MAPLE: A Framework for Active Preference Learning Guided by Large Language Models

Saaduddin Mahmud, Mason Nakamura, Shlomo Zilberstein

Manning College of Information and Computer Sciences
University of Massachusetts Amherst
{smahmud,mnakamura,shlomo}@umass.edu

Abstract

The advent of large language models (LLMs) has sparked significant interest in using natural language for preference learning. However, existing methods often suffer from high computational burdens, taxing human supervision, and lack of interpretability. To address these issues, we introduce MAPLE, a framework for large language model-guided Bayesian active preference learning. MAPLE leverages LLMs to model the distribution over preference functions, conditioning it on both natural language feedback and conventional preference learning feedback, such as pairwise trajectory rankings. MAPLE employs active learning to systematically reduce uncertainty in this distribution and incorporates a language-conditioned active query selection mechanism to identify informative and easy-to-answer queries, thus reducing the burden on humans. We evaluate MAPLE's sample efficiency and preference inference quality across two benchmarks, including a real-world vehicle route planning benchmark using OpenStreetMap data. Our results demonstrate that MAPLE accelerates the learning process and effectively improves humans' ability to answer queries.

Introduction

With recent advances in artificial intelligence, autonomous agents are being increasingly deployed in real-world applications to address complex tasks (Zilberstein 2015; Dietterich 2017). A prominent method for efficiently aligning these agents with human preferences is Active Learning from Demonstration (Active LfD) (Biyik 2022). Preference-based Active LfD, a variant of LfD, aims to infer a preference function from human-generated rankings over a set of observed behaviors using a Bayesian active learning approach.

Recent advancements in natural language processing have inspired many researchers to leverage language-based abstraction for learning human preferences (Soni et al. 2022; Guan, Sreedharan, and Kambhampati 2022). This approach offers a more flexible and interpretable way of learning preferences compared to conventional methods (Sadigh et al. 2017; Brown, Goo, and Niekum 2019; Brown et al. 2019). More recent work (Yu et al. 2023; Ma et al. 2023) has focused on utilizing large language models (LLMs), such as ChatGPT (Achiam et al. 2023), with prompting-based approaches to learn preferences from natural language instructions. However, these methods often require significant computational resources and taxing human supervision, as they lack a systematic querying approach.

To address these challenges, we introduce a novel framework: MAPLE (**M**odel-guided **A**ctive **P**reference **Le**arning). MAPLE begins by utilizing a large language model (LLM) to interpret the natural language instructions from humans and to estimate an initial distribution over preference functions. Then, it applies an active learning approach to systematically reduce uncertainty on the correct preference function. This is achieved through standard Bayesian posterior updates, conditioned on both conventional preference learning feedback, such as pairwise trajectory rankings, and linguistic feedback such as clarification or explanations of the cause behind the preference. By integrating Bayesian inference with LLMs, MAPLE achieves higher preference inference accuracy than a general-purpose pre-trained LLM while offering greater sample efficiency and flexibility than purely Bayesian methods.

To further ease human effort, MAPLE incorporates a language-conditioned in-context (Dong et al. 2024) active query selection mechanism that takes advantage of feedback on the difficulty of previous queries to choose future queries that are informative and easier to answer. MAPLE represents preference functions as a linear combination of abstract language concepts, providing a modular structure that enables the framework to acquire new concepts over time and enhance sample efficiency for future instructions. Moreover, this interpretable structure allows for human auditing of the learning process, facilitating human-guided validation (Mahmud, Saisubramanian, and Zilberstein 2023) before applying the preference function to optimize behavior.

In our experiments, we evaluate the efficacy of MAPLE in terms of sample efficiency during learning, as well as the quality of the final preference function. We use an environment based on the popular Minigrid (Chevalier-Boisvert et al. 2023) and introduce a new realistic vehicle routing benchmark based on OpenStreetMap (OpenStreetMap Contributors 2017) data, which includes text descriptions of the road network of different cities in the USA. Our evaluation shows the effectiveness of MAPLE on preference inference while improving the human's ability to answer queries.

Our contributions are threefold:

- We propose a Bayesian preference learning framework that leverages task-independent pre-trained LLMs and natural language explanations to reduce uncertainty over preference functions.
- We provide a language-conditioned in-context active query selection approach to reduce human burden.
- We perform extensive evaluations, including the design of a realistic new benchmark that can be used for future research in this area.

Related Work

Learning from demonstration Most Learning from Demonstration (LfD) algorithms learn a reward function using expert trajectories (Ng and Russell 2000; Abbeel and Ng 2004; Ziebart et al. 2008). Some of these approaches utilize a Bayesian framework to learn the reward or preference function (Ramachandran and Amir 2007; Brown et al. 2020; Mahmud, Saisubramanian, and Zilberstein 2023), and some pair it with active learning to reduce the number of human queries (Sadigh et al. 2017; Basu, Singhal, and Dragan 2018; Biyik 2022). However, these methods are unable to utilize natural language abstraction during Learning from Demonstrations, whereas our method can use both. In addition, we employ language-conditioned active learning to reduce user burden, an approach not previously explored in this context.

A related method in this area, REVEALE (Mahmud, Saisubramanian, and Zilberstein 2023), uses explanations to detect misalignment and infer a distribution over preferences. However, while REVEALE relies on feature attribution-based explanation methods and incorporates human selection or random sampling for the learning and auditing process, our approach utilizes natural language explanations and integrates active learning to enhance efficiency for preference learning.

Natural language in intention communication With the advent of natural language processing, several works have focused on directly communicating abstract concepts to agents (Tevet et al. 2022; Guo et al. 2022; Lin et al. 2022; Wang et al. 2024; Sontakke et al. 2024; Tien et al. 2024; Lou et al. 2024). The key difference is that this line of work directly conditions behavior in natural language, whereas we learn a language-abstracted preference function. This approach offers several advantages, including increased transparency, more fine-grained trade-offs between concepts, and enhanced transferability. The work most closely related to ours is (Lin et al. 2022), which infers rewards from language but restricts them to stepwise decision making.

Other lines of work (Yu et al. 2023; Ma et al. 2023) aim to learn reward functions directly by prompting LLMs. However, these methods often struggle with identifying temporally extended abstract behaviors. Furthermore, these approaches cannot utilize conventional preference feedback, whereas MAPLE can utilize both. In addition, they either lack a systematic way of acquiring human feedback or rely on data-hungry evolutionary algorithms. In contrast, our approach employs more efficient Bayesian active learning.

Abstraction in reward learning Abstraction in reward learning helps simplify complex objectives, improves generalization, and facilitates interaction between AI systems and humans. Several prior research efforts leverage abstract concepts to learn reward functions (Lyu et al. 2019; Illanes et al. 2020; Guan et al. 2021; Bobu et al. 2021; Icarte et al. 2022; Guan, Valmeekam, and Kambhampati 2022; Soni et al. 2022; Guan, Sreedharan, and Kambhampati 2022; Silver et al. 2022; Zhang et al. 2022; Bucker et al. 2023; Cui et al. 2023). Two methods closely related to our work are PRESCA (Soni et al. 2022) and RBA (Guan, Valmeekam, and Kambhampati 2022). PRESCA learns state-based abstract concepts to be avoided, while RBA learns temporally extended concepts with two variants: global (eliciting preference weights directly from humans) and local (tuning weights using binary search). Our approach also leverages temporally extended concepts but learns preference functions from natural language feedback using active learning. Unlike RBA, which relies on direct preference weights from humans or binary search, our method uses LLM-guided active learning for more expressive and informative preference elicitation while reducing human effort.

