Masked Inverse Reinforcement Learning for Language Conditioned Reward Learning

Minyoung Hwang[1], Alexandra Forsey-Smerek[1], Andreea Bobu[1]
[1]MIT CSAIL

***Abstract*—Natural language provides a flexible interface for specifying robot tasks, but language-conditioned reward learning often assumes that instructions are unambiguous and directly informative. In reality, human language is frequently ambiguous — and may specify not just what to do, but also what matters in the environment. In this work, we propose a method that leverages this duality: we use large language models (LLMs) to extract state feature-level relevance masks from language and demonstrations, and train a reward function that is both conditioned on clarified task language and explicitly invariant to irrelevant parts of the state. We show that this approach improves generalization and sample efficiency in inverse reinforcement learning, particularly in settings with ambiguous instructions, distractor objects, or limited data. Our results highlight that disambiguating language with contextual demonstrations — and using language to guide both goal inference and state abstraction — enables more robust reward learning from natural instructions.**

Inverse Reinforcement Learning, Multi-Modal Feedback, Language Conditioning, Reward Learning

I. Introduction

In robotics, natural language provides a flexible and intuitive interface for specifying the tasks. However, language-conditioned reward learning typically assumes language instructions are clear and unambiguous. In practice, human language is often inherently ambiguous – an instruction can specify not only what the robot should do but also which elements of the environment matter for the task. Addressing this ambiguity is essential for effective reward learning from limited demonstrations and generalization to novel tasks or contexts.

Language-conditioned reward learning has gained significant interest in recent robotics literature. Fu *et al.* [9] show that learning a reward model conditioned on language yields behavior that transfers to novel tasks, whereas directly training a language-conditioned policy was less effective. Poddar *et al.* [23] learns a latent space that maps language instructions into hidden states to condition the reward model. Although language is frequently used as an additional modality in robot learning, existing approaches typically treat language simply as another input to a policy or reward model, without explicitly structuring the learning around the state features indicated as important by language. Consequently, these models implicitly infer feature relevance, which can lead to spurious correlations.

To address this gap, we propose a method that leverages the duality of language instructions in reward learning: their ability to specify tasks as well as to indicate relevant environmental state features. Specifically, our approach uses large language models (LLMs) to extract explicit relevance masks at the state feature level from language instructions and demonstrations during training. These masks identify which environmental features are task-relevant according to the provided instruction. Using these masks, we train a reward function conditioned on language instructions that explicitly ignores irrelevant state features. At inference time, our model can handle not only clear instructions but also ambiguous language instructions clarified by a single demonstration per instruction.

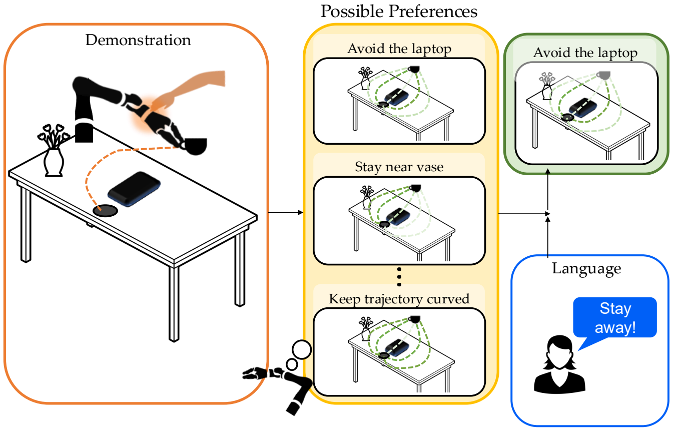


Fig. 1: Overview. For a robotic task, language can specify not only what to do, but also what matters in the environment. When there are multiple objects (e.g., vase and laptop) and human in the environment, an instruction "stay away from the laptop" states that 'laptop' is the only important feature. Even for ambiguous instructions such as "stay away", when combined with a contextual demonstration (blue trajectory), the instruction can be clarified to include the missing referent, e.g., laptop.

We introduce *Masked Inverse Reinforcement Learning* (Masked IRL), a framework that integrates human demonstrations and language instructions to explicitly guide feature selection for reward learning. Masked IRL leverages language-derived masks to dynamically gate relevant features in the reward function. For example, given the instruction "stay away from the table" in the scene in Fig. 1, our model explicitly ignores laptop-related features and irrelevant end-effector coordinates while emphasizing the vertical distance between the robot and table surface. We propose a masking loss that penalizes variations in reward predictions resulting from perturbations in state features indicated as irrelevant by the language. This builds upon the concept of contextual reliability by Ghosal *et al.* [10], explicitly training models to identify and ignore spurious or contextually irrelevant features.

