# RoboHorizon: An LLM-Assisted Multi-View World Model for Long-Horizon Robotic Manipulation

# RoboHorizon: 一种基于大型语言模型辅助的多视角世界模型，用于长时域机器人操作

Zixuan Chen [1] , Jing Huo [1] , Yangtao Chen [1] and Yang Gao [1]

陈子轩 [1]， 霍靖 [1]， 陈阳涛 [1] 和高扬 [1]

[1] State Key Laboratory for Novel Software Technology, Nanjing University, China {chenzx, huojing} @nju.edu.cn, 502023330009@smail.nju.edu.cn, gaoy@nju.edu.cn

[1] 新型软件技术国家重点实验室，南京大学，中国 {chenzx, huojing}@nju.edu.cn, 502023330009@smail.nju.edu.cn， gaoy@nju.edu.cn

## Abstract

## 摘要

Efficient control in long-horizon robotic manipulation is challenging due to complex representation and policy learning requirements. Model-based visual reinforcement learning (RL) has shown great potential in addressing these challenges but still faces notable limitations, particularly in handling sparse rewards and complex visual features in long-horizon environments. To address these limitations, we propose the Recognize-Sense-Plan-Act (RSPA) pipeline for long-horizon tasks and further introduce RoboHorizon, an LLM-assisted multi-view world model tailored for long-horizon robotic manipulation. In RoboHorizon, pre-trained LLMs generate dense reward structures for multi-stage sub-tasks based on task language instructions, enabling robots to better recognize long-horizon tasks. Keyframe discovery is then integrated into the multi-view masked autoencoder (MAE) architecture to enhance the robot's ability to sense critical task sequences, strengthening its multi-stage perception of long-horizon processes. Leveraging these dense rewards and multi-view representations, a robotic world model is constructed to efficiently plan long-horizon tasks, enabling the robot to reliably act through RL algorithms. Experiments on two representative benchmarks, RLBench and FurnitureBench, show that RoboHorizon outperforms state-of-the-art visual model-based RL methods, achieving a 23.35% improvement in task success rates on RLBench's 4 short-horizon tasks and a 29.23% improvement on 6 long-horizon tasks from RLBench and 3 furniture assembly tasks from FurnitureBench.

由于复杂的表示和策略学习需求，长时域机器人操作中的高效控制具有挑战性。基于模型的视觉强化学习 (RL) 在解决这些挑战方面展现出巨大潜力，但仍面临显著限制，尤其是在处理稀疏奖励和长时域环境中的复杂视觉特征时。为克服这些限制，我们提出了针对长时域任务的识别-感知-规划-执行 (Recognize-Sense-Plan-Act, RSPA) 流程，并进一步引入了 RoboHorizon，一种针对长时域机器人操作设计的基于大型语言模型 (LLM) 辅助的多视角世界模型。在 RoboHorizon 中，预训练的大型语言模型根据任务语言指令生成多阶段子任务的密集奖励结构，使机器人能够更好地识别长时域任务。关键帧发现随后被整合进多视角掩码自编码器 (MAE) 架构，以增强机器人感知关键任务序列的能力，加强其对长时域过程的多阶段感知。利用这些密集奖励和多视角表示，构建了机器人世界模型以高效规划长时域任务，使机器人能够通过强化学习算法可靠执行操作。在两个代表性基准 RLBench 和 FurnitureBench 上的实验表明，RoboHorizon 优于最先进的基于视觉模型的强化学习方法，在 RLBench 的 4 个短时域任务上任务成功率提升了 23.35%，在 RLBench 的 6 个长时域任务及 FurnitureBench 的 3 个家具组装任务上提升了 29.23%。

# 1 Introduction

# 1 引言

A general-purpose robotic manipulator for real-life applications should be capable of performing long-horizon tasks composed of multiple sub-task phases, such as kitchen organization or warehouse picking. For instance, kitchen organization requires a robot to complete tasks like sorting food items, placing them into the refrigerator, and cleaning the countertops, while warehouse picking might involve identifying orders, picking items, and packing them. But how can we design such a comprehensive robotic system? Traditionally, long-horizon robotic tasks are tackled using the ”Sense-Plan-Act” (SPA) pipeline [Marton, 1984; Paul, 1981; Murphy, 2019], which involves perceiving the environment, planning tasks based on a dynamic model, and executing actions through low-level controllers. A common approach to implementing this pipeline involves using visual and language encoders to extract task-relevant features for representation learning, followed by training control policies with model-based visual reinforcement learning (RL) [Dalal et al., 2021; Yamada et al., 2021; Dalal et al., 2024]. While the above solutions are

面向实际应用的通用机器人操作臂应具备执行由多个子任务阶段组成的长时域任务的能力，例如厨房整理或仓库拣选。例如，厨房整理要求机器人完成食材分类、放入冰箱及清洁台面等任务，而仓库拣选可能涉及识别订单、拣选物品和包装。但我们如何设计这样一个综合性的机器人系统呢?

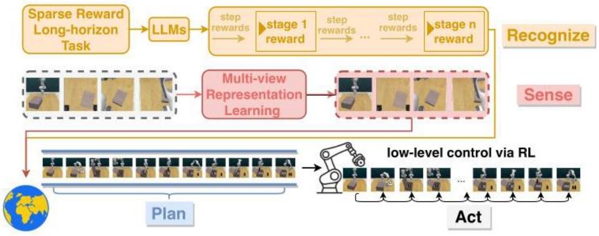


Figure 1: The proposed RSPA pipeline for long-horizon robotic manipulation.

图 1: 提出的用于长时域机器人操作的 RSPA 流程。

somewhat effective, they still face significant challenges in complex long-horizon tasks: (1) Language and visual encoders struggle to capture the hierarchical structure and dependencies of multi-stage sub-tasks in long-horizon tasks; and (2) Environmental feedback in such tasks is often sparse, while RL policies heavily relies on the rational reward structure. The former limits the robot's ability to fully understand task dynamics and environmental context, while the latter further hinders the development of stable and effective long-horizon manipulation policies.

传统上，长时域机器人任务采用“感知-规划-执行”(Sense-Plan-Act, SPA) 流程 [Marton, 1984; Paul, 1981; Murphy, 2019]，包括感知环境、基于动态模型规划任务以及通过低级控制器执行动作。实现该流程的常见方法是使用视觉和语言编码器提取与任务相关的特征进行表示学习，随后通过基于模型的视觉强化学习 (RL) 训练控制策略 [Dalal et al., 2021; Yamada et al., 2021; Dalal et al., 2024]。尽管上述方案在一定程度上有效，但在复杂的长时域任务中仍面临重大挑战:(1) 语言和视觉编码器难以捕捉长时域任务中多阶段子任务的层级结构和依赖关系；(2) 此类任务中的环境反馈通常稀疏，而强化学习策略高度依赖合理的奖励结构。前者限制了机器人对任务动态和环境上下文的全面理解，后者则进一步阻碍了稳定且有效的长时域操作策略的发展。

Our key insight is that achieving stable execution of long-horizon tasks in model-based visual RL relies on enabling robots to accurately understand tasks, perceive multistage interactions between the robot and objects in the environment, and learn stable control policies through a structured reward system. How can robots be equipped with these capabilities? We propose leveraging pre-trained large language models (LLMs) and visual demonstrations captured by multi-view cameras to empower robots, primarily because: 1) LLMs have made significant advancements in robotics, demonstrating capabilities such as step-by-step planning [Liang et al., 2023; Zeng et al., 2022; Ahn et al., 2022; Snell et al., 2022], goal-oriented dialogue [Zeng et al., 2022; Ahn et al., 2022; Huang et al., 2022], sub-goals [Huang et al., 2023; Chen et al., 2024a] and reward generation [Chiang et al., 2019; Yu et al., 2023] for robotic tasks based on language instructions. and 2) Observations from multi-camera views can significantly enhance a robot's visual manipulation capabilities, and this setup is becoming increasingly common in real-world applications. Execution trajectories captured from different viewpoints often share similar environmental dynamics and physical structures. Previous studies have explored learning control policies from multi-view offline data using model-based RL [Seo et al., 2023b] or imitation learning (IL) [Goyal et al., 2023; Shridhar et al., 2023; Ke et al., 2024].

我们的关键见解是，实现基于模型的视觉强化学习中长时域任务的稳定执行，依赖于使机器人能够准确理解任务、感知机器人与环境中物体之间的多阶段交互，并通过结构化的奖励系统学习稳定的控制策略。如何赋予机器人这些能力？我们提出利用预训练的大型语言模型 (LLMs) 和由多视角摄像头捕捉的视觉示范来增强机器人，主要原因是:1)LLMs 在机器人领域取得了显著进展，展示了基于语言指令的逐步规划 [Liang et al., 2023; Zeng et al., 2022; Ahn et al., 2022; Snell et al., 2022]、目标导向对话 [Zeng et al., 2022; Ahn et al., 2022; Huang et al., 2022]、子目标 [Huang et al., 2023; Chen et al., 2024a] 及奖励生成 [Chiang et al., 2019; Yu et al., 2023] 等能力；2) 多摄像头视角的观察能显著提升机器人的视觉操作能力，这种配置在实际应用中日益普及。从不同视角捕获的执行轨迹通常具有相似的环境动态和物理结构。先前研究已探索通过基于模型的强化学习 [Seo et al., 2023b] 或模仿学习 (IL)[Goyal et al., 2023; Shridhar et al., 2023; Ke et al., 2024] 从多视角离线数据中学习控制策略。

