Trajectory Improvement and Reward Learning from Comparative Language Feedback

Anonymous Author(s)
Affiliation
Address
email

Abstract: Learning from human feedback has gained traction in fields like robotics and natural language processing in recent years. While prior works mostly rely on human feedback in the form of comparisons, language is a preferable modality that provides more informative insights into user preferences. In this work, we aim to incorporate comparative language feedback to iteratively improve robot trajectories and to learn reward functions that encode human preferences. To achieve this goal, we learn a shared latent space that integrates trajectory data and language feedback, and subsequently leverage the learned latent space to improve trajectories and learn human preferences. To the best of our knowledge, we are the first to incorporate comparative language feedback into reward learning. Our simulation experiments demonstrate the effectiveness of the learned latent space and the success of our learning algorithms. We also conduct human subject studies that show our reward learning algorithm achieves a 23.94% higher subjective score on average and is 11.34% more time-efficient compared to preference-based reward learning, underscoring the superior performance of our method.

Keywords: Learning from Human Language Feedback, Reward Learning, Human-Robot Interaction

1 Introduction

Learning from human feedback has gained significant popularity in robotics, leading to the study of different forms of human feedback: demonstrations [1, 2, 3, 4], preference comparisons [5, 6, 7, 8], rankings [9], physical corrections [10], visual saliency maps [11], human language [12, 13], etc. Among these, preference comparisons grew popular for its simplicity and ease of use, especially compared to demonstrations [7]. Preference comparisons often involve users choosing between a pair of choices. Using these selections to learn a reward function and train a policy is known as reinforcement learning from human feedback (RLHF) [6] or more generally preference-based learning [5, 14]. It has proven applicable to a broad range of fields ranging from robotics [7, 15] to natural language processing [16], from traffic routing [17] to human-computer interaction [18].

Despite their successes, preference comparisons suffer from problems [19] such as the reliability of human data and the limited information bandwidth, i.e., each pairwise comparison contains at most 1 bit of information [20]. There has been research to provide a better interface [21], allowing the users to specify their preferences for every feature, but they require features to be designed by hand. As an alternative form of human feedback, comparative language is considerably more informative than preference comparisons, allowing users to prioritize specific aspects.

In this work, we aim to leverage comparative human language feedback to learn the human preferences, i.e., their reward functions. In pursuit of this objective, we first learn a shared latent space that aligns trajectory data and language feedback. This alignment enables the robot to comprehend human language feedback, leverage it for adapting its behavior to better align with the preferences of the human user, and efficiently learn the human's preferences. To test the effectiveness of our approach, we conduct experiments in two simulation environments and a human-subject study with a real robot. The results suggest that reward learning from comparative language feedback outper-

Submitted to the 8th Conference on Robot Learning (CoRL 2024). Do not distribute.

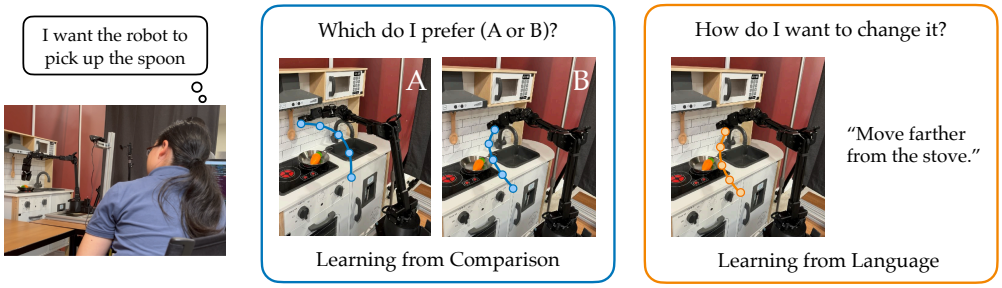


Figure 1: The figure shows an image from our human subject studies where the human user wants the robot to pick up the spoon. Compared to traditional comparison preference learning, our language preference learning enables users to give more informative feedback, which helps the robot to capture human preferences more efficiently.

forms traditional preference comparisons in performance and time-efficiency, and is favored by most of the users.

2 Related Work

Before formally defining the problem, we will first review existing works in learning from human feedback, preference-based learning, and robot learning from human language feedback.

Learning from Human Feedback. Several promising approaches leverage human feedback to train robots [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 22]. However, the most popular methods of learning from demonstrations and rankings involves a trade-off between informativeness and ease of use [6, 19, 23, 24]. Furthermore, none have taken advantage of the expressive nature of language to develop a method of reward learning from comparative language feedback. Addressing this gap will provide more scalable methods for training robots, particularly in complex environments where traditional feedback forms may be insufficient or impractical.

