LEARNING TRANSFERABLE LATENT USER PREFERENCES FOR HUMAN-ALIGNED DECISION MAKING

Alina Hyk
School of EECS
Oregon State University
hyka@oregonstate.edu

Sandhya Saisubramanian
School of EECS
Oregon State University
sandhya.sai@oregonstate.edu

ABSTRACT

Large language models (LLMs) are increasingly used as reasoning modules for autonomous agents. While they are efficient in certain tasks, LLMs often struggle to produce human-aligned solutions. Human-aligned decision making requires accounting for both explicitly stated goals and latent user preferences that shape how ambiguous situations should be resolved. Existing approaches to incorporating such preferences either rely on extensive and repeated user interactions or fail to generalize latent preferences across tasks and contexts, limiting their practical applicability. We consider a setting in which an agent uses an LLM for high-level reasoning and as an intermediary between the human and the agent. In this role, the LLM is responsible for inferring latent user preferences from limited interactions, which guides downstream decision making. We introduce CLIPR (Conversational Learning for Inferring Preferences and Reasoning), a framework that learns actionable, transferable representations of latent user preferences from minimal conversational input and applies them to both in-distribution and out-of-distribution ambiguous tasks across multiple environments. Evaluations on three datasets show that CLIPR consistently outperforms existing methods in aligning agent behavior with user preferences, while significantly reducing training, inference, and runtime costs.

1 INTRODUCTION

Large language models (LLMs) are increasingly integrated into human-AI and human-robot systems as high-level reasoning modules for task planning (Han et al., 2025; Rana et al., 2023) and mobile manipulation (Kim et al., 2024). They are used for various purposes including extracting semantic knowledge for object arrangement (Ding et al., 2023), generating executable action scripts and policy code (Inagaki et al., 2023; Liang et al., 2022), and planning under multiple constraints (Irpan et al., 2022; Singh et al., 2023).

Despite these advances, LLM-augmented agents perform poorly and exhibit misaligned behavior in novel scenarios that were not encountered during training (Agrawal, 2022). The accuracy of the LLM's output directly impacts downstream robotic planning and execution. However, language models are prone to hallucinations (Liang et al., 2024) or producing plans that are infeasible for robotic execution (Ahn et al., 2022). Further misalignment arises from incorrect interpretation of human queries and behaviors, often due to oversimplified internal models of human intent and reasoning (Ahn et al., 2022; MacMahon et al., 2006; Zhang et al., 2023). Ambiguities in user queries, especially those lacking proficiency in prompting and prior interactions with LLMs, further amplify misalignment (Wang et al., 2024). These limitations highlight a key gap in human-robot interaction (HRI): the inability of current LLM-based reasoning methods to support efficient personalization and alignment with individual user preferences, which is critical for user satisfaction, agent usefulness, and safety (Wang et al., 2023).

For example, a user request "Get me something to drink with my sandwich" may yield multiple valid responses: iced green tea, water, and coffee (Figure 1). However, the user may consistently prefer one option due to a latent preference (e.g., preferring cold drinks), making iced green tea the preferred choice. Without preference knowledge, an LLM can identify that all four options are

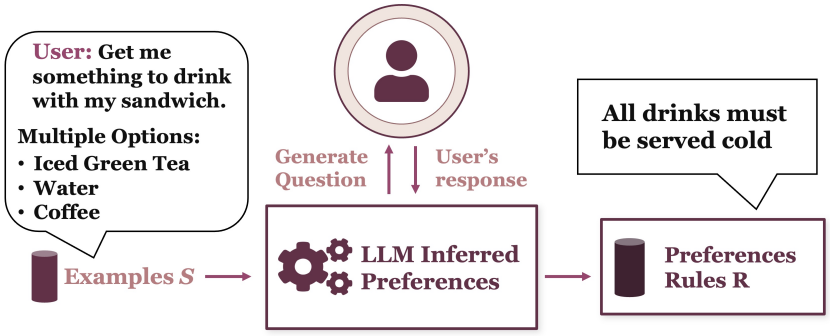

Figure 1: Interactive Preference Rule Learning.

valid but cannot determine which one is ideal. Selection among correct options would thus be near-uniform ($\sim$25% of selecting preferred drink), whereas with preference knowledge, the goal is to shift selection decisively toward the preferred option. To complete tasks in a user-aligned manner, the LLM must therefore first infer these latent preferences.

Various methods are used to improve LLM alignment, including Reinforcement Learning from Human Feedback (RLHF) (Song et al., 2024; Yuan et al., 2023; Poddar et al., 2024), in-context learning and parameter-efficient fine-tuning (Dong et al., 2024), language-based hierarchical architectures (Liu et al., 2024; Borate et al., 2025), and zero- and few-shot method (Hejna & Sadigh, 2023; Zhao et al., 2024). While these methods are promising, they are often computationally and data inefficient, impose restrictive assumptions on preference structure, and rely on oversimplified models of human behavior (Hejna & Sadigh, 2023; Ghose et al., 2025; Zhong et al., 2024).

By relying primarily on indirect preference feedback signals such as rankings, demonstrations, or annotations, existing approaches underutilize the LLM's capacity to infer intent and reason about high-level tasks through dynamic, natural-language interaction. This issue is particularly consequential in HRI settings, where social agency and humanlike communications are central to engaging interactions (Zhang et al., 2025; Dautenhahn, 2007). Supporting effective personalization requires collecting preference feedback using natural interfaces that are aligned with how users communicate their goals, expectations, and constraints (Zhang et al., 2025; Whelan et al., 2018). Such feedback methods enable agents to capture complex and nested user preferences, and contextual shifts. These factors are oversimplified in existing approaches but are critical for reliable and sustained human–robot collaboration (Di Napoli et al., 2023).

To address these gaps, we introduce CLIPR (Conversational Learning for Inferring Preferences and Reasoning), an LLM-based framework for natural, language-driven preference learning that treats the LLM as an interactive intermediary between the user and the agent. CLIPR acquires preference information via dialogue with the user and encodes it into actionable, rule-based representations that generalize across environments and tasks. The interaction is designed to capture complex dependencies and the nested nature of user preferences with minimal cognitive burden on the user. We extend CLIPR with adaptive feedback (Adaptive CLIPR) to support the continuous refinement of learned rules by incorporating mechanisms for dynamic feedback and updating preference information based on the scenarios encountered and expected improvement, which supports improving alignment over time.

