Optimizing Robot Behavior via Comparative Language Feedback

Jeremy Tien*
UC Berkeley
jtien@berkeley.edu

Zhaojing Yang*
University of Southern California
zyang966@usc.edu

Miru Jun
University of Southern California
mdjun@usc.edu

Stuart J. Russell
UC Berkeley
russell@berkeley.edu

Anca Dragan
UC Berkeley
anca@berkeley.edu

Erdem Bıyık
University of Southern California
biyik@usc.edu

ABSTRACT

Learning from human feedback has gained traction in fields like robotics and natural language processing (NLP) in recent years. While prior work mostly relies on human feedback in the form of comparisons, language can be a preferable feedback form that provides more informative insights into user preferences. This work aims to harness human language feedback to improve robot trajectories and infer human preferences. To achieve this goal, we learn a shared latent space that integrates trajectory data and language feedback. Our experiments demonstrate that by utilizing this learned latent space, we can effectively leverage human language feedback to improve robot trajectories.

ACM Reference Format:
Jeremy Tien, Zhaojing Yang*, Miru Jun, Stuart J. Russell, Anca Dragan, and Erdem Bıyık. 2024. Optimizing Robot Behavior via Comparative Language Feedback. In *Proceedings of (HRI'24 HIRL Workshop).* ACM, New York, NY, USA, 4 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Learning from human feedback has gained significant popularity in robotics over the past few years. Several different forms of human feedback have been studied: preference comparisons [1–4], rankings [5], physical corrections [6], visual saliency maps [7], human language [8, 9], etc. Among types of different human feedback, preference comparisons arose as one common approach due to its simplicity and ease from the perspective of the human users. In these methods, human users are presented with a query that consists of a pair of trajectories, requiring them to select their preferred option. Based on their selections, a reward function is learned, which is then used to train a better policy for the robot. Known as reinforcement learning from human feedback (RLHF) [2] or more generally preference-based learning [1, 10], this method has been applied successfully in a broad range of fields ranging from robotics [3, 11] to natural language processing [12], from routing [13] to human-computer interaction [14].

*The first two authors contributed equally and listed alphabetically.

HRI'24 HIRL Workshop, Mar 11–15, 2024, Boulder, CO
© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/XXXXXXX.XXXXXXX

Despite their success, preference-based learning methods suffer from several problems [15], such as the reliability of human data and the limited information bandwidth, i.e., each preference comparison requires the user to carefully analyze two trajectories but contains at most one bit of information [16]. Besides, both trajectories in a query may have their pros and cons, which makes it difficult for the user to give a single preference label. For instance, the human user may prefer the robot to move quickly but away from the fragile objects. When one trajectory moves quickly near the fragile objects and the other slowly but away from the objects, the human will have to decide the tradeoff between these two features. Instead, a better interface would allow the users to specify their preferences for each individual feature. Although there has been some research in this direction, they require features to be designed by hand [17].

As an alternative form of human feedback, language is considerably more informative than preference comparisons. It allows human users to reveal the specific aspects that they prioritize and to provide guidance on the desired direction for policy improvement. For example, it allows users to simply say "the robot should move faster" to indicate their preference about the robot's speed. While there are existing studies that leverage human language feedback to adjust robot trajectories [9, 18–20], it is noteworthy that no prior work has explored the potential of utilizing language feedback in preference-based robot learning frameworks. The incorporation of language feedback into the learning process offers a richer and more efficient understanding of users' preferences.

In this work, we aim to leverage human language feedback to improve robot trajectories with the ultimate goal of applying it for preference-based learning. In pursuit of this objective, we first learn a shared latent space that integrates trajectory data and language feedback. Such integration enables us to comprehend human language feedback and leverage it for adapting a robot's behavior to better align with the preferences of the human user. In the future, the learned features and the shared latent space will enable us to conduct preference-based learning with a single trajectory: we will use the human's language feedback to construct an imaginary trajectory in the latent space which is preferred over the trajectory shown to the human.