By explicitly leveraging multimodal human feedback, Masked IRL substantially reduces demonstration requirements, improves sample efficiency, and enhances generalization by focusing solely on relevant task features. We empirically validate our approach using a PyBullet simulation environment with a Franka Emika Panda robotic arm. Our experiments highlight Masked IRL's effectiveness in settings with ambiguous language instructions, distractor objects, and limited demonstration data, demonstrating improved data efficiency, robustness, and generalization relative to standard IRL methods. In summary, our contributions are:

- A method using large language models (LLMs) to disambiguate language instructions and explicitly extract state-feature relevance masks from instructions paired with demonstrations.
- **Masked IRL**, an IRL framework that conditions rewards on clarified instructions and explicitly enforces invariance to irrelevant state features via a novel masking loss.
- Empirical validation of Masked IRL's effectiveness on simulated robotic manipulation tasks, demonstrating improved generalization, robustness, and data efficiency compared to traditional language-conditioned IRL approaches.

II. Related Work

A. Reward Learning from Human Feedback

Inverse reinforcement learning (IRL) learns reward functions from expert demonstrations. Early works [19], [8], [1], [30] have shown promising results in robotics but suffer a trade-off between the number of expert demonstrations and identifiability [20], [26], i.e., the required amount of demonstrations to identify the true objective function is huge. One fundamental limitation of IRL is that we can only train one reward function given a set of demonstrations, thereby requiring N set of demonstrations and N training processes to train N different reward functions. Bobu *et al.* [3] separates feature learning and reward learning, and uses human trajectory similarity queries to learn a task-agnostic feature space. However, they still require multiple demonstration sets for different user preferences and cannot generalize to unseen preferences. Beyond demonstrations alone, incorporating various modalities of human feedback (e.g., pairwise trajectory comparisons, language) has been shown to improve reward learning efficiency or reduce human's cognitive effort. Reinforcement learning from human feedback (RLHF) methods [6], [5] use pairwise human preferences to guide reward learning, but these methods often require thousands of human feedback to learn a single reward function [14]. Previous works [18], [28], [15] leverage API-based LLMs to generate a reward function as a code or predict weights on sub-rewards. Yu *et al.* [28] use an LLM as a Reward Translator, mapping high-level instructions into dense reward functions that standard RL can optimize. Recent works [23], [27] combine pairwise comparisons with language. Poddar *et al.* [23] highlight the need for personalized reward learning, arguing that aggregating human preferences can obscure individual human preferences. Their method learns a variational latent user model that personalizes rewards to individual users. Yang et al. [27] incorporates comparative language feedback, where humans describe which trajectory is better and why. Their model embeds trajectory-language pairs into a shared space, enabling iterative refinement of the reward function.

B. Language-Conditioned Learning in Robotics

Integrating natural language with robot learning has gained significant interest as a way to bridge human intent with robots. Recent methods leverage language as a conditioning signal in policy learning and reward modeling. Fu *et al.* [9] propose a language-conditioned reward learning approach in which IRL is used to ground language commands, showing that the resulting reward functions transfer better to novel tasks. In parallel, systems like LILAC [7] allow human operators to provide online language corrections during task execution. While such approaches have shown promising results, they often use language merely as an auxiliary input without explicit structure for feature selection. Language has become an essential modality for training robots, as it enables humans to specify goals, provide feedback, and guide behavior. One prominent approach is to condition policies or reward functions on language instructions. Ahn et al. [2] introduce the SayCan framework, which grounds high-level instructions using a large language model (LLM) and constrains execution using a value function, allowing robots to follow abstract human commands. Huang et al. [11] show that LLMs can serve as zero-shot planners by generating structured action sequences from instructions, while Huang et al. [12] introduce Inner Monologue, a framework that integrates environment feedback into LLM planning, significantly improving long-horizon task execution. Incorporating LLMs into robotic control has also gained traction. Liang et al. [17] propose Code-as-Policies (CaP), in which LLMs generate executable code (Python functions) for robotic policies, allowing for interpretable, structured control. This approach enables robots to generalize to unseen instructions by modifying their behavior through high-level program synthesis.

Beyond LLM-based planning, recent work has explored language-conditioned reward learning. Yu et al. [28] introduce Language to Rewards, where an LLM parses high-level instructions and outputs a parametric reward function, bridging natural language and robotic reinforcement learning. Karamcheti et al. [16] propose Voltron, a vision-language model for representation learning that aligns video frames with text descriptions, facilitating language-driven imitation learning. Hwang et al. [13] learn a success detector or a reward function that understands semantic grounding of robot motions. Other approaches integrate demonstrations with corrective language feedback to directly gate task-irrelevant features [7]. In such systems, language helps the robot focus on task-relevant features, thereby reducing the number of demonstrations needed and improving generalization. This multimodal feedback approach is especially promising in robotics, where safety and efficiency are paramount. Our work builds on these ideas by combining demonstration data with language instructions to guide a feature gating mechanism, leading to a reward model that is both data-efficient and robust.