The main contributions of this paper are: (1) a natural language–based framework for learning rich, structured user preferences from minimal interaction, while reducing user burden and data requirements; (2) an adaptive interaction mechanism that selectively requests feedback through natural language dialogue and incrementally updates preference representations, achieving improved alignment with substantially lower cost than existing methods; and (3) empirical evaluation on three datasets that show that our approach outperforms the state-of-the-art baselines in improving alignment, while significantly reducing the inference and runtime costs.

2 Related Works

LLMs for Modeling Human Behavior Prior works have used LLMs to model human behaviors for human-aware decision-making in human-robot settings (Ritschel et al., 2017; Mahadevan et al., 2024). This includes using LLMs for learning and representing user mental models (Gebellí et al., 2025), simulating and approximating human behaviors for human-aware motion planner (Li et al., 2023; Sisbot et al., 2007; Park et al., 2023), and reasoning about ambiguous natural language queries for safe task execution (Yang et al., 2023). In addition, LLMs have also been used to estimate and model uncertainty, and make decisions on how to act in uncertain environments, including when to ask for clarification or assistance (Ren et al., 2023).

User Preference Representation Learning In settings that demand preference representation to be more controllable or verifiable, LLMs have been used to generate symbolic predicates in spatial planning and mobile manipulation (Han et al., 2024) or to generate a reward function (Xie et al., 2024; Peng et al., 2024; Yang et al., 2024). To overcome task-specific ambiguities, a recent work proposed constructing a knowledge base with examples as post-hoc rationalizations of human-selected safe and compliant plans (Liang et al., 2024). However, it does not provide a robust mechanism for specifying and learning individual, non-trivial preferences. While some prior works introduce more natural, interaction-based methods for preference elicitation (Bärmann et al., 2024; Han et al., 2025; Li et al., 2026), their representation of preferences is often unstructured, hindering generalizability. A recent work also uses LLMs to generate a direct set of rules that represent user preferences (Wu et al., 2023). However, they treat direct rule creation as summarization, assuming few-shot examples with correct answers have already been provided. This limits generalization, encourages memorization, and makes it difficult to capture complex or context-dependent preferences that may require clarification. One notable exception is CIPHER (Gao et al., 2024), an algorithm that directly learns and operationalizes user preferences through natural language feedback, without requiring a few-shot demonstrations of correct responses. CIPHER achieves this by reasoning over discrepancies between user expectations and actions selected. However, this approach incurs high computational costs due to its constant need for feedback and does not leverage any prior knowledge about users or past examples. In contrast, we propose a method that can extract actionable rules from preferences, such that the rules are generalizable to a family of tasks.

3 Interactive Learning of Latent User Preferences

We consider a setting in which an agent operates on tasks specified through natural-language queries, using an LLM as its reasoning module. The LLM is responsible for interpreting user requests and selecting from a predefined set of candidate actions to satisfy the query. Formally, the LLM receives a natural-language user request x along with a set of candidate actions $A_x = \{a_1, a_2, \ldots, a_k\}$. A subset of these candidates, $\mathcal{C}(x) \subseteq A_x$, are the *correct* actions, representing those actions that satisfy the explicit requirements of the request. However, not all correct actions are equally desirable. A *preferred* action, $a^* \in \mathcal{C}(x)$, is a correct action that also aligns with the user's *latent* preferences.

We propose CLIPR (Conversational Learning for Inferring Preferences and Reasoning) to improve alignment by learning user preferences as a set of representative rules. The CLIPR's protocol for learning user preferences is defined in Algorithm 1.

Example Set Initialization: CLIPR is initialized with a small example set $\mathcal{S}$ approximating the tasks the agent will assist the user with. This set need not cover all possible tasks, but should convey the agent's general purpose and the nature of expected interactions. Crucially, the candidate actions in $\mathcal{S}$ must include cases where multiple actions in $\mathcal{C}(x)$ are correct, but latent user preferences determine the preferred action a^*. In our running example of getting a drink to go with sandwich, $\mathcal{S}$ should include drink-related tasks with varied options (e.g., iced green tea, water, alcohol, hot coffee) where preferences such as favoring cold non-alcoholic drinks would determine the ideal choice. Note, however, that neither preferences nor preferred action a^* are ever exposed to the LLM in example set $\mathcal{S}$.

Interactive Preference Elicitation: Given $\mathcal{S}$, CLIPR iteratively analyzes the examples provided in $\mathcal{S}$ alongside the current dialogue history $\mathcal{D}$ (Line 3 in Alg. 1). Given an interaction budget T that denotes how many questions can be posed to the user, the LLM automatically determines what questions to ask based on $\mathcal{S}$ and $\mathcal{D}$. At the beginning of interaction, dialogue history $\mathcal{D}$ is empty

Algorithm 1: CLIPR

Input : Example set $\mathcal{S}$, Maximum interactions allowed T, language model $\mathcal{M}$
Output: Preference rules R

1 $\mathcal{D} \leftarrow \emptyset$
2 **for** $t = 1$ **to** T **do**
3 $\quad \mathcal{P} \leftarrow \text{AnalyzeExamples}(\mathcal{S}, \mathcal{D}, \mathcal{M})$
4 $\quad$ **if** $\text{IsSufficient}(\mathcal{P}, \mathcal{D}, \mathcal{M})$ **then**
5 $\quad\quad$ **break**
6 $\quad q_t \leftarrow \text{GenerateQuestionForUser}(\mathcal{P}, \mathcal{D}, \mathcal{M})$
7 $\quad a_t \leftarrow \text{CollectUserResponse}(q_t)$
8 $\quad \mathcal{D} \leftarrow \mathcal{D} \cup \{(q_t, a_t)\}$
9 $R \leftarrow \text{SynthesizeRules}(\mathcal{D}, \mathcal{S}, \mathcal{M})$
10 **return** R

(Line 1). An example of a query to the user related to the running example is "When it comes to snacks and food choices, do you generally prefer healthier options (like fresh fruits, nuts, yogurt) or are you more inclined toward indulgent treats (like cookies, chocolate, sugary cereals)?" to which the user may respond with "Yes, definitely healthier options. I am on a diet, so I try to stick to food that is healthy, but I do like sweets. When possible, I would prefer options that are both sweet and healthy". Each question-answer pair is added to $\mathcal{D}$ (Line 8) and this process repeats until the interaction budget T is exhausted or the LLM decides that a sufficient amount of information has been provided (Line 4). After interacting with the user to elicit their preferences, the LLM is prompted to synthesize the dialogue into an explicit preference rules list R (Line 9). The learned user preferences are stored as a list of concise, actionable *rules*. An example of a rule synthesized from above example interactions is *"Always prioritize healthy food options over indulgent/unhealthy alternatives"*. These rules are then applied at inference time for user-aligned behavior.

