In-situ Value-aligned Human-Robot Interactions with Physical Constraints

Hongtao Li[1,2], Ziyuan Jiao[2], Xiaofeng Liu[1,†] *Senior Member, IEEE*, Hangxin Liu[2,†], Zilong Zheng[2,†]

***Abstract*— Equipped with Large Language Models (LLMs), human-centered robots are now capable of performing a wide range of tasks that were previously deemed challenging or unattainable. However, merely completing tasks is insufficient for cognitive robots, who should learn and apply human preferences to future scenarios. In this work, we propose a framework that combines human preferences with physical constraints, requiring robots to complete tasks while considering both. Firstly, we developed a benchmark of everyday household activities, which are often evaluated based on specific preferences. We then introduced In-Context Learning from Human Feedback (ICLHF), where human feedback comes from direct instructions and adjustments made intentionally or unintentionally in daily life. Extensive sets of experiments, testing the ICLHF to generate task plans and balance physical constraints with preferences, have demonstrated the efficiency of our approach.**

I. Introduction

Equipping robots, especially service robots, with the ability to consider personalized human preferences is a challenging task. On the one hand, this is due to the subjectivity and diversity of human preferences, and on the other hand, the physical constraints of the objective world limit the realization of preferences. Imagine a scenario where robots tidy up a table, as shown in Fig. 1, where humans expect the robot to tidy up according to their preferences. Therefore, it is inappropriate to consider preferences without regard to physical constraints, or physical constraints only, but only to take both into account, i.e., behaving in accordance with human preferences while adhering to physical constraints.

One of the challenges is to learn human preferences, with existing methods mainly including learning by pairwise comparison [1, 2] and learning by LLMs [3–5]. The former typically simplifies complex preferences into a ranking function, which assigns a partial order to multiple outputs of the model. Such methods are easy to implement and have numerous applications in recommendation systems [6, 7], human-robot interaction [8], natural language processing [9, 10], and more. However, they require collecting a large amount of annotated data from humans [11–13], so what is learned reflects the shared preferences rather than individualized preferences [14]. Additionally, this form of comparison itself also sacrifices the diversity of human preferences. The latter arises from the powerful text processing and commonsense reasoning capabilities exhibited by LLMs trained on massive datasets [15, 16]. For example, TidyBot [3] utilizes the summarization capabilities of LLMs to infer human preferences for tidying up a room, while DROC [4] learns human preferences from online human language corrections.

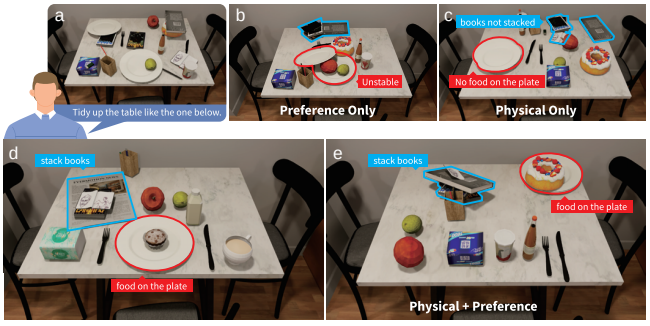

Fig. 1: **An example task of robots tidying up a table with human preferences.** (a) The messy table that needs to be cleared. (b) Considering human preferences without regard to physical constraints would result in unrealistic behavior. (c) Considering physical constraints alone would fail to meet human expectations. (d) Example of human preferences. (e) Only by balancing both, that is, behaving following human preferences while adhering to physical constraints, can the task be satisfactorily completed.

Another major challenge is the integration of learned human preferences and physical constraints. Physical constraints manifest in both task planning [17] and motion planning [18] of robots. In this paper, we only consider the physical constraints of the task planning, and the human preferences associated with it only involve the outcomes of planning. Although it is a subset of the original problem, it remains a challenging task. Due to the inherent, universal, and omnipresent nature of physical constraints, they are typically predefined in the form of hand-written rules, such as the domain description in Planning Domain Definition Language (PDDL) [19, 20], or environment modeling in reinforcement learning [21]. On the other hand, personalized human preferences are often unpredictable, ambiguous, and diverse. The heterogeneity between these two demands an appropriate way of combining them [22].

Traditional task planning methods, such as those based on PDDL or scene graphs, require converting preferences into a form recognizable by planners, such as preference predicates [23] or constraints [24]. However, this conversion often sacrifices the diversity of preferences. Another approach is to use LLMs to convert text-based human preferences into a reward function in reinforcement learning [5, 25] and learn

[1] College of Artificial Intelligence and Automation, Hohai University, Changzhou, China

[2] State Key Laboratory of General Artificial Intelligence, BIGAI, Beijing, China

This work is done during H. Li's internship at BIGAI. † indicates the corresponding authors. E-mails: {lihongtao, jiaoziyuan, liuhx, zlzheng}@bigai.ai, xfliu@hhu.edu.cn

physical constraints through a large number of trial-and-error. However, the intrinsic nature of reinforcement learning requires human preferences, serving as reward signals, to be strongly goal-oriented for specific tasks [26], thus making it less suited for learning diverse preferences.

To address the aforementioned issues, we first proposed a set of household benchmarks, collecting tasks with strong personal preferences. Then, we introduced the In-Context Learning from Human Feedback (ICLHF) algorithm, which aims to combine LLMs' preference learning capability with the ability to learn from feedback. In this approach, the LLM functions like a reinforcement learning policy model, learning human preferences through in-context learning [27, 28]. Feedback is provided in textual form, combining physical constraints and human preferences. Finally, the structure like Reinforcement Learning from Human Feedback (RLHF) of the algorithm allows for balancing physical feedback and preference feedback to generate suitable solutions. Meanwhile, the employment of in-context learning avoids training an excess of parameters for LLMs, maximizing its inferential prowess as much as possible, thus enabling in-situ personalized preference learning. To achieve generalization, learned human preferences can be easily combined into a hierarchical structure, with higher-level preferences being more adaptive. Considering the powerful capability of traditional algorithms in handling intricate manipulation tasks, we integrated a customized version of POG [29], an algorithm for efficient sequential manipulation planning on scene graphs, to enhance the preliminary plans generated by the LLM planner. Consequently, the final task plan incorporates more comprehensive geometric spatial information, thereby ensuring seamless transitions to the motion planner.