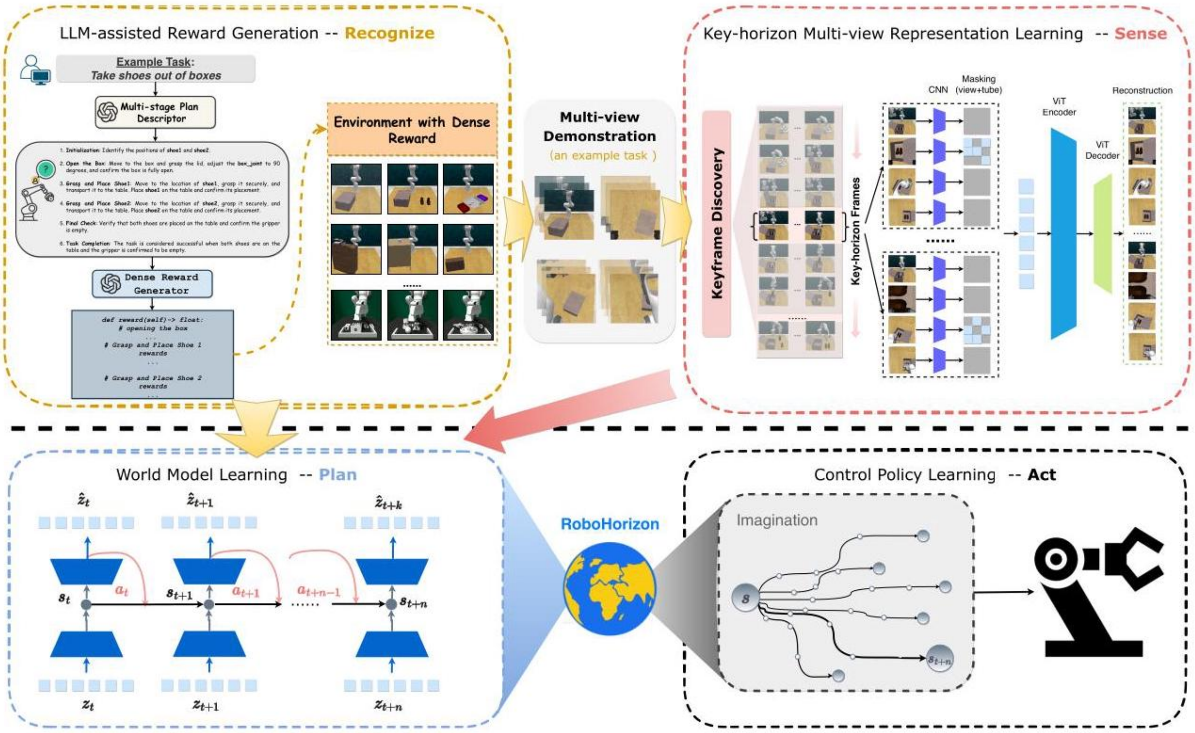

LLM-assisted Reward Generation -- Recognize
Example Task:
Take shoes out of boxes
Multi-stage Plan Descriptor
1. Initialization: Identify the positions of shoe1 and shoe2.
2. Open the Box: Move to the box and grasp the lid; adjust the box_joint to 90 degrees, and confirm the box is fully open.
3. Grasp and Place Shoe1: Move to the location of shoe1, grasp it securely, and transport it to the table. Place shoe1 on the table and confirm its placement.
4. Grasp and Place Shoe2: Move to the location of shoe2, grasp it securely, and transport it to the table. Place shoe2 on the table and confirm its placement.
5. Final Check: Verify that both shoes are placed on the table and confirm the gripper is empty.
6. Task Completion: The task is considered successful when both shoes are on the table and the gripper is confirmed to be empty.
Dense Reward Generator
def reward(self)-> float:
# opening the box
...
# Grasp and Place Shoe 1 rewards
...
# Grasp and Place Shoe 2 rewards
...
Environment with Dense Reward
......
Multi-view Demonstration
(an example task)
Key-horizon Multi-view Representation Learning -- Sense
Keyframe Discovery
Key-horizon Frames
CNN
Masking (view+tube)
ViT Encoder
ViT Decoder
Reconstruction
......
World Model Learning -- Plan
$\hat{z}_t$
$\hat{z}_{t+1}$
$\hat{z}_{t+k}$
$s_t$
$a_t$
$s_{t+1}$
$a_{t+1}$
$a_{t+n-1}$
$s_{t+n}$
$z_t$
$z_{t+1}$
$z_{t+n}$
RoboHorizon
Control Policy Learning -- Act
Imagination
$s$
$s_{t+n}$

Figure 2: RoboHorizon overview, using the long-horizon robotic manipulation task ”take shoes out of box” in RLBench as the illustration example, following the proposed RSPA pipeline.

图 2:RoboHorizon 概览，以 RLBench 中的长时域机器人操作任务“从盒子中取鞋”为示例，遵循所提 RSPA 流程。

Building on the above insight, we extend the traditional SPA pipeline into the Recognize-Sense-Plan-Act (RSPA) pipeline, as illustrated in Fig. 1, specifically designed to address the challenges in long-horizon robotic manipulation. At the core of this pipeline is RoboHorizon, an LLM-assisted multiview world model that enables robots to execute tasks effectively in complex, long-horizon robotic scenarios. To construct RoboHorizon, we first leverage pre-trained LLMs to generate a reasonable reward system for long-horizon tasks with sparse rewards. Unlike previous methods that use LLMs to generate reward signals directly applied to motion controllers [Chiang et al., 2019; Yu et al., 2023], which fail to address the complexities of multi-stage, long-horizon decision-making, our approach utilizes LLMs to divide long-horizon tasks into multi-stage sub-tasks. For each stage, dense stepwise rewards and intermediate rewards are generated, along with task reward code integrated into the environment interface, enabling robots to fundamentally recognize the long-horizon tasks they need to execute. Next, by collecting multi-view demonstrations of long-horizon manipulation tasks, we move beyond previous methods that directly perform representation learning on multi-view long-horizon demonstrations [Seo et al., 2023b; Goyal et al., 2023; Shridhar et al., 2023]. Instead, we integrate the multi-view masked autoencoder (MAE) architecture [Seo et al., 2023b; He et al., 2021] with the keyframe discovery method in multistage sub-tasks [James and Davison, 2022], proposing a novel long-horizon key-horizon multi-view representation learning method. This method enables robots to more accurately sense interactions between the robotic gripper and objects during critical multi-task stages. Building on dense reward structures and the learned key-horizon multi-view representations, we construct RoboHorizon, enabling robots to effectively and efficiently plan action trajectories for long-horizon tasks. Finally, we leverage imagined trajectories generated by the world model to train RL policies, enabling effective low-level act capabilities for robots. We evaluate RoboHorizon on 4 short-horizon and 6 long-horizon tasks from the RLBench [James et al., 2020] and 3 furniture assembly tasks from the Furni-tureBench [Heo et al., 2023], both representative testbeds for robotic manipulation under sparse rewards and multi-camera settings. RoboHorizon outperforms state-of-the-art model-based visual RL methods, with a 25.35% improvement on short-horizon RLBench tasks and 29.23% on long-horizon RLBench tasks and FurnitureBench assembly tasks.

基于上述见解，我们将传统的 SPA 流程扩展为识别-感知-规划-执行 (Recognize-Sense-Plan-Act, RSPA) 流程, 如图 1 所示, 专门针对长时域机器人操作中的挑战设计。该流程的核心是 RoboHorizon, 一种由大型语言模型辅助的多视角世界模型，使机器人能够在复杂的长时域机器人场景中有效执行任务。构建 RoboHorizon 时，我们首先利用预训练的 LLMs 为稀疏奖励的长时域任务生成合理的奖励系统。不同于以往直接使用 LLMs 生成奖励信号并应用于运动控制器的方法 [Chiang et al., 2019; Yu et al., 2023]，这些方法未能解决多阶段长时域决策的复杂性，我们的方法利用 LLMs 将长时域任务划分为多阶段子任务。针对每个阶段，生成密集的逐步奖励和中间奖励，并将任务奖励代码集成到环境接口中，使机器人能够从根本上识别其需执行的长时域任务。接着，通过收集长时域操作任务的多视角示范，我们超越了以往直接对多视角长时域示范进行表征学习的方法 [Seo et al., 2023b; Goyal et al., 2023; Shridhar et al., 2023]。相反，我们将多视角掩码自编码器 (MAE) 架构 [Seo et al., 2023b; He et al., 2021] 与多阶段子任务中的关键帧发现方法 [James and Davison, 2022] 相结合，提出了一种新颖的长时域关键视角多视角表征学习方法。该方法使机器人能够更准确地感知机器人夹持器与物体在关键多任务阶段的交互。基于密集奖励结构和学习到的关键视角多视角表征，我们构建了 RoboHorizon，使机器人能够有效且高效地规划长时域任务的动作轨迹。最后，我们利用世界模型生成的想象轨迹训练强化学习策略，实现机器人有效的低层执行能力。我们在 RLBench[James et al., 2020] 的 4 个短时域和 6 个长时域任务以及 FurnitureBench[Heo et al., 2023] 的 3 个家具组装任务上评估了 RoboHorizon，这些均为稀疏奖励和多摄像头设置下机器人操作的代表性测试平台。RoboHorizon 在短时域 RLBench 任务上取得了 25.35% 的提升，在长时域 RLBench 任务和 FurnitureBench 组装任务上提升了 29.23%，优于最先进的基于模型的视觉强化学习方法。

Our contributions can be summarized as follows: (1) We introduce a novel Recognize-Sense-Plan-Act (RSPA) pipeline for long-horizon robotic manipulation, which tightly integrates LLMs for dense reward structures generation (Recognize), key-horizon multi-view representation learning for interaction perceiving (Sense), a world model for action trajectories planning (Plan), and RL policies for robot control (Act). (2) Based on the Recognize-Sense-Plan-Act (RSPA) pipeline, we propose RoboHorizon, a robot world model specifically designed for long-horizon manipulation tasks, built upon LLM-generated dense reward structures and key-horizon multi-view representations. It enables efficient long-horizon task planning and ultimately ensures the stable execution of RL policies. (3) We provide a comprehensive empirical study of RoboHorizon's performance in both short- and long-horizon manipulation tasks, validating RoboHorizon's effectiveness enabled by the proposed RSPA pipeline.