3.1 Adaptive CLIPR: Continuous Learning through Adaptive Feedback

The rules learned with CLIPR, described above, are static. These rules may be incorrect or incomplete since they are derived based on limited interactions with the user. To overcome this limitation, we introduce an adaptive feedback mechanism that enables the revision and refinement of rules, based on agent performance. Specifically, we extend CLIPR to gather additional information from the user so as to update the rules (Figure 2). A *preference rule critic* monitors the agent performance and decides when to seek additional information from the user for updating the rules, based on the current failure rate. This yields richer, targeted feedback while reducing cost as the feedback is sought only when failure rates exceed an accepted baseline performance. This adaptive feedback mechanism also enables improved generalization and adaptability to new environments and tasks, via continual learning of preferences.

Algorithm 2 formalizes this process. Based on the initial rules learned using Algorithm 1, the model performs inference on incoming test tasks. We assume the test cases are processed in batches of size k for efficiency, with the adaptive feedback mechanism applied in feedback interval of f scenarios (where $f \geq k$). The LLM selects actions for each test case in the batch using the current rules R (Line 4), and batch accuracy is recorded. The algorithm maintains a running history of accuracy $\mathcal{H}$ of action selection, with respect to the user's (latent) preferences, which is updated after every batch (Line 15). This history provides a baseline characterized by mean $\mu_{\mathcal{H}}$ and standard deviation $\sigma_{\mathcal{H}}$, requiring at least 2 batches to establish.

The preference rule critic determines whether to intervene, based on whether the failure rate exceeds the historical baseline by a sensitivity threshold α. In our setting, the preference rule critic intervenes only when $(1 - \text{acc}_b) > (1 - \mu_{\mathcal{H}}) + \alpha \cdot \sigma_{\mathcal{H}}$ where acc_b denotes the action selection accuracy for instances in current batch b. Since $\mu_{\mathcal{H}}$ and $\sigma_{\mathcal{H}}$ are unavailable when $|\mathcal{H}| < 2$, the critic always intervenes when $|\mathcal{H}| < 2$.

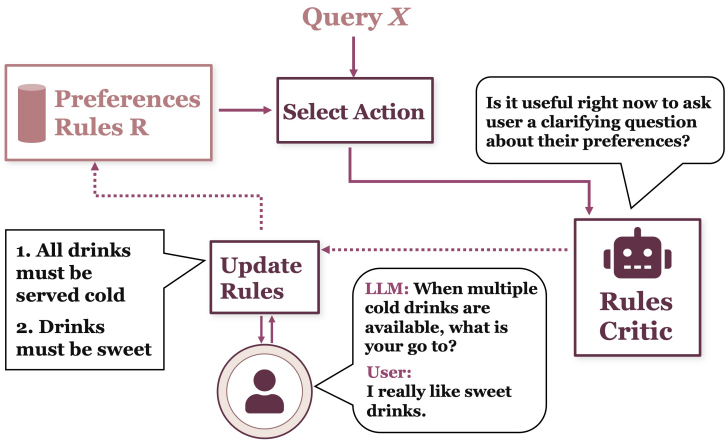

Figure 2: Rules Update via Adaptive Feedback

Algorithm 2: Adaptive CLIPR

Require : Rules R, Test scenarios $\mathcal{E}$, batch size k, feedback interval f (where $f \geq k$), sensitivity α
Definitions: $\mathcal{B}$: batch of k consecutive test scenarios; $\mathcal{H}$: history of per-batch accuracies; $\mu_{\mathcal{H}}, \sigma_{\mathcal{H}}$: mean and std. dev. of $\mathcal{H}$

1 $\mathcal{H} \leftarrow []$
2 $c \leftarrow 0$ `// scenario counter`
3 **for** *each batch* $\mathcal{B} \subset \mathcal{E}$ *of* k *scenarios* **do**
4 $\quad \text{acc}_b \leftarrow \textsc{SelectAction}(\mathcal{B}, R)$ `// accuracy of the batch`
5 $\quad c \leftarrow c + k$
$\quad$ `// Apply feedback mechanism every` f `scenarios`
6 $\quad$ **if** $c \mod f = 0$ **then**
7 $\quad\quad \text{intervene} \leftarrow |\mathcal{H}| < 2 \ \vee \ (1 - \text{acc}_b) > (1 - \mu_{\mathcal{H}}) + \alpha \cdot \sigma_{\mathcal{H}}$
8 $\quad\quad$ **if** intervene **then**
9 $\quad\quad\quad$ **if** $\textsc{Critic.AssessFailures}(\mathcal{B}, R)$ **then**
10 $\quad\quad\quad\quad$ **if** $\textsc{Critic.NeedsUserInput}()$ **then**
11 $\quad\quad\quad\quad\quad q \leftarrow \textsc{Critic.GenerateQuery}(\mathcal{B}, R)$
12 $\quad\quad\quad\quad\quad r \leftarrow \textsc{QueryUser}(q)$
13 $\quad\quad\quad\quad\quad$ **if** $\textsc{Critic.ShouldUpdateRules}(r)$ **then**
14 $\quad\quad\quad\quad\quad\quad R \leftarrow \textsc{Critic.UpdateRules}(R, q, r)$
15 $\quad \mathcal{H}.\text{append}(\text{acc}_b)$
16 **return** R

When intervening, the critic makes three sequential judgments: (1) whether to analyze specific failure instances in the batch (line 9), (2) whether user input is needed to resolve the gap (Line 10), and (3) whether the user's response will likely result in a rule update (Line 13). The sequential nature of this assessment ensure that the rules are updated only after querying the user to gather more information, thereby preventing the update of rules to overfit to test instances. If the critic decides to query the user, it generates a query (Line 11) to gather additional information, and then refines the rules accordingly (Line 14). An example of a rule update is modifying "Always prioritize healthy options, when available" to "For drinks, prioritize option that is both healthy and cold". This gated approach ensures that feedback is sought only when it is likely to yield substantive improvements. Sequential assessments further ensure that the user is not queried unnecessarily, and that if the user's feedback is unhelpful or unclear, the critic can choose not to update rules based on it.