We conducted numerous experiments to validate the effectiveness of the ICLHF algorithm in learning human preferences and combining them with physical constraints. Finally, real robot experiments demonstrate the validation of the approach on robotic hardware. The contribution of our work can be summarized as follows:

- We present a benchmark on household activities whose evaluation is based on personalized preferences.
- We introduce the ICLHF algorithm that learns human preferences in situ and combines them with physical constraints to accomplish the task.
- We conduct large-scale experiments to validate the effectiveness of ICLHF, and real-world robot experiments to demonstrate its practicality.

A. Related Works

1) LLMs for Robots: As LLMs trained on massive amounts of data exhibit powerful common-sense reasoning capabilities [15, 16], a significant amount of work focuses on utilizing them for robotic task planning [25, 30, 31]. These works can be categorized based on the format of LLMs' output into two types: action primitives based [3, 30] methods and code-based [5, 25, 31] methods. Methods based on action primitives guide LLMs to generate corresponding sequences of action primitives based on different task prompts, while methods based on code aim to leverage the programming abilities of LLMs by providing specific API instructions to output execution plans [31] or objective functions [5, 25] represented in code format.

2) Task Planning: Traditional task planning in robotics mainly includes planning with symbols [23, 24] and planning with scene graphs [29, 30, 32]. The utilization of symbols for robotic task planning derives from earlier planning problems, where algorithms such as PDDL [19, 20] standardize artificial intelligence planning. 3D scene graph [33, 34] emerges as a formidable tool for scene modeling and makes many graph operations possible due to the graph structure, such as graph edit distance [29] and graph neural networks [35]. However, it also has some issues whereby even small environments can contain hundreds of objects and complex relationships between them [36].

B. Overview

We organize the remainder of this paper as follows. Section II describes the underlying framework and the ICLHF algorithm constructed on it. Section III presents the benchmark used in this paper, and exhaustive experiments are conducted on both simulation and real environments to validate the effectiveness of the ICLHF algorithm. Finally, we conclude the paper in Section IV.

II. Method

A. Framework

The problem we consider is similar to the traditional Markov Decision Process (MDP), where the state space $\mathcal{S}$ encompasses all possible states of objects, including poses, intrinsic properties, and more. The action space $\mathcal{A}$ consists of predefined action primitives, such as group, put on, and slice. Since it is often challenging [5, 25] to accurately describe physical constraints and human preferences using the traditional scalar reward function $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, we employ a novel text-based form of reward to integrate both, defined as $R^* : \mathcal{S} \times \mathcal{A} \to \mathbb{F}^{\text{text}}$. In the simulation, physical constraints are typically provided by simulators, whereas in the real world, constraints and preferences are provided by human observers either in the form of speech (which can be converted to text) or directly as text.

Specifically, environmental feedback consists of two parts. First is the execution of actions, for example, any placement actions within the container before it is opened are considered failures. Second is the consequences of actions, such as collisions or collapses caused by the operation. Based on such feedback, an intelligent agent can generate plans with higher physical feasibility. Human feedback also includes two parts. First is direct preference instructions from humans regarding unsatisfactory aspects of the execution process, demanding the agent to respond promptly and generate plans that align with human preferences. Second is the adjustments made by humans in daily life based on their preferences, which are more implicit compared to the first type of feedback, sometimes even stemming from subconscious human behavior. Inductive learning of

