Masked IRL: LLM-Guided Reward Disambiguation from Demonstrations and Language

Minyoung Hwang, Alexandra Forsey-Smerek, Nathaniel S. Dennler, Andreea Bobu

***Abstract*—Robots can adapt to user preferences by learning reward functions from demonstrations, but with limited data, reward models often overfit to spurious correlations and fail to generalize. This happens because demonstrations show robots how to do a task but not what matters for that task, causing the model to focus on irrelevant state details. Natural language can more directly specify what the robot should focus on, and, in principle, disambiguate between many reward functions consistent with the demonstrations. However, existing language-conditioned reward learning methods typically treat instructions as simple conditioning signals, without fully exploiting their potential to resolve ambiguity. Moreover, real instructions are often ambiguous themselves, so naive conditioning is unreliable. Our key insight is that these two input types carry complementary information: demonstrations show *how* to act, while language specifies *what* is important. We propose *Masked Inverse Reinforcement Learning (Masked IRL)*, a framework that uses large language models (LLMs) to combine the strengths of both input types. Masked IRL infers state-relevance masks from language instructions and enforces invariance to irrelevant state components. When instructions are ambiguous, it uses LLM reasoning to clarify them in the context of the demonstrations. In simulation and on a real robot, Masked IRL outperforms prior language-conditioned IRL methods by up to 15% while using up to 4.7 times less data, demonstrating improved sample-efficiency, generalization, and robustness to ambiguous language.**
Project page and Code: https://github.com/MIT-CLEAR-Lab/Masked-IRL

I. Introduction

Robots can learn how to do tasks for people by learning reward functions from user demonstrations, but reward learning is fundamentally ill-posed: many reward functions can explain the same demonstration. For instance, in the example in Fig. 1, from the demonstration alone the robot could infer it should prioritize staying close to the human, avoiding the laptop, both, or something else entirely, like producing curved trajectories. While more data could help resolve this ambiguity, in practice demonstrations are costly and difficult to collect in sufficient diversity. As such, reward models often overfit, latching onto *spurious correlations* in the demonstrations rather than capturing true user intent [5].

The core issue is that, while demonstrations show robots how to perform a task, they don't explicitly convey what matters for the task. Natural language (e.g., "Stay away from my laptop") could address this challenge by directly specifying what the robot should focus on and, in principle, help disambiguate between the many reward functions consistent with the demonstrations. However, existing language-conditioned reward learning methods treat language utterances as simple conditioning signals for multitask learning [11], without fully exploiting their potential to resolve ambiguity. Moreover, real instructions are often underspecified or ambiguous: if the user in Fig. 1 simply says "Stay away", the robot cannot determine whether to avoid the laptop, table, or human. In summary, both demonstrations and language alone are insufficient for reliable reward learning.

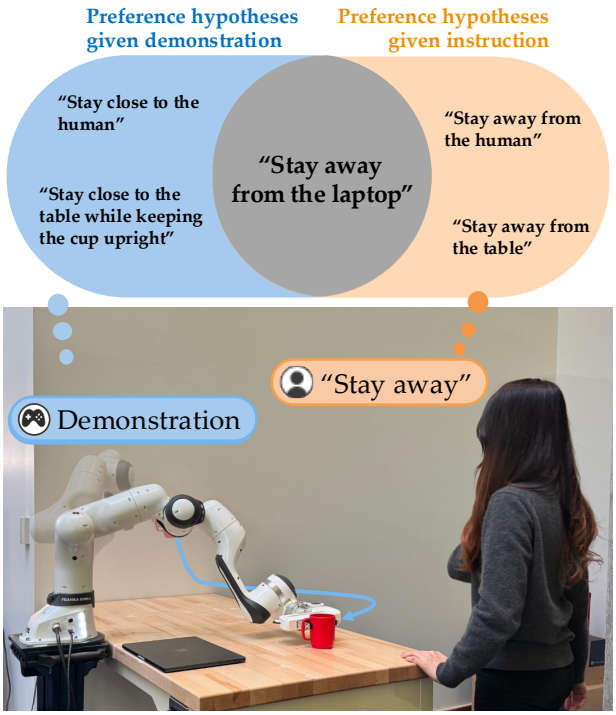


Fig. 1: Overview. Demonstrations show how to complete a task, but the same demonstration can be supported by many reward hypotheses. Language can be leveraged to disambiguate what matters in the environment. Even when both the demonstration (blue trajectory) and the associated instruction (e.g., "Stay away") are individually ambiguous, when reasoning jointly about the pair they can often disambiguate each other, revealing the intended preference ("Stay away from the laptop").

Our key insight is that these two input types are complementary: demonstrations show *how* to act, while language specifies *what* is important. From the example shown in Fig. 1, if the robot reasoned jointly about the

demonstration and the instruction, it could infer that the human meant to stay away from the laptop, and thus learn the intended preference. To enable this kind of joint reasoning, we need methods that can both extract what matters from language and clarify ambiguous instructions in the context of demonstrations.