C. Abstractions in Robot Learning

Another limitation of IRL comes from the spurious correlations of features. In many robotic tasks, not all sensory features are relevant for determining the reward. Ghosal *et al.*[10] aim to dynamically choose which features to rely on based on the current task or context. They have explored conditional gating mechanisms where a context variable modulates the importance of each input feature. Such approaches encourage sparsity in the feature set, thereby reducing the effective dimensionality of the learning problem. In robotics, this is particularly valuable since different tasks may require attention to different subsets of sensor modalities or object attributes. By integrating contextual reliability, one can obtain a more robust and interpretable model that adapts to the nuances of each task. Feature relevance varies with context, making contextual feature selection an essential component of robust learning. Unlike static feature selection, contextual feature selection dynamically selects which features are relevant based on auxiliary information such as the task or environment. In robotics, contextual feature selection is crucial for multi-task learning. Some skill learning frameworks enable robots to dynamically select relevant object attributes for different tasks, reducing learning complexity and improving generalization. Peng *et al.* [20] deals with overparameterization of reward by iteratively generating features and learning a reward on top of the current feature set. Peng *et al.* [20] uses language-guided contrastive explanations to iteratively extract and validate semantically meaningful features for the reward function. [21] uses background knowledge of language models to build state representations for unseen tasks. We aim to learn which state features matter under different user preferences, thereby improving sample efficiency and interpretability.

III. Problem Formulation

We consider the problem of learning reward functions that capture the unknown preferences held by a human given a small number of user demonstrations and language.

A. Preliminaries

We model our problem as a Markov Decision Process $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R} \rangle$ with states $s \in \mathcal{S}$, actions $a \in \mathcal{A}$, transition probability $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$, and rewards $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$. A solution to the MDP is a policy $\pi : \mathcal{S} \to \mathcal{A}$ that specifies what actions the robot should take in different states. The reward function is typically parameterized (e.g. a neural network) $\mathcal{R}_\theta(s)$, and is intended to capture the human's preference for how the robot should perform the task. To optimize task performance, the robot seeks a trajectory $\tau = \{s^0, \dots, s^T\}$ that maximizes the cumulative reward $\mathcal{R}_\theta(\tau) = \sum_{s^t \in \tau} \mathcal{R}_\theta(s^t)$ and executes the corresponding actions.

B. Maximum Entropy Inverse RL (MaxEnt IRL)

In practice, the reward function $\mathcal{R}_\theta$ is typically unknown to the robot or very challenging to manually specify. Thus, in IRL the robot's goal is to *learn* this reward function from human feedback, such as demonstrations. Given a dataset of human-demonstrated trajectories $\mathcal{D} = \{\tau_i\}_{i=1}^N$, the robot treats them as evidence of the human's preferred behavior and attempts to infer the reward parameters θ that explain the underlying objective. We adopt the maximum entropy (MaxEnt) framework for modeling human decision-making [30], [8], where the human is assumed to be a noisily optimal agent who selects trajectories with probability proportional to their exponentiated reward:

$$p(\tau \mid \theta) = \frac{e^{\mathcal{R}_\theta(\tau)}}{\int_{\bar{\tau}} e^{\mathcal{R}_\theta(\bar{\tau})} d\bar{\tau}} \propto \exp(R_\theta(\tau)) \tag{1}$$

This model captures the intuition that while humans generally act optimally, suboptimal trajectories are still possible, but occur with exponentially lower probability as their reward decreases [30]. To recover the reward parameters, we maximize the log-likelihood of the demonstrations:

$$\theta^* = \arg\max_\theta L(\theta) = \arg\max_\theta \sum_{\tau \in \mathcal{D}} \log p(\tau \mid \theta) \ . \tag{2}$$

Since the partition function in the denominator is intractable to compute exactly, we follow prior work [8], [4] and use importance sampling to approximate it. Once the reward is learned, the robot can act according to the policy that optimizes it. While MaxEnt IRL provides a principled framework for inferring rewards from demonstrations, learning a flexible reward function directly from high-dimensional states typically demands thousands of demonstrations per task [29], [24], [25], which is costly and impractical to scale. With limited data, learned rewards often capture spurious correlations between state features that accidentally co-occur with task success rather than reflecting true human intent. This fundamentally limits generalization, particularly in environments with distractors, ambiguous cues, or structural variations.