我们的贡献可总结如下:(1) 我们提出了一种新颖的识别-感知-规划-执行 (Recognize-Sense-Plan-Act, RSPA) 流水线，用于长时域机器人操作，该流水线紧密整合了用于密集奖励结构生成的大型语言模型 (LLMs)(识别)、用于交互感知的关键时域多视角表示学习 (感知)、用于动作轨迹规划的世界模型 (规划) 以及用于机器人控制的强化学习策略 (执行)。(2) 基于识别-感知-规划-执行 (RSPA) 流水线，我们提出了 RoboHorizon，一种专为长时域操作任务设计的机器人世界模型，构建于 LLM 生成的密集奖励结构和关键时域多视角表示之上。它实现了高效的长时域任务规划，并最终确保强化学习策略的稳定执行。(3) 我们对 RoboHorizon 在短时域和长时域操作任务中的性能进行了全面的实证研究，验证了由所提 RSPA 流水线赋能的 RoboHorizon 的有效性。

# 2 Related Work

# 2 相关工作

Methods for Long-horizon Manipulation Tasks Long-horizon robotic tasks are typically addressed through the ”Sense-Plan-Act” (SPA) pipeline [Marton, 1984; Paul, 1981; Murphy, 2019]. This pipeline involves comprehensive environment perception, task planning based on dynamic models of the environment, and action execution via low-level controllers. Traditional methods encompass a range of techniques, from operation planning [Taylor et al., 1987], grasp analysis [Miller and Allen, 2004] to task and motion planning (TAMP) [Garrett et al., 2021] and skill-chaining [Chen et al., 2024b]. Recent approaches, on the other hand, integrate vision-driven learning techniques [Mahler et al., 2016; Sundermeyer et al., 2021]. These algorithms enable long-horizon decision-making in complex, high-dimensional action spaces [Dalal et al., 2024]. However, they often face challenges in handling contact-rich interactions [Mason, 2001; Whitney, 2004], are prone to cascading errors arising from imperfect state estimation [Kaelbling and Lozano-Pérez, 2013], and require extensive manual engineering [Garrett et al., 2020]. Our work leverages pre-trained large language models (LLMs) for task recognition, extending the traditional SPA pipeline into the Recognize-Sense-Plan-Act (RSPA) pipeline, thereby significantly reducing the dependence on manual engineering. At the same time, our key-horizon multi-view representation learning method enhances the robot’s ability to perceive contact-rich interactions. Together, these innovations effectively address cascading failures between sub-tasks, enabling robust long-horizon planning using the developed world model.

长时域操作任务的方法长时域机器人任务通常通过“感知-规划-执行”(Sense-Plan-Act, SPA) 流水线来解决 [Marton, 1984; Paul, 1981; Murphy, 2019]。该流水线包括对环境的全面感知、基于环境动态模型的任务规划以及通过低级控制器执行动作。传统方法涵盖了从操作规划 [Taylor et al., 1987]、抓取分析 [Miller and Allen, 2004] 到任务与运动规划 (TAMP)[Garrett et al., 2021] 及技能链 [Chen et al., 2024b] 等多种技术。另一方面, 近期方法融合了基于视觉的学习技术 [Mahler et al., 2016; Sundermeyer et al., 2021], 使得在复杂高维动作空间中实现长时域决策成为可能 [Dalal et al., 2024]。然而, 这些方法常面临处理接触丰富交互的挑战 [Mason, 2001; Whitney, 2004], 易受不完美状态估计引发的级联错误影响 [Kaelbling and Lozano-Pérez, 2013], 且依赖大量人工工程 [Garrett et al., 2020]。我们的工作利用预训练大型语言模型 (LLMs) 进行任务识别, 将传统 SPA 流水线扩展为识别-感知-规划-执行 (RSPA) 流水线, 从而显著减少对人工工程的依赖。同时, 我们的关键时域多视角表示学习方法增强了机器人感知接触丰富交互的能力。这些创新共同有效解决了子任务间的级联失败, 实现了基于所构建世界模型的稳健长时域规划。

Visual Robotic Control with Multi-View Observation Building on the latest advancements in computer vision and robotics learning, numerous methods have been developed to leverage multi-view data from cameras for visual control [Akinola et al., 2020; Chen et al., 2021; Hsu et al., 2022; Chen et al., 2023; Shridhar et al., 2023; Seo et al., 2023b]. Some of these methods utilize self-supervised learning to obtain view-invariant representations [Sermanet et al., 2018], learn 3D keypoints [Chen et al., 2021; Shridhar et al., 2023; Ke et al., 2024], or perform representation learning from different viewpoints [Seo et al., 2023b] to address subsequent manipulation tasks. However, these approaches are often limited to short-horizon robotic visual control tasks and lack the ability to handle long-horizon, multi-view robotic visual representations. In contrast, our work aims to develop a framework that learns key-horizon representations for multi-stage subtasks from long-horizon, multi-view visual demonstrations, enabling robots to tackle various complex long-horizon visual control tasks.

基于多视角观测的视觉机器人控制依托计算机视觉和机器人学习的最新进展，已有众多方法利用来自多摄像头的多视角数据进行视觉控制 [Akinola et al., 2020; Chen et al., 2021; Hsu et al., 2022; Chen et al., 2023; Shridhar et al., 2023; Seo et al., 2023b]。其中一些方法采用自监督学习以获得视角不变表示 [Sermanet et al., 2018]，学习三维关键点 [Chen et al., 2021; Shridhar et al., 2023; Ke et al., 2024]，或从不同视角进行表示学习 [Seo et al., 2023b]， 以应对后续操作任务。然而， 这些方法通常局限于短时域机器人视觉控制任务，缺乏处理长时域多视角机器人视觉表示的能力。相比之下，我们的工作旨在开发一个框架，从长时域多视角视觉示范中学习多阶段子任务的关键时域表示，使机器人能够应对各种复杂的长时域视觉控制任务。

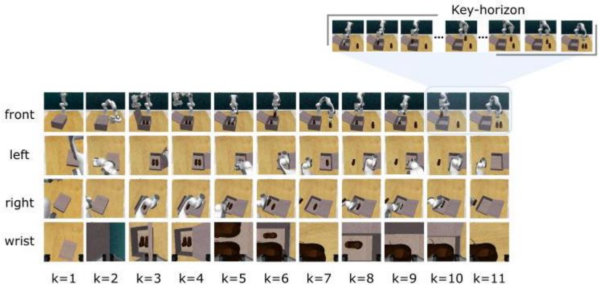


Figure 3: Visualizing the RGB observations of keyframes from four camera viewpoints for the take shoes out of the box task using the keyframe discovery method, and displaying the key-horizon between the last two keyframes from the front viewpoint.

图 3: 使用关键帧发现方法，展示“从盒子中取出鞋子”任务中来自四个摄像头视角的关键帧 RGB 观测，并显示来自正面视角的最后两个关键帧之间的关键时域。

## 3 Method

## 3 方法

In this section, we detail the construction process of Robo-Horizon, an LLM-assisted multi-view world model designed to achieve stable long-horizon robotic manipulation. First, we define the problem setting for the long-horizon manipulation tasks targeted in this work (Sec. 3.1). Next, we describe the LLM-assisted reward generation process (Recognize

本节详细介绍 Robo-Horizon 的构建过程，该模型是一种由大型语言模型辅助的多视角世界模型，旨在实现稳定的长时域机器人操作。首先，我们定义本工作所针对的长时域操作任务的问题设置 (第 3.1 节)。接着，我们描述由大型语言模型辅助的奖励生成过程 (识别

- Sec. 3.2) and key-horizon multi-view representation learning method (Sense - Sec. 3.3). Finally, we explain the development of the RoboHorizon world model (Plan - Sec. 3.4) and the implementation of robot control through RL policies (Act

- 第 3.2 节) 和关键视角多视图表示学习方法 (感知 - 第 3.3 节)。最后，我们介绍 RoboHorizon 世界模型的发展 (规划 - 第 3.4 节) 以及通过强化学习策略实现机器人控制的过程 (执行

- Sec. 3.5).

- 第 3.5 节)。

## 3.1 Problem Setup

## 3.1 问题设置

We consider a long-horizon task as a Partially Observed Markov Decision Process (POMDP) [Sutton et al., 1999] defined by $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, p_0, \mathcal{O}, p_O, \gamma)$ . Here, $\mathcal{S}$ represents the set of environment states, $\mathcal{A}$ the set of actions, $\mathcal{T}(s' \mid s, a)$ the transition probability distribution, $\mathcal{R}(s, a, s')$ the reward function, $p_0$ the initial state distribution, $\mathcal{O}$ the set of observations, $p_O(O \mid s)$ the observation distribution, and $\gamma$ the discount factor. A sub-task $\omega$ is a smaller POMDP $(\mathcal{S}, \mathcal{A}_\omega, \mathcal{T}, \mathcal{R}_\omega, p_0^\omega)$ derived from the full task's POMDP. For example, in the task "take shoes out of the box", the first sub-task is "open the box". The next sub-task, "grasp and place shoe I", can only begin after the first is completed. When multiple sub-tasks are highly sequentially dependent, this forms the long-horizon tasks we focus on. In our case, the observation space consists of all RGB images. The reward function is generated by large language models (LLMs), and the task description is provided to the agent in natural language. We also assume the availability of multi-view demonstration data for the task: $\zeta_n^v = \{o_0^v, \ldots, o_n^v\}$ , where $v \in \mathcal{V}$ represents available viewpoints, and $n$ is the total time step of the demonstrations.

我们将长时域任务视为部分可观测马尔可夫决策过程 (POMDP)[Sutton et al., 1999]，定义为 $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, p_0, \mathcal{O}, p_O, \gamma)$ 。其中，$\mathcal{S}$ 表示环境状态集合，$\mathcal{A}$ 表示动作集合，$\mathcal{T}(s' \mid s, a)$ 表示转移概率分布，$\mathcal{R}(s, a, s')$ 表示奖励函数，$p_0$ 表示初始状态分布，$\mathcal{O}$ 表示观测集合，$p_O(O \mid s)$ 表示观测分布，$\gamma$ 表示折扣因子。子任务 $\omega$ 是从完整任务 POMDP 派生出的较小 POMDP $(\mathcal{S}, \mathcal{A}_\omega, \mathcal{T}, \mathcal{R}_\omega, p_0^\omega)$ 。例如，在“从盒子里拿鞋子”任务中，第一个子任务是“打开盒子”。下一个子任务“抓取并放置鞋子 I”只能在第一个完成后开始。当多个子任务高度顺序依赖时，形成了我们关注的长时域任务。在我们的案例中，观测空间由所有 RGB 图像组成。奖励函数由大型语言模型 (LLMs) 生成，任务描述以自然语言提供给智能体。我们还假设任务具备多视角示范数据: $\zeta_n^v = \{o_0^v, \ldots, o_n^v\}$ ，其中 $v \in \mathcal{V}$ 表示可用视角，$n$ 为示范的总时间步数。

## 3.2 LLM-assisted Reward Generation - Recognize

## 3.2 LLM 辅助奖励生成 - 识别

We present the RoboHorizon overview in Fig. 1, using the long-horizon manipulation task "take shoes out of box" in RLBench as the illustration example. Specifically, given a task's language description, we prompt the LLMs to generate a corresponding task plan and encode dense rewards that align closely with each stage phase of the task. Following Yu et al. (2023), we decompose the process of translating language into rewards into two stages: multi-stage tasks description and dense reward generation. It is worth noting that while we reference the architecture of Yu et al. (2023), the internal prompts and tasks setting are entirely different.

