in obtaining a win rate of 100. In the more challenging stochastic environments, one round of feedback results in a win rate of 58%, and a second round of feedback improves the win rate. In future work, we propose to determine the number of rounds that would be sufficient, if any, for the stochastic environments, and to extend our analysis by evaluating our framework with smaller increments of stochasticity, to explore a broader range of stochastic environments. While only preliminary results, these trends imply that the LLM’s explanations enable the demonstrator to improve their demonstrations.

The violin plots in the Appendix illustrate the evolution of SHAP values across two rounds of corrective feedback in both deterministic and stochastic environments. We note that the violin plots across the deterministic and stochastic environments begin to converge after the second iteration, indicating that the pipeline effectively guides the agent in learning reward functions with similarly weighted features. For instance, the SHAP values for ‘distance to the goal’ increasingly become more negative, signifying that the agent is rewarded for moving closer to the goal. Likewise, the SHAP values for ‘distance to the monster’ turn positive, reflecting the agent’s learned behavior of avoiding the monster as it progresses through the iterations.

In Figure 2b we depict the model’s response in the deterministic environment before the demonstrator provides their first round of corrective feedback. We observe that the model’s response is informative in explaining the meaning of the different features relevant to the reward function, and contextualizes how the demonstrations need to change given the task and set of Shapley values.

Takeaways: We design a user study to evaluate our pipeline’s efficacy in enabling demonstrators to better understand a policy’s features importance, in order to provide corrective demonstrations. In a pilot study, we find that after two rounds of feedback, the policy’s win rate improves across all three environments, achieving a win rate of 100% in the case of the deterministic environment. Finally, we observe that the violin plots (found in the Appendix) across the deterministic and stochastic versions of the Wumpus World environment begin to converge after the second iteration of corrective feedback from the demonstrator, indicating that the pipeline effectively guides the agent in learning reward functions with similarly weighted features.

Stochasticity (α)	Rounds of Feedback	Total Reward	Win Percentage
0	No feedback	-7.895	51.5%
	One round of feedback	44.0	100%
	Two rounds of feedback	44.0	100%
.10	No feedback	-13.40	24.2%
	One round of feedback	-0.023	58.6%
	Two rounds of feedback	26.25	83.5%
.20	No feedback	-34.85	25.6%
	One round of feedback	0.181	58.9%
	Two rounds of feedback	8.609	66.56%

Table 1: Results comparing the pipeline with baseline on varying levels of stochasticity (the agent takes a random action with probability α ranging from 0 to 1, and takes the desired action with probability $1 - \alpha$). The total reward and win percentage are calculated on 1000 episodes.

Limitations and Future Work In addition to conducting the user study with the target population, there are various opportunities for future work. For instance, our framework could be evaluated in more environments, including partially observable and continuous environment spaces. Additional baselines could be investigated – for instance, a baseline with only the Shapley values and no additional explanations, or a baseline with neither explanations nor Shapley values – in order to determine how the LLM explanations enhance interpretability and subsequent performance. Future work could consider different ways of sampling demonstrations upon receiving the corrective feedback. Currently, the five new demonstrations are selected, and (no matter what round of feedback the demonstrator is on) the remaining ten demonstrations are samples from the existing dataset, without replacement. An alternative approach could be to weigh the demonstrations provided in more recent rounds of feedback higher than the demonstrations provided earlier. Finally, further prompt engineering has the potential to improve the interaction between the human and the system.

Broader Impact Statement This work proposes a pipeline that leverages LLMs to improve the degree to which Shapley values are informative for a non-expert demonstrator. It has the potential to contribute to aging in place as it improves the experience of non-expert demonstrators by providing guidance about how to teach a robot (specifically, about the content of their demonstrations). This is essential, as direct involvement of the care receiver or care provider in the robot learning process has the potential to improve the degree of personalization and reduce reliance on robotics experts.

This work is still in the development stage. In this proof-of-concept evaluation of the pipeline, the participants are the authors of the work. In future work, we propose to demonstrate how the framework performs with non-expert users, representative of the target audience, in order to determine if the LLM feedback would be comprehensible and useful. We anticipate that working with older adults in future work will help shape how we prompt the LLM, to encourage clear and understandable explanations that are useful to non-expert users.

Conclusion

Enabling end-users to be involved in teaching desired behaviors to the system (via demonstration) allows them to communicate their needs and preferences directly, which is integral for the deployment of assistive agents. To help demonstrators understand the key features they should focus on when providing demonstrations, we propose a framework where we utilize an LLM to provide human-interpretable explanations of Shapley values. These explanations contextualize the feature importance learned by the model with respect to the task and domain, enabling non-expert demonstrators to more effectively be involved in the agent’s learning. We demonstrate our framework’s performance both in deterministic and stochastic versions of the Wumpus World environment.

Acknowledgements

This work was supported by NSF IIS-2112633 and the NSF CAREER award IIS-2340177.

References

- Ali, T.; and Kostakos, P. 2023. HuntGPT: Integrating machine learning-based anomaly detection and explainable AI with large language models (LLMs). *arXiv preprint arXiv:2309.16021*.
- Alvanpour, A.; Das, S. K.; Robinson, C. K.; Nasraoui, O.; and Popa, D. 2020. Robot failure mode prediction with explainable machine learning. In *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, 61–66. IEEE.
- Castelvecchi, D. 2016. Can we open the black box of AI? *Nature News*, 538(7623): 20.
- Gavriilidis, K.; Munafo, A.; Pang, W.; and Hastie, H. 2023. A surrogate model framework for explainable autonomous behaviour. *arXiv preprint arXiv:2305.19724*.
- Gopalan, N.; Moorman, N.; Natarajan, M.; and Gombolay, M. C. 2022. Negative Result for Learning from Demonstration: Challenges for End-Users Teaching Robots with Task And Motion Planning Abstractions. In *Robotics: Science and Systems*.