2 PROBLEM DEFINITION AND APPROACH

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 encodes human preferences regarding the task:

$$R(s_t, a_t) = w^\top \theta(s_t, a_t) \tag{1}$$

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

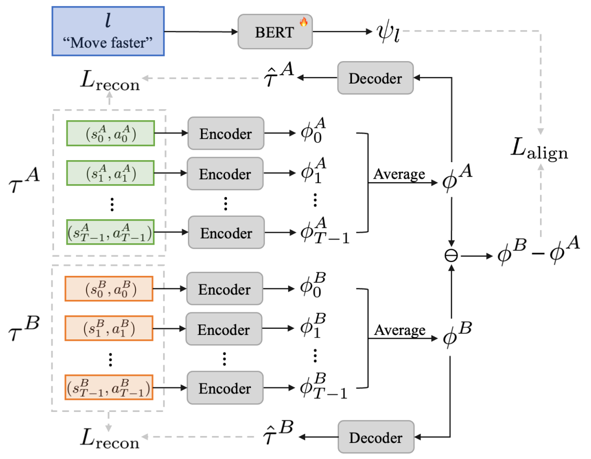

Figure 1: Overview of our architecture that learns a shared latent space between trajectories and language feedback

where θ is an unknown function that maps a state-action pair to its corresponding vector of features. Similarly, w is an unknown vector of weights that maps the features into a scalar reward value. The reward of a trajectory is then defined as:

$$R(\tau) = \sum_{t=0}^{T-1} R(s_t, a_t) = w^\top \sum_{t=0}^{T-1} \theta(s_t, a_t) = w^\top \theta(\tau) . \qquad (2)$$

For each trajectory, the human user 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. Our goal is to develop a framework where we optimize the robot's trajectory based on such feedback.

To achieve this, we take a learning-based approach: First, we collect a dataset that consists of pairs of robot trajectories and a language label describing their difference. We use this dataset to learn a shared latent space between trajectory features and language utterances. We then use this latent space to improve trajectories based on humans' language feedback.

Learning the Shared Latent Space. To learn a shared latent space for trajectories and language feedback, we propose the model shown in Figure 1. An encoder neural network encodes each state-action pair (s_t, a_t) from trajectories τ^A and τ^B to embeddings ϕ_t^A and ϕ_t^B, respectively. Our goal is to ensure these embeddings approximate $\theta(s_t^A, a_t^A)$ and $\theta(s_t^B, a_t^B)$ such that their averages $\phi^A := \frac{1}{T}\sum_{t=0}^{T-1} \phi_t^A$ (and similarly defined ϕ^B) will give $R(\tau^A)$ and $R(\tau^B)$ when multiplied with $w^\top$. Achieving this requires ϕ^A and ϕ^B to carry sufficient information about the trajectories. To encourage this, we employ an autoencoder-like architecture with a reconstruction loss:

$$L_{\text{recon}}(\tau^A, \tau^B) = d(\tau^A, \hat{\tau}^A) + d(\tau^B, \hat{\tau}^B) \qquad (3)$$

where $\hat{\tau}^A$ and $\hat{\tau}^B$ are trajectories (sequences of state-action pairs) reconstructed from ϕ^A and ϕ^B with a decoder neural network, and d is a distance function between the trajectories. In this work, we use the squared Euclidean distance between state-action pairs.

Although these reconstruction losses encourage ϕ^A and ϕ^B to carry high information about the trajectories, they do not align them with the true θ-vectors. For that, we use the language utterance l in the dataset that tells us the difference between the two trajectories τ^A and τ^B, e.g., "move faster" if τ^B is a faster trajectory than τ^A. Note this language annotation 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" even though the user wants the robot to move slowly.

We use a pretrained BERT model [21] to encode the language feedback l as ψ_l. To align features of trajectories with language feedback, we first freeze BERT model and train the trajectory encoder. Subsequently, we perform co-finetuning of both components. Now, aligning $\phi^B - \phi^A$ with ψ_l 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.