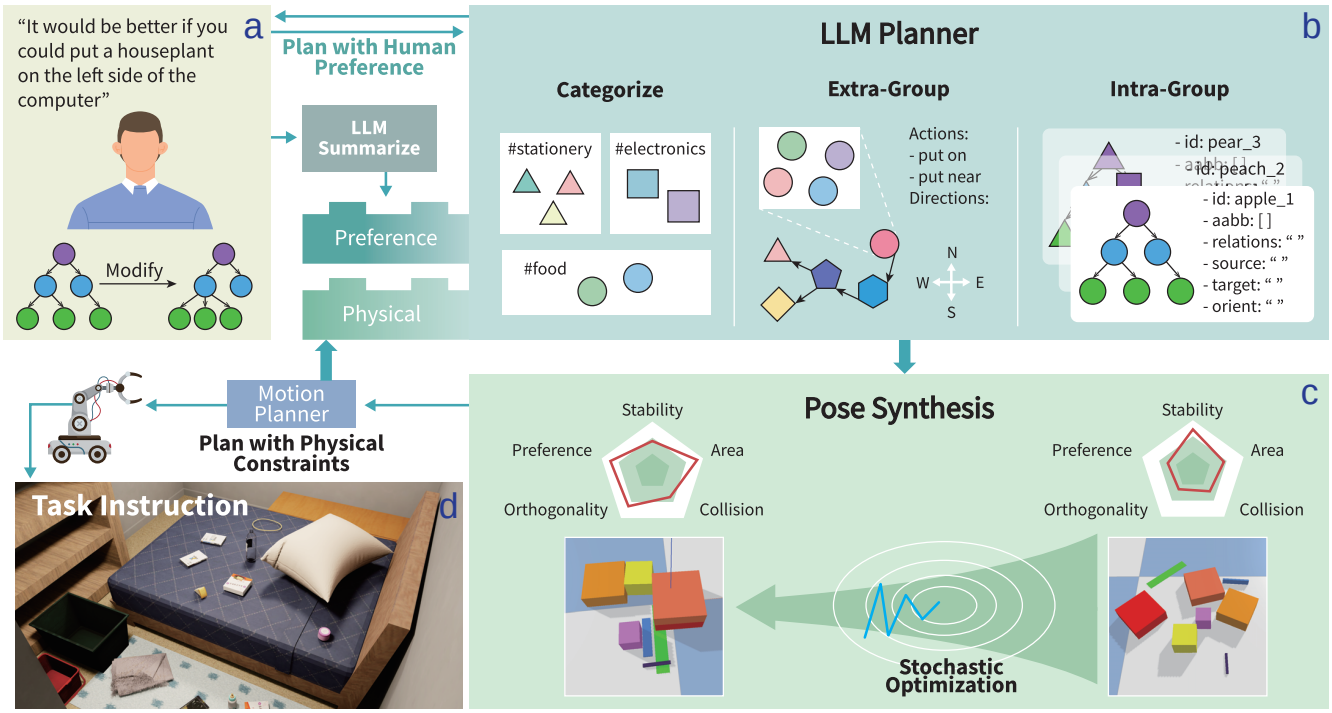

Fig. 2: **Dual loop of ICLHF**. (a-b) In the upper loop, humans give instructions or adjust the plan according to their preferences, which are then analyzed and integrated by the LLM into human preferences. (b) In the lower loop, LLM receives the initial state and task instruction as input and outputs the object attributes and relationships between objects represented by the scene graph. (c) Then, the pose synthesizer processes this scene graph and generates specific positions and rotations for the object. (d) Finally, the motion planner generates the robot's motion trajectory based on this information. After executing in the environment, the robot receives physical feedback, which is then injected into the LLM planner for subsequent planning. The whole process continues until the generated plan conforms to the physical constraints and human preferences.

these preferences poses a challenging task. Traditional algorithms [23, 24] lack effective learning methods for this, but LLMs can summarize corresponding human preferences and apply them to subsequent tasks [3].

B. ICLHF

We propose the ICLHF algorithm, which is capable of learning human preferences in situ and combining them with physical constraints to accomplish tasks. It consists of two parts: the LLM planner and the object pose synthesizer. Taking task instructions and initial states as input, the LLM planner performs in-context learning based on corresponding prompts and recorded human preferences, and outputs execution sequences along with the goal scene graph. The output scene graph is a rough version, representing objects and their attributes with nodes and carrying the relationships between objects with edges. The pose synthesizer then takes this goal scene graph as input and further synthesizes more specific object information, such as position and rotation, based on object attributes and relationships between objects. Additionally, the pose synthesizer can conduct a preliminary physical feasibility analysis on scene graphs, such as object collisions, and provide feedback on the physical aspects.

Throughout the entire process, humans can provide modification suggestions or make adjustments based on their preferences at any time. The algorithm can capture these human preferences and apply them to subsequent planning. Additionally, to utilize human preferences more efficiently, the algorithm performs periodic introspection to extract higher-level human characteristics from lower-level human preferences, typically when reaching the maximum context length of LLMs. The overall process is illustrated in Fig. 2.

1) LLM as Task Planner: The task planner adopts a top-down processing logic. Given the textual description of objects, first classify the objects. Assuming this step generates N categories, a total of $N + 1$ directed acyclic graphs will be obtained. Objects of the same category are considered as nodes within the same graph, while these N categories themselves form a graph with N nodes.

Next, consider the placement between groups, mainly involving actions of *put on* and *put near*, as well as optional orientation indications, which will provide a global placement for the N categories divided in the previous step.

Finally, groups with more than two objects will undergo more detailed operations, with the types varying according to tasks. Taking tidying up a table as an example, operations will include *put on*, *put in*, *open*, *close*, and so on. These operations either alter the relationship between objects, such as *put on* and *put in*, or change the state of a single object, like *open* and *close*.

In each of the above processes, physical feedback and preference feedback can be promptly injected to influence

subsequent planning. With physical feedback, the agent can modify parts of the plan that are not executable or unrealistic, while preference feedback will affect future planning. During the modification process, potential human preferences also need to be considered. Therefore, unlike the error handling mechanism proposed in DROC [4], which restricts retrievable history to four categories, we track the source of relationships in the scene graph that lead to errors and use them together with the relationships of neighbors as contextual input to regenerate an overall plan that better aligns with human preferences. Additionally, when the stored preferences reach the maximum token length allowed by LLMs, the planner will conduct a profile, aiming to extract more generalized features from trivial preferences.

2) POG as Pose Synthesizer: In contrast to the top-down logic of the task planner, the pose synthesizer adopts a bottom-up approach. It first analyzes the placement of objects within each group, then treats them as a whole to generate a total of N placement configurations for all groups based on their orientation and relationships with other groups.

Specifically, for the symbolic relationships generated by the task planner, we use stochastic optimization in POG [29] to determine the geometric information of the objects. To reduce computational complexity, the oriented bounding box is used instead of objects during calculations, and then the results are mapped back to the respective objects. In addition to the original objective function used in POG [29], the following additional objectives have been added.

$$\mathcal{L}_{\text{manhattan}} := \sum_{l \in \mathcal{G}} \mathbf{1}_{|l|>1} \sum_{\mathbf{m},\mathbf{n} \in l} \|\mathbf{m} - \mathbf{n}\|_1 \tag{1}$$

$$\mathcal{L}_{\text{area}} := \mathcal{L}_{\text{manhattan}} + \sum_{l \in \mathcal{G}} \mathbf{1}_{|l|>1} R(\mathbf{x}^l) \cdot R(\mathbf{y}^l) \tag{2}$$

$$\mathcal{L}_{\text{orth}} := \sigma^2(\boldsymbol{\theta}) \tag{3}$$

where l denotes the depth of nodes in the scene graph and $\mathbf{1}$ denotes the indicator function. The $\mathbf{m}, \mathbf{n}$ in Eq. (1) denote the 3D coordinates of nodes. In Eq. (2), R denotes the range, i.e. $R(\mathbf{x}) = \mathbf{x}_{\text{max}} - \mathbf{x}_{\text{min}}$, and x, y denote the x-axis and y-axis coordinates of the node, respectively. In Eq. (3), $\boldsymbol{\theta}$ is the intersection angle between the main axis of symmetry and the x-axis for each object.