- Hedlund-Botti, E.; and Gombolay, M. C. 2023. Investigating Learning from Demonstration in Imperfect and Real World Scenarios. In *Companion of the 2023 ACM/IEEE International Conference on Human-Robot Interaction*, 769–771.
- Hedlund-Botti, E.; Seelan, L.; Yang, C.; Belles, N.; Zaidi Haider, Z.; and Gombolay, M. 2024. Developing Design Guidelines for Older Adults with Robot Learning from Demonstration. In *2024 Robotics Science and Systems (RSS)*.
- Huang, S. H.; Bhatia, K.; Abbeel, P.; and Dragan, A. D. 2018. Establishing appropriate trust via critical states. In *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, 3929–3936. IEEE.
- Khediri, A.; Slimi, H.; Yahiaoui, A.; Derdour, M.; Bendjenna, H.; and Ghenai, C. E. 2024. Enhancing Machine Learning Model Interpretability in Intrusion Detection Systems through SHAP Explanations and LLM-Generated Descriptions. In *2024 6th International Conference on Pattern Analysis and Intelligent Systems (PAIS)*, 1–6. IEEE.
- Lundberg, S. M.; and Lee, S.-I. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30.
- Mahbooba, B.; Timilsina, M.; Sahal, R.; and Serrano, M. 2021. Explainable artificial intelligence (XAI) to enhance trust management in intrusion detection systems using decision tree model. *Complexity*, 2021(1): 6634811.
- Mahmud, S.; Saisubramanian, S.; Zilberstein, S.; of Massachusetts Amherst, U.; and University, O. S. 2023. RE-VEALE: Reward Verification and Learning Using Explanations. Technical report.
- Molnar, C. 2018. A guide for making black box models explainable. URL: <https://christophm.github.io/interpretable-ml-book>, 2(3): 10.
- Moorman, N.; Hedlund-Botti, E.; Schrum, M.; Natarajan, M.; and Gombolay, M. C. 2023a. Impacts of robot learning on user attitude and behavior. In *Proceedings of the 2023 ACM/IEEE International Conference on Human-Robot Interaction*, 534–543.
- Moorman, N. M.; Gopalan, N.; Singh, A.; Hedlund-Botti, E.; Schrum, M. L.; Yang, C.; Seelan, L.; and Gombolay, M. 2023b. Investigating the Impact of Experience on a User’s Ability to Perform Hierarchical Abstraction. In *2023 Robotics Science and Systems (RSS)*.
- Rodríguez-Lera, F. J.; González-Santamarta, M. A.; González-Cantón, A.; Fernández-Becerra, L.; Sobrín-Hidalgo, D.; and Guerrero-Higueras, A. M. 2024. ROXIE: Defining a Robotic eXplanation and Interpretability Engine. *arXiv preprint arXiv:2403.16606*.
- Schrum, M. L.; Hedlund-Botti, E.; and Gombolay, M. C. 2022. Towards Improving Life-Long Learning Via Personalized, Reciprocal Teaching. In *Workshop on Lifelong Learning and Personalization in Long-Term Human-Robot Interaction (LEAP-HRI), 17th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*.
- Schrum, M. L.; Hedlund-Botti, E.; Moorman, N.; and Gombolay, M. C. 2022. Mind meld: Personalized meta-learning for robot-centric imitation learning. In *2022 17th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 157–165. IEEE.
- Shah, R.; Gundotra, N.; Abbeel, P.; and Dragan, A. 2019. On the feasibility of learning, rather than assuming, human biases for reward inference. In *International conference on machine learning*, 5670–5679. PMLR.
- Silva, A.; Tambwekar, P.; Schrum, M.; and Gombolay, M. 2024. Towards Balancing Preference and Performance through Adaptive Personalized Explainability. In *Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction*, 658–668.
- Thomaz, A. 2023. Robots in Real Life: Putting HRI to Work. In *Proceedings of the 2023 ACM/IEEE International Conference on Human-Robot Interaction*, 3–3.
- Verma, M.; Bhambri, S.; and Kambhampati, S. 2024. Theory of Mind abilities of Large Language Models in Human-Robot Interaction: An Illusion? In *Companion of the 2024 ACM/IEEE International Conference on Human-Robot Interaction*, 36–45.
- WHO, W. H. O. 2023. Ageing. https://www.who.int/health-topics/ageing#tab=tab_1. Accessed: 2024-08-06.
- Wulfmeier, M.; Ondruska, P.; and Posner, I. 2015. Maximum entropy deep inverse reinforcement learning. *arXiv preprint arXiv:1507.04888*.
- Ye, R.; Xu, W.; Fu, H.; Jenamani, R. K.; Nguyen, V.; Lu, C.; Dimitropoulou, K.; and Bhattacharjee, T. 2022. Rcare world: A human-centric simulation world for caregiving robots. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 33–40. IEEE.
- Zhang, L.; and Chen, Z. 2024. Large language model-based interpretable machine learning control in building energy systems. *Energy and Buildings*, 313: 114278.
- Zhifei, S.; and Joo, E. M. 2012. A survey of inverse reinforcement learning techniques. *International journal of intelligent computing and cybernetics*, 5(3): 293–311.
- Ziebart, B. D.; Maas, A. L.; Bagnell, J. A.; Dey, A. K.; et al. 2008. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, 1433–1438. Chicago, IL, USA.