4 EXPERIMENTS

Baselines. We compare the performance of our approaches, CLIPR and Adaptive CLIPR, with that of six baselines: (1) CIPHER Gao et al. (2024) with Levenshtein distance measure between the selected action and action guided by the user preferences; (2) CIPHER Gao et al. (2024) with cosine semantic similarity distance measure between the selected action and action guided by the user preferences; (3) Introspective Planning (IP) Liang et al. (2024), being a direct replication approach of the original paper with knowledge base generation and inference replicated; (4) Introspective Planning (IP Extended) Liang et al. (2024) with knowledge base examples being provided directly, and only reasoning over the knowledge base and inference replicated from the original work; (5) In-context learning (ICL) without the correct answers in the examples; (6) In-context learning (ICL) with correct answers provided in the examples.

Datasets. We evaluate our approach on three datasets designed to test preference learning in ambiguous single-decision settings, where multiple actions satisfy explicit task requirements, but only one aligns with the user's latent preferences.

1. Kitchen Domain: In this custom domain, tasks involve an agent selecting actions to fulfill user requests for food and drink items in a household kitchen setting (e.g., "Can you get me strawberry yogurt?" or "Bring me something to drink"). The preference learning phase uses 15 examples with up to 15 user interactions permitted. Evaluation spans four conditions with increasing distribution shift: (C1) in-distribution home kitchen tasks, (C2) grocery store with identical food items, (C3) mountain climbing settings with context-appropriate food items, and (C4) out-of-distribution environments (airports, hotels, trains) with mixed food and non-food items. User preferences include brand preference, serving temperature, time-of-day beverage selections, as well as diet and alcohol restrictions.

2. Flower Shop Domain: In this custom domain, tasks involve an agent selecting actions to fulfill user requests for floral arrangements and shop management in a flower shop setting (e.g., "Which flowers should I use for the wedding centerpiece?" or "How should I water the orchid display?"). Requests are more open-ended than in Kitchen settings. The preference learning phase uses 15 examples with up to 15 user interactions permitted and distribution shift presents: (C1) in-distribution flower shop tasks with standard inventory, (C2) outdoor event venues (weddings, corporate events) with identical flowers and materials, (C3) home and garden settings with similar flowers and plants, and (C4) out-of-distribution research laboratory environments (botany, chemistry, agriculture) with some unfamiliar non-floral objects. User preferences include local grower preference, watering method by flower type, sustainability requirements, occasion-based color schemes, and allergy-driven restrictions on scented flowers indoors.

3. Mobile Manipulation Liang et al. (2024): The task involves a robot selecting objects to retrieve based on natural language commands in household environments (e.g., "Bring me that soda"). We adapted the dataset by augmenting it with ground-truth user preferences to enable evaluation of preference-aligned action selection. We evaluate the unified preferences variation, where user intent is determined by consistent priority orderings over ten object categories (e.g., Coke $>$ Pepsi $>$ Sprite for sodas; multigrain $>$ kettle $>$ jalapeño for chips; apple $>$ orange for fruits). The preference learning phase uses 11 examples with up to 10 user interactions permitted. The test set contains 309 scenarios. Dataset adaptation details are provided in the Appendix.

Evaluation Metrics. We evaluate the performance of the different methods based on two key metrics: (1) *Preference-Aligned Accuracy* measured as the proportion of responses that satisfy the conjunction of explicit constraints derived from the natural language user request query and latent preference constraints inferred from the user's preference profile. For example, given the query "bring me something to drink with my sandwich" and given that a user likes cold beverages as one of their preferences, only selecting a cold beverage option constitutes a correct response; (2) *Computational Efficiency* score for methods involving iterative preference learning (Adaptive CLIPR and CIPHER), which is measured as $E = \frac{A \cdot N}{C}$, where A denotes accuracy, C the cumulative number of LLM calls, and N the total number of action selections made by LLM.

Table 1: Mean accuracy, along with standard deviation, of different approaches evaluated on test instances across domains and LLMs. For Kitchen and Flower Shop, results are reported for in-domain, out-of-domain (OOD), and cumulative Overall accuracy. A single value overall accuracy is reported for Mobile Manipulation dataset. †For Mobile Manipulation, IP uses N=53 samples to provide a comparison against direct replication on the subset of Mobile Manipulation dataset used in the original Introspective Planning paper; all other methods use N=309 for test set, which is full subset of Mobile Manipulation dataset of ambiguous tasks. **Bold** indicates that Adaptive CLIPR outperformed all other approaches; * denotes statistical significance ($p < 0.05$, t-test) where Adaptive CLIPR exceeded either of the two best-performing non-CLIPR baselines.

	Kitchen			**Flower Shop**			**Mobile M.**
Method	In-Dom	OOD	Overall	In-Dom	OOD	Overall	Overall
Adaptive CLIPR	**95.2**±7.2*	**92.5**±8.3*	**93.4**±7.9*	90.0±10.9	**90.6**±4.2*	**90.4**±5.4*	**65.0**
CLIPR	94.4±8.3	91.2±8.3	92.3±7.8	91.7±8.3	88.3±7.7	89.2±7.3	62.1
CIPHER (Lev.)	75.0±4.4	85.8±8.0	82.1±5.3	53.3±13.9	69.8±6.4	65.7±4.9	48.9
CIPHER (Sem.)	71.1±10.9	86.1±5.1	80.9±6.7	68.3±13.7	79.2±5.6	76.4±5.5	53.1
ICL	59.2±5.2	80.4±7.6	73.2±5.5	78.3±7.5	79.4±9.5	79.2±8.8	35.3
ICL + Answers	79.2±7.7	91.2±7.0	87.1±7.1	86.7±15.1	84.4±8.9	85.0±9.7	33.7
IP Extended	72.8±11.8	82.9±1.7	79.5±4.0	81.7±3.7	82.2±2.5	82.1±2.8	50.5
IP†	72.0±9.4	86.2±3.2	81.4±3.7	83.3±0.0	82.8±3.0	82.9±2.3	39.6