To achieve this alignment between $\phi^B - \phi^A$ and ψ_l, we use the following loss function in training:

$$L_{\text{align}}(\tau^A, \tau^B, l) = -\log\left(\text{sigmoid}\left(\psi_l^\top(\phi^B - \phi^A)\right)\right) \qquad (4)$$

where the sigmoid can be considered as the probability that language feedback l aligns with the difference between τ^A and τ^B.

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

$$L(\tau^A, \tau^B, l) = L_{\text{align}}(\tau^A, \tau^B, l) + L_{\text{recon}}(\tau^A, \tau^B) , \qquad (5)$$

which we use to train the encoder, the decoder and finetune BERT.

Training this architecture gives us the ability to encode any trajectory and language utterance in the same latent space. In the next section, we demonstrate how this latent space is useful for adapting robot trajectories based on humans' language feedback.

3 EXPERIMENTS

Environment. We conduct experiments using Robosuite 1.3 [22]. We created an environment with a Jaco robot arm at a table with a cube and a bottle. The states in this environment are 65-dimensional and the actions are 4-dimensional (the end-effector of the robot is set to always point down).

Training Dataset. We automated data collection process using simulated humans. For this, we first hand-crafted true $\theta(s, a)$, i.e., relevant features in the environment:

- 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. Note that we use these features only for automatically creating a dataset — training our architecture does not require hand-designing features.

For each of these features, we created a list of adjectives and their antonyms. For example, the height feature has the adjectives {higher, taller, at a greater height} and the antonyms to those adjectives {lower, shorter, at a lesser height}. We use each of these adjectives and their antonyms to construct a language feedback with the template: "Move [adjective/antonym]", leading to an initial language feedback dataset of size 32 (22 training and 10 validation).

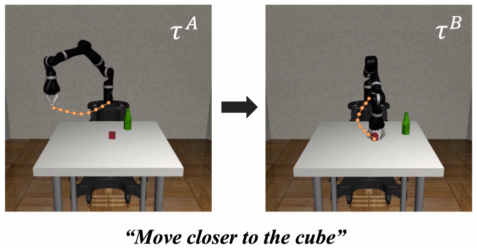

Figure 2: Each sample in the dataset consists of a pair of trajectories and a language utterance that corresponds to the difference between them.

The template varies based on specific attributes. For example, "Lift [adjective/antonym]" is used for feature height. We then augmented these datasets by using GPT 3.5 [23] to obtain a training set of 241 language feedback sentences and a validation set of 110 sentences. Of these sentences, 8 were shared between the training and validation sets since GPT 3.5 generated duplicate augmentations.

We then trained RL policies with randomized reward function weights w (via stratified sampling [24]) that operate over the handcrafted features listed above. We used these RL policies to make 218 unique rollouts with trajectory length $T = 500$. We used 196 of these trajectories as the training set and 22 as the validation set. Pairing each couple of trajectories from the training set yields to a total combination of 19,110 trajectory pairs.

Finally, we trained our architecture that learns a shared latent space between trajectories and language feedback using these 19,110 trajectory pairs augmented with the training set of language feedback sentences. An example data sample is depicted in Figure 2.

Ablation Study: BERT Variants. We trained our full architecture with 3 variants of BERT that have different model and output sizes: BERT-base, BERT-mini, and BERT-tiny [25]. We measured their success based on how well the learned models explain the language comparisons from the validation set between pairs of validation trajectories. Mathematically, we computed sigmoid $\left(\psi_l^\top(\phi^B - \phi^A)\right)$ for the validation set as the success metric, similar to L_{align} in training. The results are shown in Table 1. Noting that 0.5 would be a random-guess baseline, all variants seem to learn useful encoders. While BERT-base performs best, it is also computationally the most expensive due to larger model and output sizes.

Table 1: Validation Accuracy in Training

Variant	BERT-base	BERT-mini	BERT-tiny
Val Acc	0.871	0.862	0.861

Analysis: Improving Trajectories. To test the models trained with BERT-tiny, chosen due to its computational efficiency, we conduct an experiment where we try to iteratively improve an initial suboptimal trajectory.