我们在图 1 中展示了 RoboHorizon 的概览，以 RLBench 中的长时域操作任务“从盒子里拿鞋子”为示例。具体来说，给定任务的语言描述，我们提示大型语言模型生成相应的任务计划，并编码与任务各阶段紧密对应的密集奖励。继 Yu 等人 (2023) 之后，我们将语言转化为奖励的过程分解为两个阶段: 多阶段任务描述和密集奖励生成。值得注意的是，尽管我们参考了 Yu 等人 (2023) 的架构，但内部提示和任务设置完全不同。

As can be seen from the Recognize part in Fig. 1, in the Stage 1, we employ a pre-trained LLM as the Multi-stage Plan Descriptor that interprets and expands user input into detailed language descriptions of the required robot motions using predefined templates. To enable the multi-stage plan descriptor to generate a coherent structure for long-horizon tasks, we create a prompt template that outlines the current robot task setting. This leverages the pre-trained LLM's internal knowledge of motion planning to produce detailed motion descriptions. In the Stage 2, we deploy another LLM as the Dense Reward Generator to convert these motion descriptions into corresponding reward functions. We approach this as a coding task, leveraging the pre-trained LLM's understanding of coding and code structure. Four types of prompts guide the dense reward generator in generating the reward codes: i) task stage descriptions based on the task environment interface, ii) examples of expected reward generator responses, iii) constraints and rules for the reward encoder, and iv) specific task descriptions. Due to space limitations, example prompts for Stage 1 and Stage 2 of the take shoes out of box task are in the Appendix. Additionally, while any pre-trained language model can be used for reward generation, we find that only GPT-4o (OpenAI, 2024) reliably generates correct plans and rewards for all tasks. A detailed data flow of LLM-assisted reward generation is in the Appendix.

如图 1 中识别部分所示，在第一阶段，我们采用预训练大型语言模型作为多阶段计划描述器，利用预定义模板将用户输入解释并扩展为详细的机器人动作语言描述。为了使多阶段计划描述器生成连贯的长时域任务结构，我们设计了一个提示模板，概述当前机器人任务设置，借助预训练模型对运动规划的内在知识生成详细动作描述。在第二阶段，我们部署另一大型语言模型作为密集奖励生成器，将这些动作描述转换为对应的奖励函数。我们将此视为编码任务，利用预训练模型对编码及代码结构的理解。四类提示引导密集奖励生成器生成奖励代码:i) 基于任务环境接口的任务阶段描述，ii) 预期奖励生成器响应示例，iii) 奖励编码器的约束和规则，iv) 具体任务描述。由于篇幅限制，“从盒子里拿鞋子”任务第一阶段和第二阶段的示例提示见附录。此外，虽然任何预训练语言模型均可用于奖励生成，但我们发现仅 GPT-4o(OpenAI，2024) 能可靠地为所有任务生成正确的计划和奖励。LLM 辅助奖励生成的详细数据流程见附录。

## 3.3 Key-horizon Multi-view Representation Learning - Sense

## 3.3 关键时刻多视角表示学习 - 感知

To enable robots to learn multi-stage interaction representations from long-horizon multi-view visual demonstrations, we propose the Key-Horizon Multi-View Masked Autoen-coder (KMV-MAE) based on the MV-MAE architecture [Seo et al., 2023b]. As shown in the Sense part of Fig. 1, our KMV-MAE method extracts key-horizons from multi-view demonstrations using the keyframe discovery method [James and Davison, 2022]. We then perform view-masked training on these key-horizons and use a video masked autoencoder to reconstruct missing pixels from masked viewpoints. Following prior work [Seo et al., 2023a; Seo et al., 2023b], we mask convolutional features instead of pixel patches and predict rewards to capture fine-grained details essential for long-horizon visual control.

为了使机器人能够从长时域多视角视觉示范中学习多阶段交互表示，我们提出了基于 MV-MAE 架构 [Seo et al., 2023b] 的关键时刻多视角掩码自编码器 (Key-Horizon Multi-View Masked Autoencoder, KMV-MAE)。如图 1 中感知部分所示, 我们的 KMV-MAE 方法利用关键帧发现方法 [James and Davison, 2022] 从多视角示范中提取关键时刻。随后，我们对这些关键时刻进行视角掩码训练，并使用视频掩码自编码器从被掩码的视角重建缺失像素。继承先前工作 [Seo et al., 2023a; Seo et al., 2023b], 我们掩码的是卷积特征而非像素块，并预测奖励以捕捉对长时域视觉控制至关重要的细粒度细节。

Keyframe Discovery. The keyframe discovery method in our KMV-MAE, following previous works [James and Davison, 2022; Goyal et al., 2023; Shridhar et al., 2023; Ke et al., 2024], identifies keyframes based on near-zero joint velocities and unchanged gripper states. As illustrated in Fig. 3, this method captures each viewpoint’s keyframe $\mathcal{K}^v = \{k_1^v, k_2^v, \ldots, k_m^v\}$ in the take shoes out of the box task from the demonstration $\zeta^v$ , where $k$ represents the keyframe number. The corresponding time steps in the demonstration for each keyframe are $\{t_{k_1}, \ldots, t_{k_m}\}$ . Each adjacent keyframe pair $k_i^v$ and $k_{i+1}^v$ then forms a key-horizon $h_i = \left\{o_{t_{k_i}}^v, \ldots, o_{t_{k_{i+1}}}^v\right\}$ . Notably, the number of RGB observations in each key-horizon varies, depending on the time step difference between the adjacent keyframes in the demonstration.

关键帧发现。我们的 KMV-MAE 中的关键帧发现方法, 遵循先前工作 [James and Davison, 2022; Goyal et al., 2023; Shridhar et al., 2023; Ke et al., 2024]，基于接近零的关节速度和未变化的夹爪状态来识别关键帧。如图 3 所示，该方法捕捉了示范中“从盒子中取鞋”任务的每个视角的关键帧 $\mathcal{K}^v = \{k_1^v, k_2^v, \ldots, k_m^v\}$ ，其中 $k$ 表示关键帧编号。每个关键帧对应的示范时间步为 $\{t_{k_1}, \ldots, t_{k_m}\}$ 。每对相邻关键帧 $k_i^v$ 和 $k_{i+1}^v$ 构成一个关键时刻 $h_i = \left\{o_{t_{k_i}}^v, \ldots, o_{t_{k_{i+1}}}^v\right\}$ 。值得注意的是，每个关键时刻中的 RGB 观测数量因示范中相邻关键帧的时间步差异而异。

View&Tube Masking and Reconstruction. To extract more interaction information from multi-view long-horizon demonstrations, we propose a view&tube masking method. For each frame, we randomly mask all features from three of the four viewpoints, while the remaining viewpoint has 95% of its patches randomly masked. Across the key-horizon, the unmasked viewpoint follows the tube masking strategy [Tong et al., 2022]. This approach enhances cross-view feature learning, accounts for temporal correlations within a single viewpoint, reduces information leakage, and improves temporal feature representation. We integrate video masked autoencoding [Feichtenhofer et al., 2022; Tong et al., 2022] with the view&tube masking operation. Vision Transformer (ViT) [Dosovitskiy et al., 2020] layers encode unmasked feature sequences across all viewpoints and frames. Following Seo et al. (2023a; 2023b), we concatenate mask tokens with the encoded features and add learnable parameters for each viewpoint and frame to align features with mask tokens. Finally, ViT layers decode the features, projecting them to reconstruct pixel patches while also predicting rewards to encode task-relevant information. This representation learning process can be summarized as:

视角与时序掩码及重建。为了从多视角长时域示范中提取更多交互信息，我们提出了一种视角与时序掩码方法。对于每一帧，我们随机掩码四个视角中的三个视角的所有特征，而剩余视角则随机掩码其部分补丁 95% 。在关键时刻范围内，未掩码的视角遵循时序掩码策略 [Tong et al., 2022]。该方法增强了跨视角特征学习，考虑了单一视角内的时间相关性，减少了信息泄露，并提升了时间特征表示。我们将视频掩码自编码 [Feichtenhofer et al., 2022; Tong et al., 2022] 与视角与时序掩码操作相结合。视觉变换器 (Vision Transformer， ViT)[Dosovitskiy et al., 2020] 层对所有视角和帧的未掩码特征序列进行编码。继 Seo 等人 (2023a; 2023b) 的方法，我们将掩码标记与编码特征拼接，并为每个视角和帧添加可学习参数以对齐特征与掩码标记。最后，ViT 层解码特征，投影以重建像素补丁，同时预测奖励以编码任务相关信息。该表示学习过程可总结为:

Given demonstration videos $\zeta_n^v = \{o_0^v, \ldots, o_n^v\}$ , after $m$ keyframes $\{k_1^v, k_2^v, \ldots, k_m^v\}$ are extracted by the keyframe discovery method, they become in the form of containing $m-1$ key-horizon: $\zeta^v = \left\{h_1^v, \ldots, h_{m-1}^v\right\}_{v \in \mathcal{V}}$ from multiple viewpoints. Given LLM-assisted generated rewards $r = \{r_1, \ldots, r_n\}$ , and a mask ratio of $m$ , KMV-MAE consists of following components:

给定演示视频 $\zeta_n^v = \{o_0^v, \ldots, o_n^v\}$ ， 在通过关键帧发现方法提取出 $m$ 个关键帧 $\{k_1^v, k_2^v, \ldots, k_m^v\}$ 后，它们以包含 $m-1$ 个关键视角 (key-horizon) 的形式呈现: $\zeta^v = \left\{h_1^v, \ldots, h_{m-1}^v\right\}_{v \in \mathcal{V}}$ 来自多个视角。给定由大语言模型 (LLM) 辅助生成的奖励 $r = \{r_1, \ldots, r_n\}$ ， 以及掩码比例为 $m$ ，KMV-MAE 包含以下组件:

Convolution stem: $l_i^v = f_\phi^{\text{conv}}\left(h_i^v\right)$

卷积干线 (Convolution stem): $l_i^v = f_\phi^{\text{conv}}\left(h_i^v\right)$

View&Tube masking: $l_i^m \sim p^{\text{mask}}\left(l_i^m \mid \{h_i^v\}_{v \in \mathcal{V}}, m\right)$

视角与时序掩码 (View&Tube masking): $l_i^m \sim p^{\text{mask}}\left(l_i^m \mid \{h_i^v\}_{v \in \mathcal{V}}, m\right)$

ViT encoder: $z_i^m \sim p_\phi\left(z_i^m \mid l_i^m\right)$

视觉变换器编码器 (ViT encoder): $z_i^m \sim p_\phi\left(z_i^m \mid l_i^m\right)$

(1)

ViT decoder: $\begin{cases} \left\{\widehat{h}_i^v\right\}_{v \in \mathcal{V}} \sim p_\phi\left(\left\{\widehat{h}_i^v\right\}_{v \in \mathcal{V}} \mid z_i^m\right) \\ \widehat{r}_{t_{k_i}, t_{k_{i+1}}} \sim p_\phi\left(\widehat{r}_{t_{k_i}, t_{k_{i+1}}} \mid z_i^m\right) \end{cases}$

视觉变换器解码器 (ViT decoder): $\begin{cases} \left\{\widehat{h}_i^v\right\}_{v \in \mathcal{V}} \sim p_\phi\left(\left\{\widehat{h}_i^v\right\}_{v \in \mathcal{V}} \mid z_i^m\right) \\ \widehat{r}_{t_{k_i}, t_{k_{i+1}}} \sim p_\phi\left(\widehat{r}_{t_{k_i}, t_{k_{i+1}}} \mid z_i^m\right) \end{cases}$

Finally the model is trained to reconstruct key-horizon pixels and predict rewards, i.e., minimizing the negative log-likelihood to optimize the model parameter $\phi$ as follows:

最终，模型被训练以重建关键视角像素并预测奖励，即通过最小化负对数似然来优化模型参数 $\phi$ ，具体如下:

$$\mathcal{L}^{\text{kwvmae}}(\phi) = -\ln p_\phi\left(\{h_i^v\}_{v\in\mathcal{V}} \mid z_i^m\right) - \ln p_\phi\left(r_{t_{k_i},t_{k_{i+1}}} \mid z_i^m\right)$$

## 3.4 RoboHorizon World Model - Plan

## 3.4 RoboHorizon 世界模型 - 规划

For the Plan part in Fig. 1, we construct RoboHorizon following previous works[Seo et al., 2023a; Seo et al., 2023b], implementing it as a variant of the Recurrent State Space Model (RSSM) [Hafner et al., 2019]. The model uses frozen autoencoder representations from the previous key-horizon multi-view representation learning as inputs and reconstruction targets. RoboHorizon includes the following components:

对于图 1 中的规划部分，我们遵循先前工作 [Seo et al., 2023a; Seo et al., 2023b] 构建 RoboHorizon，将其实现为递归状态空间模型 (Recurrent State Space Model, RSSM)[Hafner et al., 2019] 的变体。该模型使用来自先前关键视角多视图表示学习的冻结自编码器表示作为输入和重建目标。RoboHorizon 包括以下组件:

Encoder:

编码器:

$$s_t \sim q_\theta\left(s_t \mid s_{t-1}, a_{t-1}, z_t\right)$$

Decoder:

解码器:

$$\begin{cases} \widehat{z}_t \sim p_\theta\left(\widehat{z}_t \mid s_t\right) \\ \widehat{r}_t \sim p_\theta\left(\widehat{r}_t \mid s_t\right) \end{cases} \tag{2}$$

Dynamics model:

动力学模型:

$$\widehat{s}_t \sim p_\theta\left(\widehat{s}_t \mid s_{t-1}, a_{t-1}\right)$$

The encoder extracts state $s_t$ from the previous state $s_{t-1}$ , previous action $a_{t-1}$ , and current autoencoder representations $z_t$ . The dynamics model predicts $s_t$ without access to $z_t$ , allowing forward prediction. The decoder reconstructs $z_t$ to provide learning signals for model states and predicts $r_t$ to compute rewards from future states without decoding future autoencoder representations. All model parameters $\theta$ are optimized jointly by minimizing the negative variational lower bound [Kingma and Welling, 2014].:

编码器从先前状态 $s_{t-1}$ 、先前动作 $a_{t-1}$ 和当前自编码器表示 $z_t$ 中提取状态 $s_t$ 。动力学模型在不访问 $z_t$ 的情况下预测 $s_t$ ，实现前向预测。解码器重建 $z_t$ 以为模型状态提供学习信号，并预测 $r_t$ 以从未来状态计算奖励，而无需解码未来的自编码器表示。所有模型参数 $\theta$ 通过最小化变分下界 (Variational Lower Bound)[Kingma and Welling, 2014] 联合优化:

$$\mathcal{L}^{\mathrm{wm}}(\theta)=-\ln p_\theta\left(z_t \mid s_t\right)-\ln p_\theta\left(r_t \mid s_t\right)$$

$$+\beta \operatorname{KL}\left[q_\theta\left(s_t \mid s_{t-1}, a_{t-1}, z_t\right) \| p_\theta\left(\widehat{s}_t \mid s_{t-1}, a_{t-1}\right)\right],$$

where $\beta$ is a scale hyperparameter.

其中 $\beta$ 是一个尺度超参数。

## 3.5 Control Policy Learning - Act

## 3.5 控制策略学习 - 执行

For the Act part in Fig. 1, we build on the approach of [Seo et al., 2023a; Seo et al., 2023b] and adopt the actor-critic framework from DreamerV2 [Hafner et al., 2021]. The goal is to train a policy that maximizes predicted future values by back-propagating gradients through the RoboHorizon world model. Specifically, we define a stochastic actor and a deterministic critic as:

对于图 1 中的 Act 部分, 我们在 [Seo 等人, 2023a; Seo 等人, 2023b] 的方法基础上, 采用了 DreamerV2 [Hafner 等人，2021] 的 actor-critic(行动者-评论家) 框架。目标是通过在 RoboHorizon 世界模型中反向传播梯度，训练一个能够最大化预测未来价值的策略。具体而言，我们定义了一个随机行动者 (stochastic actor) 和一个确定性评论家 (deterministic critic)，如下所示:

Actor: $\widehat{a}_t \sim p_\psi\left(\widehat{a}_t \mid \widehat{s}_t\right)$ Critic: $v_\xi\left(\widehat{s}_t\right) \approx \mathbb{E}_{p_\theta}\left[\sum_{i \leq t} \gamma^{i-t} \widehat{r}_i\right.$

行动者: $\widehat{a}_t \sim p_\psi\left(\widehat{a}_t \mid \widehat{s}_t\right)$ 评论家: $v_\xi\left(\widehat{s}_t\right) \approx \mathbb{E}_{p_\theta}\left[\sum_{i \leq t} \gamma^{i-t} \widehat{r}_i\right.$

Here, the sequence $\left\{\left(\widehat{s}_t, \widehat{a}_t, \widehat{r}_t\right)\right\}_{t=1}^H$ is predicted from the initial state $\widehat{s}_0$ using the stochastic actor and dynamics model from Eq. 2. Unlike previous work, we set $H$ to match the length of each key-horizon in the long-horizon task, with each key-horizon sequence having a different duration. Given the $\lambda$ -return [Schulman et al.,2015] defined as:

在这里，序列 $\left\{\left(\widehat{s}_t, \widehat{a}_t, \widehat{r}_t\right)\right\}_{t=1}^H$ 是通过使用方程 2 中的随机行动者和动力学模型，从初始状态 $\widehat{s}_0$ 预测得到的。与以往工作不同，我们将 $H$ 设置为与长时序任务中每个关键时域 (key-horizon) 长度相匹配，每个关键时域序列的持续时间各不相同。给定 $\lambda$ -回报 (return)[Schulman 等人，2015]，定义如下:

$$V_t^\lambda \doteq \widehat{r}_t + \gamma \begin{cases} (1-\lambda) v_\xi\left(\widehat{s}_{t+1}\right) + \lambda V_{t+1}^\lambda & \text{if } t < H \\ v_\xi\left(\widehat{s}_H\right) & \text{if } t = H \end{cases}$$

the critic is trained to regress the $\lambda$ -return, while the actor is trained to maximize the $\lambda$ -return with gradients backpropa-gated through the world model. To enable the robot to execute long-horizon tasks more reliably, we introduce an auxiliary behavior cloning loss that encourages the agent to learn expert actions while interacting with the environment. To achieve this, we follow the setup of [James and Davison, 2022; Seo et al., 2023b] to acquire the demonstration. Specifically, at each time step, given an expert action $a_t^{\mathrm{e}}$ , the objective of auxiliary behavior cloning is $\mathcal{L}^{\mathrm{BC}} = -\ln p_\psi\left(a_t^{\mathrm{e}} \mid s_t\right)$ . Thus, the objective for the actor network and the critic network is:

评论家被训练以回归 $\lambda$ -回报，而行动者则通过在世界模型中反向传播梯度，训练以最大化 $\lambda$ -回报。为了让机器人更可靠地执行长时序任务，我们引入了辅助行为克隆损失 (auxiliary behavior cloning loss)，鼓励智能体在与环境交互时学习专家动作。为此，我们按照 [James 和 Davison，2022；Seo 等人，2023b] 的设置获取演示数据。具体来说，在每个时间步，给定专家动作 $a_t^{\mathrm{e}}$ ，辅助行为克隆的目标为 $\mathcal{L}^{\mathrm{BC}} = -\ln p_\psi\left(a_t^{\mathrm{e}} \mid s_t\right)$ 。因此，行动者网络和评论家网络的目标为:

$$\mathcal{L}^{\text{critic}}(\xi) \doteq \mathbb{E}_{p_\theta, p_\psi}\left[\sum_{t=1}^{H-1} \frac{1}{2}\left(v_\xi\left(\widehat{s}_t\right) - \operatorname{sg}\left(V_t^\lambda\right)\right)^2\right]$$

$$\mathcal{L}_{\mathrm{BC}}^{\text{actor}}(\psi) \doteq \mathbb{E}_{p_\theta, p_\psi}\left[-V_t^\lambda - \eta \mathrm{H}\left[a_t \mid \widehat{s}_t\right]\right] + \mathcal{L}^{\mathrm{BC}}$$

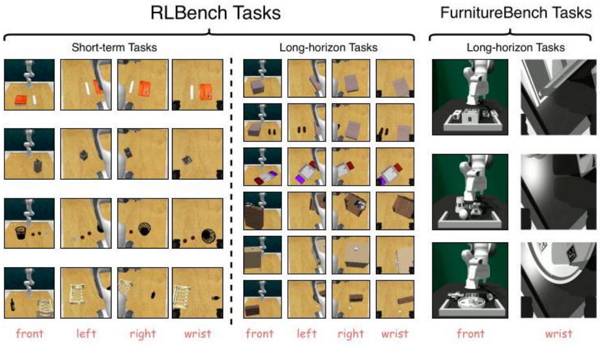


Figure 4: Visualization of multi-view demonstrations from front, left, right, and wrist cameras for 10 RLBench tasks, and from front and wrist cameras for 3 FurnitureBench tasks.

