Structured Preference Optimization for Vision-Language Long-Horizon Task Planning

Anonymous ACL submission

Abstract

Existing methods for vision-language task planning excel in short-horizon tasks but often fall short in complex, long-horizon planning within dynamic environments. These challenges primarily arise from the difficulty of effectively training models to produce high-quality reasoning processes for long-horizon tasks. To address this, we propose **S**tructured **P**reference **O**ptimization (**SPO**), which aims to enhance reasoning and action selection in long-horizon task planning through structured preference evaluation and optimized training strategies. Specifically, SPO introduces: 1) Preference-Based Scoring and Optimization, which systematically evaluates reasoning chains based on task relevance, visual grounding, and historical consistency; and 2) Curriculum-Guided Training, where the model progressively adapts from simple to complex tasks, improving its generalization ability in long-horizon scenarios and enhancing reasoning robustness. To advance research in vision-language long-horizon task planning, we introduce ExtendaBench, a comprehensive benchmark covering 1,509 tasks across VirtualHome and Habitat 2.0, categorized into ultra-short, short, medium, and long tasks. Experimental results demonstrate that SPO significantly improves reasoning quality and final decision accuracy, outperforming prior methods on long-horizon tasks and underscoring the effectiveness of preference-driven optimization in vision-language task planning. Specifically, SPO achieves a +5.98% GCR and +4.68% SR improvement in VirtualHome and a +3.30% GCR and +2.11% SR improvement in Habitat over the best-performing baselines.

1 Introduction

In autonomous systems, there is a growing demand for robots capable of executing complex, real-world tasks in domestic environments. Tasks such as organizing a room, preparing a meal, and cleaning up afterward require not only a diverse set of actions but also sophisticated long-term planning capabilities. However, current approaches struggle with long-horizon tasks due to a lack of learning in long-term planning ability and the fact that most benchmarks (Puig et al., 2018; Liao et al., 2019; Shridhar et al., 2020a,b) focus on short-term discrete tasks. This gap hinders progress toward robots capable of handling the complex, multi-step tasks demanded by real-life scenarios.

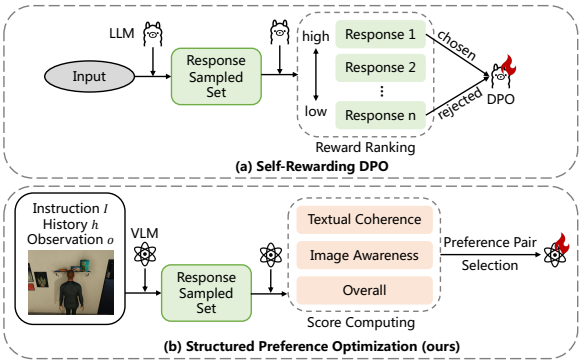


Figure 1: Comparison with existing methods. (a) Self-Rewarding DPO (Yuan et al., 2024) relies on a single reward criterion to rank sampled responses and selects both the highest-ranked (preferred) and lowest-ranked (rejected) responses for DPO training. (b) Structured Preference Optimization (ours) introduces a structured scoring framework with multiple criteria and an adaptive preference selection strategy, enabling more fine-grained and informed optimization.

Existing reasoning-based decision-making methods primarily rely on prompting strategies or environmental feedback to determine actions, often without explicitly modeling the quality of reasoning chains. While recent approaches (Yao et al., 2022; Zhao et al., 2024; Zhi-Xuan et al., 2024) leverage textual inputs for reasoning, they lack a structured mechanism to incorporate multimodal information or refine reasoning processes over extended horizons. Furthermore, prior optimization frameworks, such as Self-Rewarding DPO (Yuan et al., 2024), rely on a single reward criterion,

which may lead to suboptimal preference selection as in Figure 1.

To address these limitations, we propose Structured Preference Optimization (SPO), a novel framework designed to enhance reasoning quality and decision-making in long-horizon task planning through structured preference evaluation and progressive learning. SPO consists of two core components: 1) **Preference-Based Scoring and Optimization:** This component systematically evaluates reasoning chains based on three key criteria: task relevance, visual grounding, and historical consistency. Unlike prior approaches that rely on heuristic prompt engineering, SPO introduces a structured mechanism to construct preference pairs, enabling explicit optimization of reasoning quality. By prioritizing high-quality thought processes and systematically discarding suboptimal reasoning steps, this approach ensures more reliable and effective decision-making in long-horizon tasks. 2) **Curriculum-Guided Training:** The model undergoes progressive learning, starting with simple tasks and gradually advancing to more complex ones. By incrementally increasing task complexity during training, the model develops robust reasoning strategies that enhance its ability to generalize across diverse long-horizon tasks. This structured learning paradigm not only improves the model's adaptability but also strengthens its stability in multi-step planning, ensuring consistent performance in real-world scenarios.

Finally, to bridge the notable gap in the field regarding the absence of a benchmark tailored for long-horizon tasks, we propose ExtendaBench, a comprehensive benchmark that categorizes the task into four difficulty levels based on the number of steps required for completion, namely ultra-short, short, medium, and long. Leveraging the generative capabilities of GPT-4o (OpenAI, 2024), we create a diverse and extensive collection of tasks. These tasks undergo minimal human refinement to ensure high-quality data while significantly reducing the costs and effort associated with manual data labeling.

Our contributions can be summarized as follows:

- We introduce Structured Preference Optimization (SPO), a framework that enhances long-horizon reasoning through structured preference-based evaluation and curriculum-guided learning, enabling more effective decision-making.
- We propose ExtendaBench, a benchmark with four levels of difficulty and 1,509 tasks across VirtualHome and Habitat 2.0, providing a comprehensive evaluation suite for sustained reasoning in long-horizon task planning.
- We validate SPO through extensive experiments, demonstrating state-of-the-art performance in long-horizon task planning.

2 Related Work

2.1 Multimodal Large Language Models

The emergence of LLMs (Touvron et al., 2023; Chiang et al., 2023) has driven substantial progress in multimodal large language models (MLLMs), which aim to integrate both visual and textual modalities, advancing toward a more generalized form of intelligence. Early works such as BLIP-2 (Jian et al., 2024), MiniGPT-4 (Zhu et al., 2023), LLaVA (Liu et al., 2024), and OpenFlamingo (Awadalla et al., 2023) capitalized on pretrained vision encoders paired with LLMs, demonstrating strong performance in tasks like visual question answering and image captioning. mPLUG-Owl (Ye et al., 2023) introduces a modularized training framework to further refine cross-modal interactions. On the closed-source side, models such as GPT-4V (OpenAI, 2023) and Gemini (Team et al., 2023) pushes the boundaries of multimodal reasoning and interaction capabilities.

2.2 LLM Self-improvement

Self-improvement techniques for LLMs aim to enhance model capabilities by enabling them to learn from their own outputs. These methods often involve supervised fine-tuning (SFT) on high-quality responses generated by the models themselves (Li et al., 2023; Wang et al., 2024b) or preference optimization (Yuan et al., 2024; Rosset et al., 2024; Pang et al., 2024; Prasad et al., 2024; Zhang et al., 2024), where the model is trained to distinguish between better and worse responses. These approaches mostly employ LLM-as-a-Judge prompting (Zheng et al., 2024) or train strong reward models (Xu et al., 2023; Havrilla et al., 2024) to evaluate and filter generated data, thereby guiding the model toward improved performance.