For this experiment, we randomly initialize reward function weights w to simulate a human user. Let τ^* and τ^0 denote the optimal and the worst trajectories under reward w from the validation set. The simulated user is then shown trajectory τ^0 and asked for language feedback, which it responds with:

$$l^0 = \operatorname*{argmax}_l \, (\theta_l^* - \theta_l^0) \tag{6}$$

where θ^* and θ^0 denote the true cumulative features of τ^* and τ^0, respectively, and the subscript l chooses the entry of the vectors that correspond to the feature represented by language feedback l (and the negative of it if l is one of the antonyms). We also experiment with a softmax model instead of the argmax model to capture noisy feedback from humans.

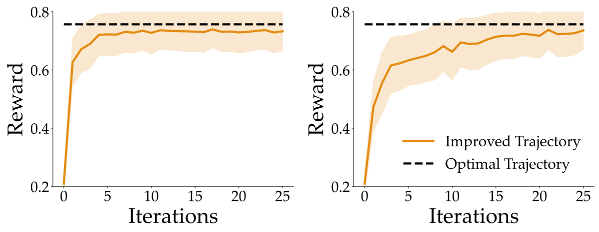

Figure 3: Results of experiments where we use simulated human language feedback to iteratively improve a robot trajectory (mean ± std over 100 runs). The dashed line represents *average* reward of optimal trajectories. (Left) Argmax human feedback model. (Right) Softmax human feedback model.

Upon receiving human's language feedback, we use our trained models to compute ϕ^0 and ψ_{l^0}. We then find the *improved trajectory* τ^1 from the validation set such that its difference with τ^0 best aligns with the human's language feedback based on cosine similarity:

$$\tau^1 = \operatorname*{argmax}_{\tau'} \frac{\psi_{l^0}^\top(\phi' - \phi^0)}{\|\phi'\|_2 \cdot \|\phi^0\|_2} \tag{7}$$

We iteratively continue this process for 15 iterations to obtain $\tau^0, \tau^1, \dots, \tau^{15}$. We repeat the full experiment over 100 random seeds. The true rewards of these trajectories are shown in Figure 3. With both argmax (noiseless) and softmax (noisy) human feedback models, we are able to iteratively improve a trajectory based on human's language feedback. As it is expected, the improvement is faster with noiseless feedback. Perhaps surprisingly, the approach does not suffer from the tradeoff between different features. For example, in this iterative approach, it would be possible to sacrifice some desired features for improving the trajectory based on the latest human feedback, which could cause some decrease in the reward. We believe we are not seeing such behavior due to the relatively small validation trajectory set size.

This result is a strong indicator that our architecture is able to learn trajectory features well and align them with language feedback. This will enable us to use these models for other human-in-the-loop learning methods, e.g., preference-based learning.

4 CONCLUSION AND FUTURE WORK

In this paper, we presented a framework that leverages human language feedback for improving robot trajectories by learning a shared latent space between trajectories and comparative language feedback. In the future, we are planning to develop preference-based learning methods that use human language feedback via our proposed architecture: consider a given trajectory with features ϕ_τ and its associated human language feedback ψ_l. We will generate an "imaginary" preferred trajectory whose features are: $\phi_\tau + \psi_l$. We will apply preference-based learning over this pair. To most efficiently achieve this, we need to develop a computational model of how humans select which feature to give language feedback about. Furthermore, we will conduct experiments in other simulation environments, as well as experiments with real robots.

REFERENCES

[1] Dorsa Sadigh, Anca D. Dragan, S. Shankar Sastry, and Sanjit A. Seshia. Active preference-based learning of reward functions. In *Proceedings of Robotics: Science and Systems (RSS)*, July 2017.

[2] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

[3] Erdem Bıyık, Dylan P Losey, Malayandi Palan, Nicholas C Landolfi, Gleb Shevchuk, and Dorsa 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.