Eq. (2) is primarily aimed at reducing the distance between objects and consists of two parts. Firstly, it constrains the distance between every two objects using Manhattan distance, as Eq. (1) shows, which, combined with Eq. (3), can make the arrangement of objects neat and in line with human preferences. Secondly, it constrains multiple objects to make them more compact as a whole. Eq. (3) aims to reduce deviations between the main axes of symmetry of objects. These metrics reflect more fundamental and general preferences, which, when combined with individual unique preferences, can model human preferences from multiple dimensions. Additionally, we extract objectives from POG [29] regarding stability and collisions to form quantitative metrics. It is worth noting that the stability and collision functions can also provide preliminary physical feedback.

III. Experiments

Our experiment needs to answer the following questions:

1) Why choose in-context learning to learn human preferences, and what advantages does it have over directly using LLMs for learning?
2) Can the LLM planner generate more detailed information based on the symbolic relationships between objects, such as object positions and rotations?
3) Can the ICLHF plan in a way that aligns with preferences while also adhering to physical constraints, and can it generalize to new scenarios with minimal effort?
4) How practical is the ICLHF algorithm, and can it be applied to real robots?

We conducted numerous experiments to answer the aforementioned questions. The remainder of this section is organized as follows. First, in Section III-A, we introduce a benchmark comprising common household tasks, each requiring specific preferences as evaluation criteria. Then, in Section III-B, experiments are conducted to validate the ability of in-context learning as a preference learner. Following that, Section III-C compares the ability of LLMs and the traditional algorithm, namely POG [29], in generating object geometric information. Section III-D carries out extensive experiments to analyze the ability of ICLHF to balance physical constraints and human preferences, as well as its generalization. Finally, Section III-E validates the effectiveness of ICLHF in real robot environments. Throughout the experiments, we utilized GPT-3.5 Turbo as the LLM planner.

A. Benchmark

From Behavior-1K [37], a collection of household activities matching human needs based on a large number of surveys, we filtered out four categories of tasks, the execution of which typically involves distinct human preferences, namely tidying up, cleaning, packing/unpacking, and loading/unloading, as shown in Fig. 3.

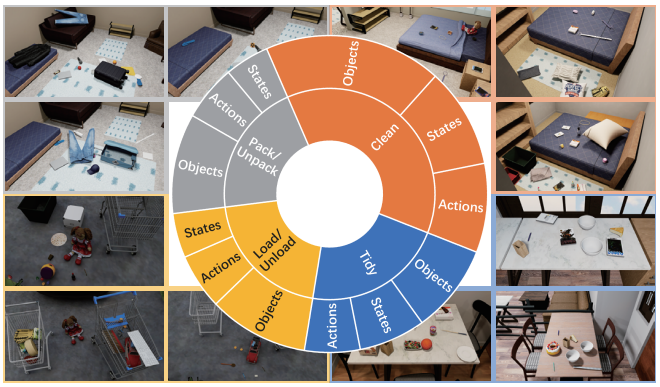

Fig. 3: Our benchmark covers four common types of tasks in household chores, the execution of which typically involves distinct human preferences, namely tidying up, cleaning, packing, and loading.

There are a total of 15 activities and 22 different scenarios in the benchmark, involving a total of 1596 objects. Table I shows the attributes of tasks in different types, each obtained by averaging and rounding across all available scenarios.

TABLE I: The four types of activities and their attributes

Activity Type	Objects	States	Actions	Amount
tidy	14	10	10	150
clean	21	16	12	100
pack/unpack	19	11	11	80
load/unload	17	12	11	80

For each type of task, we provide default human preferences. For instance, the human preference is "I prefer everything to be laid flat on the table rather than stacked together" for the task of tidying up tables. Each preference is carefully selected to ensure that there is at least one solution that aligns with the preference in the current context, while also being as general as possible to influence other types of tasks in specific scenarios.

Additionally, the benchmark considers the impact of preferences on the physical difficulty of task completion, implicitly increasing or decreasing constraints by adjusting the physical contact between objects. For example, the default preference for tidying up tables avoids stacking objects, making it easier to execute, while the default preference for unloading cars suggests placing objects in the same container, greatly reducing the feasible domain of the task. The evaluation of human preferences consists of two parts: subjective scoring and objective scoring. The subjective scoring is performed by selected participants, who rate the final RGB image from 0 to 10 based on given preferences. The objective scoring is calculated by selecting different features with varying weights according to specific preferences.

B. Symbolic Spatial Relationship Experiments

1) Settings: To test the ability of in-context learning to learn human preferences, we conducted experiments using tidying up tables as an example on the PyBullet [38] platform, involving 5 to 10 objects. This test includes two aspects. First, the ability of the algorithm to extract human preferences from modifications in object relationships, and second, the understanding and application of human preferences. To standardize the output format, in methods that do not involve in-context learning, only content related to preference learning has been removed.

Considering the inherent ability of LLMs to process semantic information, we categorize object types used in experiments into those containing semantic information, namely everyday items, and those lacking semantic information, which mainly include boxes and cylinders.