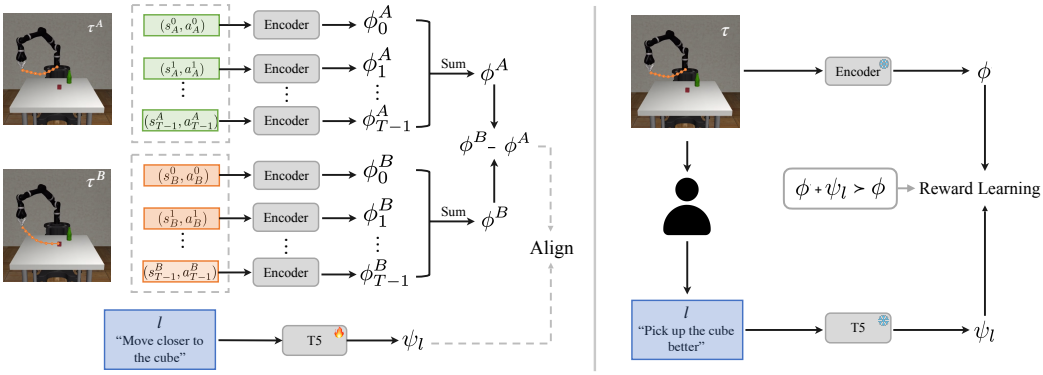
Preference-based Learning. Based on preference comparison feedback, preference-based learning is widely used for its logical intuitiveness and ease of use. Works have sought to improve its shortcomings in time- and sample-efficiency by actively generating pairs based on information gain [20] or volume removal [5], and doing these in batches of pairs [25]. However, the fundamental limitation remains that each pairwise comparison provides at most 1 bit of information [20]. In addition, users struggle to choose between two similar trajectories [7]. In contrast, our method of comparative language feedback offers more informative input after observing just one trajectory, while maintaining intuitiveness and ease of use.

Robot Learning with Human Language Feedback. Benefiting from advancements in natural language processing, works have leveraged language for adjusting robot trajectories [13, 26, 27, 28, 29, 30], fine-tuning language models [12], and reward shaping [31, 32]. For example, Shi et al. [29] uses language-conditioned behavior cloning (LCBC) for corrective language commands and improving policies. However, no prior work has explored learning the reward function of humans entirely from comparative language feedback, which offers benefits over policy learning with generalizability and explainability. Additionally, our method iteratively updates the learned reward function for better preference alignment, providing an advantage over one-shot corrections[13, 27].

3 Problem Definition

Having reviewed the related literature, we are now ready to formally define the problem. We model the robot as a decision making agent in a standard finite-horizon Markov decision process (MDP). Each robot trajectory consists of state-action pairs for T time steps[1]: $\tau = \{(s_0, a_0), (s_1, a_1), ..., (s_{T-1}, a_{T-1})\}$. A reward function $R(s_t, a_t)$, which is only known by the

[1]Our work trivially extends to the cases where trajectory length is not fixed.



(a) Model Architecture (b) Language-based Reward Learning

Figure 2: Overview of our approach. (a) Architecture of the model that learns a shared latent space between trajectories and language feedback. (b) Language-based Reward Learning.

human user, encodes human preferences regarding the task. The reward of a trajectory is defined as:

$$R(\tau) = \sum_{t=0}^{T-1} R(s_t, a_t) \tag{1}$$

For each trajectory, the human may provide language feedback l that attempts to improve the trajectory (based on the reward function) in one aspect, e.g., speed, distance to objects, height of the robot's arm, etc. As an example, the human might say "move farther from the stove" (see Figure 1) if the robot's trajectory would be higher in that way. Our goal is to develop a framework where we learn the reward function based on such feedback.

Unfortunately, this task is doomed without any information about how language feedback relates to the different aspects of the task. In this work, we learn this relation with an offline training phase where we collect a dataset that consists of pairs of robot trajectories and a language label describing how the two trajectories differ. We use this dataset to learn encoders that map trajectories and language feedback onto a shared latent space. We leverage these encoders to learn the preferences (reward functions) of different users based on their language feedback.

In the next section, we detail our approach to learning the latent space and explain how this learned latent space is utilized for iterative trajectory improvements and reward learning.

4 Our Approach

As shown in Figure 2, our approach is composed of two stages: First, we learn a shared latent space where robot trajectories and human language feedback are aligned. Second, we leverage this learned latent space to (1) improve robot trajectory or (2) learn human preferences.

4.1 Learning the Shared Latent Space

To learn a shared latent space for trajectories and language feedback, we collect a dataset of (τ^A, τ^B, l) tuples, where τ^A and τ^B are a pair of trajectories and l is a language utterance that describes the difference between the two trajectories. Note that this language utterance does not necessarily align with the human's preferences about the robot: it just describes a difference between the trajectories — it is possible to have $l =$"move faster" if the robot moves faster in τ^B than τ^A even though users want the robot to move slowly.

After collecting such a dataset, we propose the model visualized in Figure 2 to learn the shared latent space between trajectories and language utterances. Similar to [33], we use a neural network to encode each state-action pair (s_t, a_t) from the pair of trajectories, τ^A and τ^B, to embeddings ϕ_t^A and ϕ_t^B, respectively. We want these embeddings to contain information about aspects of the trajectories that the human may care and give feedback about.

To achieve this, we align the difference between the embeddings ϕ^A and ϕ^B with embeddings ψ_l of the language feedback l by using the following loss function:

$$L_{\text{align}}(\tau^A, \tau^B, l) = -\log\left(\text{sigmoid}\left(\psi_l^\top(\phi^B - \phi^A)\right)\right) \tag{2}$$

where the sigmoid can be considered as the probability that the language utterance l is inputted for the pair of trajectories τ^A and τ^B. Intuitively, when $\phi^B - \phi^A$ and ψ_l align, i.e., have the same direction for fixed magnitudes, the loss is minimized. This alignment will enable us to acquire the embedding of an improved trajectory as $\phi^A + \psi_l$ when given an initial trajectory τ^A and language feedback l.