Some works use offline behavior datasets or demonstrations to learn various skills (Lee and Popović 2010; Wang et al. 2017; Zhou and Dragan 2018; Peng et al. 2018; Luo et al. 2020; Chebotar et al. 2021; Peng et al. 2021), which complement our approach. While MAPLE may also utilize such datasets in pre-training, the main purpose of MAPLE is to encode human preferences using natural language.

Active learning Previous works have explored different acquisition functions for active learning, typically focusing on selecting queries that maximize certain uncertainty quantization metrics. These metrics include predictive entropy (Gal, Islam, and Ghahramani 2017), uncertainty volume reduction (Sadigh et al. 2017), uncertainty weighted density score (Wray and Zilberstein 2016), mutual information maximization (Biyik et al. 2019), and maximizing the variation ratio (Gal, Islam, and Ghahramani 2017). Our approach complements these methods by integrating language-conditioned query selection to reduce the user burden. Although any of these methods can be paired with MAPLE, we opt for the variation ratio because of its ease of calculation and high effectiveness.

LLMs and Probabilistic Reasoning Many previous and contemporary research has examined the integration of probabilistic reasoning with large language models (LLMs) for active querying in classification and preference learning tasks (Rao and Daumé III 2018; Yu et al. 2020; Piriyakulkij, Kuleshov, and Ellis 2023; Grand et al. 2024; Andukuri et al. 2024). Among these, the work of Piriyakulkij, Kuleshov, and Ellis (2023) is particularly relevant, as it focuses on actively learning a binary preference function for single-step decision-making problems by combining LLMs and probabilistic reasoning. However, there are two key differences from our approach: their method does not leverage linguistic explanations for preference learning (relying solely on yes/no feedback) and does not utilize in-context learning for query selection to reduce user burden.

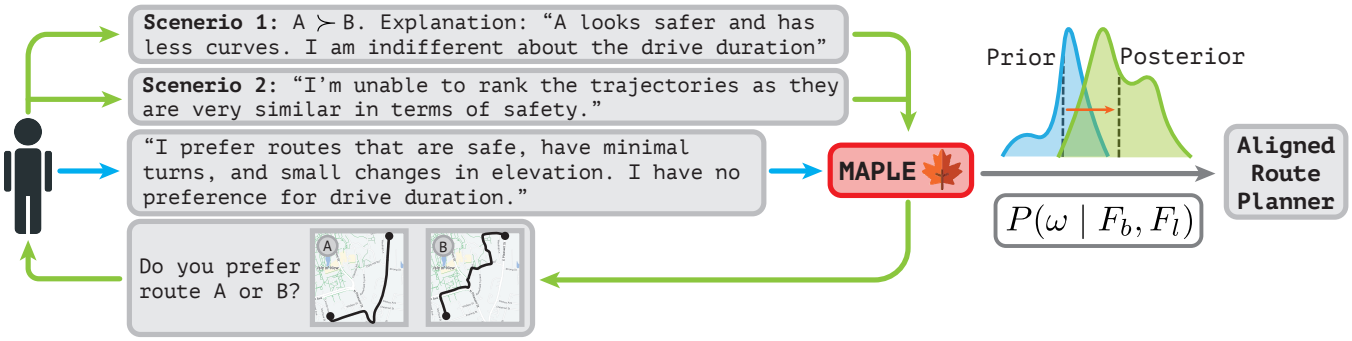

Figure 1: Application of MAPLE to the Natural Language Vehicle Routing Task.

Background

Markov decision process (MDP) A Markov Decision Process (MDP) M is represented by the tuple $M = (S, A, T, S_0, R, \gamma)$, where S is the set of states, A is the set of actions, $T : S \times A \times S \to [0, 1]$ is the transition function, S_0 is the initial state distribution, and $\gamma \in [0, 1)$ is the discount factor. A history h_t is a sequence of states up to time t, $(s_0, \ldots, s_t)$[1]. The reward function $R : H \times A \to [-R_{\max}, R_{\max}]$ maps histories and actions to rewards. For some problems, a goal function $G : H \to [0, 1]$ is provided that maps histories to goal achievements. In such problems, the reward function is typically $R : H \times A \to [-R_{\max}, 0]$ and $R(h_t \cup s_g, a) = 0$ for $G(h_t \cup s_g) = 1$. A policy $\pi : H \times A \to [0, 1]$ is a mapping from histories to a distribution over actions. The policy π induces a value function $V^\pi : S \to \mathbb{R}$, which represents the expected cumulative return $V^\pi(s)$ that the agent can achieve from state s when following policy π. An optimal policy π^* maximizes the expected cumulative return $V^*(s)$ from any state s, particularly from the initial state s_0.

Bayesian Preference Learning A preference function ω maps a trajectory τ to a real number representing how well the trajectory aligns with a human's objectives. The goal of preference learning is to infer this function from human feedback, enabling it to be directly used as the reward function in a Markov Decision Process (MDP) or to construct a reward function (discussed further in later sections).

One common approach is to learn this function from a dataset of pairwise preferences, represented as

$$\mathcal{D} = \{(\tau_1^1 \succ \tau_1^2), (\tau_2^1 \succ \tau_2^2), \ldots, (\tau_n^1 \succ \tau_n^2)\}.$$

Here, τ_i^1 and τ_i^2 are two trajectories, and $\tau_i^1 \succ \tau_i^2$ indicates that τ_i^1 is preferred over τ_i^2.

Using a Bayesian framework, as outlined in Ramachandran and Amir (2007), preference learning defines a probability distribution over preference functions conditioned on the dataset $\mathcal{D}$ via Bayes' rule:

$$P(\omega \mid \mathcal{D}) \propto P(\mathcal{D} \mid \omega) P(\omega).$$

The likelihood function $P(\mathcal{D} \mid \omega)$ is modeled in various ways. Following BREX (Brown et al. 2020), we use the Bradley–Terry model (Bradley and Terry 1952), which defines the likelihood as:

$$P(\mathcal{D} \mid \omega) = \prod_{(\tau_i^1 \succ \tau_i^2) \in \mathcal{D}} \frac{e^{\beta\omega(\tau_i^1)}}{e^{\beta\omega(\tau_i^1)} + e^{\beta\omega(\tau_i^2)}}. \quad (1)$$

Here, $\beta \in [0, \infty)$ is the inverse-temperature parameter.

Variation Ratio Given a conditional probability distribution $P(\cdot \mid X)$ with support $\{y_i\}_{i=0}^k$, the *variation ratio* of an input X is defined as follows:

$$\text{Variation\_Ratio}(X) = 1 - \arg\max_{y_i} P(y_i \mid X)$$

Problem Formulation

MAPLE We define a MAPLE problem instance as the tuple $(M_{|R}, \mathcal{H}, \mathbb{L}, C, \Omega, D_\tau)$, where:

- $M_{|R}$ is an MDP with an undefined or partially defined[2] reward function R.
- $\mathcal{H}$ is the human interaction function that acts as the interface between the human and MAPLE. Humans provide their feedback, preferences, and explanations in response to natural language queries posed by MAPLE.
- $\mathbb{L}$ is the LLM interaction function that generates natural language LLM queries and returns structured output.
- C is an expanding set of concepts in natural language $\{c_1, c_2, \ldots, c_n\}$. We also use $C(\cdot)$ to refer to a mapping model that takes a trajectory embedding $\phi(\tau)$ and a natural language concept embedding $\psi(c_i)$ and maps them to a numeric value indicating the degree to which the trajectory τ satisfies the concept c_i. For non-Markovian concepts, $\phi(\cdot)$ may be a sequence model such as a transformer. For Markovian concepts, we can define $C(\phi(\tau), \psi(c_i)) = \sum_{s \in \tau} C(\phi(s), \psi(c_i))$, where $\phi(s)$ is the state embedding.
- Ω is the space of all preference functions. In MAPLE, preference functions ω over a trajectory τ are modeled as a linear combination of the concepts and their associated weights:

$$\omega(\tau) = \sum_{c_i \in C} \omega_{c_i} \cdot C(\phi(\tau), \psi(c_i)) \quad (2)$$

[1] A history is can also be a sequence of observations.