The evaluation criteria are divided into three levels: scene graph, action sequence, and preference. The scene graph includes stability and area, the action sequence includes execution efficiency and feasibility, and the preference includes learning and application. In the evaluation of the scene graph, the stability cost function is defined as

$$\mathcal{L}_{\text{stab}} := \frac{\sum_o \text{Mass}_o \cdot \|\text{CoM}_o\|_2 + \|\sum_o \text{Mass}_o \cdot \text{CoM}_o\|_2}{\sum_o \text{Mass}_o} \quad (4)$$

where Mass_o denotes the mass of object o, and CoM_o represents the center of mass of object o relative to the base object. The area cost function is defined as Eqs. (1) and (2). The corresponding scores are scaled and transformed into a range of 0 to 10 through min-max normalization. The execution efficiency of the action sequence is inversely proportional to the length of the task plan, and feasibility includes logical feasibility and physical feasibility. Logical feasibility analyzes whether the inherent logic of the plan is correct, including the format of instructions, while physical feasibility analyzes whether the plan can be successfully executed physically. Preference learning utilizes Sentence-Transformers [39] to measure the cosine similarity between learned preferences and the ground truth. The application of preferences tests the algorithm's understanding of preferences by evaluating its application to new scenarios. The overall score is calculated as the average of the subjective and objective scores from the benchmark. The scaling method for the objective score follows the same approach as the stability.

TABLE II: Results of task planning without semantic information (average across 5 to 10 objects)

Criteria		With ICL	Without ICL
Goal	Stability ↑	7.18	**7.28**
	Area ↓	**8.45**	8.61
Sequence	Length ↓	14.81	**14.67**
	Feasibility ↑	**12.98**	8.44
Preference	Learn ↑	**0.95**	0.47
	Apply ↑	**85.48**	66.67

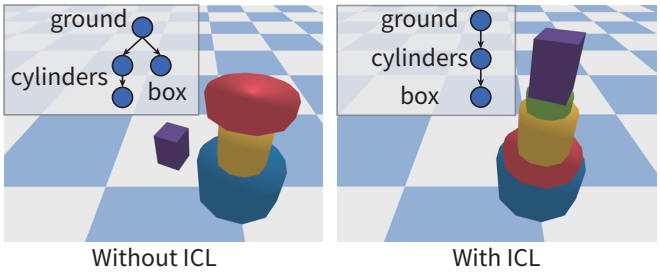

Fig. 4: The example results generated with or without in-context learning, where humans prefer mixing all objects together.

2) Results: Table II displays the corresponding results when there is no semantic information. In this experiment, 5 to 10 objects are randomly sampled within appropriate ranges of categories and sizes. When the number of objects in a category is less than one-third of the total, or greater than two-thirds, the preference is to mix boxes and cylinders, meaning there exist instances of one category of objects placed on top of another. In all other cases, the preference is to separate boxes and cylinders. From Table II, it can be seen that using in-context learning greatly improves the feasibility of generating plans, as well as preference learning and application. Fig. 4 shows visual examples of scenarios requiring mixed objects.

Table III presents the results with semantic information. When identical objects are present, the preference is set to disallow stacking identical objects together. From Table III, it can be observed that the overall feasibility of the plan

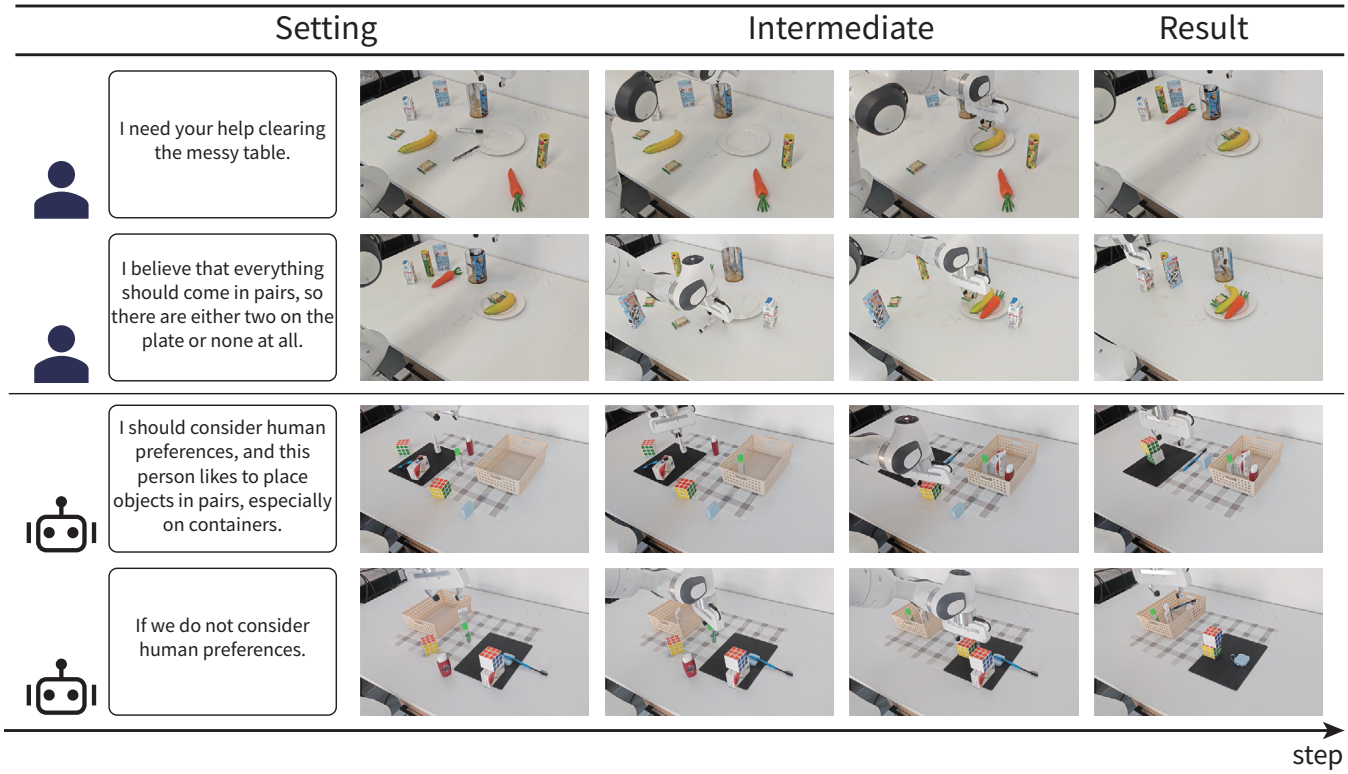

Fig. 5: The results of two sets of real robots tidying up a table. The first was configured with zero-shot learning, while the second used one-shot learning. Results without preference were also provided for comparison.