We introduce *Masked Inverse Reinforcement Learning (Masked IRL)*, a multitask reward learning framework that integrates demonstrations and natural language instructions to overcome the limitations of existing language-conditioned reward learning. Whereas prior approaches use language solely to condition rewards across multiple preferences, Masked IRL additionally exploits language to resolve ambiguity when instructions are underspecified. Specifically, our method uses LLMs in two ways: (i) to infer state-relevance masks from language instructions, enabling a masking loss that enforces invariance to irrelevant state components and reduces spurious correlations; and (ii) to clarify ambiguous instructions by using information from demonstrations, allowing reward models to remain reliable even when language is underspecified. In both simulation and real-robot experiments with a 7DoF arm, we show that combining these complementary forms of human feedback enables our method to recover more generalizable rewards while requiring up to 4.7× fewer demonstrations than prior language-conditioned approaches.

In summary, our contributions are: **(1)** Introducing language-guided state relevance masks and a novel masking loss that improves sample efficiency in IRL, **(2)** Developing an LLM-based disambiguation mechanism that clarifies underspecified instructions using demonstrations, and **(3)** Demonstrating robust generalization in both simulation and real-robot experiments.

II. Related Work

Reward Learning from Human Feedback. An effective approach for learning robot tasks is inferring a policy or reward function from human inputs like demonstrations [25], corrections [2, 17], teleoperation [26], comparisons [7], or trajectory rankings [6, 10], among others [9, 4]. To learn in a tractable way from such human data, classical IRL methods rely on hand-specified feature functions [25], but poorly chosen features risk misalignment with human intent that produces unsafe behavior [16]. Deep IRL methods mitigate this assumption by learning directly from raw state, but require a large numbers of demonstrations to avoid overfitting to spurious correlations in irrelevant state components [5]. These challenges make standard IRL impractical for settings where robots must adapt to diverse user preferences.

To reduce the human burden of collecting demonstrations, reinforcement learning from human feedback (RLHF) uses pairwise trajectory comparisons [7]. While these labels are easier for humans to provide than demonstrations, they contain at most one bit of information about the human's internal reward. Thus, RLHF often requires thousands of feedback queries to learn a single reward function [13]. Recent work explores leveraging API-based LLMs to generate reward functions directly [24, 14], for example by translating high-level instructions into dense rewards that can be optimized with RL [24]. Other approaches focus on personalization, either by separating feature learning from reward learning [3, 5] or by modeling latent user preferences from feedback [23].

Although these methods improve efficiency for personalization, they still require training a new reward function for each user preference that do not generalize to unseen instructions. In contrast, our work learns a *single* language-conditioned reward model that generalizes across preferences by using language as structured supervision.

Language-Conditioned Learning in Robotics. Language provides a natural interface for specifying goals, feedback, and constraints in robot learning, and recent work has explored conditioning policies and rewards on natural language [11, 8]. Fu et al. [11] introduced a language-conditioned reward learning approach, grounding instructions through IRL to improve transfer to novel tasks. Systems such as LILAC [8] allow operators to provide online language corrections during task execution, demonstrating how language can adapt behavior in real time. However, these methods assume instructions are clear and unambiguous, which limits their adaptability when language is vague or context-dependent.

While LLMs have recently bridged language and robotic control through high-level planning [1, 15] and reward specification [24, 12], existing approaches typically treat instructions as static inputs. Consequently, these frameworks often fail to reason about state relevance or dynamically resolve underspecified commands. In contrast to existing language-conditioned learning methods, our work uses LLMs not only to condition a shared reward model, but also to structure learning by generating state relevance masks and clarifying ambiguous instructions. This enables us to use language both as a conditioning signal and as a supervisory cue for which state elements matter, reducing data requirements.

III. Problem Formulation

Our goal is to learn a single reward function that captures diverse human task preferences from a minimal amount of language-labeled demonstrations.

Preliminaries. We build on the IRL framework where a human's task preference is represented as a reward function in a Markov Decision Process (MDP) [20] $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{T}, r \rangle$ 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 r :

$\mathcal{S} \to \mathbb{R}$. A solution to the MDP is a policy $\pi : \mathcal{S} \to \mathcal{A}$ that maximizes the reward and specifies what actions the robot should take in every state. The robot executes trajectories $\tau = \{s^0, \ldots, s^T\}$ according to the policy.

Since the human's reward function is not known *a priori*, IRL attempts to learn it from data. In realistic settings, robots must handle *many different user preferences*, each corresponding to a different underlying reward. Training a separate reward model for each preference requires extensive data, motivating a multitask formulation where a single model can generalize across preferences.