To address this, we propose leveraging natural language as an additional, structured form of supervision. Our key insight is that language plays a dual role in reward learning: 1) it *conveys information about the human's intent*, enabling a shared reward model to generalize across tasks via language conditioning; and 2) it *implicitly communicates which aspects of the state are task-relevant*, providing a signal for filtering out irrelevant environmental variation. By exploiting this natural duality, we learn a language-conditioned reward function that both shares structure across tasks and ignores spurious correlations, resulting in more generalizable rewards from significantly fewer demonstrations.

IV. Method

We present Masked Inverse Reinforcement Learning for Language Conditioned Reward Learning (Masked IRL), a method which leverages demonstrations paired with human language instructions to learn a language-conditioned reward function. Our approach exploits language's two distinct signals: language commands condition the preference captured by the reward model, and a language-informed masking loss is used to enforce invariance to task-irrelevant state aspects. We generate this mask directly from language commands and implement a masking loss that forces the reward function to ignore spurious state elements. By combining this masking loss with a language-conditioned architecture, Masked IRL achieves improved sample efficiency, requiring fewer demonstrations to learn generalizable rewards.

A. Preliminaries

We assume the human maintains a set of ground truth state features $\phi(s)$ which are only known to the human, not the observing agent. We assume the ground truth reward for preference i is a function of these features, $\mathcal{R}_i^*(\phi(s))$, where $\mathcal{R}_i^*(\tau) = \sum_{s^t \in \tau} \mathcal{R}_i^*(\phi(s^t))$. Given a set of training preferences $\mathcal{P}_{train} = \{1, 2, ..., N\}$, we collect a training dataset $\mathcal{D} = \{\tau_i, \ell_i\}_{i=1}^N$, where each paired demonstration τ_i and language command ℓ_i correspond to preference $i \in \mathcal{P}_i$. We aim to learn a general reward function $\mathcal{R}_\theta(s|\ell_j)$ that captures the ground truth reward for a new preference j where $j \notin \mathcal{P}_{train}$. Our goal is to learn a reward function that can generalize to unseen preferences given just a single language command ℓ_j. Since we lack access to the ground truth state features, our inferred reward is state-based $\mathcal{R}_\theta(\tau|\ell_j) = \sum_{s^t \in \tau} \mathcal{R}_\theta(s^t|\ell_j)$. We assume that all ground truth training and test preferences are functions of the same set of ground truth human features, representing a consistent intermediate representation unknown to the agent. We use language commands in our training dataset in two distinct ways. First we condition our model on these language inputs, following established practices in prior methods. Novel to our approach is our second usage – we convert language commands into state-based masks that inform a specialized training loss, promoting invariance to irrelevant state elements.

B. Language for State Masking

We extract state relevance from language by translating language commands into binary feature masks. For each demonstration-language pair $\{\tau, \ell\} \in \mathcal{D}$ we use language command ℓ to generate a binary mask $m \in \{0, 1\}^d$, where d is the dimension of the input state s. Each mask element is 1 for state indices relevant to the specified preference, and 0 otherwise. We augment our dataset with these language-generated masks to create $\mathcal{D}' = \{\tau_i, \ell_i, m_i\}_{i=1}^N$. These masks are produced by leveraging large language models ..

To ensure that the reward model is invariant to features deemed irrelevant by the language command, we introduce a masking loss. Let $s^{(j)}$ denote a perturbed version of state $s \in \tau$, where element j is modified (such as through the addition of Gaussian noise) and all other elements remain unchanged. The masking loss becomes

$$\mathcal{L}_{\text{mask}}(\theta) = \sum_{\tau, \ell, m \in \mathcal{D}'} \sum_{s \in \tau} \sum_{j=1}^{d} (1 - m_j) \Big(R_\theta(s^{(j)} \mid \ell) - R_\theta(s \mid \ell) \Big)^2, \tag{3}$$

where m_j represents the j^{th} element of m. This loss term penalizes changes in the reward when irrelevant features are perturbed, forcing the reward model to ignore these features.

The final training loss becomes

$$\mathcal{J}(\theta) = \mathcal{L}_{\text{IRL}}(\theta) + \lambda \, \mathcal{L}_{\text{mask}}(\theta), \tag{4}$$

where $\lambda > 0$ is a hyperparameter controlling the trade-off between fitting the demonstrations and enforcing invariance to irrelevant state elements.