5 Results and Discussion

Accuracy: The preference-aligned accuracy values of different approaches, across three datasets, are reported in Table 1. Results in Kitchen and Flower Shop domains are averaged across five LLMs (Claude Opus 4.5, Claude Sonnet 4.5, GPT-5-nano, GPT-5.2, and GPT-4-1106-preview). For Mobile Manipulation, only GPT-4-1106-preview was evaluated due to implementation constraints. For Adaptive CLIPR, $\alpha = 2.5$ was used for the evaluation. The results show that CLIPR with adaptive feedback (Adaptive CLIPR) consistently and statistically significantly ($p < 0.05$, t-test) outperforms the other methods, except in the the flower shop in-distribution testing category where it performed second best to CLIPR without the adaptive learning feedback loop. This demonstrates the advantages of Adaptive CLIPR over alternative methods that include continuous learning mechanisms, outperforming both variants of CIPER, as well as oracle based in-context learning approaches, where correct answers were provided to the model at inference. Additionally, even without feedback, CLIPR's rule-based preference representation demonstrates superior performance compared to other methods, including oracle ICL with correct answers, suggesting that both rule based representation and adaptive feedback mechanisms offer improvements over the state-of-the-art baseline approaches.

Computational Efficiency: The results in Table 2 show that Adaptive CLIPR is more cost efficient than CIPHER, across all domains. The results demonstrate that the combination of a feedback gate and a multi step decision making process performed by a rules critic that regulates the amount of LLM calls is sufficient to correct rules and improve performance, allowing Adaptive CLIPR to significantly improve its performance. Additionally, we would like to note that when compering CLIPR without adaptive feedback to other approaches without continuous learning or feedback, CLIPR's initial rules learning approach is also more efficient in terms of cost/performance ratio compared to other baselines. For instance, while the Introspective Planning approach requires 196 API calls, CLIPR requires a maximum of 11 in our implementation during training, while outperforming the Introspective Planning approach by nearly 15%. Therefore both CLIPR and Adaptive CLIPR provide a measurable improvement in efficiency that becomes especially valuable at larger scales, as well as in settings where feedback is only needed occasionally and it is essential for the approach to gather it precisely when needed, while remaining cost aware.

Effect of α: To understand the effect of α on Adaptive CLIPR's performance, we vary α between 0.05 and 3.0 and report the average accuracy, along with standard deviation, on Kitchen and FlowerShop domain (Figure 3). While there are variations between different LLM performances, the results show that Adaptive CLIPR's performance within each LLM model remains stable across varying α levels.

Table 2: Computational Efficiency Score Comparison for Methods with Continuous Feedback (Adaptive CLIPR and two variations of CIPER). The scores were computed based on the overall performance of a given method within each of the three domains, as reported in Table 1

Method	Kitchen	Flower Shop	Mobile M.
Adaptive CLIPR	**0.74**	**0.65**	**0.56**
CIPHER (Lev.)	0.38	0.29	0.20
CIPHER (Sem.)	0.28	0.26	0.19

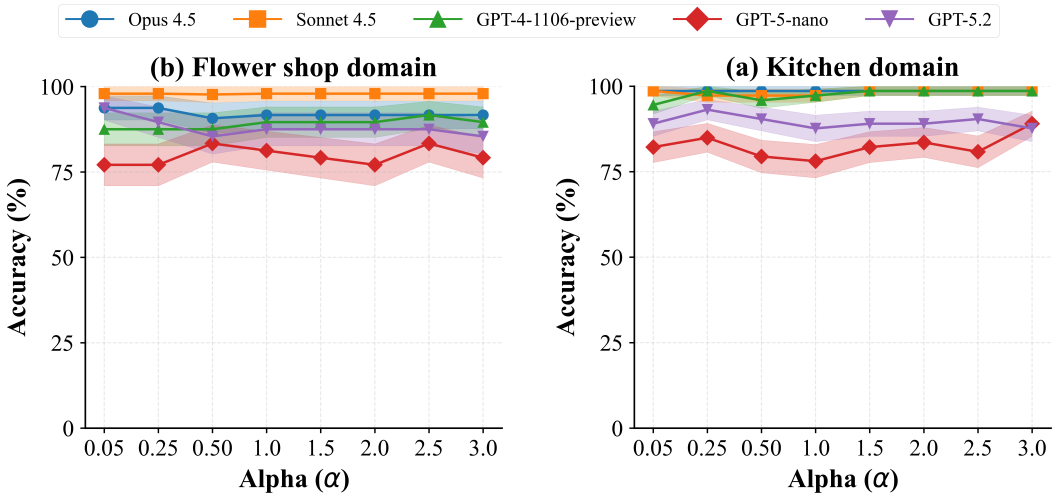

Figure 3: Effect of varying α on Adaptive CLIPR's accuracy, using five LLMs on Kitchen and Flower Shop domains.

Ablation Study One concern with Adaptive CLIPR is that it may become overly dependent on the quality of initially collected preferences during training. However, in real-world settings, users tend not to provide optimal responses, which can significantly affect (adaptive) CLIPR's ability to acquire a high-quality initial set of rules for preference representation. Therefore, we evaluate Adaptive CLIPR's performance using the best-performing LLM (Anthropic Opus 4.5) in a zero-shot setting with (1) no initial preference rules provided and (2) initial rules provided that directly contradict all true user preferences, simulating the most extreme cases of poor-quality initial rules acquired during the training stage (Figure 4). Contradicted rules were generated using LLM, prompting it to create a negation of the correct rules that were learned using the CLIPR approach.