Fig. 6: The example of using LLM-GROP and POG to generate object poses.

is improved when using everyday objects, while the use of in-context learning still enhances preference learning and application capabilities. Additionally, we observed that as the number of objects increases, the logical feasibility of the plans generated by the LLM significantly decreases, with frequent occurrences of hallucinations and inconsistencies. This issue is particularly evident in methods that do not involve in-context learning, where the LLM often generates nonexistent actions or operation logic in an attempt to handle human preferences.

TABLE III: Results of task planning with semantic information (average across 5 to 10 objects)

Criteria		With ICL	Without ICL
Goal	Stability ↑	**8.69**	8.62
	Area ↓	8.02	**7.98**
Sequence	Length ↓	**12.32**	12.65
	Feasibility ↑	**26.45**	10.74
Preference	Learn ↑	**0.86**	0.39
	Apply ↑	**93.33**	89.02

C. Geometric Spatial Relationship Experiments

1) Settings: We aim to have the LLM generate more precise geometric spatial relations based on symbolic spatial relationships using the algorithm proposed in LLM-GROP [40], and we compare these results with traditional optimization-based algorithms. The experimental setup, based on LLM-GROP [40], involves a service robot tasked with arranging a dining table. For testing, we sample 3 to 5 objects from 7 categories, totaling 26.

The evaluation criteria for this experiment include success rate and orthogonality. The success rate is defined as whether the method can place all objects on the table without collisions. Orthogonality is defined as in Eq. (3).

2) Results: Table IV shows the results. Orthogonality is transformed into scores using min-max scaling. It can be observed that even with only 3 to 5 objects, the success rate of LLM-GROP [40] is below 60%, while POG [29] achieves 100%. Additionally, POG [29] outperforms in orthogonality, indicating a neater placement of objects. Fig. 6 illustrates examples of object poses generated using LLM-GROP [40] and POG [29], respectively.

TABLE IV: Results of geometric spatial planning (average across 3 to 5 objects)

Criteria	LLM-GROP	POG
Orthogonality Score (·/10)	5.45	**7.04**
Success Rate (%)	56.67	**100**

D. Simulation Experiments

1) Settings: We conducted ablation experiments on the OmniGibson [37] platform to validate the ICLHF algorithm's ability to plan in accordance with human preferences while adhering to physical constraints. The experiments are based on the benchmark we previously proposed and are divided into four categories: tidying up, unloading, unpacking, and cleaning, each with corresponding human preferences. To enhance the complexity of the experiment, we integrated various human preferences from previous tasks into a room tidying task. This enabled us to analyze how the algorithm balances complex and diverse preferences in more realistic scenarios. Additionally, we observed that LLM itself possesses many common human preferences, so the preferences used in the experiments have distinct personalities. The number of objects in the experiments ranges from 5 to 15, with their categories and poses sampled within appropriate ranges. The method for evaluating preferences is similar to the previous one, supplemented with quantitative scores.

2) Results: Fig. 7 presents the results of the experiments, visualizing some scenarios and supplemented with quantitative scores. The fourth row of RGB images depicts a scene that combines the preferences from the scenes in the first three rows. It can be seen that plans generated solely under physical constraints do not meet specific human preferences, while considering preferences alone may result in impractical plans. Only the ICLHF algorithm, which simultaneously considers both physical constraints and human preferences, is capable of addressing this challenge. Additionally, the algorithm demonstrates strong generalization abilities, meaning previously learned human preferences can be applied to unknown scenarios.

The computational complexity primarily involves LLM and the pose synthesis module. The latter's efficiency is improved through the use of oriented bounding boxes and parallel processing of different groups, averaging 0.6 seconds in current experiments.

E. Real Robot Experiments

Lastly, we conducted manipulation experiments using a Franka Research 3 manipulator, where the task was to tidy up a table. Initially, the robotic arm followed physical constraints to tidy up the table efficiently. Subsequently, human preferences were explicitly expressed, and the robotic arm adjusted its actions accordingly. When faced with a new scenario, the robotic arm planned actions based on previously learned human preferences while strictly adhering to physical constraints. As a comparison, results without considering human preferences were provided for evaluation. The experimental results are shown in Fig. 5, demonstrating the robot's excellent task completion and adaptability.