Language-Conditioned Reward Learning (LC-RL). Language offers a natural interface for multitask reward learning by conditioning the reward on user preferences in the form of language commands [11]. Specifically, we consider the setting where a robot must learn a language-conditioned reward function that captures a set of human preferences $\mathcal{P} = \{P_1, \ldots, P_N\}$. For each preference $P_i \in \mathcal{P}$, the human gives a set $\mathcal{D}_i = \{(\tau_i^k, \ell_i^k)\}_{k=1}^{M_i}$ of *language-labeled demonstrations*. The overall training dataset is then $\mathcal{D} = \bigcup_{i=1}^{N} \mathcal{D}_i$.

We parameterize the reward as a language-conditioned function $r_\theta(s \mid \ell)$, and aim to learn θ from demonstration-language pairs. The robot can then perform the task according to the preference represented by language command ℓ by selecting a trajectory τ that maximizes the cumulative reward $\mathcal{R}_\theta(\tau \mid \ell) = \sum_{s \in \tau} r_\theta(s \mid \ell)$.

Using language-labeled demonstrations $\mathcal{D}$, the robot infers reward parameters θ that define the human's underlying objective function. Inspired by prior work on language-conditioned reward learning [11], we train our reward model using the standard Maximum Entropy IRL objective [25]. We model the human as a noisily rational agent who selects trajectories with probability proportional to their exponentiated reward:

$$p(\tau \mid \ell, \theta) = \frac{e^{\mathcal{R}_\theta(\tau|\ell)}}{\int_{\bar{\tau}} e^{\mathcal{R}_\theta(\bar{\tau}|\ell)} d\bar{\tau}} \propto \exp(\mathcal{R}_\theta(\tau \mid \ell)) \ , \quad (1)$$

where ℓ captures the human's personal preference. To recover the reward parameters, we minimize the negative log-likelihood of the demonstrations via gradient descent:

$$\theta^* = \arg\min_\theta \mathcal{L}_{\text{IRL}}(\theta) = \arg\min_\theta \Big(- \sum_{\tau, \ell \in \mathcal{D}} \log p(\tau \mid \ell, \theta) \Big) \ . \quad (2)$$

To optimize this objective, we approximate the intractable integral in Eq. (1) using importance sampling as in prior work [5].

Limitations of LC-RL. While LC-RL provides a principled framework for inferring rewards from language-demonstration pairs, LC-RL requires high sample complexity [11] and often leads to spurious correlations and overfitting in low-data regimes. Furthermore, the inherent ambiguity of natural language makes it an unreliable standalone signal for capturing precise human intent.

In this work we address these challenges by leveraging two complementary properties of language and demonstrations to improve sample efficiency and avoid spurious correlations. First, language plays a dual role in reward learning: it specifies not only *what task the human wants the robot to do*, enabling a single reward model to generalize across tasks, but also implicitly indicates *which aspects of the environment matter for the task*, providing a signal for filtering out irrelevant state components and improving sample efficiency. Second, when language commands are ambiguous, examining language in the context of demonstrations can ground instructions and resolve ambiguity.

IV. Method

We present **Masked Inverse Reinforcement Learning (Masked IRL)**, a method that leverages demonstrations paired with language instructions to efficiently learn a language-conditioned reward function. Our key contribution is to exploit common-sense priors encoded in LLMs in two complementary ways: (1) to generate *relevance masks* from demonstrations paired with language (Sec. IV-A), which define a masking loss that enforces invariance to irrelevant state components (Sec. IV-B); and (2) to enable training on *ambiguous instructions* by reasoning about demonstrations and language in context, enabling robust reward learning even when language underspecifies the preference (Sec. IV-C). The underlying structure of our approach is a language-conditioned reward model that captures shared structure across multiple preferences (Sec. IV-D). Masked IRL remains robust under limited feedback and leverages LLM reasoning to resolve ambiguity in natural language commands. Fig. 2 summarizes our Masked IRL pipeline.

A. Generating State Masks from Language

While language conditioning provides a shared reward model across preferences, it does not by itself prevent the model from exploiting spurious correlations in irrelevant state components. To address this, we leverage LLMs to generate *state relevance masks* that indicate which elements of the state vector s are relevant to the instruction in context. For each demonstration-language pair $\{\tau, \ell\} \in \mathcal{D}$ we query an LLM with both the command ℓ and a description of the robot and environment state, asking it to identify which state components matter for satisfying the instruction. The LLM outputs a binary mask $m \in \{0, 1\}^d$, indicating which components of the d state dimensions are relevant to the instruction, where the jth mask element $m^{(j)} = 1$ if component j is relevant, and $m^{(j)} = 0$ otherwise. For example, given the instruction "Stay away from the laptop" and a demonstration showing the robot trajectory moving around the laptop, an accurate output masks all state elements except the end effector