Instead of training a language encoder from scratch, which would require a prohibitively large and diverse dataset of (τ^A, τ^B, l) tuples, we use a pretrained T5 model [34]. To align embeddings of trajectories with those of language feedback, we first freeze T5 model and train the trajectory encoder. Subsequently, we perform co-finetuning of both components.

Additionally, we incorporate a regularization term into the loss function to prevent the model from overfitting. This regularization term attempts to constrain the norms of trajectory embeddings to remain below 1 and the norm of language embeddings to be close to 1:

$$L_{\text{reg}}(\tau^A, \tau^B, l) = a \cdot \left(\max\{\|\phi^A\| - 1, 0\} + \max\{\|\phi^B\| - 1, 0\}\right) + b \cdot \|\psi_l - 1\|^2 \tag{3}$$

where a and b are two hyperparameters.

Overall, the objective we use in the model training is made up of two terms:

$$L(\tau^A, \tau^B, l) = L_{\text{align}}(\tau^A, \tau^B, l) + L_{\text{reg}}(\tau^A, \tau^B, l), \tag{4}$$

which we use to train the trajectory encoder and finetune T5. Training this architecture gives us the ability to encode any trajectory and language utterance in the same latent space. In the next subsection, we demonstrate how this latent space is useful for iteratively improving robot trajectories and for learning human preferences based on their language feedback.

4.2 Utilizing the Learned Latent Space

The shared latent space aligns robot trajectories with human language feedback. This alignment enables an intuitive understanding of user preferences. We will now explore two primary ways to leverage this learned latent space: first, to iteratively improve the robot's trajectory, and second, to accurately learn user preferences.

4.2.1 Iterative Trajectory Improvement

Firstly, we leverage the latent space to iteratively improve an initial suboptimal robot trajectory. We start with showing the initial trajectory τ^0 to the user and asking for language feedback l^0. Upon receiving human's language feedback, we use our trained encoders to compute the trajectory embedding ϕ^0 and language embedding ψ_{l^0}. We then find the *improved trajectory* τ^1 such that its difference with τ^0 best aligns with the human's language feedback based on cosine similarity:

$$\tau^1 = \underset{\tau'}{\text{argmax}} \frac{\psi_{l^0}^\top(\phi' - \phi^0)}{\|\phi'\|_2 \cdot \|\phi^0\|_2} \tag{5}$$

We iteratively continue this process for N iterations to obtain $\tau^0, \tau^1, \ldots, \tau^N$. In this work, we search over a discrete, predefined set of trajectories to solve the optimization in Eq. (5) for computational efficiency. It is, however, possible to use reinforcement learning or model-predictive control algorithms to solve this optimization at the expense of increased computational cost.

4.2.2 Reward Learning from Comparative Language Feedback

In addition to improving trajectories, we also utilize the learned latent space to learn the user's preference, i.e., their reward function. Previous approaches ask users to label their preference from a pair of trajectories, which requires the users to watch two trajectories in order to give at most 1 bit of information. In our language-based reward learning approach, for each query i, we only show users one trajectory τ_i to collect language feedback l_i based on their preferences. Given one

trajectory, we construct an improved trajectory $\hat{\tau}_i$: $\phi_{\hat{\tau}_i} = \phi_{\tau_i} + \psi_{l_i}$. Note that $\hat{\tau}_i$ is just an imaginary trajectory that maps to $\phi_{\tau_i} + \psi_{l_i}$ in the learned latent space. Then following the Bradley-Terry model [35], which is commonly used in preference-based learning, we can model a preference predictor with the reward function r_ξ as:

$$P_\xi(\hat{\tau}_i \succ \tau_i) = \frac{\exp r_\xi(\phi_{\hat{\tau}_i})}{\exp r_\xi(\phi_{\hat{\tau}_i}) + \exp r_\xi(\phi_{\tau_i})} \tag{6}$$

where $\hat{r_\xi}$ is a neural network parameterized with ξ. The reward function $\hat{r_\xi}$ is trained by minimizing the following negative loglikelihood loss:

$$L_{\texttt{Implicit}} = -\frac{1}{n}\sum_{i=0}^{n-1} \log P_\xi(\hat{\tau}_i \succ \tau_i)\,. \tag{7}$$

We now make an important observation about the language feedback. When the user tells the robot to move farther from the stove, it indeed indicates a preference between the original trajectory and the improved trajectory that moves farther from the stove. However, this feedback contains much more information than this comparison. The user could use any comparative language utterance to teach the robot, but they specifically selected one about the distance to the stove. This means, with high probability, the improvement the robot may get from this feedback is higher than that of any other comparative language feedback.