2.3 Embodied Task Planning

Traditional robotics planning methods have relied on search algorithms in predefined domains (Fikes and Nilsson, 1971; Garrett et al., 2020; Jiang et al., 2018), but face scalability challenges in complex

environments with high branching factors (Puig et al., 2018; Shridhar et al., 2020a). Heuristics have helped alleviate these limitations, leading to advancements (Baier et al., 2009; Hoffmann, 2001; Helmert, 2006; Bryce and Kambhampati, 2007). More recently, learning-based methods like representation learning and hierarchical strategies have emerged, showing effectiveness in complex decision-making (Eysenbach et al., 2019; Xu et al., 2018, 2019; Srinivas et al., 2018; Kurutach et al., 2018; Nair and Finn, 2019; Jiang et al., 2019). The advent of LLMs has further revolutionized planning by enabling task decomposition and robust reasoning (Li et al., 2022; Huang et al., 2022b; Ahn et al., 2022; Valmeekam et al., 2022; Silver et al., 2022; Song et al., 2023; Rana et al., 2023; Driess et al., 2023; Liu et al., 2023b; Wu et al., 2023; Wake et al., 2023; Chen et al., 2023; Bhat et al., 2024; Zhi-Xuan et al., 2024). Other works focus on translating natural language into executable code and formal specifications (Vemprala et al., 2023; Liang et al., 2023; Silver et al., 2023; Xie et al., 2023; Skreta et al., 2023; Liu et al., 2023a; Zhang and Soh, 2023; Ding et al., 2023b,a; Zhao et al., 2024). Some approaches fine-tune LLMs for better performance (Driess et al., 2023; Qiu et al., 2023), while others opt for few-shot or zero-shot methods (Huang et al., 2022b,a; Singh et al., 2023) to avoid the resource demands of model training. In contrast, our method introduces multimodal preference optimization, fine-grained preference scoring, and curriculum-guided optimization.

3 Preliminaries

Direct Preference Optimization (DPO) (Rafailov et al., 2024) is a reinforcement learning-free approach that optimizes a model's policy using preference-labeled data. Instead of relying on an explicit reward model, DPO directly enforces preference ordering by encouraging the model to assign higher probabilities to preferred outputs over less preferred ones.

Given a dataset $D = \{(x, y^+, y^-)\}$, where y^+ is the preferred response and y^- is the less preferred response for input x, DPO optimizes the following contrastive ranking loss:

$$L_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\,\mathbb{E}_{(x,y^+,y^-)\sim D}\left[\log \sigma\left(\beta \log \left(\frac{\pi_\theta(y^+ \mid x)}{\pi_\theta(y^- \mid x)}\right)\right)\right], \quad (1)$$

where σ is the sigmoid function, and β is a scaling factor controlling preference sharpness.

4 Structured Preference Optimization

The Structured Preference Optimization (SPO) framework enhances long-horizon task planning by introducing a structured evaluation mechanism for reasoning quality and a progressive training strategy to improve model generalization. Unlike standard preference optimization, which lacks explicit reasoning quality assessment and task complexity adaptation, SPO systematically refines the model's reasoning capabilities through Preference-Based Scoring and Optimization and Curriculum-Guided Training. The overview of our framework is shown in Figure 2.

4.1 Preference-Based Scoring and Optimization

The structured preference-based optimization mechanism evaluates and ranks reasoning chains based on explicit criteria. Unlike standard preference optimization, which treats reasoning as a single scalar preference, SPO decomposes reasoning quality into multiple dimensions and optimizes the model's decision-making accordingly.

4.1.1 Structured Preference Evaluation

Instead of relying on external annotations, SPO adopts a self-evaluation approach, where the vision-language model (sLVLM) itself serves as the judge to assess reasoning quality. Given a generated reasoning chain R_i, the model evaluates it based on the task context, which includes: task instruction (I), current image observation (o), and history of executed actions (h). Using this structured input, the model assigns two separate scores to assess different aspects of reasoning quality:

- Textual Coherence (S_{text}): Evaluates the logical consistency of the reasoning chain, ensuring that each step is task-relevant and maintains historical consistency with prior steps. This prevents reasoning errors such as goal misalignment or contradictions in multi-step plans.
- Image Awareness (S_{image}): Measures whether the reasoning chain sufficiently incorporates relevant information from the visual observations, ensuring that decisions are grounded in the environment rather than relying solely on textual priors.

To obtain these scores, the model is prompted with an evaluation query p, where the model M

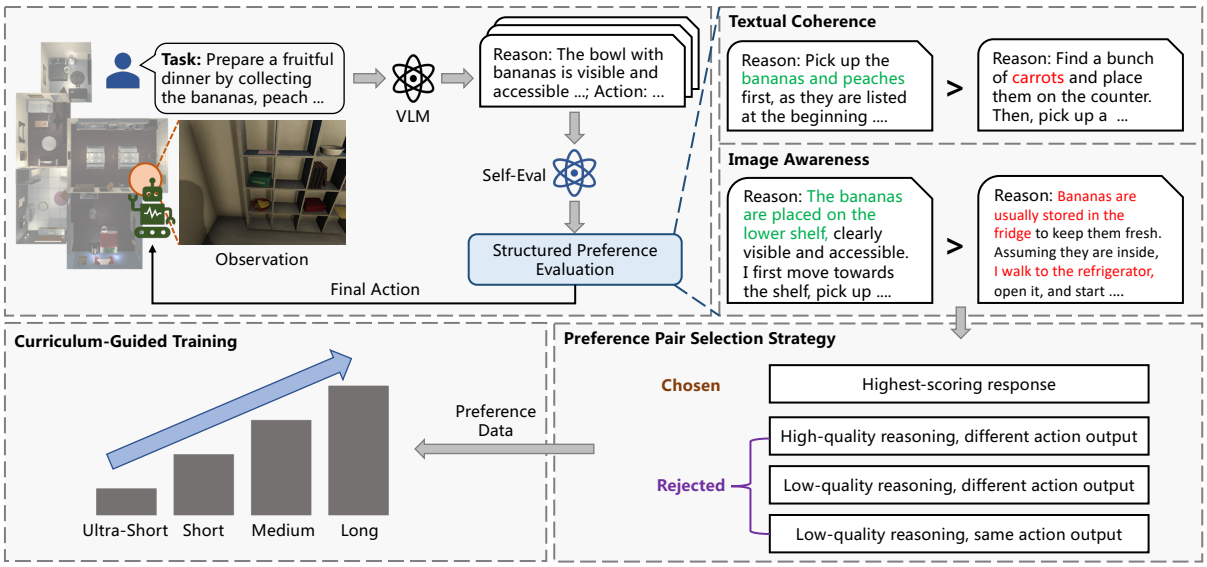


Figure 2: Overview of the Structured Preference Optimization.

estimates reasoning quality as follows:

$$S_{\text{text}} = M(p_{\text{text}}, R_i, I, h), \tag{2}$$

$$S_{\text{image}} = M(p_{\text{image}}, R_i, I, o, h), \tag{3}$$

where p_{text} and p_{image} are evaluation prompts designed to assess textual coherence and image awareness, respectively. The overall preference score can then be computed as either a weighted combination:

$$S(R_i) = w_1 \mathcal{S}_{\text{text}} + w_2 \mathcal{S}_{\text{image}}, \tag{4}$$

where w_1 and w_2 are weighting factors that control the relative contribution of textual coherence and image awareness. Alternatively, instead of using a predefined weighted sum, the model directly provides an overall preference score:

$$S(R_i) = M(p_{\text{overall}}, R_i, I, o, h), \tag{5}$$

where p_{overall} is an evaluation prompt requesting a single comprehensive score. Empirically, we found that using the model to generate the overall preference score yields better optimization results compared to manually setting weighting factors.

4.1.2 Preference Pair Selection Strategy

To refine the model's reasoning capabilities, SPO constructs structured preference pairs from model-generated samples, ensuring that the optimization process explicitly accounts for both reasoning quality and action selection. Unlike prior methods, which simply select the highest-scoring reasoning chain as the positive sample and the lowest-scoring reasoning chain as the negative sample, SPO introduces a targeted preference selection strategy that prevents the model from over-optimizing reasoning at the cost of decision accuracy.