[2] A partially defined reward function may be a sparse reward function that encodes signals for goals but not path preferences.

- D_τ is a dataset of unlabeled trajectories $\{\tau_1, \tau_2, \ldots, \tau_m\}$.

The objective of MAPLE is to model the repeated interaction between a human and an agent, where the human communicates their task objective $\mathcal{O}^{\mathcal{H}}_{\mathcal{T}}$ in natural language, and the agent is responsible for completing the task in alignment with that objective. MAPLE accomplishes this by actively learning a symbolic preference function ω using large language models (LLMs), enabling the agent to optimize its behavior according to this function to ensure that its actions align with human preferences.

Motivating example Consider an intelligent route planning system that takes a source, a destination, and user preferences about the route in natural language, as illustrated in Figure 1. Datasets for several preference-defining concepts such as speed, safety, battery friendliness, smoothness, autopilot friendliness, and scenic views can be easily obtained and used to pre-train the concept mapping function $C(\cdot)$. The goal of MAPLE is to take natural language instructions from a human and map them to a preference function ω interactively so that a search algorithm can optimize it to find the preferred route. MAPLE incorporates preference feedback on top of natural language feedback to address issues such as hallucination (Huang et al. 2023) and calibration required to directly apply a pre-trained LLM to unseen tasks. Additionally, MAPLE allows the human to skip difficult queries and applies in-context learning to identify which query to present, making the system more human-friendly. Furthermore, the preference function inference process in MAPLE is fully interpretable, enabling a human to audit the process thoroughly and provide the necessary feedback for improvement. Finally, the interaction with the human is repeated, allowing MAPLE to acquire and memorize new concepts and become more efficient for future tasks.

Detailed Description of the Proposed Method

A key innovation of MAPLE is the integration of conventional feedback from the preference learning literature with more expressive linguistic feedback, formally captured within a Bayesian framework introduced in REVEALE (Mahmud, Saisubramanian, and Zilberstein 2023):

$$P(\omega \mid F_b, F_l) \propto P(F_b \mid \omega) P(F_l \mid \omega) P(\omega) \tag{3}$$

Above, F_b represents the set of feedback observed in conventional preference learning algorithms, specifically in the context of this paper pairwise trajectory ranking.[3] F_l denotes the set of linguistic feedback. We can rewrite the equation as:

$$P(\omega \mid F_b, F_l) \propto \underbrace{P(F_b \mid \omega)}_{\text{Bradley-Terry Model}} \underbrace{P(\omega \mid F_l)}_{\text{LLM}} \underbrace{P(F_l)}_{\text{Uniform}} \tag{4}$$

$$\propto \underbrace{P(F_b \mid \omega)}_{\text{Bradley-Terry Model}} \underbrace{P(\omega \mid F_l)}_{\text{LLM}} \tag{5}$$

Here, the likelihood of F_b given ω is defined using the Bradley-Terry Model. The likelihood of ω given F_l is estimated using an LLM. Beyond incorporating linguistic feedback via LLMs, MAPLE advances conventional active learning methods. Conventional active learning typically focuses on selecting queries that reduce the maximum uncertainty of the posterior but lacks a flexible mechanism to account for human capability in responding to certain types of queries. MAPLE's Oracle-guided active query selection enhances any conventional acquisition function by leveraging linguistic feedback to alleviate the human burden associated with difficult queries. In the rest of this section, we provide more details on MAPLE, particularly Algorithms 1 and 2.

[3]MAPLE can handle any conventional feedback for which $P(F_b \mid \omega)$ is defined.

Algorithm 1: MAPLE

Require: Human instruction $\mathcal{O}^{\mathcal{H}}_{\mathcal{T}}$, Acquisition function $\mathcal{A}_f$, Number of LLM query K

1: $F_b, F_q \leftarrow \emptyset, \emptyset$
2: $F_l \leftarrow \{\mathcal{O}^{\mathcal{H}}_{\mathcal{T}}\}$
3: $\Omega_{\mathcal{T}} \leftarrow \{\omega_i\}_{i=0}^{n} \sim P(\omega \mid F_l)$
4: **while** condition not met **do**
5: $\quad Q \leftarrow \{(\tau_i, \tau_j) : \tau_i, \tau_j \in \mathcal{D}_\tau \wedge i \neq j \wedge (\tau_i, \tau_j) \notin F_b\}$
6: $\quad q \leftarrow$ Query Selection($\mathcal{A}_f$, Q, F_q, $\Omega_{\mathcal{T}}$, $\mathbb{L}$, K)
7: $\quad (f_b, f_l, f_q) \leftarrow \mathcal{H}(q)$
8: $\quad F_b, F_l, F_q \leftarrow F_b \cup \{f_b\}, F_l \cup \{f_l\}, F_q \cup \{f_q\}$
9: $\quad \Omega_{\mathcal{T}} \leftarrow \{\omega_i\}_{i=0}^{n} \sim P(\omega \mid F_b, F_l)$
10: **end while**
11: **return** $\Omega_{\mathcal{T}}$

Initialization

MAPLE starts by taking natural language instruction about task preference $\mathcal{O}^{\mathcal{H}}_{\mathcal{T}}$ and initializes the pairwise preference feedback set F_b, linguistic feedback F_l, and feedback on query difficulty F_q (lines 1-2, Algorithm 1). After that, the initial set of weights is sampled using the LLM from the distribution $P(\omega \mid F_l)$ as F_b is still empty (line 3, Algorithm 1). To sample ω from $P(\omega \mid F_l)$, we explore two sampling strategies described below.

Preference weight sampling from LLM We directly prompt the LLM $\mathbb{L}$ to provide linear weights ω over the abstract concepts. Specifically, we provide $\mathbb{L}$ with a prompt containing the task description $\mathcal{T}$, a list of known concepts C, human preference $\mathcal{O}^{\mathcal{H}}_{\mathcal{T}}$, and examples of pairs of instruction weights, along with additional answer generation instructions G. The LLM processes this prompt and returns an answer $\mathcal{O}^{\mathbb{L}}_{\omega_i}$:

$$\mathcal{O}^{\mathbb{L}}_{\omega_i} \leftarrow \mathbb{L}(\text{prompt}(\mathcal{T}, C, D_{\mathcal{I}}, G, \mathcal{O}^{\mathcal{H}}_{\mathcal{T}}))$$

We can take advantage of the text generation temperature to collect a diverse set of samples. We define the set of all the generated weights as $\mathcal{O}^{\mathbb{L}}_{\omega}$. Then, $P(\omega_j \mid F_l)$ can be modeled for any arbitrary ω_j as follows:

$$P(\omega_j \mid F_l) = \exp\left(-\beta_l \mathbb{E}_{\omega_i \in \mathcal{O}^{\mathbb{L}}_{\omega}}[\text{Distance}(\omega_i, \omega_j)]\right) \tag{6}$$

In this case, Euclidean or Cosine distance can be applied.