Mathematically, this indicates a preference for $\hat{\tau}_i$ over $\tilde{\tau}_i : \phi_{\tilde{\tau}_i} = \phi_{\tau_i} + \psi_{\tilde{l}_i}$ where $\tilde{l}_i$ is any language utterance other than l_i. Again we apply the Bradley-Terry model to utilize this implicit preference:

$$P_\xi(\hat{\tau}_i \succ \tilde{\tau}_i) = \frac{\exp r_\xi(\phi_{\hat{\tau}_i})}{\exp r_\xi(\phi_{\hat{\tau}_i}) + \exp r_\xi(\phi_{\tilde{\tau}_i})} \tag{8}$$

Based on this, we sample k language feedback from a set of pre-collected language utterances other than l_i, and minimize the following loss about the chosen language feedback:

$$L_{\texttt{Explicit}} = -\frac{1}{k}\sum_{j=0}^{k-1} \log P_\xi(\hat{\tau}_i \succ \tilde{\tau}_{i,j}) \tag{9}$$

Overall, the loss for reward learning from comparative language feedback is as follows:

$$L_{\texttt{Reward}} = L_{\texttt{Implicit}} + L_{\texttt{Explicit}} = -\frac{1}{n}\sum_{i=0}^{n-1}\left(\log P_\xi(\hat{\tau}_i \succ \tau_i) + \frac{1}{k}\sum_{j=0}^{k-1}\log P_\xi(\hat{\tau}_i \succ \tilde{\tau}_{i,j})\right)\,. \tag{10}$$

Training the reward function with this loss enables us to efficiently capture human preferences through comparative language feedback.

5 Experiments

We conducted simulation experiments and human subject studies to validate our methods with diverse tasks and features that humans may give feedback about.

5.1 Simulation Experiments

Synthetic humans. For simulation experiments, we synthesize the comparative language feedback from a simulated human with a simplified reward function R:

$$R(s_t, a_t) = w^\top \theta(s_t, a_t) \tag{11}$$

where θ is a function that maps a state-action pair to a vector of high-level features (e.g., speed, distance to the stove). We similarly define $\theta(\tau)$ to denote the sum of features over a trajectory τ. w is a vector of weights that maps the features to a scalar reward value. Both θ and w are unknown to our algorithm.

Given a true reward function w, let τ^* be the optimal trajectory from the validation set under reward w. Then, the synthetic human gives the noisy language feedback l^0 when shown trajectory τ^0:

$$l^0 \sim \ell\left(\text{softmax}(w \odot (\theta_l^* - \theta_l^0))\right) \tag{12}$$

where θ^* and θ^0 denote the true cumulative features of τ^* and τ^0 respectively, $\odot$ denotes element-wise multiplication, and the function ℓ takes a sample from the output of the softmax and outputs a language feedback that corresponds to the sampled feature.

Environments. We experimented in two simulation environments: Robosuite [36] and Meta-World [37]. Robosuite has a Jaco robot arm at a table with a cube and a bottle, and the task is to pick up the cube. The states are given as 640×480 RGB images (resized to 224×224) and 25-dimensional proprioception. The action space is 4-dimensional (the end-effector always points down). Meta-World has a Sawyer robot arm at a table, and the task is to push a button on the side of the table. The states are given as 480×320 RGB images (resized to 224×224) and 20-dimensional proprioception. The action space is 4-dimensional (the end-effector always points down).

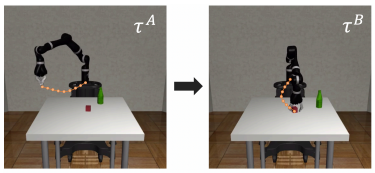


Figure 3: Each dataset sample is a pair of trajectories and a language feedback.

Learning the Latent Space. Lists of language feedback were created for hand-crafted features (see Appendix A) to indicate a change (e.g., {'Move higher', 'Move lower'} for height). They were augmented with GPT 3.5 [38] to 629 sentences for Robosuite and 660 for Meta-World. The splits are 480, 74, 75 and 492, 84, 84 for the training, validation, and test sets. Note these features are only for synthetic dataset creation — training our architecture does not require hand-designing features.

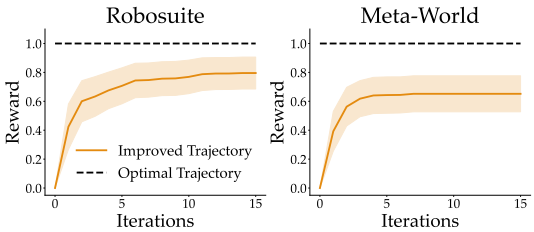


Figure 4: Results of experiments where we use simulated human language feedback to iteratively improve a robot trajectory (mean $\pm$ std over 100 runs). The dashed line represents *average* reward of optimal trajectories.

We trained RL policies with randomized weights w using stratified sampling [39] over the features, then generated rollout trajectories with timesteps $T = 500$ for each environment. For Robosuite, we generated 448 unique rollouts and 324 for Meta-World. The splits were 359, 44, 45 and 260, 32, 32 for training, validation, and test sets. Trajectories were paired within each set and matched with a language feedback that describes the change in each feature from τ^A to τ^B (see Figure 3).

We train the encoders with the loss in Eq. (4), and use accuracy as the metric to evaluate:

$$\text{Acc} = \frac{|\{(\tau^A, \tau^B, l) \mid \psi_l^\top(\phi^B - \phi^A) > 0\}|}{\text{Total number of samples}} \,. \tag{13}$$

Training the encoders with the training set of trajectories and language utterances, we observed a test accuracy of 0.8489 in Robosuite and 0.8288 in Meta-World. This indicates that the trajectory and language embeddings are well-aligned.