Given a set of generated reasoning chains $\{R_i\}$ for the same task input (I, o, h), the model self-evaluates each reasoning chain using the scoring mechanism described in Structured Preference Evaluation. The positive sample R^+ is selected as the highest-scoring reasoning chain, and in cases where multiple chains achieve the same highest score, we choose the one where the final action appears most frequently across all generated samples. This ensures that the model prioritizes common and stable action choices, reducing the risk of selecting an outlier action due to randomness in generation. For the negative sample R^-, instead of always selecting the lowest-scoring reasoning chain, SPO considers different selection strategies to ensure both reasoning quality and action feasibility are optimized. The negative sample is chosen from one of the following categories:

- High-quality reasoning, different action output: A reasoning chain that is coherent but results in a different final action from R^+. This prevents the model from focusing solely on reasoning quality while ignoring the correctness of the final decision.
- Low-quality reasoning, different action output: A reasoning chain with poor reasoning that also leads to incorrect final action. This helps the model distinguish between poor reasoning leading to incorrect decisions and high-quality thought processes.
- Low-quality reasoning, same action output: A reasoning chain that is less coherent but pro-

duces the same final action as R^+. This prevents the model from blindly optimizing for reasoning quality without considering whether the final decision remains valid.

4.1.3 Preference Optimization

Once the structured preference pairs (R^+, R^-) are selected, SPO directly applies DPO to align the model's policy with the preferred reasoning chains. The optimization follows the original DPO contrastive ranking loss (referencing Eq. 1, adapted to our task setting with inputs (I, o, h)):

$$L_{\text{pref}} = -\mathbb{E}_{(I,o,h,R^+,R^-)\sim D}\left[\log \sigma\left(\beta \log\left(\frac{\pi_\theta(R^+ \mid I, o, h)}{\pi_\theta(R^- \mid I, o, h)}\right)\right)\right]. \quad (6)$$

4.2 Curriculum-Guided Training

To facilitate structured learning, SPO categorizes tasks into four levels: ultra-short, short, medium, and long-horizon tasks. Instead of training on all task types simultaneously, SPO follows a progressive training strategy to gradually expose the model to increasing task complexity while preventing catastrophic forgetting.

Training is divided into four stages, where the model starts with ultra-short tasks and progressively incorporates more complex tasks in each subsequent stage. At every stage, a certain amount of previously learned tasks is retained to reinforce fundamental reasoning skills and prevent the model from overfitting to newly introduced tasks. This approach ensures that earlier-learned reasoning strategies remain effective as the model learns to handle longer task horizons.

A key challenge in curriculum learning is stabilizing the transition between different difficulty levels without disrupting previously learned decision patterns. To address this, SPO maintains a dynamic balance between newly introduced tasks and previously learned ones. During each training phase, the model is exposed to a mixture of current-stage tasks and replayed tasks from earlier stages, ensuring that it can generalize across task difficulties while refining long-horizon reasoning capabilities.

5 ExtendaBench

The ExtendaBench task corpus is developed using two distinct approaches tailored to each simulator. For VirtualHome (Puig et al., 2018), we leverage GPT-4o's advanced generative capabilities to create diverse and complex tasks. In contrast, tasks for Habitat 2.0 (Szot et al., 2021) are systematically collected using pre-defined templates.

5.1 VirtualHome

The ExtendaBench task corpus is developed using tailored approaches for each simulator, both leveraging GPT-4o's advanced generative capabilities. For VirtualHome (Puig et al., 2018), we utilize GPT-4o to directly generate diverse and complex tasks, allowing for a wide range of scenarios. For Habitat 2.0 (Szot et al., 2021), GPT-4o is used to generate pre-defined templates as well as to create specific task instances from these templates, resulting in systematically varied tasks with extended action sequences that are suitable for long-horizon planning.

Task Proposal The initial phase begins within the confines of VirtualHome, a simulated environment, where a varied collection of objects sets the stage for a multitude of task scenarios. By employing GPT-4o as a task generator, we design tasks focusing on object manipulation, striving for a wide array of task varieties and complexities. This method ensures an exhaustive representation of scenarios that closely mimic real-world challenges. To facilitate the generator's task creation, we provide prompts that are carefully constructed to inspire a broad range of tasks.

Review In the subsequent phase, GPT-4o undertakes the generation of detailed action plans for the devised tasks, meticulously outlining the steps required for successful task execution. To ensure the feasibility and coherence of these tasks, we introduce an additional examiner of scrutiny, also powered by GPT-4o. This examiner evaluates each task and its associated action plan for clarity, necessity, and coherence of steps, as well as the relevance and practicality of the actions and items involved, ensuring they belong to the simulated environment VirtualHome. It also assesses each step for common sense applicability, providing constructive feedback for further refinement.

Refinement After undergoing expert scrutiny, the generator refines the tasks and their corresponding action plans. Subsequent simulation of these revised tasks and plans enables further improvements based on simulator feedback. Tasks that are successfully executed within the simulator receive preliminary approval. Nevertheless, to guar-

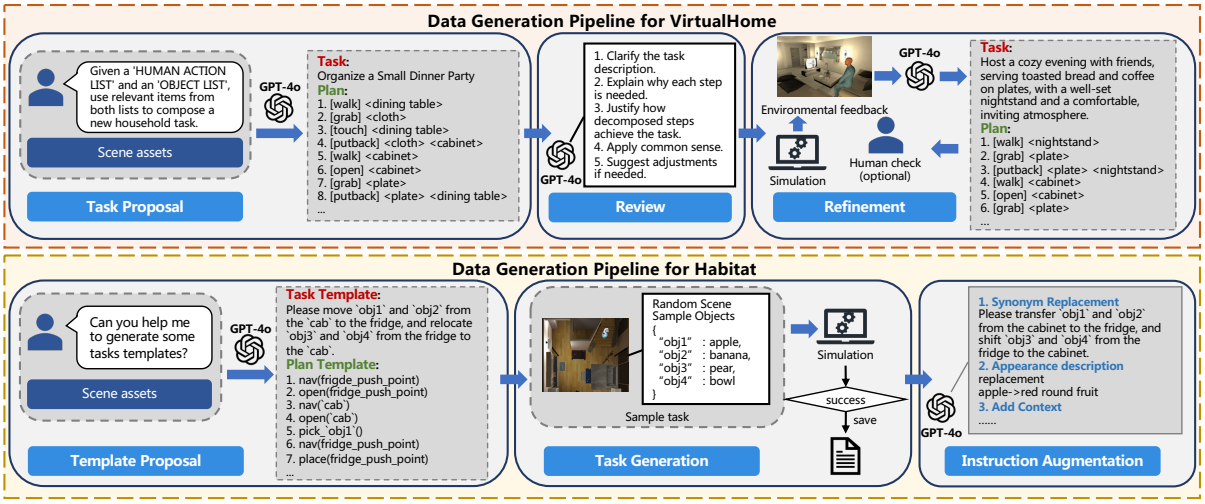


Figure 3: The process of generating tasks in ExtendaBench.

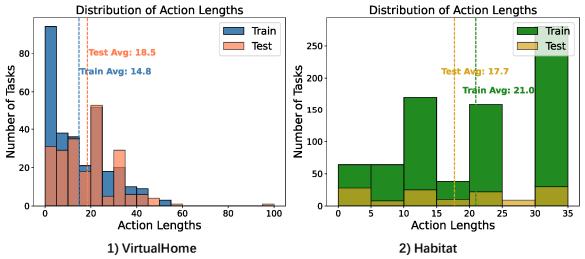


Figure 4: Distribution of action lengths in our benchmark.

antee optimal quality and applicability, we subject each task to a rigorous manual review, evaluating them for practicality and realism. Tasks that do not achieve success in the simulation are minimally modified by human according to the simulator's feedback, focusing on enhancing their realism and feasibility.

The multi-stage process, with minimal human intervention, is designed to ensure the reliability and quality of the tasks and their associated plans. The whole process of generating tasks in benchmark is shown in Figure 3.

5.2 Habitat 2.0

Inspired by Language Rearrangement (Szot et al., 2023), we propose a novel data generation pipeline for Habitat 2.0 that leverages GPT-4o to create diverse, long-horizon tasks, as illustrated in Figure 3. Our approach consists of three main stages:

Template Proposal GPT-4o generates initial task templates based on scene assets. These templates define general task structures (e.g., moving objects between locations) and serve as the basis for generating varied instructions.

Task Generation Using the task templates, we sample objects within random scenes to generate specific tasks with extended action sequences. This phase results in more complex task plans that evaluate an agent's capacity for long-term planning and adaptability.