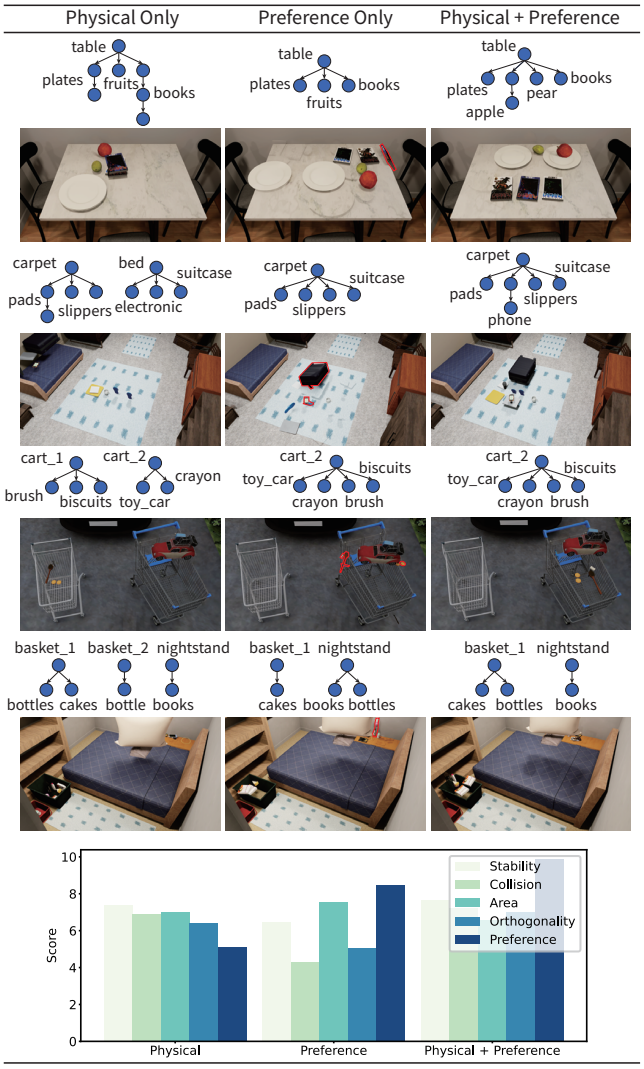

Fig. 7: The results of the ablation experiments include visualizations of selected scenes, supplemented by quantitative data analysis. Preferences for tidying involve laying objects flat on the table. Preferences for unloading entail placing all items in the same cart, and unpacking preferences dictate avoiding placing items unrelated to sleeping on the bed. The ICLHF algorithm, which integrates both physical constraints and human preferences, can generate plans that are physically feasible while also aligning with human preferences.

IV. Conclusion

In this paper, we introduce a dual-loop planning framework that integrates physical constraints and human preferences, offering a novel human-in-the-loop paradigm. Based on this framework, we propose the In-Context Learning from Human Feedback (ICLHF) algorithm, which can learn human preferences in situ and make plans that adhere to physical constraints while aligning with preferences. To validate the effectiveness of the proposed algorithm, we introduce a novel benchmark that incorporates personalized preferences into the evaluation process. We conduct extensive experiments to verify the capabilities of the ICLHF algorithm across various aspects. Finally, real robot experiments demonstrate its practicality in robotic hardware.

Acknowledgment

The authors thank the reviewers for their insightful suggestions to improve the manuscript. This work presented herein is supported by the National Natural Science Foundation of China (62376031).

REFERENCES

[1] J. Fürnkranz and E. Hüllermeier, "Pairwise preference learning and ranking," in *Machine Learning: ECML 2003*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 145–156.

[2] ——, "Preference learning and ranking by pairwise comparison," in *Preference Learning*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 65–82.

[3] J. Wu, R. Antonova, A. Kan, M. Lepert, A. Zeng, S. Song, J. Bohg, S. Rusinkiewicz, and T. Funkhouser, "Tidybot: Personalized robot assistance with large language models," *Autonomous Robots*, vol. 47, no. 8, pp. 1087–1102, 2023.

[4] L. Zha, Y. Cui, L.-H. Lin, M. Kwon, M. G. Arenas, A. Zeng, F. Xia, and D. Sadigh, "Distilling and retrieving generalizable knowledge for robot manipulation via language corrections," in *International Conference on Robotics and Automation (ICRA)*, 2024, pp. 15 172–15 179.

[5] Y. J. Ma, W. Liang, G. Wang, D.-A. Huang, O. Bastani, D. Jayaraman, Y. Zhu, L. Fan, and A. Anandkumar, "Eureka: Human-level reward design via coding large language models," in *International Conference on Learning Representations (ICLR)*, 2024.

[6] Z. Zhao, H. Lu, D. Cai, X. He, and Y. Zhuang, "User preference learning for online social recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 9, pp. 2522–2534, 2016.

[7] J. He, X. Li, and L. Liao, "Category-aware next point-of-interest recommendation via listwise bayesian personalized ranking," in *International Joint Conference on Artificial Intelligence (IJCAI)*, 2017, pp. 1837–1843.

[8] M. Cakmak, S. S. Srinivasa, M. K. Lee, J. Forlizzi, and S. Kiesler, "Human preferences for robot-human hand-over configurations," in *International Conference on Intelligent Robots and Systems (IROS)*, 2011, pp. 1986–1993.

[9] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei, "Deep reinforcement learning from human preferences," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017.

[10] D. J. H. III and D. Sadigh, "Few-shot preference learning for human-in-the-loop rl," in *Conference on Robot Learning (CoRL)*. PMLR, 2023, pp. 2014–2025.

[11] J. Ji, M. Liu, J. Dai, X. Pan, C. Zhang, C. Bian, B. Chen, R. Sun, Y. Wang, and Y. Yang, "Beavertails: Towards improved safety alignment of llm via a human-preference dataset," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 36, 2023, pp. 24 678–24 704.

[12] Y. Kirstain, A. Polyak, U. Singer, S. Matiana, J. Penna, and O. Levy, "Pick-a-pic: An open dataset of user preferences for text-to-image generation," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 36, 2023, pp. 36 652–36 663.

[13] X. Wu, K. Sun, F. Zhu, R. Zhao, and H. Li, "Human preference score: Better aligning text-to-image models with human preference," in *International Conference on Computer Vision (ICCV)*, October 2023, pp. 2096–2105.

[14] M. Bakker, M. Chadwick, H. Sheahan, M. Tessler, L. Campbell-Gillingham, J. Balaguer, N. McAleese, A. Glaese, J. Aslanides, M. Botvinick, and C. Summerfield, "Fine-tuning language models to find agreement among humans with diverse preferences," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, 2022, pp. 38 176–38 189.

[15] J. Wei, X. Wang, D. Schuurmans, M. Bosma, b. ichter, F. Xia, E. Chi, Q. V. Le, and D. Zhou, "Chain-of-thought prompting elicits reasoning in large language models," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, 2022, pp. 24 824–24 837.