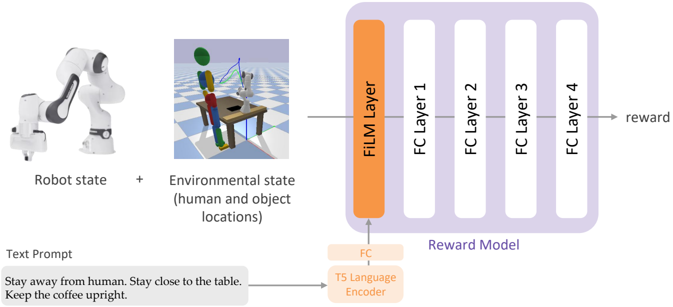

Robot state
+
Environmental state (human and object locations)
FiLM Layer
FC Layer 1
FC Layer 2
FC Layer 3
FC Layer 4
reward
Reward Model
Text Prompt
Stay away from human. Stay close to the table. Keep the coffee upright.
FC
T5 Language Encoder

Fig. 2: Network Architecture. We condition the reward model on language instructions using FiLM layers. The conditioned reward model infers a scalar reward of a robot's 9-dim state.

C. Masked IRL for Language Conditioned Reward Learning

We pair our masking loss with a language-conditioned architecture to additionally leverage the intent captured by language instructions. Specifically, we apply Feature-wise Linear Modulation (FiLM) [22] to the first fully connected (FC) layer of the reward model (see Fig. 2) to condition the reward model based on the language inputs. This FiLM layer applies language-dependent affine transformations to intermediate network features, allowing language commands to dynamically modulate reward components directly. As opposed to simply concatenating the language command with the input state, FiLM targets conditioning input to explicitly modulate intermediate network features, providing the ability to scale features, negate them, or shut them off entirely. This method enables using language for a dual purpose: both as a gating mechanism that filters out irrelevant state aspects, and as an adaptation function that adjusts intermediate feature weights based on the preference captured in language. Algorithm 1 shows the training procedure for Masked IRL.

D. Clarifying Ambiguous Language Instructions

For ambiguous language instructions, we systematically generate the instructions within two types: (1) referent omitted and (2) expression omitted. Referent omitted instructions do not include the object that the user actually cares about, and only include instructions such as "stay away", "stay close", and "carry it upright". Expression omitted instructions have the information about what object the user wants to refer to, but does not mention how the user wants the relationship between the robot and the object to be. For instance, "table", "laptop", or "human" can be expression omitted instructions. To generate state masks from ambiguous instructions, we provide the information of a demonstration trajectory as tabular data in text, along with the instruction to an LLM. We use Chain-of-thought reasoning to let LLM generate the response step-by-step, including its reasoning process to generate the clarified language instruction. For instance, given the instruction 'stay away' and a demonstration where the robot moves away from the table, the LLM might reason: 'The robot avoids the table. Therefore, the instruction likely refers to avoiding the table.' The clarified instruction becomes 'stay away from the table', which is then mapped to a binary mask emphasizing the end effector's z position and de-emphasizing human or laptop locations. Then, we query the LLM again to convert the clarified language instruction into a 9-dim state mask that represents the importance of each state dimension.

Algorithm 1: Masked IRL with Language Conditioning

```
Input: Demonstrations $\{(\tau_i, \theta_i)\}_{i=1}^{N}$, training
       trajectories $\mathcal{T}$, language encoder $E$, reward
       network $R$, learning rate $\eta$, iterations $I$, batch
       size $B$, masked loss weight $\lambda$, noise scale $\sigma$.
for epoch 1 to I do
    Shuffle demo and training indices
    for each minibatch $b$ of size $B$ do
        Form demo inputs: $X_d^b = \{(\bar{s}_i, c_i)\}$ and
          compute cost $C_d^b = R(X_d^b)$
        Form training inputs $X_t^b = \{(\bar{s}_j, c_j)\}$ with cost
          $C_t^b = R(X_t^b)$
        Compute maxent loss:
          $\mathcal{L}_{\text{IRL}}^b = \text{mean}(C_d^b) + \log\Big(\text{mean}(\exp(-C_t^b))\Big)$
          Perturb demo states:
          $\bar{s}'_i = \bar{s}_i + \epsilon,\ \epsilon \sim \mathcal{N}(0, \sigma^2 I)$ (only in
          dimensions where $\Pi(\theta_i) = 0$);
        Compute perturbed cost $C'^{\,b}_d = R(\{\bar{s}'_i, c_i\})$ and
          masked loss:
          $\mathcal{L}_{\text{mask}}^b = \text{mean}\Big(\big|C_d^b - C'^{\,b}_d\big|\Big)$
          Update parameters:
          $\theta \leftarrow \theta - \eta\nabla\Big(\mathcal{L}_{\text{IRL}}^b + \lambda\,\mathcal{L}_{\text{mask}}^b\Big)$
    end
end
return $\theta$.
```

V. Experiments

We evaluate our method on a robotic task to move a coffee from a start to a goal location in a PyBullet simulator, where there is a human, a table, and a laptop in the environment. Each state consists of the position and rotation of the robot's end effector, objects (table and laptop), and a human in the environment. In each task, only a subset of features is relevant to the reward. Human instructions (e.g., "stay away from the laptop") are provided to guide the feature gating.