Instruction Augmentation To increase task diversity, we apply various transformations to the instructions. These include synonym replacement, appearance description alterations (e.g., "apple" to "red round fruit"), and additional contextual details. This augmentation, powered by GPT-4o, allows us to expand the instruction set, testing the agent's understanding and flexibility in interpreting varied language inputs.

5.3 Dataset Statistics

The categorization within ExtendaBench is defined by the length of the action sequence required to accomplish a task, distributed as follows:

- Ultra-Short Tasks: Tasks that can be completed in fewer than 10 actions.
- Short Tasks: Tasks requiring 10 to 20 actions for completion.
- Medium Tasks: Tasks necessitating 20 to 30 actions to finish.
- Long Tasks: Tasks that demand more than 30 actions to complete.

Distribution of action lengths is shown in Figure 4. The VirtualHome set includes a total of 605 tasks, with 220 ultra-short tasks, 128 short tasks, 155 medium tasks, and 102 long tasks. Similarly, the Habitat 2.0 set comprises 904 tasks, distributed as 161 ultra-short tasks, 243 short tasks, 190 medium tasks, and 310 long tasks.

Table 1: Comparison with existing methods using Qwen2.5-VL 7B as the baseline on different sets of our ExtendaBench in VirtualHome.

	Ultra-Short		Short		Medium		Long		Average	
Method	GCR	SR	GCR	SR	GCR	SR	GCR	SR	GCR	SR
Baseline	57.32	35.00	42.72	9.62	30.57	3.33	27.47	0	39.52	11.99
CoT (Wei et al., 2022)	68.66	41.67	35.46	3.85	36.36	1.67	20.45	0	40.23	11.80
Self-Rewarding (Yuan et al., 2024)										
Iteration 1	62.13	35.00	42.89	7.69	36.38	3.33	22.55	0	40.99	11.51
Iteration 2	59.15	31.67	44.47	11.54	29.92	3.33	28.80	0	40.58	11.64
Iteration 3	58.93	35.00	48.16	9.62	32.56	3.33	27.46	0	41.78	11.99
Iterative RPO (Pang et al., 2024)										
Iteration 1	67.03	38.33	41.59	7.69	26.97	1.67	26.06	0	40.41	11.92
Iteration 2	72.31	43.33	40.06	1.92	31.66	3.33	22.33	0	41.73	12.15
Iteration 3	59.86	31.67	46.42	11.54	34.02	6.67	29.06	0	42.34	12.47
SPO (1 iteration)	71.53	48.33	48.96	13.46	38.92	3.33	31.41	2.17	47.71	16.83

Table 2: Results using Qwen2.5-VL 7B as the baseline on different sets of our ExtendaBench in Habitat.

	Ultra-Short		Short		Medium		Long		Average	
Method	GCR	SR	GCR	SR	GCR	SR	GCR	SR	GCR	SR
Baseline	41.67	33.33	14.39	8.57	3.17	0	2.48	0	15.43	10.48
CoT (Wei et al., 2022)	42.36	50.00	9.49	8.57	7.51	0	6.10	0	16.36	14.64
Self-Rewarding (Yuan et al., 2024)										
Iteration 1	43.06	36.11	13.33	8.57	3.32	0	2.48	0	15.55	11.17
Iteration 2	43.06	33.33	14.19	8.57	4.34	0	2.48	0	16.01	10.48
Iteration 3	44.44	36.11	13.59	8.57	3.63	0	2.48	0	16.03	11.17
Iterative RPO (Pang et al., 2024)										
Iteration 1	45.14	36.11	12.90	8.57	3.43	0	2.86	0	16.08	11.17
Iteration 2	33.33	38.89	12.17	11.43	11.35	0	7.62	0	16.12	12.58
Iteration 3	38.54	44.44	15.06	8.57	10.49	0	6.76	0	17.71	13.25
SPO (1 iteration)	52.08	50.00	14.78	11.43	10.51	0	6.67	0	21.01	15.36

6 Experiments

6.1 Experimental Setup

For the VirtualHome set, we designate 218 tasks as the test set, with the remaining tasks serving as the training set. The Habitat 2.0 set also includes 120 test tasks. As our approach is unsupervised, we do not utilize the training set data for model training.

Evaluation Metrics To assess system efficacy, we employ success rate (SR) and goal conditions recall (GCR) (Singh et al., 2023) as our primary metrics. SR measures the proportion of executions where all key goal conditions (changing from the beginning to the end during a demonstration) are satisfied. GCR calculates the discrepancy between the expected and achieved end state conditions, relative to the total number of specific goal conditions needed for a task. A perfect SR score of 1 corresponds to achieving a GCR of 1, indicating flawless task execution. Results of SR and GCR are both reported in %.

6.2 Comparison with Existing Methods

We compare SPO with existing long-horizon reasoning methods, including Chain-of-Thought (CoT) (Wei et al., 2022), as well as the multi-modal VLM-based versions of Self-Rewarding (Yuan et al., 2024) and Iterative RPO (Pang et al., 2024), using Qwen2.5-VL 7B as the baseline. The evaluations are conducted on ExtendaBench in VirtualHome (Table 1) and Habitat (Table 2), covering tasks of increasing complexity from Ultra-Short to Long.

Across both benchmarks, CoT significantly improves over the baseline in Habitat, achieving higher SR across all task levels, demonstrating that

Table 3: Ablation studies of different modules in VirtualHome.

			Ultra-Short		Short		Medium		Long		Average	
Textual	Image	Curriculum	GCR	SR	GCR	SR	GCR	SR	GCR	SR	GCR	SR
✗	✗	✗	67.25	36.67	34.50	1.92	34.58	3.33	23.09	0	39.86	10.48
✓	✗	✗	71.70	43.33	45.52	9.62	30.57	1.67	16.71	0	41.13	13.65
✗	✓	✗	69.71	40.00	42.11	1.92	25.98	3.33	26.16	0	40.99	11.31
✓	✓	✗	70.52	43.33	43.89	7.69	33.96	3.33	27.90	0	44.07	13.59
✓	✓	✓	71.53	48.33	48.96	13.46	38.92	3.33	31.41	2.17	47.71	16.83

Table 4: Average performance of preference pair selection strategy in VirtualHome.

pair selection	GCR	SR
✗	40.04	11.73
✓	41.13	13.65

explicit reasoning helps in shorter-horizon environments. However, its effectiveness diminishes on longer tasks, where structured multi-step planning becomes necessary. In VirtualHome, CoT provides a slight improvement over the baseline, but its SR drops significantly for more complex tasks.

Self-Rewarding and Iterative RPO introduce iterative refinement mechanisms, leading to gradual improvements in GCR and SR, particularly on short and medium tasks. However, their impact remains limited for long-horizon planning, with SR reaching 0% on long tasks in both VirtualHome and Habitat, indicating difficulties in maintaining coherent reasoning across extended steps.

In contrast, SPO achieves the best overall performance across both environments, consistently outperforming all baselines. In VirtualHome (Table 1), SPO achieves 47.71% GCR and 16.83% SR, surpassing Iterative RPO (42.34% GCR, 12.47% SR) and Self-Rewarding (41.78% GCR, 11.99% SR). Similarly, in Habitat (Table 2), SPO achieves 21.01% GCR and 15.36% SR, outperforming all other methods, including Iterative RPO (17.71% GCR, 13.25% SR) and Self-Rewarding (16.03% GCR, 11.17% SR). Notably, SPO surpasses CoT in overall performance, achieving superior results without requiring iterative refinement.

6.3 Ablation Study

To evaluate the contributions of different components in SPO, we conduct an ablation study on VirtualHome, selectively removing textual coherence scoring, image awareness scoring, and curriculum-guided training. The results in Table 3 show that removing textual coherence scoring leads to the most significant performance drop, especially on short, medium, and long tasks, indicating its critical role in maintaining reasoning consistency. Removing image awareness scoring also results in a decline, particularly on long tasks, where integrating visual observations becomes more important. Without curriculum learning, performance on medium and long tasks deteriorates, demonstrating that progressive training helps the model handle more complex task sequences. The full SPO model achieves the highest performance, with 47.71% GCR and 16.83% SR, confirming that structured preference learning and curriculum-guided training together enable more effective long-horizon task planning.