The results demonstrate an overall accuracy improvement of +18.8% in the Flower Shop domain and +9.6% in the Kitchen domain going from the no-rules zero-shot baseline to CLIPR, with both gains statistically significant ($p < 0.05$). Crucially, despite starting without initial preference rules, CLIPR achieved final accuracies of 91.8% and 89.6% across both domains—results that are comparable to the CLIPR with no-feedback baseline as reported in 1. In the second setting with incorrect rules that contradict true user preferences, CLIPR outperforms baselines in both domains, both under incorrect rules zero-shot baselines (with gains of +89% and +54.1%, both statistically significant at $p < 0.1$) and the no-rules baseline (with gains of +6.8% and +2.1%). Further, such improvement is achieved while maintaining a high computational efficiency of 0.77%–0.82%, demonstrating that its adaptive learning feedback loop can very effectively recover even from the most substantially compromised initial rules.

These results strongly suggest that Adaptive CLIPR maintains robust performance even when initial rules are severely deficient due to human error or unclear communication. This is critical, as human feedback is rarely perfect in real-life settings, making CLIPR a very practical and robust solution.

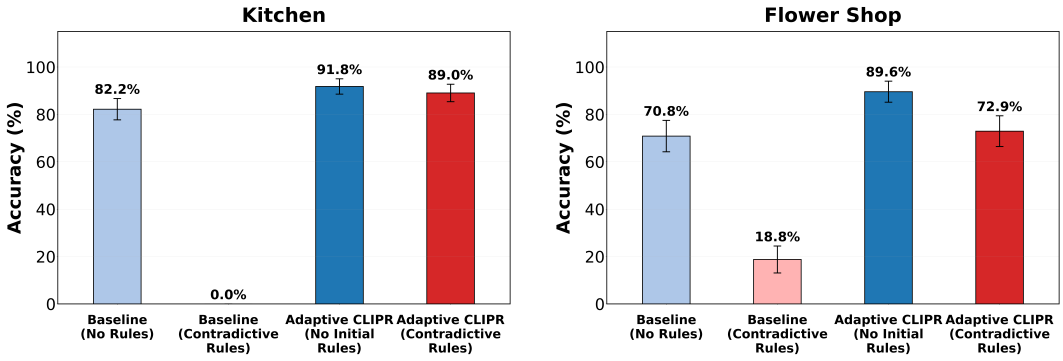

Figure 4: Comparison of zero-shot (baseline) versus Adaptive CLIPR performance on accuracy starting from no rules or contradictive rules. Comparison performed using the Anthropic Opus 4.5 model on Kitchen and Flower Shop domains, with feedback intervals of $f = 5$ and $f = 4$, respectively, and $\alpha = 1.5$ and $\alpha = 0.5$, respectively, to account for the smaller sample size of the Flower Shop domain and balance feedback session options across the two domains. In the Kitchen domain, the Adaptive CLIPR critic intervened 5 times for the no rules setting and 2 times for the contradictive rules setting. In Flower Shop domain, the Adaptive CLIPR critic intervened 2 times for both no rules and contradictive rules setting.

6 Conclusion and Future Work

This paper presents a conversational learning framework to infer actionable rules, from user interactions with the LLM, for user-aligned task completion. We also present an adaptive feedback method that enables continual revision of rules to improve alignment over time. Our evaluation across three domains using in-distribution and out-of-distribution instances demonstrate the efficiency of our approach in improving alignment while being computationally efficient. The results also indicate that LLMs can be effective in maintaining a consistent internal model of user intent.

There are, however, important limitations and open questions. The current formulation assumes that user preferences can be expressed as relatively stable constructs. While this holds for many practical settings, preferences in real environments may be context-dependent, conflicting, or temporally evolving. Extending CLIPR to represent substantially more complex, conditional, and task-dependent preferences, as well as handling preference uncertainty explicitly, remains an important direction for future work. Additionally, while the current experiments focus on single-user settings, multi-user or shared-environment scenarios introduce new challenges around preference aggregation and conflict resolution.

Acknowledgments

This work was supported in part by National Science Foundation award 2416459.

References

Pulkit Agrawal. The task specification problem. In *Proceedings of the 5th Conference on Robot Learning (CoRL)*, volume 164 of *Proceedings of Machine Learning Research*, pp. 1745–1751. PMLR, 2022.

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

Leonard Bärmann, Rainer Kartmann, Fabian Peller-Konrad, Jan Niehues, Alex Waibel, and Tamim Asfour. Incremental learning of humanoid robot behavior from natural interaction and large language models. *Frontiers in Robotics and AI*, 11:1455375, 2024. doi: 10.3389/frobt.2024.1455375.

Suraj Borate et al. Llm-based generalizable hierarchical task planning and execution for heterogeneous robot teams with event-driven replanning. *arXiv preprint arXiv:2511.22354*, 2025.

Kerstin Dautenhahn. Socially intelligent robots: dimensions of human–robot interaction. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 362(1480):679–704, 2007.

Claudia Di Napoli, Giovanni Ercolano, and Silvia Rossi. Personalized home-care support for the elderly: a field experience with a social robot at home. *User Modeling and User-Adapted Interaction*, 33:405–440, 2023.

Yan Ding, Xiaohan Zhang, Chris Paxton, and Shiqi Zhang. Task and motion planning with large language models for object rearrangement. *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2086–2092, 2023. URL https://api.semanticscholar.org/CorpusID:257496672.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, et al. A survey on in-context learning. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pp. 1107–1128, 2024.

Ge Gao, Alexey Taymanov, Eduardo Salinas, Paul Mineiro, and Dipendra Misra. Aligning llm agents by learning latent preference from user edits. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9798331314385.

Ferran Gebellí, Lavinia Hriscu, Raquel Ros, Séverin Lemaignan, Alberto Sanfeliu, and Anaís Garrell. Personalised explainable robots using llms. In *2025 20th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 1304–1308, 2025. doi: 10.1109/HRI61500.2025.10974125.

Debasmita Ghose et al. Open-ended goal inference through actions and language for human-robot collaboration. *arXiv preprint arXiv:2512.04453*, 2025.

Dongge Han, Trevor McInroe, Adam Jelley, Stefano V. Albrecht, Peter Bell, and Amos Storkey. Llm-personalize: Aligning llm planners with human preferences via reinforced self-training for housekeeping robots. In *Proceedings of the 31st International Conference on Computational Linguistics (COLING)*, pp. 1465–1474, Abu Dhabi, UAE, 2025.