图 4: 展示了 10 个 RLBench 任务中来自前方、左侧、右侧和手腕摄像头的多视角演示，以及 3 个 FurnitureBench 任务中来自前方和手腕摄像头的演示可视化。

where sg is a stop gradient operation, $\eta$ is a scale hyperparam-eter for an entropy $\mathrm{H}\left[a_t \mid \widehat{s}_t\right]$ . Thus, with the generated dense reward structure, the training objective for the sense, plan, and act processes in RoboHorizon is to minimize the following objective function:

其中 sg 是停止梯度操作，$\eta$ 是熵 $\mathrm{H}\left[a_t \mid \widehat{s}_t\right]$ 的缩放超参数。因此，利用生成的密集奖励结构，RoboHorizon 中感知、规划和执行过程的训练目标是最小化以下目标函数:

$$\mathcal{L}^{\text{RoboHorizon}} = \underbrace{\mathcal{L}^{\text{kwymae}}}_{\text{Sense}} + \underbrace{\mathcal{L}^{\text{wm}}}_{\text{Plan}} + \underbrace{\mathcal{L}^{\text{critic}} + \mathcal{L}^{\text{actor}}_{\text{BC}}}_{\text{Act}} \tag{3}$$

# 4 Experiments

# 4 实验

We design our experiments to explore the following questions: (i) How does RoboHorizon, following the RSPA pipeline, perform compared to SPA-driven model-based reinforcement learning (RL) baselines in short- and long-horizon manipulation tasks? (ii) If SPA-based baseline methods also adopt stepwise rewards generated by LLMs but do not aware the staged reward for each sub-task stage, does RoboHorizon still maintain its advantage? (iii) To what extent do RoboHorizon's key design components impact its overall performance?

我们设计实验以探究以下问题:(i)RoboHorizon 遵循 RSPA 流程，在短时序和长时序操作任务中，与基于 SPA 的模型强化学习 (RL) 基线相比表现如何? (ii) 如果基于 SPA 的基线方法也采用由 LLM 生成的逐步奖励，但未能感知每个子任务阶段的分阶段奖励，RoboHorizon 是否仍能保持优势? (iii)RoboHorizon 的关键设计组件对其整体性能有多大影响?

Environmental Setup For quantitative evaluation, we adopt a demonstration-driven RL setup to address visual robotic manipulation tasks in RLBench [James et al., 2020] and Furni-tureBench [Heo et al., 2023]. In both benchmarks, we rely on limited environment interactions and expert demonstrations. All experiments use only RGB observations from each camera, without incorporating proprioceptive state or depth information. Following previous studies [James and Davison, 2022; Seo et al., 2023b], we populate the replay buffer with expert demonstrations, and the RL agent outputs relative changes in the gripper's position. For all tasks, 50 expert demonstrations are provided for each camera view. For FurnitureBench, we use a low-randomness environment initialization setup. Due to space limitations, more detailed experimental setups are provided in the Appendix.

环境设置为了进行定量评估，我们采用基于示范驱动的强化学习 (RL) 设置，解决 RLBench [James et al., 2020] 和 FurnitureBench [Heo et al., 2023] 中的视觉机器人操作任务。在这两个基准测试中，我们依赖有限的环境交互和专家示范。所有实验仅使用来自各摄像头的 RGB 观测，不包含本体感知状态或深度信息。遵循先前研究 [James and Davison, 2022; Seo et al., 2023b]，我们用专家示范填充重放缓冲区，RL 智能体输出夹爪位置的相对变化。所有任务中，每个摄像头视角提供 50 个专家示范。对于 FurnitureBench，我们采用低随机性的环境初始化设置。由于篇幅限制，更详细的实验设置见附录。

Multi-view Camera Setup We adopt a multi-view observation and single-view control approach [Seo et al., 2023b], suitable for scenarios where multiple cameras are available during training, but the robot relies on a single camera during deployment. For RLBench tasks, we use multi-view data from front, wrist, left, and right cameras to enhance the robot's perception of long-horizon tasks and the environment, while training a RL agent that operates solely on front camera input. For FurnitureBench tasks, we use multi-view data from front and wrist cameras with the same training and control setup. We conduct experiments on 10 representative tasks in RLBench, including 4 short-horizon tasks (phone on base, take umbrella out of stand, put rubbish in bin, stack wine) and 6 long-horizon tasks (take shoes out of box, put shoes in box, empty container, put books on bookshelf, put item in drawer, slide cabinet open and place cups), as well as 3 long-horizon furniture assembly tasks in FurnitureBench (cabinet, lamp, round table), as shown in Fig. 4. In these tasks, the front, left, right cameras provide a wide view of the robot's workspace, while the wrist camera offers close-up views of the target objects.

多视角摄像头设置我们采用多视角观测与单视角控制的方法 [Seo et al., 2023b]，适用于训练阶段可用多摄像头但部署时机器人仅依赖单摄像头的场景。对于 RLBench 任务，我们使用来自前视、腕部、左侧和右侧摄像头的多视角数据，以增强机器人对长时序任务和环境的感知，同时训练仅基于前视摄像头输入操作的 RL 智能体。对于 FurnitureBench 任务，我们使用来自前视和腕部摄像头的多视角数据，训练和控制设置相同。我们在 RLBench 中对 10 个代表性任务进行了实验，包括 4 个短时序任务 (手机放底座、从伞架取伞、将垃圾放入垃圾桶、叠放酒瓶) 和 6 个长时序任务 (从盒子取鞋、将鞋放入盒子、清空容器、将书放入书架、将物品放入抽屉、滑动柜门并放置杯子)，以及 FurnitureBench 中的 3 个长时序家具组装任务 (柜子、灯、圆桌)，如图 4 所示。在这些任务中，前视、左视、右视摄像头提供机器人工作空间的广角视野，腕部摄像头则提供目标物体的特写视角。

Table 1: Success Rates (%) on 4 short-horizon tasks ( in RLBench ) and 9 long-horizon tasks ( 6 in RLBench , 3 in FurnitureBench ). Results are averaged over 5 seeds.

表 1:4 个短时序任务 (RLBench) 和 9 个长时序任务 (RLBench 中 6 个，FurnitureBench 中 3 个) 的成功率 (%)。结果为 5 个随机种子的平均值。

| Model | Short-Horizon Tasks | | | | Long-Horizon Tasks (RLBench) | | | | | | FurnitureBench | | | Average | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Phone on Base | Take Umbrella | Put Rubbish in Bin | Stack Wine | Take Shoes | Put Shoes | Empty Container | Put Books | Put Item | Slide Cabinet & Place Cups | Cabinet | Lamp | Round Table | Short Avg. | Long Avg. |
| TCN+WM | 5.2 | 4.1 | 2.4 | 2.2 | 0 | 0 | 0 | 0.4 | 0 | 0 | 1.0 | 6.5 | 8.2 | 3.48 | 1.79 |
| CLIP+WM | 13.3 | 15.7 | 12.2 | 11.8 | 0 | 0 | 0 | 2.2 | 0 | 0 | 3.8 | 11.7 | 15.5 | 13.25 | 3.69 |
| MAE+WM | 20.4 | 19.1 | 18.6 | 19.6 | 0 | 0 | 0 | 2.5 | 0 | 0 | 8.5 | 16.3 | 20.9 | 19.43 | 5.36 |
| MWM | 32.5 | 30.8 | 28.4 | 29.7 | 1.2 | 0.8 | 2.1 | 3.3 | 1.1 | 0.5 | 15.2 | 27.5 | 32.0 | 30.35 | 9.30 |
| MV-MWM | 52.6 | 50.3 | 48.9 | 49.1 | 3.5 | 2.7 | 4.6 | 10.4 | 5.2 | 2.9 | 26.6 | 43.5 | 46.7 | 50.23 | 16.23 |
| RoboHorizon | 78.4 | 75.2 | 74.8 | 73.9 | 36.5 | 31.2 | 40.5 | 58.4 | 48.2 | 33.6 | 41.0 | 58.5 | 61.3 | 75.58(25.35%↑)45.47 (29.23% ↑) | |

| 模型 | 短期任务 | | | | 长期任务 (RLBench) | | | | | | 家具基准 | | | 平均值 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 基座上的电话 | 拿伞 | 把垃圾放进垃圾桶 | 叠酒瓶 | 拿鞋 | 放鞋 | 清空容器 | 放书 | 放物品 | 滑动柜子并放置杯子 | 柜子 | 灯 | 圆桌 | 短期平均 | 长期平均 |
| TCN+WM | 5.2 | 4.1 | 2.4 | 2.2 | 0 | 0 | 0 | 0.4 | 0 | 0 | 1.0 | 6.5 | 8.2 | 3.48 | 1.79 |
| CLIP+WM | 13.3 | 15.7 | 12.2 | 11.8 | 0 | 0 | 0 | 2.2 | 0 | 0 | 3.8 | 11.7 | 15.5 | 13.25 | 3.69 |
| MAE+WM | 20.4 | 19.1 | 18.6 | 19.6 | 0 | 0 | 0 | 2.5 | 0 | 0 | 8.5 | 16.3 | 20.9 | 19.43 | 5.36 |
| MWM | 32.5 | 30.8 | 28.4 | 29.7 | 1.2 | 0.8 | 2.1 | 3.3 | 1.1 | 0.5 | 15.2 | 27.5 | 32.0 | 30.35 | 9.30 |
| MV-MWM | 52.6 | 50.3 | 48.9 | 49.1 | 3.5 | 2.7 | 4.6 | 10.4 | 5.2 | 2.9 | 26.6 | 43.5 | 46.7 | 50.23 | 16.23 |
| RoboHorizon | 78.4 | 75.2 | 74.8 | 73.9 | 36.5 | 31.2 | 40.5 | 58.4 | 48.2 | 33.6 | 41.0 | 58.5 | 61.3 | 75.58(25.35%↑)45.47 (29.23% ↑) | |

Baselines To compare with SPA-driven model-based RL methods using manually defined rewards, we select MV-MWM [Seo et al., 2023b] and MWM [Seo et al., 2023a] as baselines. The former lacks key-horizon representation learning, while the latter lacks multi-view key-horizon representation learning. Both baselines rely on manually defined rewards and the same amount of training data. Additionally, we adopt various representation learning methods to train world models, further demonstrating the effectiveness of our key-horizon multi-view representation learning in long-horizon tasks. The comparison methods include CLIP+WM [Radford et al., 2021], MAE+WM [He et al., 2021], and TCN+WM [Sermanet et al., 2018]. Specifically, RoboHorizon, MV-MWM, MWM, and TCN+WM learn representations from scratch, whereas CLIP+WM and MAE+WM use frozen pretrained representations. More details about the experimental baselines are provided in the Appendix.