To evaluate the impact of preference pair selection, we conduct an ablation study on the Textual Coherence model (row 2 in Table 3). WApplying pair selection on the Textual Coherence model improves GCR by +1.09% and SR by +1.92%. This demonstrates that structured preference selection enhances decision accuracy.

7 Conclusion

We introduce Structured Preference Optimization (SPO), a method for improving long-horizon vision-language task planning through structured preference learning and curriculum-guided training. Unlike existing methods that struggle with multi-step decision-making, SPO systematically evaluates reasoning chains based on textual coherence and image awareness, ensuring high-quality reasoning and action selection. Additionally, curriculum-guided training progressively adapts the model from simpler to more complex tasks, enhancing generalization and robustness in long-horizon scenarios. To support research in this area, ExtendaBench provides a benchmark spanning VirtualHome and Habitat simulators with tasks of increasing difficulty. Experimental results show that SPO outperforms prior methods, particularly in long-horizon task planning, demonstrating improved reasoning consistency and decision-making accuracy.

References

Pravesh Agrawal, Szymon Antoniak, Emma Bou Hanna, Devendra Chaplot, Jessica Chudnovsky, Saurabh Garg, Theophile Gervet, Soham Ghosh, Amélie Héliou, Paul Jacob, et al. 2024. Pixtral 12b. *arXiv preprint arXiv:2410.07073*.

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

Anas Awadalla, Irena Gao, Josh Gardner, Jack Hessel, Yusuf Hanafy, Wanrong Zhu, Kalyani Marathe, Yonatan Bitton, Samir Gadre, Shiori Sagawa, et al. 2023. Openflamingo: An open-source framework for training large autoregressive vision-language models. *arXiv preprint arXiv:2308.01390*.

Jorge A Baier, Fahiem Bacchus, and Sheila A McIlraith. 2009. A heuristic search approach to planning with temporally extended preferences. *Artificial Intelligence*, 173(5-6):593–618.

Vineet Bhat, Ali Umut Kaypak, Prashanth Krishnamurthy, Ramesh Karri, and Farshad Khorrami. 2024. Grounding llms for robot task planning using closed-loop state feedback. *arXiv preprint arXiv:2402.08546*.

Daniel Bryce and Subbarao Kambhampati. 2007. A tutorial on planning graph based reachability heuristics. *AI Magazine*, 28(1):47–47.

Siwei Chen, Anxing Xiao, and David Hsu. 2023. Llm-state: Expandable state representation for long-horizon task planning in the open world. *arXiv preprint arXiv:2311.17406*.

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. 2024a. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*.

Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, et al. 2024b. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *arXiv preprint arXiv:2404.16821*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Yan Ding, Xiaohan Zhang, Saeid Amiri, Nieqing Cao, Hao Yang, Andy Kaminski, Chad Esselink, and Shiqi Zhang. 2023a. Integrating action knowledge and llms for task planning and situation handling in open worlds. *arXiv preprint arXiv:2305.17590*.

Yan Ding, Xiaohan Zhang, Chris Paxton, and Shiqi Zhang. 2023b. Task and motion planning with large language models for object rearrangement. *arXiv preprint arXiv:2303.06247*.

Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. 2023. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Ben Eysenbach, Russ R Salakhutdinov, and Sergey Levine. 2019. Search on the replay buffer: Bridging planning and reinforcement learning. *Advances in Neural Information Processing Systems*, 32.

Richard E Fikes and Nils J Nilsson. 1971. Strips: A new approach to the application of theorem proving to problem solving. *Artificial intelligence*, 2(3-4):189–208.

Caelan Reed Garrett, Tomás Lozano-Pérez, and Leslie Pack Kaelbling. 2020. Pddlstream: Integrating symbolic planners and blackbox samplers via optimistic adaptive planning. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 30, pages 440–448.

Alex Havrilla, Sharath Raparthy, Christoforus Nalmpantis, Jane Dwivedi-Yu, Maksym Zhuravinskyi, Eric Hambro, and Roberta Raileanu. 2024. Glore: When, where, and how to improve llm reasoning via global and local refinements. *arXiv preprint arXiv:2402.10963*.

Malte Helmert. 2006. The fast downward planning system. *Journal of Artificial Intelligence Research*, 26:191–246.

Jörg Hoffmann. 2001. Ff: The fast-forward planning system. *AI magazine*, 22(3):57–57.

Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. 2022a. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International Conference on Machine Learning*, pages 9118–9147. PMLR.

Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. 2022b. Inner monologue: Embodied reasoning through planning with language models. *arXiv preprint arXiv:2207.05608*.

Yiren Jian, Chongyang Gao, and Soroush Vosoughi. 2024. Bootstrapping vision-language learning with decoupled language pre-training. *Advances in Neural Information Processing Systems*, 36.

Yiding Jiang, Shixiang Shane Gu, Kevin P Murphy, and Chelsea Finn. 2019. Language as an abstraction for hierarchical deep reinforcement learning. *Advances in Neural Information Processing Systems*, 32.

Yuqian Jiang, Shiqi Zhang, Piyush Khandelwal, and Peter Stone. 2018. Task planning in robotics: an empirical comparison of pddl-based and asp-based systems. *arXiv preprint arXiv:1804.08229*.

Thanard Kurutach, Aviv Tamar, Ge Yang, Stuart J Russell, and Pieter Abbeel. 2018. Learning plannable representations with causal infogan. *Advances in Neural Information Processing Systems*, 31.

Shuang Li, Xavier Puig, Chris Paxton, Yilun Du, Clinton Wang, Linxi Fan, Tao Chen, De-An Huang, Ekin Akyürek, Anima Anandkumar, et al. 2022. Pretrained language models for interactive decision-making. *Advances in Neural Information Processing Systems*, 35:31199–31212.

Xian Li, Ping Yu, Chunting Zhou, Timo Schick, Omer Levy, Luke Zettlemoyer, Jason Weston, and Mike Lewis. 2023. Self-alignment with instruction backtranslation. *arXiv preprint arXiv:2308.06259*.

Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. 2023. Code as policies: Language model programs for embodied control. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9493–9500. IEEE.

Yuan-Hong Liao, Xavier Puig, Marko Boben, Antonio Torralba, and Sanja Fidler. 2019. Synthesizing environment-aware activities via activity sketches. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6291–6299.

Bo Liu, Yuqian Jiang, Xiaohan Zhang, Qiang Liu, Shiqi Zhang, Joydeep Biswas, and Peter Stone. 2023a. Llm+ p: Empowering large language models with optimal planning proficiency. *arXiv preprint arXiv:2304.11477*.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2024. Visual instruction tuning. *Advances in neural information processing systems*, 36.

Zeyi Liu, Arpit Bahety, and Shuran Song. 2023b. Reflect: Summarizing robot experiences for failure explanation and correction. *arXiv preprint arXiv:2306.15724*.

Suraj Nair and Chelsea Finn. 2019. Hierarchical foresight: Self-supervised learning of long-horizon tasks via visual subgoal generation. *arXiv preprint arXiv:1909.05829*.

OpenAI. 2023. Gpt-4 technical report. *ArXiv*, abs/2303.08774.

OpenAI. 2024. Hello gpt-4o. Accessed: 2024-05-26.

Richard Yuanzhe Pang, Weizhe Yuan, Kyunghyun Cho, He He, Sainbayar Sukhbaatar, and Jason Weston. 2024. Iterative reasoning preference optimization. *arXiv preprint arXiv:2404.19733*.

Archiki Prasad, Weizhe Yuan, Richard Yuanzhe Pang, Jing Xu, Maryam Fazel-Zarandi, Mohit Bansal, Sainbayar Sukhbaatar, Jason Weston, and Jane Yu. 2024. Self-consistency preference optimization. *arXiv preprint arXiv:2411.04109*.

Xavier Puig, Kevin Ra, Marko Boben, Jiaman Li, Tingwu Wang, Sanja Fidler, and Antonio Torralba. 2018. Virtualhome: Simulating household activities via programs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8494–8502.