A. Dataset.

We generate a dataset of 100 start-goal pairs for a task of moving a coffee mug, each with 100 robot trajectories, in PyBullet simulator. We also generate 242 language instructions that are mapped into ground truth reward functions that define human preferences. For clear language instructions, we construct the dataset with 193 train instructions and 49 test instructions. Each instruction has a corresponding 5-dim theta value that describes human's ground truth reward function. We use GPT-4o API to infer the state mask only from each clear instruction, without any information about the ground truth reward. For inferring state masks from ambiguous instructions, we pair each instruction with its corresponding expert demonstration and pass the information of the language instruction and demonstration to GPT-4o as described in Section IV-D. We train each model with 1 to 10 demonstrations per human preference.

B. Baselines

Traditional IRL learns a single reward function from demonstrations, without contextual modulation. This often results in a reward model that uses all features indiscriminately, making it vulnerable to spurious correlations when demonstrations cover multiple tasks or environments. To demonstrate the effectiveness of masking loss, we compare Masked IRL and MaxEnt IRL on two different types of reward model - single model and multiple model. We refer to 'single model' as a language-conditioned reward model, regardless of the usage of masking loss. 'Multiple model' refers to a set of language-unconditioned reward models, where each element of the set is a reward model that corresponds to a specific language instruction, i.e., user preference.

For multiple model approaches, we compare:

- **MaxEnt IRL (No Language, multiple model).** We train a 3-layer MLP that inputs a 9-dimensional state and outputs a scalar reward value for each state. We train this baseline with standard maximum entropy loss.
- **Masked IRL (No Language, multiple model).** Same architecture as the baseline but uses the weighted masking loss in addition to the maximum entropy loss for training.

For single model approaches, we compare:

- **MaxEnt IRL (Language-conditioned, single model).** The 3-layer MLP reward model is conditioned on language embedding using FiLM. We use the standard maximum entropy loss function to train this model.
- **Masked IRL (Language-conditioned, single model).** Same architecture as the baseline but uses the weighted masking loss in addition to the maximum entropy loss for training.

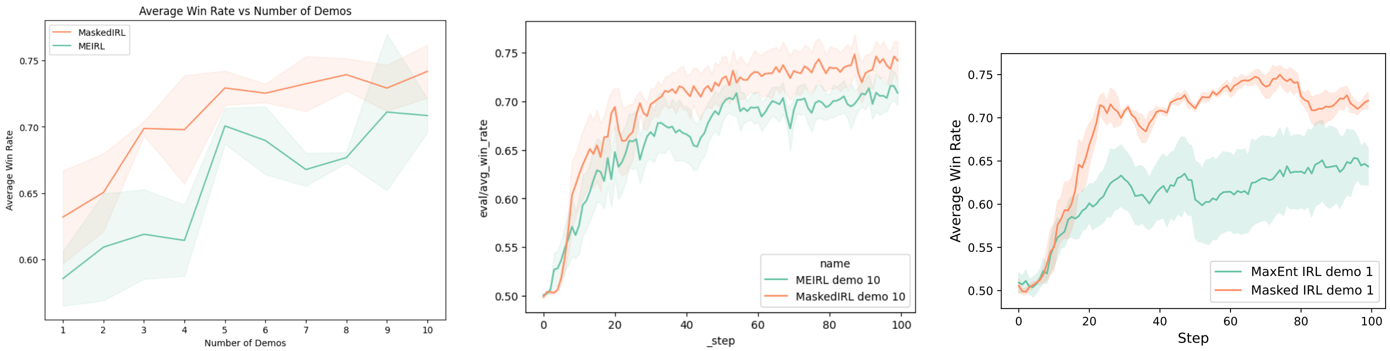
For multiple model methods, we only evaluate on seen human preferences, since unseen human preferences do not have any corresponding trained reward model. However, for single model methods, we evaluate on unseen human preferences, i.e., language instructions.

C. Evaluation Metrics.

We evaluate all models by calculating the average win rate, where the average win rate measures how often our learned reward model correctly prefers better trajectories compared to ground-truth preferences. We run all experiments with 5 different random seeds (12345, 23451, 34512, 45123, and 51234) and show the average and standard error across seeds.