基线为了与使用手动定义奖励的基于 SPA 的模型强化学习方法进行比较，我们选择了 MV-MWM [Seo et al., 2023b] 和 MWM [Seo et al., 2023a] 作为基线。前者缺乏关键视野 (key-horizon) 表示学习，后者缺乏多视角关键视野表示学习。两者均依赖手动定义的奖励和相同数量的训练数据。此外，我们采用多种表示学习方法训练世界模型，进一步证明了我们关键视野多视角表示学习在长视野任务中的有效性。比较方法包括 CLIP+WM [Radford et al., 2021]、MAE+WM [He et al., 2021] 和 TCN+WM [Sermanet et al., 2018]。具体而言，RoboHorizon、MV-MWM、MWM 和 TCN+WM 从零开始学习表示，而 CLIP+WM 和 MAE+WM 使用冻结的预训练表示。关于实验基线的更多细节见附录。

## 4.1 Performance Comparison

## 4.1 性能比较

In this section, we conduct experiments on 4 short-horizon and 6 long-horizon tasks from RLBench, as well as 3 furniture assembly tasks from FurnitureBench, to address the three initial questions.

本节中，我们在 RLBench 中的 4 个短视野任务和 6 个长视野任务，以及 FurnitureBench 中的 3 个家具组装任务上进行了实验，以回答最初的三个问题。

SPA-driven model-based RL baselines vs. RoboHorizon Table 1 compares the success rates of RoboHorizon and five SPA-driven baselines on 4 short-horizon tasks (in RLBench) and 9 long-horizon tasks ( 6 in RLBench, 3 in FurnitureBench). RoboHorizon outperforms all baseline methods, achieving the highest average success rate across all tasks. Specifically, it exceeds MV-MWM by 25.35% on the 4 short-horizon tasks and by 29.23% on the 9 long-horizon tasks. This result shows that with LLM-assisted reward structures and key-horizon multi-view representation learning, RoboHorizon excels in both short-horizon pick-and-place tasks and long-horizon tasks with multiple sub-tasks and numerous target objects, achieving more stable manipulation. While MV-MWM benefits from multi-view representation learning, outperforming MWM, MAE+WM, CLIP+WM, and TCN+WM in representation, it still struggles with long-horizon tasks. Additionally, the manually designed reward structures are not robust, leading to inconsistent performance across all tasks for MV-MWM, MWM, MAE+WM, CLIP+WM, and TCN+WM.

基于 SPA 的模型强化学习基线与 RoboHorizon 的比较表 1 比较了 RoboHorizon 与五个基于 SPA 的基线在 4 个短视野任务 (RLBench) 和 9 个长视野任务 (RLBench 中 6 个，FurnitureBench 中 3 个) 上的成功率。RoboHorizon 优于所有基线方法，在所有任务中取得了最高的平均成功率。具体来说，在 4 个短视野任务中，其成功率比 MV-MWM 高出 25.35%，在 9 个长视野任务中高出 29.23%。该结果表明，借助大语言模型 (LLM) 辅助的奖励结构和关键视野多视角表示学习，RoboHorizon 在短视野的抓取放置任务和包含多个子任务及众多目标物体的长视野任务中均表现出色，实现了更稳定的操作。虽然 MV-MWM 受益于多视角表示学习，在表示能力上优于 MWM、MAE+WM、CLIP+WM 和 TCN+WM, 但其在长视野任务中仍存在困难。此外, 手动设计的奖励结构不够鲁棒, 导致 MV-MWM、MWM、MAE+WM、CLIP+WM 和 TCN+WM 在所有任务中的表现不稳定。

These experimental results effectively answer our first question: RoboHorizon outperforms SPA-driven baselines with manually defined rewards in both short- and long-horizon tasks, with a particularly strong advantage in long-horizon scenarios. This outcome strongly validates the capability of our proposed RSPA pipeline in handling long-horizon tasks.

这些实验结果有效回答了我们的第一个问题:RoboHorizon 在短视野和长视野任务中均优于使用手动定义奖励的基于 SPA 的基线方法，且在长视野场景中优势尤为显著。该结果有力验证了我们提出的 RSPA 流程在处理长视野任务中的能力。

SPA-driven baselines with LLM-generated stepwise rewards vs. RoboHorizon To compare the performance of SPA-driven baselines with LLM-generated stepwise rewards but without staged rewards for each sub-task phase against RoboHorizon, we replace the manually defined reward system in SPA-driven baselines with the stepwise rewards generated during RoboHorizon's Recognize phase using LLM assistance. Fig. 5 shows the success rate comparison between RoboHorizon and five SPA-driven baselines enhanced with LLM-generated stepwise rewards (as indicated by ”LLM” in the legend) across 4 short-horizon tasks and 9 long-horizon tasks. It also illustrates the performance improvement of SPA-driven baselines with LLM-generated stepwise rewards compared to manually defined rewards. The results show that while these baseline methods achieve some improvement in task success rates when using LLM-generated stepwise rewards, demonstrating the effectiveness of well-designed stepwise rewards aligned with motion planning in helping robots understand operational tasks, their performance gains in long-horizon tasks remain very limited without considering staged rewards for each sub-task phase. Furthermore, regardless of the task, they fail to surpass the full RoboHorizon framework driven by our proposed RSPA pipeline.

基于 SPA 的基线方法 (使用 LLM 生成的逐步奖励) 与 RoboHorizon 的比较为了比较使用 LLM 生成的逐步奖励但未针对每个子任务阶段设计分阶段奖励的基于 SPA 的基线方法与 RoboHorizon 的性能，我们将基于 SPA 的基线中的手动定义奖励系统替换为 RoboHorizon 识别阶段通过 LLM 辅助生成的逐步奖励。图 5 展示了 RoboHorizon 与五个增强了 LLM 生成逐步奖励 (图例中标注为“LLM” ) 的基于 SPA 基线在 4 个短视野任务和 9 个长视野任务上的成功率比较，同时展示了基于 SPA 基线使用 LLM 生成逐步奖励相较于手动定义奖励的性能提升。结果表明，虽然这些基线方法在使用 LLM 生成的逐步奖励时在任务成功率上有所提升，证明了与运动规划相匹配的良好设计的逐步奖励有助于机器人理解操作任务，但在未考虑每个子任务阶段的分阶段奖励时，其在长视野任务中的性能提升仍然非常有限。此外，无论任务如何，它们均未能超越由我们提出的 RSPA 流程驱动的完整 RoboHorizon 框架。

This result clearly answers our second question: By using LLM-generated stepwise rewards aligned with motion planning, SPA-driven baseline methods achieve certain performance improvements in both short-horizon and

该结果明确回答了我们的第二个问题: 通过使用与运动规划相匹配的 LLM 生成逐步奖励, 基于 SPA 的基线方法在短视野和

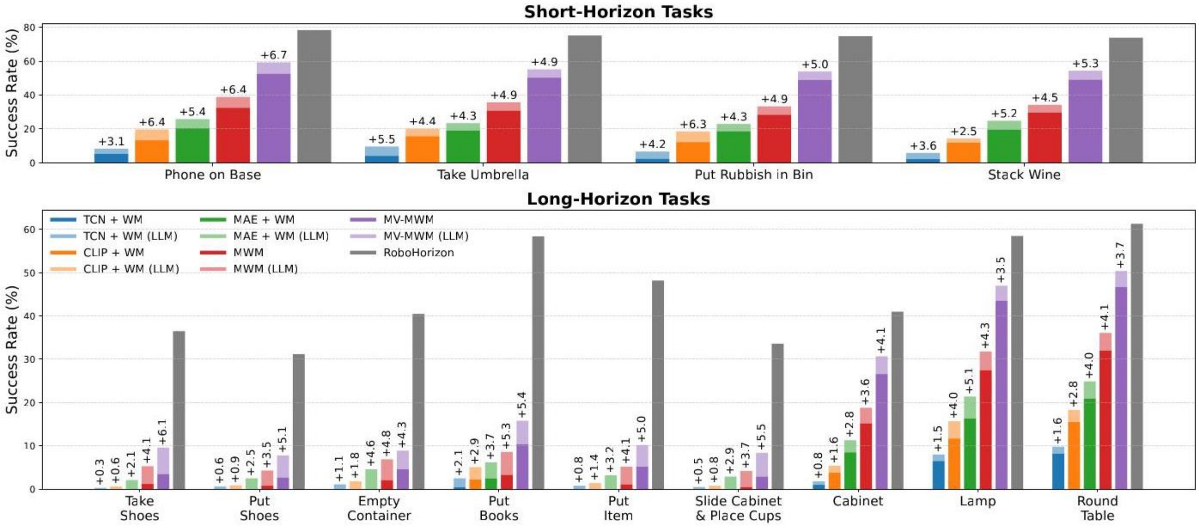


Figure 5: SPA-driven baselines with LLM-generated dense rewards vs. RoboHorizon.