Jielin Qiu, Mengdi Xu, William Han, Seungwhan Moon, and Ding Zhao. 2023. Embodied executable policy learning with language-based scene summarization. *arXiv preprint arXiv:2306.05696*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Krishan Rana, Jesse Haviland, Sourav Garg, Jad Abou-Chakra, Ian Reid, and Niko Suenderhauf. 2023. Sayplan: Grounding large language models using 3d scene graphs for scalable task planning. *arXiv preprint arXiv:2307.06135*.

Corby Rosset, Ching-An Cheng, Arindam Mitra, Michael Santacroce, Ahmed Awadallah, and Tengyang Xie. 2024. Direct nash optimization: Teaching language models to self-improve with general preferences. *arXiv preprint arXiv:2404.03715*.

Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. 2020a. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10740–10749.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. 2020b. Alfworld: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768*.

Tom Silver, Soham Dan, Kavitha Srinivas, Joshua B Tenenbaum, Leslie Pack Kaelbling, and Michael Katz. 2023. Generalized planning in pddl domains with pretrained large language models. *arXiv preprint arXiv:2305.11014*.

Tom Silver, Varun Hariprasad, Reece S Shuttleworth, Nishanth Kumar, Tomás Lozano-Pérez, and Leslie Pack Kaelbling. 2022. Pddl planning with pretrained large language models. In *NeurIPS 2022 Foundation Models for Decision Making Workshop*.

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

Marta Skreta, Naruki Yoshikawa, Sebastian Arellano-Rubach, Zhi Ji, Lasse Bjørn Kristensen, Kourosh Darvish, Alán Aspuru-Guzik, Florian Shkurti, and Animesh Garg. 2023. Errors are useful prompts: Instruction guided task programming with verifier-assisted iterative prompting. *arXiv preprint arXiv:2303.14100*.

Chan Hee Song, Jiaman Wu, Clayton Washington, Brian M Sadler, Wei-Lun Chao, and Yu Su. 2023. Llm-planner: Few-shot grounded planning for embodied agents with large language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2998–3009.

Aravind Srinivas, Allan Jabri, Pieter Abbeel, Sergey Levine, and Chelsea Finn. 2018. Universal planning networks: Learning generalizable representations for visuomotor control. In *International Conference on Machine Learning*, pages 4732–4741. PMLR.

Andrew Szot, Alexander Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Singh Chaplot, Oleksandr Maksymets, et al. 2021. Habitat 2.0: Training home assistants to rearrange their habitat. *Advances in neural information processing systems*, 34:251–266.

Andrew Szot, Max Schwarzer, Harsh Agrawal, Bogdan Mazoure, Rin Metcalf, Walter Talbott, Natalie Mackraz, R Devon Hjelm, and Alexander T Toshev. 2023. Large language models as generalizable policies for embodied tasks. In *The Twelfth International Conference on Learning Representations*.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Qwen Team. 2025. Qwen2.5-vl.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Karthik Valmeekam, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. 2022. Large language models still can't plan (a benchmark for llms on planning and reasoning about change). *arXiv preprint arXiv:2206.10498*.

Sai Vemprala, Rogerio Bonatti, Arthur Bucker, and Ashish Kapoor. 2023. Chatgpt for robotics: Design principles and model abilities. *Microsoft Auton. Syst. Robot. Res*, 2:20.

Naoki Wake, Atsushi Kanehira, Kazuhiro Sasabuchi, Jun Takamatsu, and Katsushi Ikeuchi. 2023. Chatgpt empowered long-step robot control in various environments: A case application. *arXiv preprint arXiv:2304.03893*.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. 2024a. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Tianlu Wang, Ilia Kulikov, Olga Golovneva, Ping Yu, Weizhe Yuan, Jane Dwivedi-Yu, Richard Yuanzhe Pang, Maryam Fazel-Zarandi, Jason Weston, and Xian Li. 2024b. Self-taught evaluators. *arXiv preprint arXiv:2408.02666*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Jimmy Wu, Rika Antonova, Adam Kan, Marion Lepert, Andy Zeng, Shuran Song, Jeannette Bohg, Szymon Rusinkiewicz, and Thomas Funkhouser. 2023. Tidybot: Personalized robot assistance with large language models. *arXiv preprint arXiv:2305.05658*.

Yaqi Xie, Chen Yu, Tongyao Zhu, Jinbin Bai, Ze Gong, and Harold Soh. 2023. Translating natural language to planning goals with large-language models. *arXiv preprint arXiv:2302.05128*.

Danfei Xu, Roberto Martín-Martín, De-An Huang, Yuke Zhu, Silvio Savarese, and Li F Fei-Fei. 2019. Regression planning networks. *Advances in Neural Information Processing Systems*, 32.

Danfei Xu, Suraj Nair, Yuke Zhu, Julian Gao, Animesh Garg, Li Fei-Fei, and Silvio Savarese. 2018. Neural task programming: Learning to generalize across hierarchical tasks. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3795–3802. IEEE.

Jing Xu, Andrew Lee, Sainbayar Sukhbaatar, and Jason Weston. 2023. Some things are more cringe than others: Preference optimization with the pairwise cringe loss. *arXiv preprint arXiv:2312.16682*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, et al. 2023. mplug-owl: Modularization empowers large language models with multimodality. *arXiv preprint arXiv:2304.14178*.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. 2024. Self-rewarding language models. *arXiv preprint arXiv:2401.10020*.

Bowen Zhang and Harold Soh. 2023. Large language models as zero-shot human models for human-robot interaction. *arXiv preprint arXiv:2303.03548*.

Xuan Zhang, Chao Du, Tianyu Pang, Qian Liu, Wei Gao, and Min Lin. 2024. Chain of preference optimization: Improving chain-of-thought reasoning in llms. *arXiv preprint arXiv:2406.09136*.

Zirui Zhao, Wee Sun Lee, and David Hsu. 2024. Large language models as commonsense knowledge for large-scale task planning. *Advances in Neural Information Processing Systems*, 36.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2024. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36.

Tan Zhi-Xuan, Lance Ying, Vikash Mansinghka, and Joshua B Tenenbaum. 2024. Pragmatic instruction following and goal assistance via cooperative language-guided inverse planning. *arXiv preprint arXiv:2402.17930*.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*.

A More Details for ExtendaBench

A.1 Statistics

A.1.1 Overview

Table 5 provides a summary of key characteristics of the VirtualHome and Habitat datasets in our ExtendaBench, highlighting differences in scene complexity, task variety, and action requirements. The VirtualHome dataset consists of 7 distinct scenes with a total of 390 objects, supporting 294 task types across 605 instructions. In VirtualHome, the simulator provides 16 unique executable actions, enabling a broader range of task interactions. In contrast, the Habitat dataset features 105 scenes with 82 distinct objects, enabling 20 task types across 904 instructions. The Habitat simulator supports 6 unique executable actions.

Table 5: Overview of scene and task characteristics in VirtualHome and Habitat.