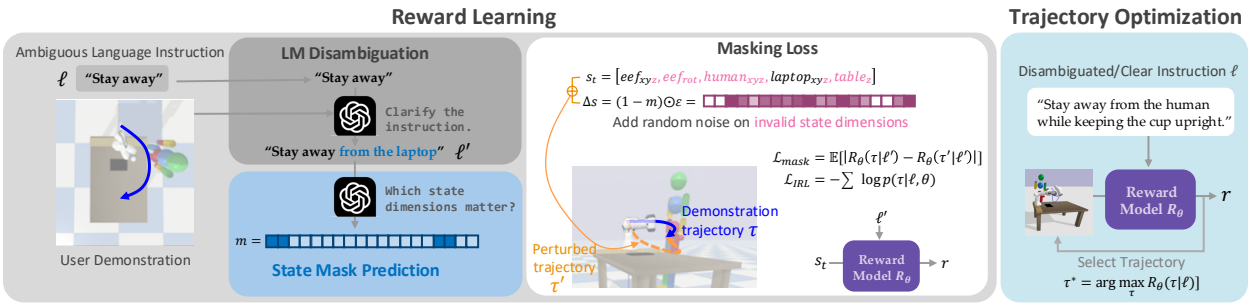


Fig. 2: System Overview. We clarify ambiguous language instructions using demonstrations and LLM reasoning. We then map disambiguated instructions into state masks, which guide the reward model through a masking loss that enforces invariance to irrelevant state dimensions during training. We train the reward model with the weighted sum of the masking loss and the IRL loss. Using the learned reward model, we can perform trajectory optimization by selecting the trajectory with the highest reward.

and laptop positions (state mask prediction module in Fig. 2). We augment our training data with these LLM-generated masks, yielding $\mathcal{D}' = \{(\tau_i, \ell_i, m_i)\}_{i=1}^{M}$. We use GPT-4o for state mask prediction (see our project page for full prompts).

B. State Masking Loss

A naive way to use the state masks is *explicit masking*, where we set irrelevant dimensions (corresponding to $m^{(j)} = 0$) to zero. However, this approach makes the model highly sensitive to errors in mask generation, completely discarding useful state input if a state element is incorrectly masked out. Instead, we *implicitly mask* irrelevant components by enforcing invariance through a masking loss, allowing the reward function to learn to ignore them without hard deletion. Formally, let $s^{(j)}$ denote a perturbed version of state $s \in \tau$, where only element j is modified by adding random noise ε. While various noise distributions can be used (e.g., Gaussian, uniform), we use $\varepsilon \sim \text{Uniform}(0, 1)$ in our experiments. We define the masking loss $\mathcal{L}_{\text{mask}}(\theta)$:

$$\mathbb{E}_{\tau,\ell,m\in\mathcal{D}'} \sum_{s\in\tau} \sum_{j=1}^{d} (1 - m^{(j)}) \left| r_\theta(s^{(j)} \mid \ell) - r_\theta(s \mid \ell) \right|,$$

which penalizes changes in the reward when irrelevant components are perturbed. The full training objective becomes a combination of the original LC-RL loss function and the masking loss,

$$\mathcal{J}(\theta) = \mathcal{L}_{\text{IRL}}(\theta) + \lambda\, \mathcal{L}_{\text{mask}}(\theta), \tag{3}$$

where $\lambda > 0$ is a hyperparameter controlling the trade-off between fitting the demonstrations and enforcing invariance to irrelevant state elements. We empirically compare the proposed implicit masking to naive explicit masking in Sec. V.

C. Clarifying Ambiguous Language Instructions

Natural language commands are often underspecified (e.g., "Stay away" without specifying to what), creating ambiguity for reward learning. We leverage LLM reasoning abilities to jointly consider language and demonstrations and hypothesize possible disambiguations.

Following Peng et al. [18], who showed that contrasting human demonstrations with nominal robot behavior helps recover intent, we provide the LLM with: (i) a task and environment description, (ii) the language utterance ℓ, (iii) a state-based representation of the demonstration τ, and (iv) a state-based representation of the shortest-path trajectory between the same start and end points, which we call the *reference trajectory*. We prompt the LLM to infer clarified commands that explain the difference between the demonstration and the reference trajectory. For example, given the instruction "Stay away" and a demonstration where the robot moves away from the table, the LLM may be able to reason the missing referent is the table, producing the disambiguated instruction, "Stay away from the table."

When multiple clarifications are possible (e.g., the command is "Stay away" and the demonstration avoids many objects), we instruct the LLM to return all disambiguations. This serves as a form of data augmentation, generating more demonstration-language pairings.

Finally, we generate state relevance masks from disambiguated commands using the same procedure as in Sec. IV-B. We can now train the model conditioning on disambiguated instructions rather than the original ambiguous ones. We use GPT-5 for language disambiguation (see our project page for full prompts).