Distribution weight sampling using LLM The second approach we explore is distribution modeling using an LLM. Here, we use similar prompts as in the previous approach; however, we instruct the LLM to generate parameters for

$P(\omega \mid F_l)$. For example, for the weight of each concept $\omega_{c_i} \in \omega$, we prompt $\mathbb{L}$ to generate a range $\omega_{c_i}^{\text{range}} = [\omega_{c_i}^{\min}, \omega_{c_i}^{\max}]$. Then we can define $P(\omega \mid F_l)$ as follows:

$$P(\omega \mid F_l) = \begin{cases} 1, & \text{if } \omega_{c_i} \in \omega_{c_i}^{\text{range}}, \forall \omega_{c_i} \in \omega \\ 0, & \text{otherwise.} \end{cases} \quad (7)$$

We can similarly model this for other forms of distribution, such as the Gaussian distribution. Once the initialization process is complete, MAPLE iteratively reduces its uncertainty using human feedback.

LLM-Guided Active Preference Learning

After initialization, MAPLE iteratively follows three steps: 1) query selection, 2) human feedback collection, and 3) preference posterior update, discussed below.

Oracle-guided active query selection (OAQS) At the beginning of each iteration, MAPLE selects a query q (a pair of trajectories) (lines 5-6, Algorithm 1) from $\mathcal{D}_\tau$ that would reduce uncertainty the most while mitigating query difficulty based on human feedback. The query selection process is described in Algorithm 2, which starts by sorting all the queries based on an acquisition function $\mathcal{A}_f$. In this paper, we use the variation ratio for its flexibility and high efficacy. In particular, for trajectory ranking queries, the score for (τ_i, τ_j) is calculated as:

$$\mathbb{E}_{\omega \sim \Omega_\mathcal{T}}[1 - \max(P(\tau_i \succ \tau_j \mid \omega), P(\tau_j \succ \tau_i \mid \omega))]$$

Note that other acquisition functions can also be used. Once sorted, OAQS iterates over the top K queries and selects the first query that the oracle (in our case, an LLM) evaluates to be answerable by the human (lines 2-11). Finally, Algorithm 2 returns the least difficult query q among the top K queries selected by $\mathcal{A}_f$. We now analyze the performance of OAQS based on the characterization of the oracle.

Definition 1 *Let Q denote the set of all possible queries, and $Q_\mathcal{A} \subseteq Q$ represent the subset of queries answerable by $\mathcal{H}$. The **absolute query success rate (AQSR)** is defined as the probability that a randomly selected query q belongs to $Q_\mathcal{A}$, i.e., $P(q \in Q_\mathcal{A})$.*

Definition 2 *The **query success rate (QSR)** of a query selection strategy is defined as the probability that a query q, selected by the strategy, belongs to $Q_\mathcal{A}$, i.e., $P(q \in Q_\mathcal{A} \mid \textit{strategy})$.*

Proposition 1 *The QSR of a **random query selection strategy**: $P(q \in Q_\mathcal{A} \mid \textit{random}) = AQSR$*

Proposition 2 *Assuming the independence of AQSR from acquisition function ranking, the QSR of a **top-query selection strategy**, which always selects the highest-rated query by $\mathcal{A}_f$, $P(q \in Q_\mathcal{A} \mid \textit{top}) = AQSR$.*

Proposition 3 *The QSR of the OAQS strategy is given by*

$$AQSR \cdot Y_1 \cdot \frac{1 - [AQSR \cdot (1 - Y_0 - Y_1) + Y_0]^K}{1 - [AQSR \cdot (1 - Y_0 - Y_1) + Y_0]},$$

where $Y_0 = P(\mathbb{L}(F_q, q \notin Q_\mathcal{A}) = \textit{False})$
and $Y_1 = P(\mathbb{L}(F_q, q \in Q_\mathcal{A}) = \textit{True})$.
Here, we assume the independence of AQSR, Y_0, and Y_1 from acquisition function ranking.

Algorithm 2: Oracle-Guided Query Selection

Require: Acquisition function $\mathcal{A}_f$, List of queries Q, Query preference feedback F_q, Weights from current posterior $\Omega_\mathcal{T}$, Oracle $\mathbb{L}$, Number of Oracle queries K

```
Q_sort ← sort(Q|A_f, Ω_T)
Q_top ← Q_sort[0 : K]
for q ∈ Q_top do
    s_q ← L(prompt(F_q, q))
    if s_q is True then
        return q
    end if
end for
return Q_sort[0]
```

Corollary 1 *Based on Proposition 3, the OAQS will have a higher QSR than the random query selection strategy and the top-query selection strategy iff, $Y_0 + Y_1 > 1$ as $K \to \infty$.*

Definition 3 *The **optimal query success rate (OQSR)** of a strategy is defined as the probability that the strategy returns the query q^* with the highest value according to an acquisition function $\mathcal{A}_f$, among all answerable queries, i.e.,*

$$P(q^* = \arg\max_{q \in Q} \mathcal{A}_f(q) \mathbb{I}(q \in Q_\mathcal{A})),$$

where q^* is the query returned by the strategy.

Proposition 4 *Under the assumption of Proposition 2, the OQSR of a **random query selection strategy** equals $1/|Q|$.*

Proposition 5 *Under the assumption of Proposition 2, the OQSR of a **top-query selection strategy** equals the AQSR.*

Proposition 6 *Under the assumption of Proposition 3, the OQSR of the **OAQS** strategy is given by*

$$OQSR = AQSR \cdot Y_1 \cdot \frac{1 - [(1 - AQSR)Y_0]^K}{1 - (1 - AQSR)Y_0}.$$

Corollary 2 *Based on Proposition 6, the OAQS strategy will have a higher OQSR than the top-query selection strategy if $(1 - AQSR)Y_0 + Y_1 > 1$ as $K \to \infty$, and higher OQSR than the random query selection strategy if $\text{AQSR} \cdot Y_1 > \frac{1-(1-\text{AQSR})Y_0}{|Q|}$ as $K \to \infty$.*

Human feedback collection MAPLE queries the human $\mathcal{H}$ using the query q returned by Algorithm 2 to collect feedback. For each query q, MAPLE provides a pair of trajectories, and $\mathcal{H}$ returns an answer $\mathcal{O}_\tau^\mathcal{H} = (f_b, f_l, f_q)$, where f_b is binary feedback, f_l is an optional natural language explanation associated with that feedback—possibly empty if the human does not provide an explanation—and f_q is an optional natural language feedback about the difficulty of the query. Each piece of feedback is then added to the corresponding feedback set (lines 7-8, Algorithm 1).

LLM-guided posterior update Once feedback is added to the set, we update our current weight sample $\Omega_\mathcal{T}$ by sampling $P(\omega \mid F_b, F_l) \propto P(F_b \mid \omega) P(\omega \mid F_l)$ using MCMC sampling, where $P(\omega \mid F_b)$ is given by Equation 1, and $P(\omega \mid F_l)$ is given by Equations 6 and 7.