	VirtualHome	Habitat
Scene Number	7	105
Scene Objects	390	82
Task Type	294	20
Instructions	605	904
Action Number	16	6

A.1.2 Data Distribution Across Sets

VirtualHome For the VirtualHome dataset, tasks are categorized into ultra short, short, medium, and long. Each category includes a portion reserved for testing, with the remaining used for training. The distribution is as follows:

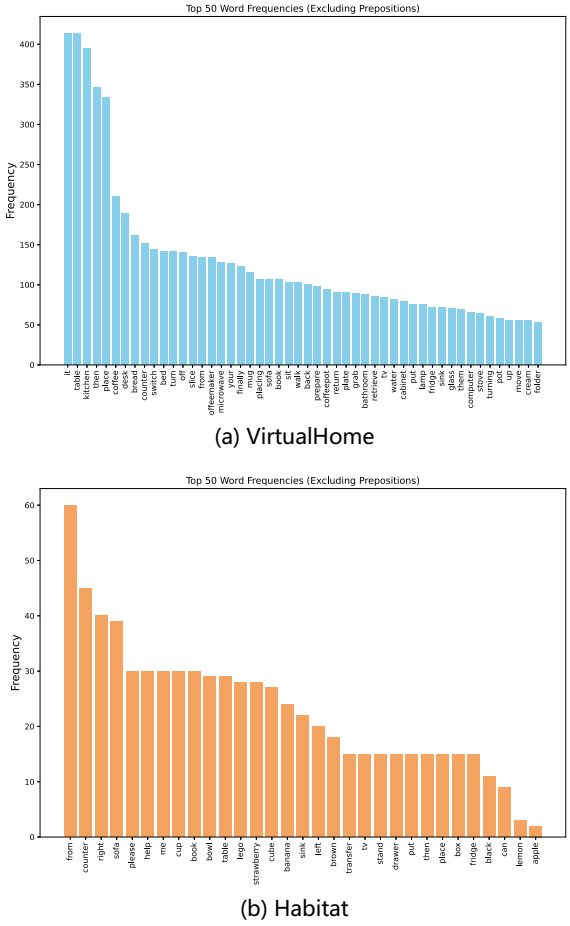
- Ultra short: This category contains 220 tasks in total, with 46 allocated for testing and 174 for training.
- Short: A total of 128 tasks, with 60 reserved for testing and 68 for training.
- Medium: Comprising 155 tasks, with 52 for testing and 103 for training.
- Long: The most complex category, including 102 tasks in total, with 60 allocated for testing and 42 for training.

Habitat For Habitat, the dataset is similarly divided into four categories based on task length: ultra short, short, medium, and long. For each category, a portion of the tasks is allocated for testing, and the remaining are used for training. The details are as follows:

- Ultra short: This category contains 161 tasks, with 36 reserved for testing and 125 for training.
- Short: There are 243 tasks, of which 35 are for testing and 208 for training.
- Medium: A total of 190 tasks, including 31 for testing and 159 for training.
- Long: The largest category, comprising 310 tasks, with 30 allocated for testing and 280 for training.

A.1.3 Word Frequency Distribution

Figure 5 presents the top 50 most frequent words, excluding prepositions, in the datasets generated for VirtualHome and Habitat environments. Subfigure (a) shows the word frequencies from VirtualHome, highlighting terms associated with common objects and actions, such as "table," "kitchen," and "place," reflecting its simulation of domestic scenarios. Subfigure (b) illustrates the word frequencies for Habitat, where terms like "from," "counter," and "cup" dominate, indicating tasks involving object interaction and spatial relationships.



(a) VirtualHome

(b) Habitat

Figure 5: Word frequency analysis for ExtendaBench.

A.2 Visualization

To better understand the structure and diversity of the tasks generated for VirtualHome and Habitat, we provide visualizations of selected examples in Figures 6, 7, 8 and 9. These examples illustrate the capability of our benchmark to handle long-horizon tasks, requiring multiple sequential steps to complete a given instruction. In Figure 6, we showcase an example task generated in the VirtualHome environment, where the agent is tasked with collecting and arranging multiple objects to prepare a dinner scene. Additionally, Figure 7 provides another example in VirtualHome, demonstrating a more complex cooking and meal arrangement task involving multiple appliances and detailed object placements. Figure 8 illustrates an example in Habitat 2.0, where the agent transfers various objects across different locations, including countertops, drawers, and sofas. Furthermore, Figure 9 shows another Habitat example, featuring a detailed multi-object transfer task requiring precise placement across multiple furniture items. The visualizations emphasize the diversity and complexity of the generated tasks.

B More Details for Experiments

B.1 Experimental Setup

For data generation, we produce $K = 5$ responses per prompt, employing a sampling temperature of 0.7 and a top-p value of 0.95. The generated dataset is then used to train the model for 3 epochs. During training, the learning rate is set to $2\mathrm{e}{-5}$. For LoRA, we use a rank value of 16, an alpha parameter of 32, and a dropout rate of 0.05. Unlike the Self-Rewarding framework, which involves iterative training where the trained model is used to re-label data and retrain in a loop, our approach trains the model only once, simplifying the training process while maintaining effectiveness. All experiments are conducted on 1 L40 GPU.

B.2 Comparison of Vision-Language Models

To assess the capabilities of various small-scale vision-language models (sLVLMs) on long-horizon task planning, we evaluated six models on our ExtendaBench benchmark in VirtualHome, spanning ultra-short to long tasks. Table 6 presents the comparative performance in terms of GCR and SR across different task horizons. The results indicate that while all models perform well on ultra-short tasks, performance drops sharply as task complexity increases, with SR reaching 0% on long tasks for most models. Among them, Qwen2.5-VL 7B achieves the highest average GCR and SR, demonstrating the best overall performance in long-horizon task planning.

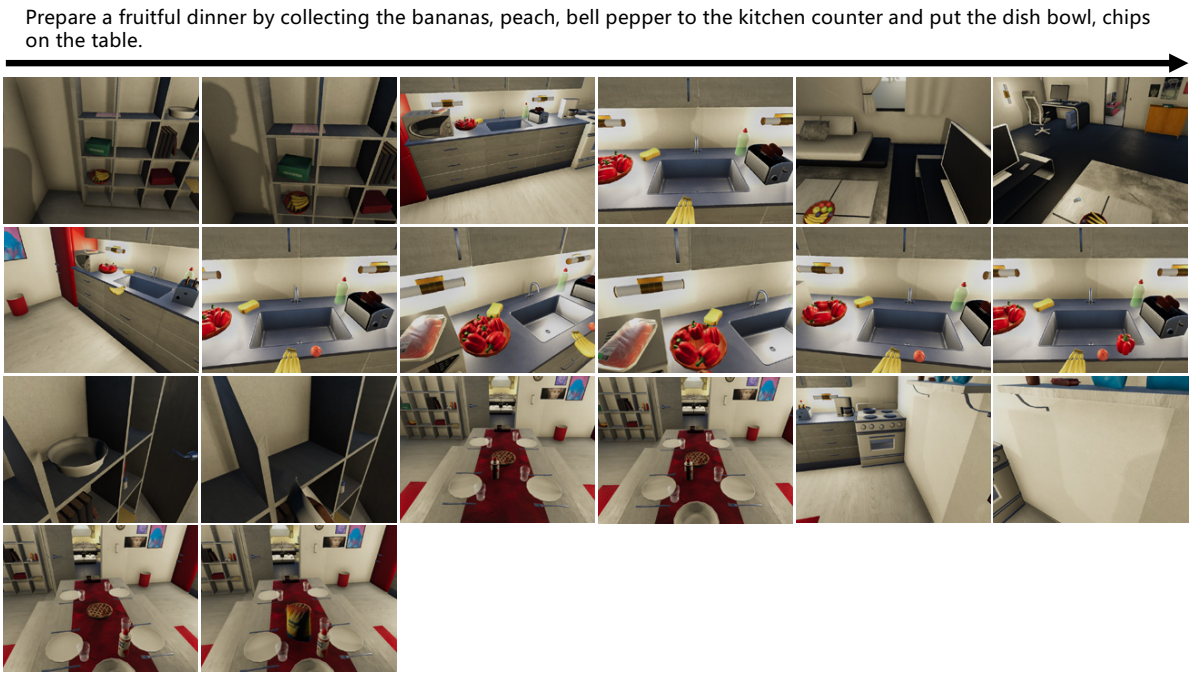


Figure 6: Generated task example in VirtualHome.

Table 6: Comparison of various small vision-language models on different sets of our ExtendaBench in VirtualHome.