Iterative Trajectory Improvement. Next, we conducted iterative trajectory improvement experiments based on the method we presented in Section 4.2.1. We set the number of iteration steps $N = 15$ and repeat the full experiment over 100 random seeds. The true rewards of these trajectories are shown in Fig. 4. In both environments, we consistently improve the trajectories, which showcases the effectiveness of the learned latent space and the improvement algorithm. However, it can be noted that our algorithm cannot reach the performance of the optimal trajectory. This is because every improvement iteration is completely independent from the previous iterations and the robot may get stuck in a loop between two good, but non-optimal, trajectories.

Our reward learning algorithm presented in Section 4.2.2, on the other hand, utilizes the entire history of human feedback, so it should not suffer from this problem. We will now demonstrate this.

Reward Learning. We again randomly initialize the reward weights w to simulate human feedback. We use the loss in Eq. (9) to learn the reward function. We adopt cross-entropy $CE(P_w, P_\xi)$ as the

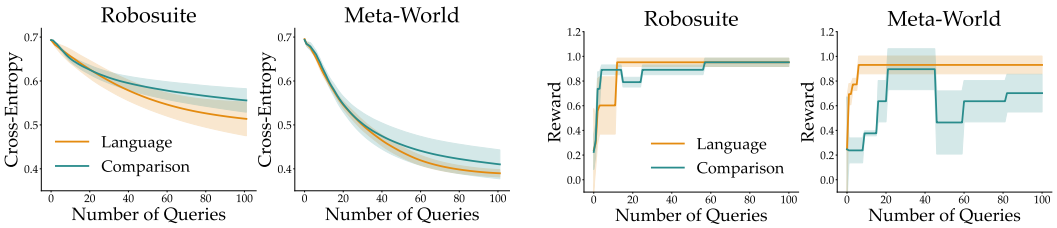
evaluation metric, where

$$P_w(\tau^A \succ \tau^B) = \frac{\exp w^\top \theta(\tau^A)}{\exp w^\top \theta(\tau^A) + \exp w^\top \theta(\tau^B)}, \; P_\xi(\tau^A \succ \tau^B) = \frac{\exp r_\xi(\phi^A)}{\exp r_\xi(\phi^A) + \exp r_\xi(\phi^B)} \quad (14)$$

Another metric is the true reward value of the optimal trajectory selected from the test set based on the learned reward, reflecting how close the learned reward is to the true reward. All reward values are normalized between 0 and 1.

We compare our method against reward learning from comparisons, where for each query, the simulated user chooses a preferred trajectory from pair (τ^A, τ^B) and with probability shown in Eq. (14).

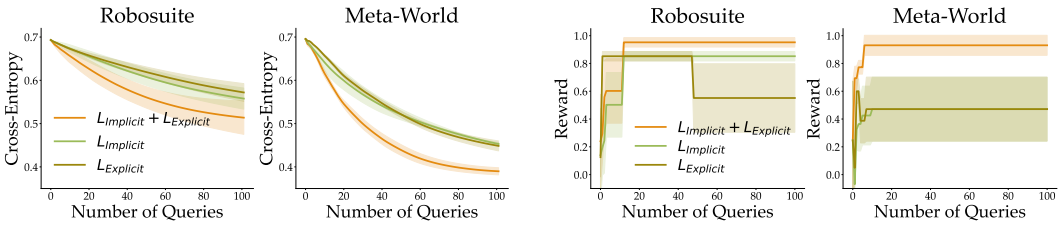
As indicated in Figure 5a, the cross-entropy of our method decreases faster than that of comparison preference learning in both environments, demonstrating that our approach converges more quickly than the baseline. Meanwhile, Figure 5b shows that the true reward of optimal trajectories reaches a value of 1 in a shorter time with our language-based method, especially in Meta-World, further supporting that it learns the reward function more efficiently.



(a) Cross-Entropy (b) True Reward Values of Optimal Trajectory

Figure 5: Results of reward learning, averaged over 3 seeds. (a) Cross Entropy: our method converges faster. (b) True Reward of Optimal Trajectory: Our approach.

Ablation Study. We conducted an ablation study to evaluate the different components of the loss function (Eq. (9)). As indicated in Fig. 6a, the combination of both components outperforms using each component individually, demonstrating the validity of our loss function design. Also, using both components fits the true reward function better, as shown in Fig. 6b.



(a) Cross-Entropy (b) True Reward Values of Optimal Trajectory

Figure 6: Results of Ablation Study. (a) Baseline Comparison: our method consistently outperforms the baseline. (b) Ablation Study: Using two components is better than applying each of them alone.

5.2 User Studies

To verify the effectiveness of our approach, we conducted human subject studies by recruiting 10 subjects (4 female, 6 male) from varying backgrounds and observing them to interact with our real robot setup. Our study has been approved by an IRB office.

Real Robot Setup. We used a WidowX 250 6DOF robot with the setup of [40] in front of a kitchen set (Figure 7). The task is to pick up a spoon while avoiding going over a pan on the stove. The states are given as 480×320 images resized to 224×224, and 22-dimensional proprioception. The actions are 7-dimensional (6DOF+gripper). To train the latent space, we collected 321 trajectories with varying levels of success at picking up the spoon, avoiding the pan, and speed. These trajectories were divided into 192 for training, 64 for validation, and 65 for testing. 496 GPT-augmented sentences were split into 297 for training, 99 for validation, and 100 for testing. These sentences were paired with the trajectories in the same manner as in the simulation experiments. Another set of 32 trajectories were collected for the purpose of user studies.