Muzhi Han, Yifeng Zhu, Song-Chun Zhu, Ying Nian Wu, and Yuke Zhu. Interpret: Interactive predicate learning from language feedback for generalizable task planning. In *Robotics: Science and Systems (RSS)*, 2024.

Donald Joseph III Hejna and Dorsa Sadigh. Few-shot preference learning for human-in-the-loop rl. In *Proceedings of The 6th Conference on Robot Learning (CoRL)*, volume 205 of *Proceedings of Machine Learning Research*, pp. 2014–2025. PMLR, 2023.

T Inagaki, Akari Kato, Koichi Takahashi, Haruka Ozaki, and Genki N. Kanda. Llms can generate robotic scripts from goal-oriented instructions in biological laboratory automation. 2023. URL https://doi.org/10.48550/arXiv.2304.10267.

Alex Irpan, Alexander Herzog, Alexander Toshkov Toshev, Andy Zeng, Anthony Brohan, Brian Andrew Ichter, Byron David, Carolina Parada, Chelsea Finn, Clayton Tan, Diego Reyes, Dmitry Kalashnikov, Eric Victor Jang, Fei Xia, Jarek Liam Rettinghouse, Jasmine Chiehju Hsu, Jornell Lacanlale Quiambao, Julian Ibarz, Kanishka Rao, Karol Hausman, Keerthana Gopalakrishnan, Kuang-Huei Lee, Kyle Alan Jeffrey, Linda Luu, Mengyuan Yan, Michael Soogil Ahn, Nicolas Sievers, Nikhil J Joshi, Noah Brown, Omar Eduardo Escareno Cortes, Peng Xu, Peter Pastor Sampedro, Pierre Sermanet, Rosario Jauregui Ruano, Ryan Christopher Julian, Sally Augusta Jesmonth, Sergey Levine, Steve Xu, Ted Xiao, Vincent Olivier Vanhoucke, Yao Lu, Yevgen Chebotar, and Yuheng Kuang. Do as i can, not as i say: Grounding language in robotic affordances. In *Proceedings of the Conference on Robot Learning*, 2022. URL https://doi.org/10.48550/arXiv.2204.01691.

Callie Y. Kim, Christine Lee, and Bilge Mutlu. Understanding large-language model (llm)-powered human-robot interaction. In *Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. ACM/IEEE, 2024.

Belinda Z. Li, Alex Tamkin, Noah Goodman, and Jacob Andreas. Eliciting human preferences with language models. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2310.11589.

Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. Camel: communicative agents for "mind" exploration of large language model society. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Jacky Liang, Wenlong Huang, F. Xia, Peng Xu, Karol Hausman, Brian Ichter, Peter R. Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9493–9500, 2022. URL https://doi.org/10.48550/arXiv.2209.07753.

Kaiqu Liang, Zixu Zhang, and Jaime Fernández Fisac. Introspective planning: Aligning robots' uncertainty with inherent task ambiguity. *Advances in Neural Information Processing Systems 37*, 2024. URL https://api.semanticscholar.org/CorpusID:270226510.

Jijia Liu et al. Llm-powered hierarchical language agent for real-time human-ai coordination. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 1219–1228. IFAAMAS, 2024.

Matt MacMahon, Brian Stankiewicz, and Benjamin Kuipers. Walk the talk: Connecting language, knowledge, and action in route instructions. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI)*, pp. 1475–1482, 2006.

Karthik Mahadevan, Jonathan Chien, Noah Brown, Zhuo Xu, Carolina Parada, Fei Xia, Andy Zeng, Leila Takayama, and Dorsa Sadigh. Generative expressive robot behaviors using large language models. In *Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction*, HRI '24, pp. 482–491, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400703225. doi: 10.1145/3610977.3634999. URL https://doi.org/10.1145/3610977.3634999.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, UIST '23, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701320. doi: 10.1145/3586183.3606763. URL https://doi.org/10.1145/3586183.3606763.

Andi Peng, Belinda Z. Li, Ilia Sucholutsky, Nishanth Kumar, Julie Shah, Jacob Andreas, and Andreea Bobu. Adaptive language-guided abstraction from contrastive explanations. In *CoRL*, pp. 3425–3438, 2024. URL https://proceedings.mlr.press/v270/peng25c.html.

Sriyash Poddar, Yanming Wan, Hamish Ivison, Abhishek Gupta, and Natasha Jaques. Personalizing reinforcement learning from human feedback with variational preference learning. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9798331314385.

Krishan Rana, Jesse Haviland, Sourav Garg, Jad Abou-Chakra, Ian Reid, and Niko Suenderhauf. Sayplan: Grounding large language models using 3d scene graphs for scalable robot task planning. In Jie Tan, Marc Toussaint, and Kourosh Darvish (eds.), *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 23–72. PMLR, 06–09 Nov 2023. URL https://proceedings.mlr.press/v229/rana23a.html.

Allen Z. Ren, Anushri Dixit, Alexandra Bodrova, Sumeet Singh, Stephen Tu, Noah Brown, Peng Xu, Leila Takayama, Fei Xia, Jake Varley, Zhenjia Xu, Dorsa Sadigh, Andy Zeng, and Anirudha Majumdar. Robots that ask for help: Uncertainty alignment for large language model planners. In *Proceedings of the Conference on Robot Learning (CoRL)*, 2023.

Hannes Ritschel, Tobias Baur, and Elisabeth André. Adapting a robot's linguistic style based on socially-aware reinforcement learning. In *2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pp. 378–384, 2017. doi: 10.1109/ROMAN.2017.8172330.

Ishika Singh, Valts Blukis, Arsalan Mousavian, Ankit Goyal, Danfei Xu, Jonathan Tremblay, Dieter Fox, Jesse Thomason, and Animesh Garg. Progprompt: Generating situated robot task plans using large language models. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11523–11530, 2023. doi: 10.1109/ICRA48891.2023.10161317.

Emrah Akin Sisbot, Luis F. Marin-Urias, Rachid Alami, and Thierry Simeon. A human aware mobile robot motion planner. *IEEE Transactions on Robotics*, 23(5):874–883, 2007. doi: 10.1109/TRO.2007.904911.