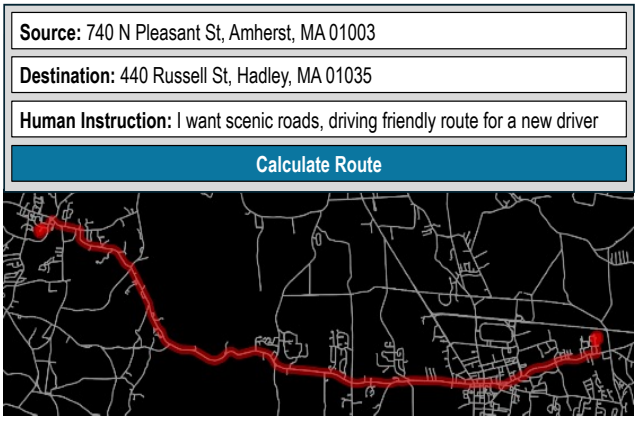

Figure 2: OpenStreetMap Routing

Figure 3: HomeGrid

Stopping criteria While MAPLE can employ various stopping criteria for active query generation, in this paper, we use a fixed budget approach, where MAPLE operates within a predefined maximum query limit.

Handling unknown concepts It should be noted that humans may provide instructions $\mathcal{O}_T^{\mathcal{H}}$ that cannot be adequately captured by the concepts available in the concept maps. Although this case is beyond the scope of this paper, several remedies exist in the literature to address this issue. LLMs can be prompted to add new concepts when generating weights. By leveraging the generalization capability of $C(\cdot)$, we can attempt to apply these new concepts directly. If the new concept is significantly different from those in C, few-shot learning techniques can be employed. In particular, during interactions, if a new concept is important, we can use nonparametric few-shot learning from human feedback, such as nearest-neighbor search, to improve concept mapping (Tian et al. 2024). Finally, if a new concept arises repeatedly, it can be added to the concept map by retraining C with data collected from multiple interactions through few-shot learning, as considered in (Soni et al. 2022).

Policy Optimization

The method for utilizing the weights generated by MAPLE to optimize the policy varies based on the trajectory encoding and the chosen policy solver algorithm. For example, for Markovian preferences, the weights can be used directly with an MDP solver. In non-Markovian settings, the weights can be used to rank trajectories and directly align the policy with algorithms such as DPO (Rafailov et al. 2024), or train a dense reward function (Guan, Valmeekam, and Kambhampati 2022) using preference learning algorithms such as TREX (Brown et al. 2019), and then use that reward function with reinforcement learning algorithms.

Experiments

In this section, we describe a comprehensive evaluation of MAPLE within the two environments detailed below. It is important to note that none of the models used in our experiments were fine-tuned for the tasks. We ran the local language model, specifically Mistral-7B-instruct-v0.3 (4-bit quantization), on a computer equipped with 64GB RAM and an Nvidia RTX 4090 24GB graphics card. For larger models, we relied on public API infrastructure. Due to space constraints, we present results using preference weight sampling as it outperformed distribution weight sampling in both benchmarks.

OpenStreetMap Routing We use OpenStreetMap to generate routing graphs for different states in the United States. The environment (Fig. 2) includes a concept mapping function capable of using ten different concepts: 1) Time, 2) Speed, 3) Safety, 4) Scenic, 5) Battery Friendly, 6) Gas Station Nearby, 7) Charging Station Nearby, 8) Human Driving Friendly, 9) Battery ReGen Friendly, and 10) Autopilot Friendly. The goal is to find a route between a given source and destination that aligns with user preferences. To generate D_τ, we used 200 pairs of random sources and destinations with weights randomly sampled from Ω. For modeling human interaction, we utilized two different datasets, each containing 50 human interaction templates. Each interaction includes a ground truth preference weight and a natural language description generated based on those weights. The first dataset, called "Clear," provides clear, knowledgeable instructions. The second dataset, called "Natural," obfuscates the "Clear" dataset with more natural language typical of everyday conversation and contextual information, for example:

> **Clear:** "*I prefer routes that are safe and scenic, with a moderate focus on speed and low importance on time.*"
>
> **Natural:** "*I'm planning a weekend drive to enjoy the countryside, so I'm not in a hurry. I want the route to be as safe as possible because I'll be driving with my family. It would be great if the drive is scenic too, so we could take in the beautiful views along the way. Speed isn't a top concern, and we're really just out to enjoy the journey rather than worry about how long it takes to get there.*"

For modeling f_l, the human clarifies the type of car (gas, autonomous, or electric) with a probability of 0.2 per feedback. For f_q, the human cannot answer when the two concepts with the highest weights (based on the ground truth) in both trajectories are closer than a predefined threshold.

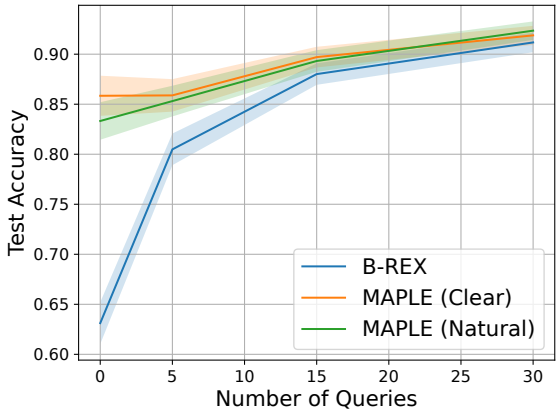

Figure 4: Test accuracy (OSM Routing)

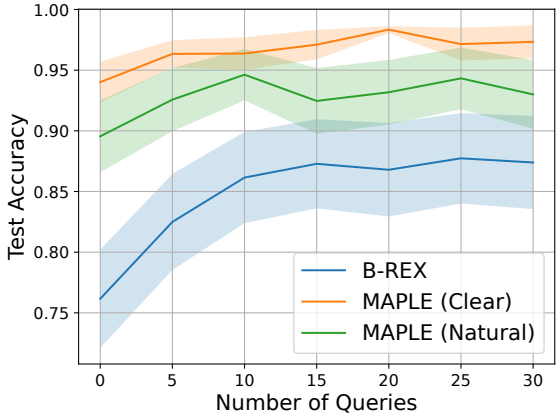

Figure 5: Test accuracy (HomeGrid)

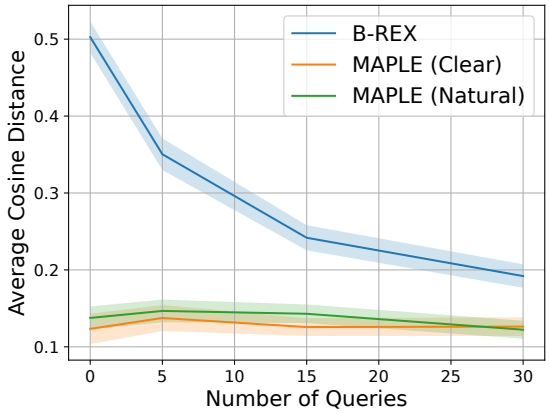

Figure 6: Cosine distance (OSM Routing)

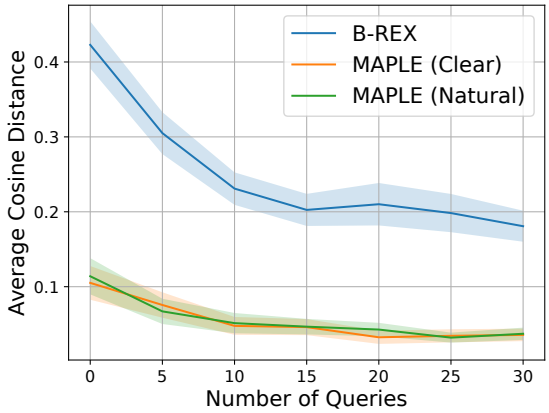

Figure 7: Cosine distance (HomeGrid)