Figure 7: (Left) Closeup view of the kitchen set with the spoon hanging on the wall, above the pan on the stove. (Right) WidowX 250 6DOF robot arm, user, and text prompt for experiments.

Study 1: Trajectory Improvement. Users start with a suboptimal trajectory from the dataset and give language feedback for a maximum of 10 iterations until they are satisfied.

Study 2: Preference Learning. For 20 iterations, users give feedback (preference comparison or comparative language) on random trajectories and rate the optimal trajectory from the dataset with respect to the learned reward after every 5 iterations. To avoid bias, 5 users began with the comparison based method and 5 with the language method.

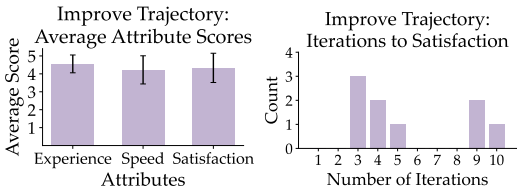


Figure 8: (Left) Average attribute scores. (Right) Iterations to satisfaction.

After each study, users were asked to complete post-study surveys (see Appendix C).

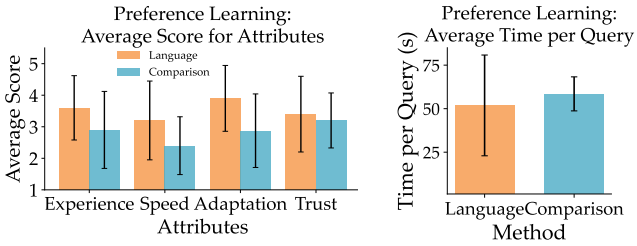


Figure 9: (Left) User ratings. (Right) Time per query.

Table 1: User satisfaction in aspects, not mutually exclusive between methods

	Satisfactory	
Method	Language	Comparison
Time Efficient	9 / 10	1 / 10
Adaptable	6 / 10	8 / 10
Convenient	8 / 10	3 / 10

Results. Our post-study surveys reveal that the first study has consistently positive responses for user experience and speed of adaptation, but has two peaks in iterations to satisfaction (Fig. 8). We conjecture that the dataset of 32 trajectories may not have contained trajectories that the users desired. As for the second study, our language-based method scored better than the comparison-based method **for all attributes**. The average score of the language-based method over all attributes is **23.94%** higher than the comparison-based method , illustrating the superior performance of our approach. In terms of time-efficiency, our method takes **11.34%** less time for users to answer each query as shown in Fig. 9. Indeed, users found the language method to be more time efficient and adaptable but found comparison method to be more convenient (Table 1).

6 Conclusion

In this paper, we presented a robot learning framework to learn human preferences using comparative language feedback. To achieve this, we aligned trajectories with language feedback in a shared latent space, then used it to improve trajectories and learn preferences. Experiments in simulation environments and real-world user studies suggest that our approach consistently outperforms comparison preference learning and is favored by most users for aspects such as time efficiency and adaptability.

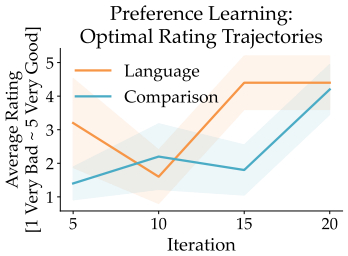


Figure 10: Average ratings of the optimal trajectory shown every 5 iterations for each method.

Limitations and Future Work. The first limitation lies in the correspondence between language feedback and objects in the scene. Additionally, we adopt a naive approach and sample queries randomly. Based on these limitations, the future works include: (1) Use image annotation to generate language feedback containing all objects. (2) Implement active learning to reduce the number of iterations required to reach the optimal trajectory.

References

[1] D. Sadigh, S. Sastry, S. Seshia, and A. Dragan. Planning for autonomous cars that leverage effects on human actions. 06 2016. doi:10.15607/RSS.2016.XII.029.

[2] S. Ross, G. Gordon, and D. Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.

[3] M. Kelly, C. Sidrane, K. Driggs-Campbell, and M. J. Kochenderfer. Hg-dagger: Interactive imitation learning with human experts. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8077–8083. IEEE, 2019.

[4] R. Hoque, A. Balakrishna, E. Novoseller, A. Wilcox, D. S. Brown, and K. Goldberg. Thriftydagger: Budget-aware novelty and risk gating for interactive imitation learning. *arXiv preprint arXiv:2109.08273*, 2021.

[5] D. Sadigh, A. D. Dragan, S. S. Sastry, and S. A. Seshia. Active preference-based learning of reward functions. In *Proceedings of Robotics: Science and Systems (RSS)*, July 2017. doi:10.15607/RSS.2017.XIII.053.

[6] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

[7] E. Bıyık, D. P. Losey, M. Palan, N. C. Landolfi, G. Shevchuk, and D. Sadigh. Learning reward functions from diverse sources of human feedback: Optimally integrating demonstrations and preferences. *The International Journal of Robotics Research*, 41(1):45–67, 2022.