D. Language-Conditioned Reward Model Architecture

The backbone of Masked IRL is a language-conditioned reward model with an inductive bias for sample-efficient conditioning. We encode input natural language ℓ with a pretrained T5 transformer [21] into an embedding h_{lang}. We incorporate h_{lang} with the input state s via Feature-wise Linear Modulation (FiLM) [19]. Specifically, h_{lang} is mapped through two MLPs to produce scaling and shifting parameters $\gamma, \beta \in \mathbb{R}^d$, which transform the state input s: $h_{\text{fused}} = \gamma \odot s + \beta$. This allows the instruction to directly modulate how reward compo-

nents are computed. Compared to simple concatenation of state and language inputs, FiLM provides a more structured and efficient interface for conditioning. The fused representation h_{fused} is then passed through a four-layer MLP, which maps the modulated state to a scalar reward value. We freeze the pretrained language encoder during training the reward model.

V. Experiments

We aim to evaluate the efficacy of Masked IRL to learn from limited and potentially ambiguous language and demonstrations. Our investigation seeks to answer the following research questions:

RQ1. Does the proposed masking loss allow Masked IRL to efficiently learn human preferences from language and demonstrations?

RQ2. Do demonstrations allow us to effectively disambiguate underspecified or ambiguous language?

RQ3. Do our findings replicate on a physical robot interacting with a human?

Environments. We evaluate our research questions on an object handover task using a Franka Emika robot arm in simulation and the real world. The goal is to deliver a coffee mug from a start location to a goal location in an environment that includes a table, a laptop, and a human. The state is a 19-dimensional vector consisting of the position and rotation of the robot's end effector, objects (table and laptop), and a human in the environment. Depending on the human preference, only a subset of these state components is relevant for the reward function. This setup provides a realistic scenario where preferences can be naturally expressed in language (e.g., "Stay away from the laptop") and grounded in demonstrations.

A. RQ1: Efficiency of the Masking Loss

Experimental procedure. Our simulated experiment is conducted in the PyBullet Simulator (Fig. 2). We simulate human reward functions based on five semantic features of the robot's trajectory: distance from the table, distance from the human, distance from the laptop, distance from the human's face, and mug orientation. Each ground truth preference is represented by a weight vector, where each feature is assigned a positive unit weight, a negative unit weight, or marked as irrelevant. Positive weights indicate a preference for proximity (e.g., keeping the mug close to the table), negative weights indicate avoidance (e.g., staying away from the laptop), and zero weights indicate irrelevance. This formulation yields 242 distinct preferences. We sample from this set of preferences for training and evaluating learned reward models.

We generate a trajectory dataset to train our reward models by sampling 20 object configurations and 10 start-goal pairs per configuration. For each start-goal pair, we generate 5 robot trajectories by smoothly perturbing shortest path trajectories with random noise in joint space. Each trajectory is paired with a language instruction that corresponds to a subset of the five features that describe the ground truth reward.

Baselines. To answer **RQ1**, we are interested in evaluating the effectiveness of the masking loss to learn reward functions. We consider three variants of incorporating language into the loss used to train our reward networks: **(1)** *Implicit Mask*, the loss proposed as in Sec. IV-B; **(2)** *Explicit Mask*, directly zeroing out the state dimension with the mask $(s \odot m)$; and **(3)** *None*, where the only language information the reward model receives is via the FiLM conditioning layer, as in LC-RL.

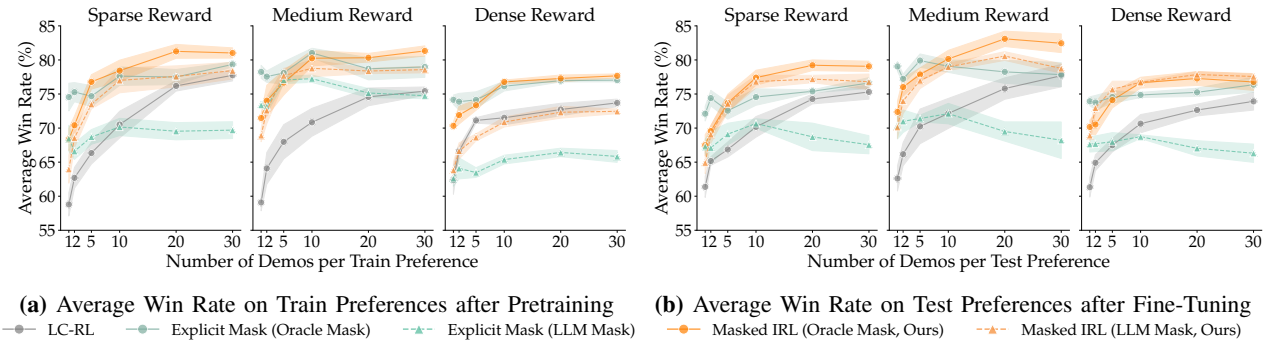
Because LLMs may incorrectly infer the state mask, we additionally ablate on how the mask is generated at two variations: **(1)** *LLM-generated*, the mask generated by the LLM as described in Sec. IV-A; and (2) *Oracle*, a ground truth mask determined by the ground-truth human preference. Combining all valid variants, we evaluate across the following five algorithms: **(1)** *Masked IRL (Oracle Mask)*, **(2)** *Masked IRL (LLM Mask)*, **(3)** *Explicit Mask (Oracle Mask)*, **(4)** *Explicit Mask (LLM Mask)*, and **(5)** *LC-RL* [11].