[16] S. Yao, D. Yu, J. Zhao, I. Shafran, T. Griffiths, Y. Cao, and K. Narasimhan, "Tree of thoughts: Deliberate problem solving with large language models," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 36, 2023, pp. 11 809–11 822.

[17] N. T. Dantam, Z. K. Kingston, S. Chaudhuri, and L. E. Kavraki, "An incremental constraint-based framework for task and motion planning," *International Journal of Robotics Research (IJRR)*, vol. 37, no. 10, pp. 1134–1151, 2018.

[18] T. Marcucci, M. Petersen, D. von Wrangel, and R. Tedrake, "Motion planning around obstacles with convex optimization," *Science Robotics*, vol. 8, no. 84, p. eadf7843, 2023.

[19] D. M. McDermott, "The 1998 ai planning systems competition," *AI Magazine*, vol. 21, no. 2, p. 35, 2000.

[20] M. Fox and D. Long, "Pddl2. 1: An extension to pddl for expressing temporal planning domains," *Journal of Artificial Intelligence Research*, vol. 20, pp. 61–124, 2003.

[21] D. Reda, T. Tao, and M. van de Panne, "Learning to locomote: Understanding how environment design matters for deep reinforcement learning," in *Proceedings of the 13th ACM SIGGRAPH Conference on Motion, Interaction and Games*, 2020.

[22] J. A. Baier and S. A. McIlraith, "Planning with preferences," *AI Magazine*, vol. 29, no. 4, pp. 25–36, 2008.

[23] G. Canal, G. Alenyà, and C. Torras, "Adapting robot task planning to user preferences: an assistive shoe dressing example," *Autonomous Robots*, vol. 43, no. 6, pp. 1343–1356, 2019.

[24] J. Kim, C. Banks, and J. Shah, "Collaborative planning with encoding of users' high-level strategies," in *AAAI Conference on Artificial Intelligence (AAAI)*, vol. 31, no. 1, 2017.

[25] W. Yu, N. Gileadi, C. Fu, S. Kirmani, K.-H. Lee, M. G. Arenas, H.-T. L. Chiang, T. Erez, L. Hasenclever, J. Humplik, B. Ichter, T. Xiao, P. Xu, A. Zeng, T. Zhang, N. Heess, D. Sadigh, J. Tan, Y. Tassa, and F. Xia, "Language to rewards for robotic skill synthesis," in *Conference on Robot Learning (CoRL)*. PMLR, 2023, pp. 374–404.

[26] J. Eschmann, "Reward function design in reinforcement learning," *Reinforcement learning algorithms: Analysis and Applications*, pp. 25–33, 2021.

[27] Q. Dong, L. Li, D. Dai, C. Zheng, J. Ma, R. Li, H. Xia, J. Xu, Z. Wu, T. Liu, *et al.*, "A survey on in-context learning," *arXiv preprint arXiv:2301.00234*, 2022.

[28] S. Min, X. Lyu, A. Holtzman, M. Artetxe, M. Lewis, H. Hajishirzi, and L. Zettlemoyer, "Rethinking the role of demonstrations: What makes in-context learning work?" in *Annual Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2022.

[29] Z. Jiao, Y. Niu, Z. Zhang, S.-C. Zhu, Y. Zhu, and H. Liu, "Sequential manipulation planning on scene graph," in *International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 8203–8210.

[30] K. Rana, J. Haviland, S. Garg, J. Abou-Chakra, I. Reid, and N. Suenderhauf, "Sayplan: Grounding large language models using 3d scene graphs for scalable robot task planning," in *Conference on Robot Learning (CoRL)*. PMLR, 2023, pp. 23–72.

[31] I. Singh, V. Blukis, A. Mousavian, A. Goyal, D. Xu, J. Tremblay, D. Fox, J. Thomason, and A. Garg, "Progprompt: Generating situated robot task plans using large language models," in *International Conference on Robotics and Automation (ICRA)*, 2023, pp. 11 523–11 530.

[32] Y. Zhu, J. Tremblay, S. Birchfield, and Y. Zhu, "Hierarchical planning for long-horizon manipulation with geometric and symbolic scene graphs," in *International Conference on Robotics and Automation (ICRA)*, 2021, pp. 6541–6548.

[33] I. Armeni, Z.-Y. He, J. Gwak, A. R. Zamir, M. Fischer, J. Malik, and S. Savarese, "3d scene graph: A structure for unified semantics, 3d space, and camera," in *International Conference on Computer Vision (ICCV)*, October 2019.

[34] X. Chang, P. Ren, P. Xu, Z. Li, X. Chen, and A. Hauptmann, "A comprehensive survey of scene graphs: Generation and application," *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 45, no. 1, pp. 1–26, 2023.

[35] J. Yang, J. Lu, S. Lee, D. Batra, and D. Parikh, "Graph r-cnn for scene graph generation," in *European Conference on Computer Vision (ECCV)*, September 2018.

[36] C. Agia, K. M. Jatavallabhula, M. Khodeir, O. Miksik, V. Vineet, M. Mukadam, L. Paull, and F. Shkurti, "Taskography: Evaluating robot task planning over large 3d scene graphs," in *Conference on Robot Learning (CoRL)*. PMLR, 2022, pp. 46–58.

[37] C. Li, R. Zhang, J. Wong, C. Gokmen, S. Srivastava, R. Martín-Martín, C. Wang, G. Levine, M. Lingelbach, J. Sun, *et al.*, "Behavior-1k: A benchmark for embodied ai with 1,000 everyday activities and realistic simulation," in *Conference on Robot Learning (CoRL)*. PMLR, 2023, pp. 80–93.

[38] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," http://pybullet.org, 2016–2021.

[39] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," in *Annual Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019.

[40] Y. Ding, X. Zhang, C. Paxton, and S. Zhang, "Task and motion planning with large language models for object rearrangement," in *International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 2086–2092.