[8] N. Wilde, E. Biyik, D. Sadigh, and S. L. Smith. Learning reward functions from scale feedback. *CoRR*, abs/2110.00284, 2021. URL `https://arxiv.org/abs/2110.00284`.

[9] V. Myers, E. Biyik, N. Anari, and D. Sadigh. Learning multimodal rewards from rankings. In *Proceedings of the 5th Conference on Robot Learning (CoRL)*, 2021.

[10] A. Bajcsy, D. P. Losey, M. K. O'malley, and A. D. Dragan. Learning robot objectives from physical human interaction. In *Conference on Robot Learning*, pages 217–226. PMLR, 2017.

[11] A. Liang, J. Thomason, and E. Biyik. Visarl: Visual reinforcement learning guided by human saliency. In *ICRA2023 Workshop on Pretraining for Robotics (PT4R)*, 2023.

[12] J. A. Campos and J. Shern. Training language models with language feedback. In *ACL Workshop on Learning with Natural Language Supervision. 2022.*, 2022.

[13] P. Sharma, B. Sundaralingam, V. Blukis, C. Paxton, T. Hermans, A. Torralba, J. Andreas, and D. Fox. Correcting robot plans with natural language feedback. *arXiv preprint arXiv:2204.05186*, 2022.

[14] C. Wirth, R. Akrour, G. Neumann, J. Fürnkranz, et al. A survey of preference-based reinforcement learning methods. *Journal of Machine Learning Research*, 18(136):1–46, 2017.

[15] Y. Wang, Z. Sun, J. Zhang, Z. Xian, E. Biyik, D. Held, and Z. Erickson. Rl-vlm-f: Reinforcement learning from vision language foundation model feedback. In *International Conference on Machine Learning (ICML)*, 2024.

[16] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

[17] E. Bıyık, D. A. Lazar, R. Pedarsani, and D. Sadigh. Incentivizing efficient equilibria in traffic networks with mixed autonomy. *IEEE Transactions on Control of Network Systems*, 8(4): 1717–1729, 2021.

[18] N. Dennler, D. Delgado, D. Zeng, S. Nikolaidis, and M. Matarić. The rosid tool: Empowering users to design multimodal signals for human-robot collaboration. In *18th International Symposium on Experimental Robotics (ISER)*, 2023.

[19] S. Casper, X. Davies, C. Shi, T. K. Gilbert, J. Scheurer, J. Rando, R. Freedman, T. Korbak, D. Lindner, P. Freire, T. Wang, S. Marks, C.-R. Segerie, M. Carroll, A. Peng, P. Christoffersen, M. Damani, S. Slocum, U. Anwar, A. Siththaranjan, M. Nadeau, E. J. Michaud, J. Pfau, D. Krasheninnikov, X. Chen, L. Langosco, P. Hase, E. Biyik, A. Dragan, D. Krueger, D. Sadigh, and D. Hadfield-Menell. Open problems and fundamental limitations of reinforcement learning from human feedback. *Transactions on Machine Learning Research (TMLR)*, 2024.

[20] E. Biyik, M. Palan, N. C. Landolfi, D. P. Losey, and D. Sadigh. Asking easy questions: A user-friendly approach to active reward learning. In *Proceedings of the 3rd Conference on Robot Learning (CoRL)*, 2019.

[21] C. Basu, M. Singhal, and A. D. Dragan. Learning from richer human guidance: Augmenting comparison-based learning with feature queries. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, pages 132–140, 2018.

[22] J. Spencer, S. Choudhury, M. Barnes, M. Schmittle, M. Chiang, P. Ramadge, and S. Srinivasa. Expert intervention learning: An online framework for robot learning from explicit and implicit human feedback. *Autonomous Robots*, pages 1–15, 2022.

[23] P. Abbeel and A. Ng. Apprenticeship learning via inverse reinforcement learning. *Proceedings, Twenty-First International Conference on Machine Learning, ICML 2004*, 09 2004. doi:10.1007/978-0-387-30164-8_417.

[24] B. Akgün, M. Cakmak, K. Jiang, and A. L. Thomaz. Keyframe-based learning from demonstration. *International Journal of Social Robotics*, 4:343 – 355, 2012. URL https://api.semanticscholar.org/CorpusID:10004846.

[25] E. Bıyık, N. Anari, and D. Sadigh. Batch active learning of reward functions from human preferences. *ACM Transactions on Human-Robot Interaction*.

[26] A. Bucker, L. Figueredo, S. Haddadinl, A. Kapoor, S. Ma, and R. Bonatti. Reshaping robot trajectories using natural language commands: A study of multi-modal data alignment using transformers. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 978–984. IEEE, 2022.

[27] A. Bucker, L. Figueredo, S. Haddadin, A. Kapoor, S. Ma, S. Vemprala, and R. Bonatti. Latte: Language trajectory transformer. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7287–7294. IEEE, 2023.

[28] J. Yow, N. P. Garg, M. Ramanathan, W. T. Ang, et al. Extract–explainable trajectory corrections from language inputs using textual description of features. *arXiv preprint arXiv:2401.03701*, 2024.

[29] L. X. Shi, Z. Hu, T. Z. Zhao, A. Sharma, K. Pertsch, J. Luo, S. Levine, and C. Finn. Yell at your robot: Improving on-the-fly from language corrections, 2024.