Each variant uses the model architecture in Sec. IV-D.

Evaluation metrics. We evaluate our approach using the *average win rate*: given two trajectories sampled from the test set, the learned reward predicts which one is preferred, and we score agreement with the ground-truth reward. This measures how often the learned reward model correctly prefers better trajectories compared to ground-truth preferences. We measure the average win rate on three different reward densities: *sparse*, *medium*, and *dense*. The density of the ground truth reward model is defined based on the number of nonzero preference weights a simulated human has for the five semantic features (sparse: 1, 2, medium: 3, dense: 4, 5). We run all experiments with 5 different random seeds and show the average and standard error across seeds.

Results. We first evaluate our proposed masking loss function against naive language conditioning on input layers. Fig. 3 shows that Masked IRL with both Oracle and LLM-generated masks consistently matches or outperforms the language-conditioned baseline (LC-RL) across different reward densities, both for train and test preferences. This demonstrates that naively conditioning the reward model on language is insufficient because the model can easily overfit to spurious correlations. In contrast, Masked IRL's masking loss penalizes sensitivity to irrelevant state elements. The masked loss enables the reward model to focus on task-relevant dimensions and improves both robustness and generalization.

Another key benefit of Masked IRL is its improved sample efficiency. As shown in Fig. 3, Masked IRL has



(a) Average Win Rate on Train Preferences after Pretraining

(b) Average Win Rate on Test Preferences after Fine-Tuning

Fig. 3: Performance Across Reward Densities. The average win rate of across all methods for different reward densities after (a) pretraining on 40 train preferences for 1k epochs and (b) fine-tuning on 30 test preferences for 100 epochs. All models are trained with 10 demonstrations per user preference and evaluated with unseen trajectories with novel object configurations. The shaded region indicates standard error across five different seeds.

a larger area under the win rate curve as the number of demonstrations increases. Because the masking loss discourages dependence on irrelevant state dimensions, the model can extract more useful information from fewer demonstrations. In practice, this means Masked IRL achieves strong generalization even with as few as five demonstrations per preference, while the other baselines require substantially more data — up to 33% more for Explicit Mask and 4.7 times more for LC-RL on average — to reach comparable performance. This efficiency is particularly valuable in robotics, where collecting demonstrations from humans is time-consuming.

Both explicit masking and Masked IRL outperform LC-RL when oracle masks are provided, but performance diverges under noisy LLM-generated masks, shown in Fig. 3. Explicit masking with LLM masks performs poorly, especially as the number of demonstrations increases, likely because hard-masking prohibits the model learning from state components that are potentially relevant to the preference due to noise. In contrast, Masked IRL remains robust with LLM masks: the masking loss encourages the model to adapt to multiple preferences even with imperfect supervision, preventing collapse and yielding stable gains over LC-RL.

B. RQ2: Robustness to Language Ambiguity

Experimental Procedure. We use the same demonstrations as described in Sec. V-A, but demonstrations are instead paired with *ambiguous instructions* based on the ground truth preferences. We procedurally generate ambiguous instructions that deliberately underspecify the user's preference in two ways naturally done by humans [22]: (1) *referent-omitted* commands, which specify a relation without the object (e.g., "Stay close"), and (2) *expression-omitted* commands, which specify an object but not the relation (e.g., "Table"). Because simultaneously omitting referents or expressions for multiple features would yield contrived and linguistically unnatural commands (e.g., "Stay away from this and stay away from that and stay close to another one"), we restrict our evaluation to sparse rewards. In this experiment, only a single feature is active at a time, allowing us to generate ambiguous commands that are both natural and representative of how users might underspecify preferences. Specifically, we evaluate six different sparse rewards, each defined by a positive or negative weighting over one of the three features: distance to the table, distance to the laptop, and distance to the human. For each preference, we assess our disambiguation method on both referent-omitted and expression-omitted instructions, paired with 10 demonstrations.

Baselines. To answer **RQ2**, we are interested in learning user preferences when given ambiguous language inputs. We consider the same three variants of incorporating language into the loss as Sec. V-A: **(1)** *Masked IRL*, **(2)** *Explicit Mask*, and **(3)** *LC-RL*. Our proposed approach performs a disambiguation process described in Sec. IV-C. The disambiguated instructions are used to predict state masks, following Sec. IV-B. To evaluate the effectiveness of the disambiguation step we use two variants of incorporating the instruction information: **(1)** *Disambiguated Instructions* (DI), the proposed disambiguation pipeline; and (2) *Ambiguous Instructions* (AI), directly calculating the mask from ambiguous instructions without first disambiguating the instructions.