HomeGrid The HomGrid environment (Fig. 3) is a simplified Minigrid (Chevalier-Boisvert et al. 2023) setting designed to simulate a robot performing household tasks (Lin et al. 2023). It features a discrete, finite action space and a partially observable language-based state space for a 3×3 grid, detailing the objects and floors in each grid square, within a truncated 12×14 grid. The abstract concepts include: 1) avoiding objects such as tables and chairs, 2) avoiding walls, 3) avoiding placing objects such as bottles and plates on the floor, 4) avoiding placing objects on the stove, and 5) avoiding placing objects on the left chairs. A total of 60 trajectories were manually generated to update the posterior distribution of the weights ω for each method. For modeling f_l, the human highlights the concept that was the most influential for their preference. The modeling of f_q follows a similar approach to that used in OSM Routing.

Experimental Results

We use three key metrics for evaluation: 1) the cosine distance between inferred preference weights (MAP of the distribution) and ground truth preference weights; 2) preference prediction accuracy, which evaluates the model's ability to generalize and accurately predict human preferences from an unseen set of trajectories; and 3)The policy cost difference compares the true cost of two policies: one trained using the ground truth preference function and the other trained using the learned preference function, with both evaluated against the ground truth preference function.

Impact of linguistic feedback Figures 4, 6, and Table 1, present the results of the OSM routing domain. In this experiment, we did not apply OAQS; instead, we selected queries randomly from the dataset to isolate the impact of language. Several noteworthy insights emerge from the results. First, observe that MAPLE, utilizing only linguistic feedback, outperforms B-REX on both the natural and clear datasets when provided with at least 10 instances of conventional feedback, highlighting the effectiveness of combining complex language feedback with conventional feedback. Additionally, as the feedback increases, the accuracy of B-REX begins to approach that of MAPLE. This sug-

Model	OSM Routing Test Accuracy	OSM Routing Expected Cost Δ
B-REX	0.79 ± 0.04	0.66 ± 0.10
Mistral-7B-Ins.	0.85 ± 0.02	0.34 ± 0.07
GPT-4o	$\mathbf{0.88 \pm 0.01}$	$\mathbf{0.22 \pm 0.05}$
GPT-4o-mini	0.87 ± 0.02	0.29 ± 0.07
Gemini-1.5-Pro	0.86 ± 0.01	0.26 ± 0.05
Mistral-Large-2	0.85 ± 0.02	0.27 ± 0.06

Table 1: Test accuracy and cost for different models with 5-feedback (OSM Routing, Natural)

Model	HomeGrid Test Accuracy	HomeGrid Expected Cost Δ
B-REX	0.82 ± 0.04	1.40 ± 0.24
Mistral-7B-Ins.	$\mathbf{0.92 \pm 0.02}$	0.67 ± 0.13
GPT-4o	0.87 ± 0.03	0.89 ± 0.20
GPT-4o-mini	0.83 ± 0.04	3.90 ± 0.66
Gemini-1.5-Pro	0.87 ± 0.04	1.00 ± 0.18
Mistral-Large-2	0.85 ± 0.04	$\mathbf{0.55 \pm 0.14}$

Table 2: Test accuracy and cost for different models with 5-feedback (HomeGrid, Natural)

gests that MAPLE is particularly advantageous when feedback is limited, such as in online settings where the agent must quickly infer rewards. Examining the cosine distance offers further insight. Language alone appears to be almost sufficient to align the reward angle, as the cosine distance remains static despite the increasing number of queries. This suggests that preference feedback is more effective in calibrating the magnitude of the preference vector than its direction. In contrast, while B-REX achieves good accuracy with large amounts of feedback, it seems to exhibit significant misalignment, which could suggest overfitting and potential failure in out-of-distribution scenarios. Lastly, we evaluated existing publicly available models and found that both GPT-4o and GPT-4o-mini outperformed other models. However, the smaller local model (Mistral-7B Instruct) proved to be competitive, so we used it to generate all the results shown in Figures 4, 5, 6, and 7.

Figures 5, 7, and Table 2 show the results of the HomeGrid experiments. In this environment, we observe that natural instructions do affect performance, but MAPLE still significantly outperforms B-REX in both datasets. Notably, the Mistral-Large-2 models surpassed B-REX by a wide margin, achieving nearly one-third of the cost difference. Surprisingly, GPT-4-mini performed poorly, with a cost difference that was worse than that of B-REX. This is due to the inference of highly misaligned preference weights for certain instructions. In this environment, we also see that most of the angle alignment was done using language feedback, and B-REX remains highly misaligned even after 30 rounds of feedback.

Impact of OAQS The results of the OAQS using an LLM as an oracle are shown in Figures 8 and 9. In this experiment, we use K=10. In the routing environment, the Active Query Success Rate (AQSR) is approximately 0.64, while in the HomeGrid environment, it is 0.46. We first evaluated the ability of various models for the selection of in-context queries (Table 3) using a dataset of 500 queries. The Mistral-7B model, used in the previous experiment, failed to meet the condition $Y_0 + Y_1 > 1$ in both environments. The Gemini-1.5-Pro model showed the best overall performance among the publicly available models and was used to generate Figures 8 and 9.

Figures 8 and 9 compare the test accuracy of Active B-REX with MAPLE, with and without OAQS. In both environments, MAPLE with OAQS achieved the highest performance, with a significant margin in the OSM routing environment. We also calculated the Query Success Rate (QSR) for all three algorithms: 0.43 for Active B-REX, 0.43 for MAPLE without OAQS, and 0.58 for MAPLE with OAQS in the routing domain. The QSR was lower than the AQSR

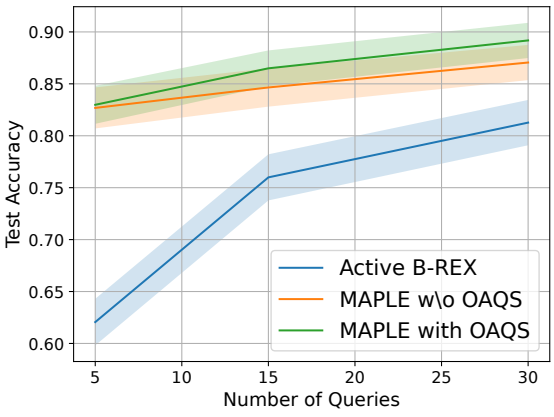

Figure 8: Efficacy of OAQS (OSM Routing, Natural)

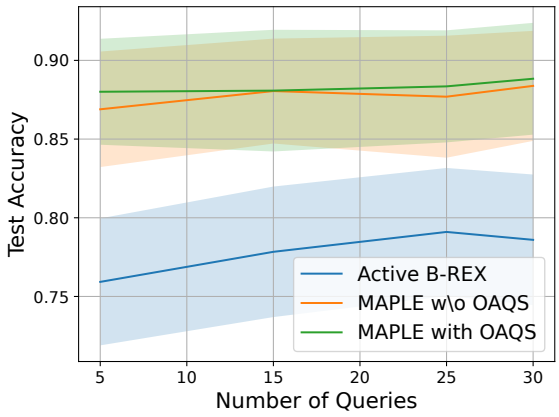

Figure 9: Efficacy of OAQS (HomeGrid, Natural)