D. Results

The effectiveness of Masking Loss on Multiple Model. Fig. 4 shows the effect of having masking loss in multiple model methods. Interestingly, the performance improvement by using masking loss is maximized when the number of valid features for the ground truth reward of the simulated human is minimized to 1. As the number of valid features increases, the



(a) Single model performance per # demonstrations

(b) Single model learning curve when # demonstrations = 10

(c) Single model learning curve when # demonstrations = 1

Fig. 3: Experiment Results. (a) shows the average win rate of single model methods and compares the performance between Masked IRL and MaxEnt IRL. (b) shows the average win rate of single model methods given 10 demonstrations per user preference as training proceeds. (c) shows the learning curve in case given 1 demonstrations per user preference. All error bars show the standard error across 5 different seeds (12345, 23451, 34512, 45123, and 51234). 'MEIRL' in the figures denote MaxEnt IRL.

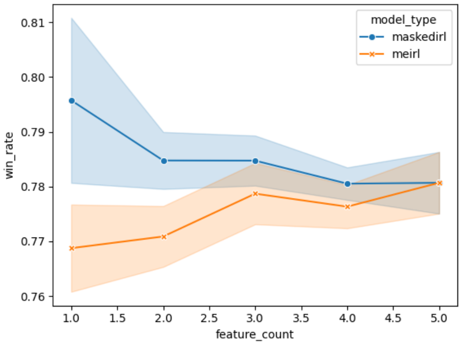


Fig. 4: Multiple model performance per feature counts. Comparing Masked IRL to MaxEnt IRL for multiple model baselines, the average win rate is improved the most when there are least number of valid features in the ground truth reward of the simulated human. As the number of valid features increase from 1 to 5, the performance gap between Masked IRL and MaxEnt IRL decreases.

gap between Masked IRL and MaxEnt IRL decreases. This is a desired behavior because when all features are valid, i.e., all state dimensions are relevant to the instruction,

The effectiveness of Masking Loss on Single Model. Fig. 3 (a) shows the average win rate over the number of demonstrations. As the number of demonstrations increases, both Masked IRL and MaxEnt IRL shows an overall trend of increasing performance. In all numbers of demonstrations, Masked IRL outperforms MaxEnt IRL. As shown in Fig. 3 (b), Masked IRL not only converges to a higher average win rate, but also converges faster than MaxEnt IRL. Also, Fig. 3 (c) shows an interesting result that given a single demonstration per user preference, Masked IRL shows more stable performance with lower variance while generalizing better to unseen preferences.

Performance on ambiguous instructions. When we use our Masked IRL single model trained with 10 demonstrations per human preference to evaluate trajectories given a single ambiguous language instruction and an expert demonstration to disambiguate language, we get an average win rate of 63.1% on the instructions. The lower performance compared to the performance on clear test instructions may be due to the inaccuracy of clarifying ambiguous instructions to clear instructions using LLMs.

Future Work and Limitations Although our Masked IRL framework effectively improves generalization and sample efficiency, several limitations remain. First, our reliance on LLMs introduces potential inaccuracies in generating relevance masks, particularly when instructions are ambiguous or nuanced, which can affect the overall robustness of the reward model. Future work could explore methods for refining mask accuracy through interactive human feedback or advanced prompting strategies. Additionally, our current evaluations focus on relatively constrained robotic tasks; extending the approach to more complex, dynamic, or multi-agent environments could further validate the generality of Masked IRL. Lastly, investigating ways to integrate explicit uncertainty estimation in the masking process could enhance the reliability of our approach in real-world deployments.

References

[1] Pieter Abbeel and Andrew Y. Ng. Apprenticeship Learning via Inverse Reinforcement Learning. In *Proceedings of the International Conference on Machine learning*, 2004.

[2] Michael Ahn, Anthony Brohan, Noah Brown, and et al. Do as i can, not as i say: Grounding language in robotic affordances. In *Conference on Robot Learning (CoRL)*, 2022.

[3] Andreea Bobu, Yi Liu, Rohin Shah, Daniel S Brown, and Anca D Dragan. Sirl: Similarity-based implicit representation learning. In *Proceedings of the 2023 ACM/IEEE International Conference on Human-Robot Interaction*, pages 565–574, 2023.

[4] Andreea Bobu, Marius Wiggert, Claire Tomlin, and Anca D Dragan. Inducing structure in reward learning by learning features. *The International Journal of Robotics Research*, 41(5):497–518, 2022.

[5] Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, et al. Open problems and fundamental limitations of reinforcement learning from human feedback. *arXiv preprint arXiv:2307.15217*, 2023.

[6] Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 4299–4307, 2017.

[7] Yuchen Cui, Siddharth Karamcheti, et al. "no, to the right" – online language corrections for robotic manipulation via shared autonomy. In *Proceedings of HRI 2023*, 2023.