[4] Nils Wilde and Erdem Biyik. Learning reward functions from scale feedback. In *Proceedings of the 5th Conference on Robot Learning (CoRL)*, 2021.

[5] Vivek Myers, Erdem Biyik, Nima Anari, and Dorsa Sadigh. Learning multimodal rewards from rankings. In *Proceedings of the 5th Conference on Robot Learning (CoRL)*, 2021.

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

[7] Anthony Liang, Jesse Thomason, and Erdem Biyik. Visarl: Visual reinforcement learning guided by human saliency. In *ICRA2023 Workshop on Pretraining for Robotics (PT4R)*, 2023.

[8] Jon Ander Campos and Jun Shern. Training language models with language feedback. In *ACL Workshop on Learning with Natural Language Supervision. 2022.*, 2022.

[9] Pratyusha Sharma, Balakumar Sundaralingam, Valts Blukis, Chris Paxton, Tucker Hermans, Antonio Torralba, Jacob Andreas, and Dieter Fox. Correcting robot plans with natural language feedback. *arXiv preprint arXiv:2204.05186*, 2022.

[10] Christian Wirth, Riad Akrour, Gerhard Neumann, Johannes Fürnkranz, et al. A survey of preference-based reinforcement learning methods. *Journal of Machine Learning Research*, 18(136):1–46, 2017.

[11] Yufei Wang, Zhanyi Sun, Jesse Zhang, Zhou Xian, Erdem Biyik, David Held, and Zackory Erickson. Rl-vlm-f: Reinforcement learning from vision language foundation model feedback. *arXiv preprint arXiv:2402.03681*, 2024.

[12] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

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

[14] Nathaniel Dennler, David Delgado, Daniel Zeng, Stefanos Nikolaidis, and Maja Matarić. The rosid tool: Empowering users to design multimodal signals for human-robot collaboration. In *18th International Symposium on Experimental Robotics (ISER)*, 2023.

[15] Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jeremy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, Tony Wang, Samuel Marks, Charbel-Raphaël Segerie, Micah Carroll, Andi Peng, Phillip Christoffersen, Mehul Damani, Stewart Slocum, Usman Anwar, Anand Siththaranjan, Max Nadeau, Eric J. Michaud, Jacob Pfau, Dmitrii Krasheninnikov, Xin Chen, Lauro Langosco, Peter Hase, Erdem Biyik, Anca Dragan, David Krueger, Dorsa Sadigh, and Dylan Hadfield-Menell. Open problems and fundamental limitations of reinforcement learning from human feedback. *Transactions on Machine Learning Research (TMLR)*, 2024.

[16] Erdem Biyik and Malayandi Palan. Asking easy questions: A user-friendly approach to active reward learning. In *Proceedings of the 3rd Conference on Robot Learning*, 2019.

[17] Chandrayee Basu, Mukesh Singhal, and Anca 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.

[18] Arthur Bucker, Luis Figueredo, Sami Haddadinl, Ashish Kapoor, Shuang Ma, and Rogerio 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.

[19] Arthur Bucker, Luis Figueredo, Sami Haddadin, Ashish Kapoor, Shuang Ma, Sai Vemprala, and Rogerio Bonatti. Latte: Language trajectory transformer. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7287–7294. IEEE, 2023.

[20] J Yow, Neha Priyadarshini Garg, Manoj Ramanathan, Wei Tech Ang, et al. Extract–explainable trajectory corrections from language inputs using textual description of features. *arXiv preprint arXiv:2401.03701*, 2024.

[21] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[22] Yuke Zhu, Josiah Wong, Ajay Mandlekar, Roberto Martín-Martín, Abhishek Joshi, Soroush Nasiriany, and Yifeng Zhu. robosuite: A modular simulation framework and benchmark for robot learning. In *arXiv preprint arXiv:2009.12293*, 2020.

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

[24] Mykel J Kochenderfer and Tim A Wheeler. *Algorithms for optimization*. Mit Press, 2019.

[25] Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Well-read students learn better: On the importance of pre-training compact models. *arXiv preprint arXiv:1908.08962*, 2019.