Evaluation metrics. For evaluating average win rate, we provide ambiguous instructions to models labeled "AI" and disambiguated instructions to models labeled "DI". We further evaluate LLM disambiguation performance with two additional metrics: *instruction accuracy* and mask-based *Precision, Recall, and F1* scores. We define a disambiguation query as correct if the generated set of command candidates includes an instruction semantically equivalent to the ground-truth clarified command. To account for LLM stochasticity, we report the average accuracy across five independent query rounds. For training the DI reward model baselines, we select the clarified instructions from the single most accurate round.

Instruction Type	Precision	Recall	F1 Score
Ambiguous	0.531 ± 0.003	0.910 ± 0.009	0.670 ± 0.005
Disambiguated	0.705 ± 0.001	0.882 ± 0.024	0.783 ± 0.009
Clear	0.789 ± 0.017	1.000 ± 0.000	0.882 ± 0.010

TABLE I: State Mask Prediction from Different Instruction Types. Disambiguated instructions improve all metrics over ambiguous instructions. Errors denote standard errors across five runs.

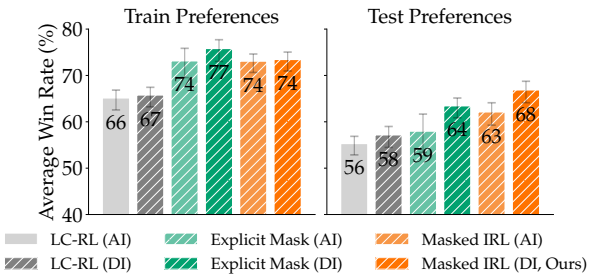


Fig. 4: Performance on ambiguous language. AI and DI denote models trained with ambiguous and disambiguated instructions, respectively. While LC-RL only uses language to condition the reward model, both Explicit Mask and Masked IRL significantly outperform LC-RL on train preferences, demonstrating the benefit of our masking approach. On unseen test preferences, both language disambiguation and masking are important, where Masked IRL using disambiguated instructions show the highest performance.

Results. Over the five rounds of 6 preferences, the average instruction accuracy of the language disambiguation step of our pipeline was 76.4%, and the average number of disambiguated instruction candidates per ambiguous command-demo pair was 1.12. We also measure the state mask prediction performance from clear, ambiguous, and disambiguated instructions, as shown in Table I. While clear instruction leads to the highest performance in all metrics, disambiguated instructions show 16.9% higher F1 score than ambiguous instructions. Fig. 4 shows the performance of reward learning using ambiguous or disambiguated instructions. On test preferences, using disambiguated language improves performance for all methods: LC-RL, Explicit Mask, and Masked IRL. Masked IRL trained with disambiguated instructions shows the highest generalization performance, showing 21.4% higher average win rate than LC-RL trained with ambiguous instructions.

C. RQ3: Evaluation in the Real World

For **RQ3**, we are interested in the efficacy of our approach in the real world, where human demonstrations may be suboptimal with respect to specified preferences. We conduct experiments on the robot using the same set of models as in Sec. V-A: LC-RL, {Explicit Masking, Implicit Masking} × {Oracle Mask, LLM Mask}.

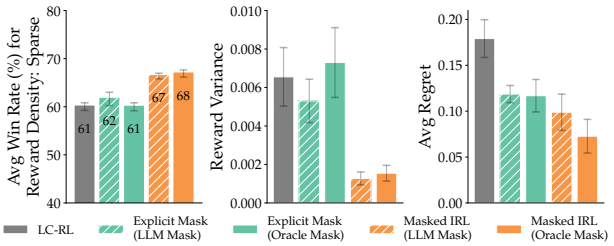


Fig. 5: Zero-shot Performance on Test Preferences with Real Robot. Masked IRL achieves higher win rates, lower reward variance given perturbation on irrelevant state dimensions, and lower win rates on optimized trajectories than baselines, showing its effectiveness in transferring to novel preferences without additional training.

Experimental Procedure. For real world experiments, we collect $1,200$ demonstrations evenly distributed over 50 preferences. Each preference comprises of two demonstrations for each of 12 object configurations, with both demonstrations sharing the same start-goal pair, randomly sampled from nine possible locations. Two experts provided demonstrations by kinesthetically guiding the robot according to a given preference.

Evaluation metrics. In addition to average win rate, we evaluate *average reward variance* and *average regret* of trajectories optimized with learned rewards. We evaluate average reward variance by adding Gaussian noise sampled from $\mathcal{N}(0,1)$ to irrelevant state dimensions. This procedure is repeated five times, and the variances of the resulting rewards are averaged to obtain the final measure. To evaluate average regret, we first perform discrete optimization over the set of test trajectories to choose the most optimal trajectory with learned reward models given test preferences. We calculate the regret by calculating the difference of the ground truth rewards between the chosen trajectory and the actual optimal trajectory that maximizes the ground truth reward function.