Model	OSM Routing Y_0	OSM Routing Y_1	HomeGrid Y_0	HomeGrid Y_1
Mistral-7B-Ins.	0.30	0.65	0.17	0.77
GPT-4o	**0.84**	**0.43**	0.53	0.58
GPT-4o-mini	0.56	0.48	0.36	0.75
Gemini-1.5-Pro	0.62	0.72	**0.50**	**0.87**
Mistral-Large-2	0.50	0.78	0.49	0.84

Table 3: Performance of different models for in-context query selection (See Propositions 3 and 6)

for the top-query selection strategy due to a violation of the independence assumption, suggesting that the variation ratio is more likely to select more challenging queries. We refer to this experimental metric as the Effective Query Success Rate (EQSR). Based on Proposition 3, the QSR for MAPLE with OAQS should be 0.77, but it was observed to be lower for the same reason. Replacing AQSR with EQSR in Proposition 3 gives us a value of 0.59, which closely matches the experimental value. Therefore, we conclude that EQSR is a more practical metric to estimate the success of a model based on Y_0 and Y_1. This phenomenon is also observed in HomeGrid. Finally, in the HomeGrid environment, the overall EQSR was low, around 0.2; therefore, even with OAQS, we saw a modest increase of 1-3 feedback signals after 30 queries, which was not enough to create a large margin, and therefore we see only a modest difference between MAPLE with and without OAQS.

Conclusion

We introduce MAPLE, a novel framework for active preference learning guided by large language models (LLMs). By combining the capabilities of LLMs with language-conditioned query selection mechanisms, MAPLE systematically reduces uncertainty in inferred user preferences while simultaneously minimizing the burden on human feedback. Through experiments in two domains, including a real-world vehicle route planning task using OpenStreetMap data, we demonstrated the superior sample efficiency and high-quality preference inference of the framework. These results highlight the potential of integrating active learning with LLM-guided mechanisms in preference-driven decision-making systems.

Our study reveals key areas for further exploration. Although MAPLE effectively identifies and prioritizes informative queries, its performance can be constrained in settings where user feedback is ambiguous or noisy. In future work, we plan to extend MAPLE to more complex environments and tasks, explore different types of linguistic feedback, and conduct user studies to evaluate its usability and effectiveness. We are also interested in leveraging multimodal feedback (e.g., combining textual input with visual or haptic cues) to improve alignment in scenarios where textual feedback alone is insufficient.

Acknowledgments

This research was supported in part by the U.S. Army DEVCOM Analysis Center (DAC) contract W911QX23D0009, and by NSF grants 2321786, 2326054, and 2416459.

References

Abbeel, P.; and Ng, A. Y. 2004. Apprenticeship learning via inverse reinforcement learning. In *21st International Conference on Machine learning*.

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

Andukuri, C.; Fränken, J.-P.; Gerstenberg, T.; and Goodman, N. 2024. STaR-GATE: Teaching Language Models to Ask Clarifying Questions. In *1st Conference on Language Modeling*.

Basu, C.; Singhal, M.; and Dragan, A. D. 2018. Learning from richer human guidance: Augmenting comparison-based learning with feature queries. In *13th International Conference on Human-Robot Interaction*.

Biyik, E. 2022. *Learning preferences for interactive autonomy*. Ph.D. thesis, Stanford University.

Biyik, E.; Palan, M.; Landolfi, N. C.; Losey, D. P.; and Sadigh, D. 2019. Asking easy questions: A user-friendly approach to active reward learning. In *3rd Annual Conference on Robot Learning*.

Bobu, A.; Paxton, C.; Yang, W.; Sundaralingam, B.; Chao, Y.-W.; Cakmak, M.; and Fox, D. 2021. Learning perceptual concepts by bootstrapping from human queries. *arXiv preprint arXiv:2111.05251*.

Bradley, R. A.; and Terry, M. E. 1952. Rank Analysis of Incomplete Block Designs: I. The Method of Paired Comparisons. *Biometrika*.

Brown, D. S.; Goo, W.; and Niekum, S. 2019. Better-than-demonstrator imitation learning via automatically-ranked demonstrations. In *3rd Annual Conference on Robot Learning*.

Brown, D. S.; Goo, W.; Prabhat, N.; and Niekum, S. 2019. Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations. In *36th International Conference on Machine Learning*.

Brown, D. S.; Niekum, S.; Coleman, R.; and Srinivasan, R. 2020. Safe imitation learning via fast Bayesian reward inference from preferences. In *37th International Conference on Machine Learning*.

Bucker, A.; Figueredo, L. F. C.; Haddadin, S.; Kapoor, A.; Ma, S.; Vemprala, S.; and Bonatti, R. 2023. LATTE: LAnguage Trajectory TransformEr. In *IEEE International Conference on Robotics and Automation*.

Chebotar, Y.; Hausman, K.; Lu, Y.; Xiao, T.; Kalashnikov, D.; Varley, J.; Irpan, A.; Eysenbach, B.; Julian, R.; Finn, C.; et al. 2021. Actionable models: Unsupervised offline reinforcement learning of robotic skills. *arXiv preprint arXiv:2104.07749*.

Chevalier-Boisvert, M.; Dai, B.; Towers, M.; Perez-Vicente, R.; Willems, L.; Lahlou, S.; Pal, S.; Castro, P. S.; and Terry, J. 2023. Minigrid & Miniworld: Modular & customizable reinforcement learning environments for goal-oriented tasks. In *Advances in Neural Information Processing Systems*.

Cui, Y.; Karamcheti, S.; Palleti, R.; Shivakumar, N.; Liang, P.; and Sadigh, D. 2023. "No, to the Right" – Online language corrections for robotic manipulation via shared autonomy. *arXiv preprint arXiv:2301.02555*.

Dietterich, T. G. 2017. Steps toward robust artificial intelligence. *AI Magazine*.

Dong, Q.; Li, L.; Dai, D.; Zheng, C.; Ma, J.; Li, R.; Xia, H.; Xu, J.; Wu, Z.; Chang, B.; Sun, X.; Li, L.; and Sui, Z. 2024. A Survey on In-context Learning. In *Conference on Empirical Methods in Natural Language Processing*.

Gal, Y.; Islam, R.; and Ghahramani, Z. 2017. Deep Bayesian active learning with image data. In *International Conference on Machine Learning*.

Grand, G.; Pepe, V.; Andreas, J.; and Tenenbaum, J. B. 2024. A Llama Sunk My Battleship! Asking Rational Questions with LLMs via Bayesian Inference. In *The First Workshop on System-2 Reasoning at Scale, NeurIPS 2024*.

Guan, L.; Sreedharan, S.; and Kambhampati, S. 2022. Leveraging approximate symbolic models for reinforcement learning via skill diversity. *arXiv preprint arXiv:2202.02886*.

Guan, L.; Valmeekam, K.; and Kambhampati, S. 2022. Relative behavioral attributes: Filling the gap between symbolic goal specification and reward learning from human preferences. *arXiv preprint arXiv:2210.15906*.

Guan, L.; Verma, M.; Guo, S. S.; Zhang, R.; and Kambhampati, S. 2021. Widening the pipeline in human-guided reinforcement learning with explanation and context-aware data augmentation. *Advances in Neural Information Processing Systems*.