	Ultra-Short		Short		Medium		Long		Average	
	GCR	SR	GCR	SR	GCR	SR	GCR	SR	GCR	SR
InternVL2 8B (Chen et al., 2024b)	39.56	11.67	20.31	0	13.88	0	16.56	0	22.58	2.92
Pixtral 12B (Agrawal et al., 2024)	53.62	28.33	28.25	0	19.34	0	19.00	0	30.05	7.08
Qwen2-VL 7B (Wang et al., 2024a)	57.54	28.33	39.27	5.77	28.26	0	26.50	0	37.89	8.53
Llama-3.2 11B (Dubey et al., 2024)	51.69	25.00	29.70	3.85	28.77	0	18.48	0	32.16	7.21
InternVL2.5 8B (Chen et al., 2024a)	55.71	28.33	30.03	0	16.61	0	21.55	0	30.98	7.08
Qwen2.5-VL 7B (Team, 2025)	57.32	35.00	42.72	9.62	30.57	3.33	27.47	0	39.52	11.99

C Prompts

C.1 Prompts for Generating Data

C.1.1 VirtualHome

Task Proposal

Follow these steps to generate your answer:
1. Think about the task generation:
- Design a task with more than 30 sequential steps.
- Use only actions from the "HUMAN ACTION LIST" and objects from the "OBJECT LIST."
- Ensure the task involves at least 12 distinct objects from the "OBJECT LIST."
2. Provide a detailed task description:
- Output a comprehensive description of the task.
- Include all subtasks and the required objects.
3. Decompose the task step by step:
- Break the task into individual steps.
- After completing each step, analyze and output what needs to be done next.
- Include reasoning for each subsequent step before outputting it.
Important rules:
- You have only two hands. Each time you grab an object, one hand becomes unavailable until you put the object back.
- Track the number of free hands after each action. Ensure you have at least one free hand before interacting with any object.
- Use only actions from the "HUMAN ACTION LIST" and objects from the "OBJECT LIST."

Figure 7: Generated task example in VirtualHome.

- The task must maintain a strong sequential relationship between its decomposed steps, ensuring logical and coherent progression.

Review

Follow these steps to verify the given task and decomposed steps step by step.

- Think about whether the task description is detailed enough to make it clear to a household agent what needs to be done, including every objects in decomposed steps. Give your reasons for this as well as your answer, if the answer is no, give a more detailed description of the task.
- Think and output the reasons why each step is necessary to complete the task.
- Think and output that each step is coherent with a necessary back-and-forth relationship between them.
- Think and output the reasons why the decomposed steps accomplish the task.
- verify the actions in decomposed steps only come from "HUMAN ACTION LIST."
- verify the objects in decomposed steps only come from "OBJECT LIST." The inclusion of any additional objects or locations is strictly prohibited.

Figure 8: Generated task example in Habitat.

- Think and output the reasons why each step make common sense.
- verify that each step is compliant with the rule of [walk] object before interacting with it.

If the verification passes, return true, otherwise return false and then give your adjustment.

Refinement

This is the feedback and observation based on your steps that have been executed:
[feedback]
<image>

Please perform the following steps based on the feedback:

1. Please think about and output the reason why the steps failed to execute.
2. Based on the reasons why the steps failed, think about and output the reasons why this task is feasible given the rules, and output yes or no.
3. if the task is feasible, output your modifications to the failed step.

C.1.2 Habitat

Template Proposal

You are a robot task generator that can generate robot task templates of different lengths based on given robot actions and examples.

The actions you can use include:

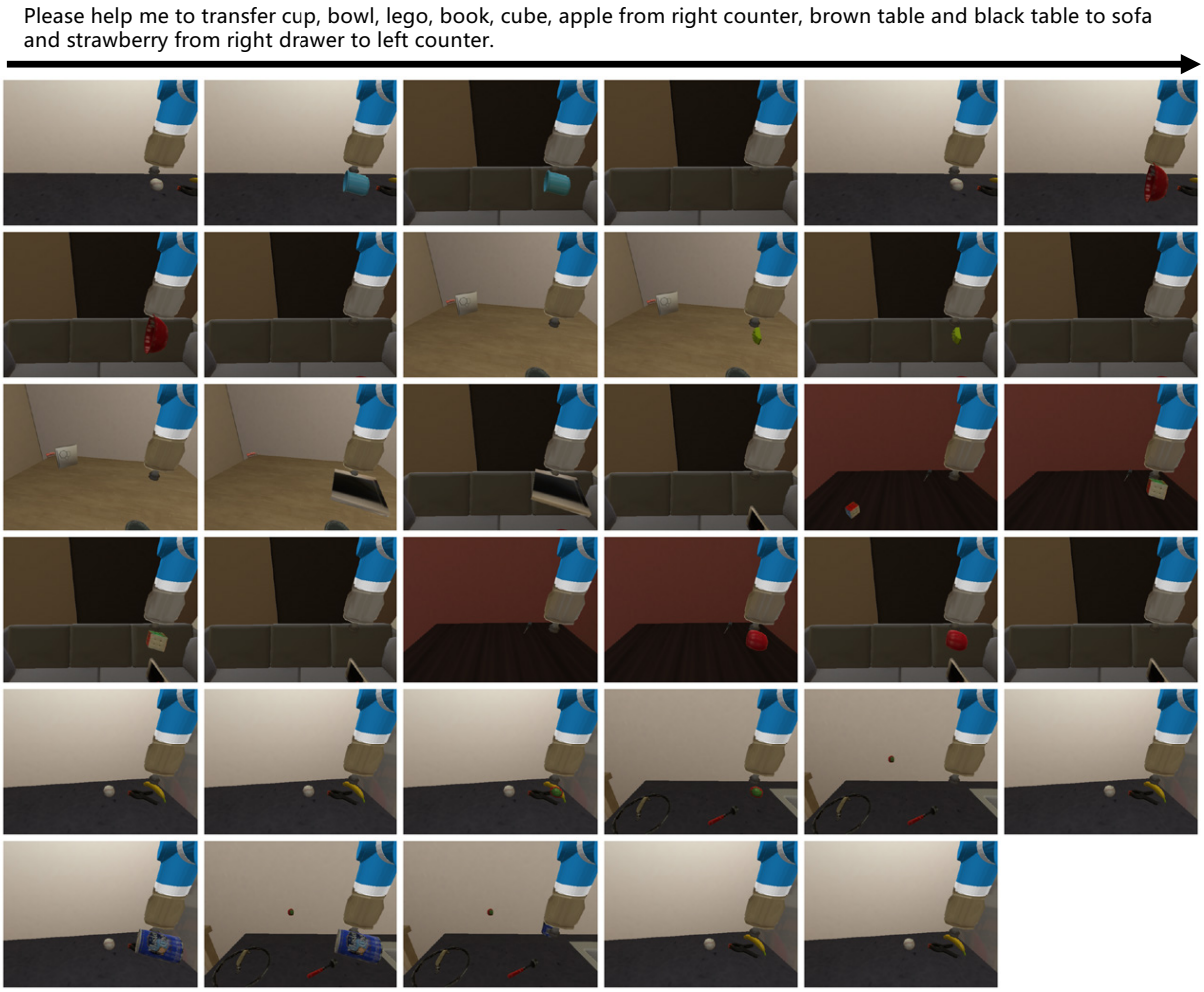


Figure 9: Generated task example in Habitat.

1.nav(obj or receptacle) is used by the robot to navigate to the corresponding object or receptacle
2.pick(obj) is used by the robot to grab an object
...

Rules:
1. You need to output five parts, including instructions, task planning, replaceable objects and target states.
2. If the object or receptacle in the instructions and task planning can be replaced, use plus pronouns to replace it.

Instruction Augmentation

You are a task instruction rewriter, and you can rewrite and expand the robot's task instructions according to the given rewriting rules.

Rules:
1. You can use the verbs of the task instructions Use synonyms to replace, for example, change move to reposition.
2. You can replace the objects used in the task instructions, replace the objects with corresponding colors or appearance descriptions, such as changing apple to a red round Fruit.
3. Add some context descriptions, for example, in "Please put an apple on the table for me," change it to "I want to eat an

apple, please put an apple on the table for me" to make the instruction longer.

Now Please help me rewrite the following instructions:

D Limitation

While the our SPO demonstrates significant improvements in long-horizon task planning, our work currently focuses on smaller versions of large vision-language models. Although this enables more efficient experimentation, it may limit the generalizability of our findings to larger models. Future research could extend SPO to larger-scale models to explore its full potential.

E License

The dataset is published under CC BY-NCSA 4.0 license, which means everyone can use this dataset for non-commercial research purposes.