Zero-shot generalization to real robot. We further validate Masked IRL on a real Franka Panda robot. As shown in Fig. 5, Masked IRL achieves higher average win rates and lower reward variance than all baselines, demonstrating that this method transfers to real world human demonstrations without additional fine-tuning or architecture changes. Masked IRL additionally shows significantly lower reward variance compared to LC-RL and Explicit Mask, demonstrating that masking loss effectively enforces invariance to irrelevant state changes. These results highlight the generalization of our approach: language-guided implicit masking makes the learned rewards more robust to distributional shifts in the real world. Furthermore, the rightmost plot in Fig. 5 shows that Masked IRL achieves 59.4% and 44.8% lower average reward regret than LC-RL using oracle and LLM masks, respectively. This demonstrates that Masked IRL learns

rewards that lead to better optimized trajectories than baseline approaches for reward learning.

VI. Discussion

Conclusion. Reward learning from demonstrations is often ambiguous and susceptible to overfitting, since demonstrations show how to act but not what matters. To address this, we propose Masked IRL, which leverages LLMs to generate state relevance masks and incorporates a masking loss that enforces invariance to irrelevant state dimensions. Combined with an LLM-based disambiguation of underspecified instructions, this approach improves sample efficiency, robustness, and generalization, outperforming prior language-conditioned IRL methods in both simulation and real-robot experiments with up to 4.7 times fewer demonstrations.

Limitations and Future Work. 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. While we focus on manipulation, the framework naturally extends to other domains where humans can provide both behavioral feedback (e.g., demonstrations, corrections) and semantic feedback (e.g., language, gaze, or gestures). 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] 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.

[2] 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 (CoRL)*, pages 217–226. PMLR, 2017.

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

[4] Andreea Bobu, Andi Peng, Pulkit Agrawal, Julie A. Shah, and Anca D. Dragan. Aligning human and robot representations. In *HRI*, 2024.

[5] 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.

[6] Daniel Brown, Russell Coleman, Ravi Srinivasan, and Scott Niekum. Safe imitation learning via fast Bayesian reward inference from preferences. In *ICML*, 2020.

[7] 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*, 2017.

[8] 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.

[9] Nathaniel Dennler, Stefanos Nikolaidis, and Maja Matarić. Contrastive learning from exploratory actions: Leveraging natural interactions for preference elicitation. In *2025 20th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 778–788. IEEE, 2025.

[10] Nathaniel Dennler, Zhonghao Shi, Yiran Tao, Andreea Bobu, Stefanos Nikolaidis, and Maja Matarić. Improving through interaction: Searching behavioral representation spaces with cma-es-ig. *arXiv preprint arXiv:2603.09011*, 2026.

[11] Justin Fu, Anoop Korattikara, Sergey Levine, and Sergio Guadarrama. From language to goals: Inverse reinforcement learning for vision-based instruction following. In *ICLR*, 2019.

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

[13] Minyoung Hwang, Gunmin Lee, Hogun Kee, Chan Woo Kim, Kyungjae Lee, and Songhwai Oh. Sequential preference ranking for efficient reinforcement learning from human feedback. *NeurIPS*, 36, 2023.

[14] Minyoung Hwang, Luca Weihs, Chanwoo Park, Kimin Lee, Aniruddha Kembhavi, and Kiana Ehsani. Promptable behaviors: Personalizing multi-objective rewards from human preferences. In *CVPR*, 2024.

[15] 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 *ICRA*, 2023.

[16] Inês Lourenço, Andreea Bobu, Cristian R. Rojas, and Bo Wahlberg. Diagnosing and repairing feature representations under distribution shifts. In *62nd IEEE Conference on Decision and Control, CDC 2023, Singapore, December 13-15, 2023*, pages 3638–3645. IEEE, 2023.

[17] Jordan Abi Nader, David Lee, Nathaniel Dennler, and Andreea Bobu. Quicklap: Quick language-action preference learning for autonomous driving agents. *arXiv preprint arXiv:2511.17855*, 2025.

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

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

[20] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[21] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67, 2020.

[22] Yanming Wan, Yue Wu, Yiping Wang, Jiayuan Mao, and Natasha Jaques. Infer human's intentions before following natural language instructions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 25309–25317, 2025.

[23] Zhaojing Yang, Miru Jun, Jeremy Tien, Stuart J. Russell, Anca D. Dragan, and Erdem Bıyık. Trajectory improvement and reward learning from comparative language feedback. In *Proceedings of the Conference on Robot Learning (CoRL)*, 2024.

[24] Wenhao Yu, Nimrod Gileadi, Chuyuan Fu, and et al. Language to rewards for robotic skill synthesis. In *CoRL*, 2023.

[25] 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.

[26] Matthew Zurek, Andreea Bobu, Daniel S. Brown, and Anca D. Dragan. Situational confidence assistance for lifelong shared autonomy. In *IEEE International Conference on Robotics and Automation, ICRA 2021, Xi'an, China, May 30 - June 5, 2021*, pages 2783–2789. IEEE, 2021.