[8] Chelsea Finn, Sergey Levine, and Pieter Abbeel. Guided cost learning: Deep inverse optimal control via policy optimization. In *International conference on machine learning*, pages 49–58. PMLR, 2016.

[9] Justin Fu et al. From language to goals: Inverse reinforcement learning for vision-based instruction following. In *International Conference on Learning Representations (ICLR)*, 2019.

[10] Gaurav Ghosal, Amrith Setlur, Daniel S. Brown, Anca D. Dragan, and Aditi Raghunathan. Contextual reliability: When different features matter in different contexts. In *Proceedings of ICML*, 2023.

[11] Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International Conference on Machine Learning (ICML)*, 2022.

[12] Wenlong Huang, Fei Xia, Ted Xiao, and et al. Inner monologue: Embodied reasoning through planning with language models. In *Conference on Robot Learning (CoRL)*, 2022.

[13] Minyoung Hwang, Joey Hejna, Dorsa Sadigh, and Yonatan Bisk. Motif: Motion instruction fine-tuning. *IEEE Robotics and Automation Letters*, 2025.

[14] Minyoung Hwang, Gunmin Lee, Hogun Kee, Chan Woo Kim, Kyungjae Lee, and Songhwai Oh. Sequential preference ranking for efficient reinforcement learning from human feedback. *Advances in Neural Information Processing Systems*, 36:49088–49099, 2023.

[15] Minyoung Hwang, Luca Weihs, Chanwoo Park, Kimin Lee, Aniruddha Kembhavi, and Kiana Ehsani. Promptable behaviors: Personalizing multi-objective rewards from human preferences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16216–16226, 2024.

[16] Siddharth Karamcheti, Suraj Nair, Annie S. Chen, Thomas Kollar, Chelsea Finn, Dorsa Sadigh, and Percy Liang. Language-driven representation learning for robotics. In *Robotics: Science and Systems (RSS)*, 2023.

[17] Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[18] Yecheng Jason Ma, William Liang, Guanzhi Wang, De-An Huang, Osbert Bastani, Dinesh Jayaraman, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Eureka: Human-level reward design via coding large language models. *arXiv preprint arXiv:2310.12931*, 2023.

[19] Andrew Y. Ng and Stuart Russell. Algorithms for inverse reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, 2000.

[20] Andi Peng, Belinda Z Li, Ilia Sucholutsky, Nishanth Kumar, Julie A Shah, Jacob Andreas, and Andreea Bobu. Adaptive language-guided abstraction from contrastive explanations. *Conference on Robot Learning (CoRL)*, 2024.

[21] Andi Peng, Ilia Sucholutsky, Belinda Li, Theodore Sumers, Thomas Griffiths, Jacob Andreas, and Julie Shah. Learning with language-guided state abstractions. In *International Conference on Learning Representations*, 2024.

[22] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[23] Sriyash Poddar, Yanming Wan, Hamish Ivison, Abhishek Gupta, and Natasha Jaques. Personalizing reinforcement learning from human feedback with variational preference learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[24] Rouhollah Rahmatizadeh, Pooya Abolghasemi, Ladislau Bölöni, and Sergey Levine. Vision-based multi-task manipulation for inexpensive robots using end-to-end learning from demonstration. In *2018 IEEE International Conference on Robotics and Automation, ICRA 2018, Brisbane, Australia, May 21-25, 2018*, pages 3758–3765. IEEE, 2018.

[25] Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. In Hadas Kress-Gazit, Siddhartha S. Srinivasa, Tom Howard, and Nikolay Atanasov, editors, *Robotics: Science and Systems XIV, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA, June 26-30, 2018*, 2018.

[26] Theodore Sumers, Robert Hawkins, Mark K Ho, Tom Griffiths, and Dylan Hadfield-Menell. How to talk so ai will learn: Instructions, descriptions, and autonomy. *Advances in Neural Information Processing Systems*, 35:34762–34775, 2022.

[27] Zhaojing Yang, Miru Jun, Jeremy Tien, Stuart J. Russell, Anca D. Dragan, and Erdem Bıyık. Trajectory improvement and reward learning from comparative language feedback. 2024.

[28] Wenhao Yu, Nimrod Gileadi, Chuyuan Fu, and et al. Language to rewards for robotic skill synthesis. In *Conference on Robot Learning (CoRL)*, 2023.

[29] Tianhao Zhang, Zoe McCarthy, Owen Jow, Dennis Lee, Xi Chen, Ken Goldberg, and Pieter Abbeel. Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. In *2018 IEEE International Conference on Robotics and Automation, ICRA 2018, Brisbane, Australia, May 21-25, 2018*, pages 1–8. IEEE, 2018.

[30] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, et al. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pages 1433–1438. Chicago, IL, USA, 2008.