[30] M. Han, Y. Zhu, S.-C. Zhu, Y. N. Wu, and Y. Zhu. Interpret: Interactive predicate learning from language feedback for generalizable task planning. *arXiv preprint arXiv:2405.19758*, 2024.

[31] P. Goyal, S. Niekum, and R. J. Mooney. Using natural language for reward shaping in reinforcement learning, 2019.

[32] P. Goyal, S. Niekum, and R. J. Mooney. Pixl2r: Guiding reinforcement learning using natural language by mapping pixels to rewards, 2020.

[33] S. M. Katz, A. Maleki, E. Bıyık, and M. J. Kochenderfer. Preference-based learning of reward function features. *arXiv preprint arXiv:2103.02727*, 2021.

[34] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

[35] R. A. Bradley and M. E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

[36] Y. Zhu, J. Wong, A. Mandlekar, R. Martín-Martín, A. Joshi, S. Nasiriany, and Y. Zhu. robosuite: A modular simulation framework and benchmark for robot learning. In *arXiv preprint arXiv:2009.12293*, 2020.

[37] T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pages 1094–1100. PMLR, 2020.

[38] OpenAI. Gpt-3.5. Online. https://platform.openai.com/docs/models/gpt-3-5.

[39] M. J. Kochenderfer and T. A. Wheeler. *Algorithms for optimization*. Mit Press, 2019.

[40] F. Ebert, Y. Yang, K. Schmeckpeper, B. Bucher, G. Georgakis, K. Daniilidis, C. Finn, and S. Levine. Bridge data: Boosting generalization of robotic skills with cross-domain datasets. *arXiv preprint arXiv:2109.13396*, 2021.

A Dataset Features

Robosuite Features

- The height of the robot's end-effector
- The speed of the end-effector
- The distance between the end-effector and the bottle
- The distance between the end-effector and the cube
- How well the robot lifts the cube (i.e., level of success of cube-lifting task)

The level of success of the cube-lifting task is quantified by: 1) whether or not the cube is lifted above the height of a success threshold or 2) a weighted sum of the distance to the cube and whether or not the end-effector is grasping the cube.

Meta-World Features

- The height of the robot's end-effector.
- The velocity of the robot's end-effector.
- The distance between the end-effector and the button.

WidowX Features

- The velocity of the end-effector
- The distance between the end-effector and the pan
- How well the robot picks up the spoon

B User Study Surveys

Here are the questions in our user study surveys.

Pre-Study Survey. To assess the subjects' experience and level of skill with robotics and machine learning, we conducted a short pre-study survey before any of the experiments.

1. "Age"
2. "Gender"
3. "Race"
4. "Highest level of education"
5. "Please describe your level of robotics experience, on a scale from 1 to 5."
6. "Please describe your level of machine learning experience, on a scale from 1 to 5."
7. "Have you ever interacted with a robot before? Please describe:"

Post-Study Survey — Improve Trajectory. Users filled out this post-study survey after completing the Improve Trajectory experiment.

1. "Are you satisfied with the final trajectory?" [1 Completely unsatisfied - 5 Completely satisfied]
2. "How many iterations did it take for the robot to adapt to your feedback?"
3. "How fast does the robot adapt to your feedback?" [1 Very slow - 5 Very fast]
4. "How did the robot's learning capabilities affect your interaction experience?" [1 Negatively - 5 Positively]
5. "Overall, do you think the approach is effective?" [Y/N]
6. "Do you have any other comments?"

Post-Study Survey — Preference Learning. Users filled out this post-study survey after completing two methods of the preference learning experiment.

1. ”What did you like about these methods? For each aspect below, please click the circle if you found the method satisfactory in that regard:” [Adaptability, Time efficiency, Convenience, None of them]
2. ”Could you explain the reasons?”
3. ”How long did it take for the robot to adapt to your feedback?” [1 Very slowly - 5 Very quickly]
4. ”How did the robot’s learning capabilities affect your interaction experience?” [1 Negatively - 5 Positively]
5. ”Did you notice the robot learning or adapting to your behavior?” [1 Not at all - 5 Constantly]
6. ”How did the robot’s learning process affect your level of trust in its capabilities?” [1 Negatively - 5 Positively]
7. ”Overall, which method do you prefer?” [Pair-wise comparison/Language feedback]
8. ”Why do you prefer this method?”
9. ”Is there anything you wish the system would do but currently does not?”
10. ”Do you have any other comments?”

C Experiments — Additional Figures

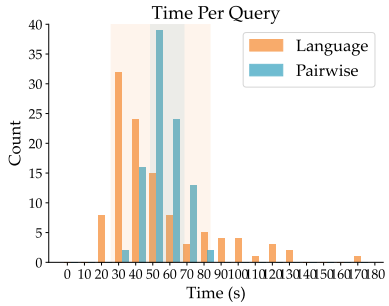


Figure 11: Time per query for both methods and averages, for 5 distinct users, indicated by color.

The error bar in Figure 11 indicate user-level differences of feedback length and time. The average of time of language-based reward learning is less than the comparison preference learning, illustrating that our method outperforms baseline in terms of the time-efficiency. However, the variance of language-based reward learning is larger than the baseline, which means users’ thinking time varies a lot based on the trajectory shown.