Fengyu Song, Bowen Yu, Mingyuan Li, Hong Yu, Fei Huang, Yi Li, and Hao Wang. Preference ranking optimization for human alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 18990–18998, 2024. doi: 10.1609/aaai.v38i17.29865.

Ben Wang et al. Task supportive and personalized human-large language model interaction: A user study. In *Proceedings of the 2024 Conference on Human Information Interaction and Retrieval (CHIIR)*. ACM, 2024.

Yufei Wang et al. Aligning large language models with human: A survey. *arXiv preprint arXiv:2307.12966*, 2023.

Sally Whelan, Kathy Murphy, Elizabeth Barrett, Christoph Kruber, Amanda Santorelli, and Carolyn Penstein Rosé. Factors affecting the acceptability of social robots by older adults including people with dementia or cognitive impairment: A literature review. *International Journal of Social Robotics*, 10:643–668, 2018.

Jimmy Wu, Rika Antonova, Adam Kan, Marion Lepert, Andy Zeng, Shuran Song, Jeannette Bohg, Szymon Rusinkiewicz, and Thomas Funkhouser. Tidybot: Personalized robot assistance with large language models. *Autonomous Robots*, 47:1087–1102, 2023. doi: 10.1007/s10514-023-10139-z.

Tianbao Xie, Siheng Zhao, Chen Henry Wu, Yitao Liu, Qian Luo, Victor Zhong, Yanchao Yang, and Tao Yu. Text2reward: Reward shaping with language models for reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=tUM39YTRxH`.

Zhaojing Yang, Miru Jun, Jeremy Tien, Stuart J. Russell, Anca Dragan, and Erdem Biyik. Trajectory improvement and reward learning from comparative language feedback. In *8th Annual Conference on Robot Learning*, 2024.

Ziyi Yang, Shreyas Sundara Raman, Ankit Shah, and Stefanie Tellex. Plug in the safety chip: Enforcing constraints for llm-driven robot agents. *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 14435–14442, 2023. URL `https://api.semanticscholar.org/CorpusID:262044464`.

Zheng Yuan, Hongyi Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. RRHF: Rank Responses to Align Language Models with Human Feedback without tears. *arXiv e-prints*, art. arXiv:2304.05302, April 2023. doi: 10.48550/arXiv.2304.05302.

Amy W. Zhang, Raquel Queiroz, and Sarah Sebo. Balancing user control and perceived robot social agency through the design of end-user robot programming interfaces. In *Proceedings of the 20th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 899–908. IEEE, 2025.

Bowen Zhang, Harold Soh, et al. Large language models as zero-shot human models for human-robot interaction. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023.

Siyan Zhao, John Dang, and Aditya Grover. Group preference optimization: Few-shot alignment of large language models. In *Proceedings of the International Conference on Learning Representations*, 2024. doi: 10.48550/arXiv.2310.11523. arXiv:2310.11523.

Yifan Zhong, Chengdong Ma, Xiaoyuan Zhang, Ziran Yang, Haojun Chen, Qingfu Zhang, Siyuan Qi, and Yaodong Yang. Panacea: Pareto alignment via preference adaptation for llms. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, 2024.

7 APPENDIX

7.1 INTROSPECTIVE PLANNING DATASET AND REPLICATION DETAILS

Task 2 establishes deterministic ground-truth labels by resolving ambiguous scenarios through a fixed preference hierarchy. For each scenario, we detect the request type from the task text via keyword matching, then retrieve the corresponding priority list. The label is assigned to the first item in this list that appears in the scenario's available options (Algorithm 3), simulating a user with consistent, known preferences. Table 3 details the complete preference orderings.

Algorithm 3: Unified Preference Assignment for Ambiguous Scenarios

Input : Scenario $s = (t, \mathcal{A}, y_{\text{orig}})$ where t is task text, $\mathcal{A}$ is the set of available options, y_{orig} is the original label

Input : Preference hierarchies $\mathcal{P} = \{r_i : (p_1^{(i)}, p_2^{(i)}, \ldots, p_{k_i}^{(i)})\}_{i=1}^{|\mathcal{R}|}$ mapping request types to ordered preference lists

Input : Keyword sets $\mathcal{K} = \{r_i : \{w_1^{(i)}, w_2^{(i)}, \ldots\}\}_{i=1}^{|\mathcal{R}|}$ mapping request types to trigger words

Output: Unified label y^*

```
// Step 1: Detect request type via keyword matching
```
$t_{\text{lower}} \leftarrow \text{LOWERCASE}(t)$
$r^* \leftarrow$ `null`
for $(r, K) \in \mathcal{K}$ **do**
  for $w \in K$ **do**
    if $w \in t_{lower}$ **then**
      $r^* \leftarrow r$ `// e.g., ''cola'', ''soda'', ''chips''`
      break

```
// Step 2: Resolve preference using first-available policy
```
if $r^* \neq$ `null` **then**
  $(p_1, p_2, \ldots, p_k) \leftarrow \mathcal{P}[r^*]$ `// Retrieve ordered preferences`
  for $j = 1$ **to** k **do**
    if $\text{LOWERCASE}(p_j) \in \{\text{LOWERCASE}(a) : a \in \mathcal{A}\}$ **then**
      $y^* \leftarrow \text{MATCHCASE}(p_j, \mathcal{A})$
      return y^*

```
// Step 3: Fallback if no preference match
```
return $y^* \leftarrow y_{orig}$

Table 3: Preference hierarchies and detection keywords. For each request type, we define trigger keywords for detection and an ordered preference list for label assignment.

Request Type	Detection Keywords	Priority Order
COLA	"cola"	Coke ≻ Pepsi
SODA	"soda"	Coke ≻ Pepsi ≻ Sprite ≻ Orange Soda
CAFFEINATED	"caffein*"	RedBull ≻ Coke ≻ Pepsi
SWEET_DRINK	"sweet" ∧ "drink"	Coke ≻ Sprite ≻ Pepsi ≻ Orange Soda
CHIPS	"chip*"	Multigrain ≻ Rice ≻ Jalapeño ≻ Kettle
SNACK	"snack"	Multigrain Chips ≻ Energy Bar ≻ Apple
DRAWER	"drawer"	Top Drawer ≻ Bottom Drawer