Guo, C.; Zou, S.; Zuo, X.; Wang, S.; Ji, W.; Li, X.; and Cheng, L. 2022. Generating diverse and natural 3D human motions from text. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Huang, L.; Yu, W.; Ma, W.; Zhong, W.; Feng, Z.; Wang, H.; Chen, Q.; Peng, W.; Feng, X.; Qin, B.; et al. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*.

Icarte, R. T.; Klassen, T. Q.; Valenzano, R.; and McIlraith, S. A. 2022. Reward machines: Exploiting reward function structure in reinforcement learning. *Journal of Artificial Intelligence Research*.

Illanes, L.; Yan, X.; Icarte, R. T.; and McIlraith, S. A. 2020. Symbolic plans as high-level instructions for reinforcement learning. In *30th International Conference on Automated Planning and Scheduling*.

Lee, S. J.; and Popović, Z. 2010. Learning behavior styles with inverse reinforcement learning. *ACM Transactions on Graphics*.

Lin, J.; Du, Y.; Watkins, O.; Hafner, D.; Abbeel, P.; Klein, D.; and Dragan, A. 2023. Learning to model the world with language. *arXiv preprint arXiv:2308.01399*.

Lin, J.; Fried, D.; Klein, D.; and Dragan, A. 2022. Inferring rewards from language in context. *arXiv preprint arXiv:2204.02515*.

Lou, X.; Zhang, J.; Wang, Z.; Huang, K.; and Du, Y. 2024. Safe reinforcement learning with free-form natural language constraints and pre-trained language models. *arXiv preprint arXiv:2401.07553*.

Luo, Y.-S.; Soeseno, J. H.; Chen, T. P.-C.; and Chen, W.-C. 2020. CARL: Controllable agent with reinforcement learning for quadruped locomotion. *ACM Transactions on Graphics*.

Lyu, D.; Yang, F.; Liu, B.; and Gustafson, S. 2019. SDRL: interpretable and data-efficient deep reinforcement learning leveraging symbolic planning. In *33rd AAAI Conference on Artificial Intelligence*.

Ma, Y. J.; Liang, W.; Wang, G.; Huang, D.-A.; Bastani, O.; Jayaraman, D.; Zhu, Y.; Fan, L.; and Anandkumar, A. 2023. Eureka: Human-level reward design via coding large language models. *arXiv preprint arXiv:2310.12931*.

Mahmud, S.; Saisubramanian, S.; and Zilberstein, S. 2023. Explanation-guided reward alignment. In *32nd International Joint Conference on Artificial Intelligence*.

Ng, A. Y.; and Russell, S. J. 2000. Algorithms for inverse reinforcement learning. In *17th International Conference on Machine Learning*.

OpenStreetMap Contributors. 2017. Planet dump retrieved from planet.osm.org. https://www.openstreetmap.org.

Peng, X. B.; Kanazawa, A.; Malik, J.; Abbeel, P.; and Levine, S. 2018. SFV: Reinforcement learning of physical skills from videos. *ACM Transactions On Graphics*.

Peng, X. B.; Ma, Z.; Abbeel, P.; Levine, S.; and Kanazawa, A. 2021. AMP: Adversarial motion priors for stylized physics-based character control. *ACM Transactions On Graphics*.

Piriyakulkij, W. T.; Kuleshov, V.; and Ellis, K. 2023. Active preference inference using language models and probabilistic reasoning. *Foundation Models for Decision Making Workshop at NeurIPS 2023*.

Rafailov, R.; Sharma, A.; Mitchell, E.; Manning, C. D.; Ermon, S.; and Finn, C. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*.

Ramachandran, D.; and Amir, E. 2007. Bayesian inverse reinforcement learning. In *20th International Joint Conference on Artifical intelligence*.

Rao, S.; and Daumé III, H. 2018. Learning to Ask Good Questions: Ranking Clarification Questions using Neural Expected Value of Perfect Information. In *56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Sadigh, D.; Dragan, A. D.; Sastry, S. S.; and Seshia, S. A. 2017. Active preference-based learning of reward functions. In *Robotics: Science and Systems XIII*.

Silver, T.; Athalye, A.; Tenenbaum, J. B.; Lozano-Perez, T.; and Kaelbling, L. P. 2022. Learning neuro-symbolic skills for bilevel planning. *arXiv preprint arXiv:2206.10680*.

Soni, U.; Thakur, N.; Sreedharan, S.; Guan, L.; Verma, M.; Marquez, M.; and Kambhampati, S. 2022. Towards customizable reinforcement learning agents: Enabling preference specification through online vocabulary expansion. *arXiv preprint arXiv:2210.15096*.

Sontakke, S.; Zhang, J.; Arnold, S.; Pertsch, K.; Biyik, E.; Sadigh, D.; Finn, C.; and Itti, L. 2024. RoboCLIP: One demonstration is enough to learn robot policies. *Advances in Neural Information Processing Systems*.

Tevet, G.; Raab, S.; Gordon, B.; Shafir, Y.; Cohen-Or, D.; and Bermano, A. H. 2022. Human motion diffusion model. *arXiv preprint arXiv:2209.14916*.

Tian, S.; Li, L.; Li, W.; Ran, H.; Ning, X.; and Tiwari, P. 2024. A survey on few-shot class-incremental learning. *Neural Networks*.

Tien, J.; Yang, Z.; Jun, M.; Russell, S. J.; Dragan, A.; and Biyik, E. 2024. Optimizing robot behavior via comparative language feedback. In *3rd HRI Workshop on Human-Interactive Robot Learning*.

Wang, Y.; Sun, Z.; Zhang, J.; Xian, Z.; Biyik, E.; Held, D.; and Erickson, Z. 2024. RL-VLM-F: Reinforcement learning from vision language foundation model feedback. *arXiv preprint arXiv:2402.03681*.

Wang, Z.; Merel, J. S.; Reed, S. E.; de Freitas, N.; Wayne, G.; and Heess, N. 2017. Robust imitation of diverse behaviors. *Advances in Neural Information Processing Systems*.

Wray, K. H.; and Zilberstein, S. 2016. A POMDP formulation of proactive learning. In *30th AAAI Conference on Artificial Intelligence*, 3202–3208.

Yu, L.; Chen, H.; Wang, S. I.; Lei, T.; and Artzi, Y. 2020. Interactive Classification by Asking Informative Questions. In *58th Annual Meeting of the Association for Computational Linguistics*.

Yu, W.; Gileadi, N.; Fu, C.; Kirmani, S.; Lee, K.-H.; Arenas, M. G.; Chiang, H.-T. L.; Erez, T.; Hasenclever, L.; Humplik, J.; et al. 2023. Language to rewards for robotic skill synthesis. *arXiv preprint arXiv:2306.08647*.

Zhang, R.; Bansal, D.; Hao, Y.; Hiranaka, A.; Gao, J.; Wang, C.; Martín-Martín, R.; Fei-Fei, L.; and Wu, J. 2022. A dual representation framework for robot learning with human guidance. In *6th Annual Conference on Robot Learning*.

Zhou, A.; and Dragan, A. D. 2018. Cost functions for robot motion style. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Ziebart, B. D.; Maas, A. L.; Bagnell, J. A.; and Dey, A. K. 2008. Maximum entropy inverse reinforcement learning. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence*.

Zilberstein, S. 2015. Building strong semi-autonomous systems. In *29th AAAI Conference on Artificial Intelligence*.