图 5: 基于 SPA 的基线方法 (使用 LLM 生成的密集奖励) 与 RoboHorizon 的比较。

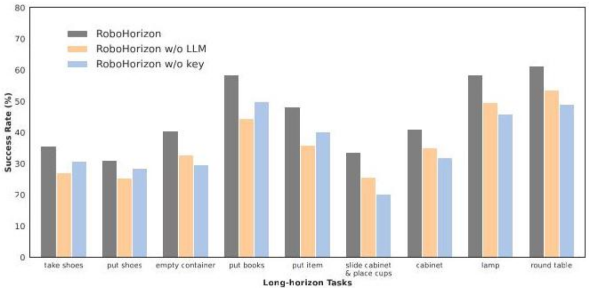


Figure 6: Ablation of key designs in RoboHorizon across 9 long-horizon tasks.

图 6:RoboHorizon 关键设计在 9 个长视野任务上的消融实验。

long-horizon tasks. However, due to their lack of consideration for staged rewards in each sub-task phase and their limited ability to learn complex long-horizon task representations, RoboHorizon still maintains a significant performance advantage. This finding further highlights the importance of the LLM-assisted reward generation mechanism and key-horizon multi-view representation learning modules in the RSPA-driven RoboHorizon framework. These modules not only enhance the robot's ability to interpret task instructions but also improve its perception of tasks and target objects in complex long-horizon scenarios.

长时任务。然而，由于其未能考虑每个子任务阶段的分阶段奖励，且在学习复杂长时任务表示方面能力有限，RoboHorizon 仍保持显著的性能优势。该发现进一步凸显了 LLM 辅助奖励生成机制和关键时域多视角表示学习模块在 RSPA 驱动的 RoboHorizon 框架中的重要性。这些模块不仅增强了机器人对任务指令的理解能力，还提升了其在复杂长时场景中对任务和目标物体的感知能力。

Ablation Study of RoboHorizon To evaluate the impact of RoboHorizon's two key designs, LLM-assisted reward generation (recognition) and key-horizon multi-view representation learning (perception), on performance, we design two comparison methods: RoboHorizon w/o LLM, which removes LLM-assisted reward generation and relies on manually designed reward structures, and RoboHorizon w/o key, which omits key-horizon multi-view representation learning and uses the MV-MWM method for world model construction. Fig. 6 shows the ablation study results across nine long-horizon tasks. The results demonstrate that both LLM-assisted reward generation and key-horizon multi-view representation learning are critical to RoboHorizon's performance, with each excelling in different scenarios. In tasks such as taking shoes out of a box, putting shoes in a box, placing books on a bookshelf, and putting items in a drawer, where objects are dispersed and representation learning is easier, RoboHorizon w/o key performs better. In contrast, in tasks like emptying a container, sliding a cabinet open and placing cups, and installing a cabinet, lamp, and round table, where objects are densely packed with more distractions, RoboHorizon w/o LLM performs better.

RoboHorizon 消融研究为评估 RoboHorizon 两个关键设计——LLM 辅助奖励生成 (识别) 和关键时域多视角表示学习 (感知) 对性能的影响，我们设计了两种对比方法:RoboHorizon w/o LLM，去除 LLM 辅助奖励生成，依赖手工设计的奖励结构；以及 RoboHorizon w/o key，省略关键时域多视角表示学习，采用 MV-MWM 方法构建世界模型。图 6 展示了九个长时任务的消融结果。结果表明，LLM 辅助奖励生成和关键时域多视角表示学习对 RoboHorizon 性能均至关重要，且各自在不同场景中表现优异。在如从盒子中取鞋、将鞋放入盒子、将书放入书架和将物品放入抽屉等物体分散且表示学习较易的任务中，RoboHorizon w/o key 表现更好。相反，在如清空容器、滑动柜门放置杯子以及安装柜子、灯具和圆桌等物体密集且干扰较多的任务中，RoboHorizon w/o LLM 表现更佳。

These findings clearly answer the third key question: Both LLM-assisted reward generation and key-horizon multiview representation learning are indispensable for Robo-Horizon's overall performance. In tasks with dispersed objects, LLM-assisted reward generation plays a more significant role, whereas in tasks with densely distributed objects, key-horizon multi-view representation learning is more crucial.

这些发现明确回答了第三个关键问题:LLM 辅助奖励生成和关键时域多视角表示学习对于 Robo-Horizon 的整体性能都是不可或缺的。在物体分散的任务中，LLM 辅助奖励生成起着更重要的作用；而在物体密集分布的任务中，关键时域多视角表示学习更为关键。

## 5 Conclusion and Discussion

## 5 结论与讨论

In this paper, we propose a novel Recognize-Sense-Plan-Act (RSPA) pipeline for long-horizon manipulation tasks. The RSPA pipeline decomposes complex long-horizon tasks into four stages: recognizing tasks, sensing the environment, planning actions, and executing them effectively. Based on this pipeline, we develop RoboHorizon, an LLM-assisted multiview world model. RoboHorizon uses its LLM-assisted reward generation module for accurate task recognition and its key-horizon multi-view representation learning module for comprehensive environment perception. These components enable RoboHorizon to build a robust world model that supports stable task planning and effective action execution through reinforcement learning algorithms. Experiments on RLBench and FurnitureBench show that RoboHorizon significantly outperforms state-of-the-art baselines in long-horizon tasks. Future work will focus on two key areas: enhancing the LLM-assisted reward generation in RoboHorizon by incorporating human feedback to create a closed-loop process that strengthens the framework's adaptability to multiple tasks, and applying the RSPA pipeline and RoboHorizon model to real-world long-horizon robotic manipulation tasks to improve the framework's sim-to-real transfer capabilities.

本文提出了一种新颖的识别-感知-规划-执行 (Recognize-Sense-Plan-Act, RSPA) 流水线，用于长时操作任务。RSPA 流水线将复杂的长时任务分解为四个阶段: 任务识别、环境感知、动作规划和有效执行。基于该流水线，我们开发了 RoboHorizon，一种 LLM 辅助的多视角世界模型。RoboHorizon 利用其 LLM 辅助奖励生成模块实现精准的任务识别，利用关键时域多视角表示学习模块实现全面的环境感知。这些组件使 RoboHorizon 构建了稳健的世界模型，支持通过强化学习算法实现稳定的任务规划和有效的动作执行。在 RLBench 和 FurnitureBench 上的实验表明，RoboHorizon 在长时任务中显著优于最先进的基线方法。未来工作将聚焦两个关键方向: 通过引入人类反馈增强 RoboHorizon 中的 LLM 辅助奖励生成，构建闭环过程以提升框架对多任务的适应性；以及将 RSPA 流水线和 RoboHorizon 模型应用于现实世界的长时机器人操作任务，提升框架的仿真到现实转移能力。

## References

## 参考文献

[Ahn et al., 2022] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. arXiv preprint arXiv:2204.01691, 2022. 2

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. 按我所能，而非我所言: 将语言基础植入机器人可供性。arXiv 预印本 arXiv:2204.01691, 2022. 2

[Akinola et al., 2020] Iretiayo Akinola, Jacob Varley, and Dmitry Kalashnikov. Learning precise 3d manipulation from multiple uncalibrated cameras. In $ICRA$ , pages 4616- 4622. IEEE, 2020. 3

Iretiayo Akinola, Jacob Varley, and Dmitry Kalashnikov. 从多台未校准摄像头学习精确的三维操作。发表于 $ICRA$ ，页 4616-4622。IEEE, 2020. 3

[Chen et al., 2021] Boyuan Chen, Pieter Abbeel, and Deepak Pathak. Unsupervised learning of visual 3 d keypoints for control. In ICLR, pages 1539-1549. PMLR, 2021. 3

Boyuan Chen, Pieter Abbeel, and Deepak Pathak. 无监督学习视觉控制关键点。发表于 ICLR，页 1539-1549。PMLR, 2021. 3

[Chen et al., 2023] Zixuan Chen, Wenbin Li, Yang Gao, and Yiyu Chen. Tild: Third-person imitation learning by estimating domain cognitive differences of visual demonstrations. In AAMAS, pages 2421-2423, 2023. 3

Zixuan Chen, Wenbin Li, Yang Gao, and Yiyu Chen. TILD: 通过估计视觉示范的领域认知差异进行第三人称模仿学习。发表于 AAMAS，页 2421-2423，2023. 3

[Chen et al., 2024a] Yangtao Chen, Zixuan Chen, Junhui Yin, Jing Huo, Pinzhuo Tian, Jieqi Shi, and Yang Gao. Grav-mad: Grounded spatial value maps guided action diffusion for generalized 3 d manipulation. arXiv preprint arXiv:2409.20154, 2024. 2

陈阳涛, 陈子轩, 尹俊辉, 霍晶, 田品卓, 石杰琦, 高扬. Grav-mad: 基于空间价值图引导的动作扩散用于广义 3 d 操作. arXiv 预印本 arXiv:2409.20154, 2024. 2

[Chen et al., 2024b] Zixuan Chen, Ze Ji, Jing Huo, and Yang Gao. Scar: Refining skill chaining for long-horizon robotic manipulation via dual regularization. In NeurIPS, 2024. 3

陈子轩, 纪泽, 霍晶, 高扬. Scar: 通过双重正则化优化长时域机器人操作的技能链. 见 NeurIPS, 2024. 3

[Chiang et al., 2019] Hao-Tien Lewis Chiang, Aleksandra Faust, Marek Fiser, and Anthony Francis. Learning navigation behaviors end-to-end with autorl. IEEE Robotics and Automation Letters, 4(2):2007-2014, 2019. 2

姜浩天, Aleksandra Faust, Marek Fiser, Anthony Francis. 使用 AutoRL 端到端学习导航行为. IEEE 机器人与自动化快报, 4(2):2007-2014, 2019. 2

[Dalal et al., 2021] Murtaza Dalal, Deepak Pathak, and Russ R Salakhutdinov. Accelerating robotic reinforcement learning via parameterized action primitives. NeurIPS, 34:21847-21859, 2021. 1

Murtaza Dalal, Deepak Pathak, Russ R Salakhutdinov. 通过参数化动作原语加速机器人强化学习. NeurIPS, 34:21847-21859, 2021. 1

[Dalal et al., 2024] Murtaza Dalal, Tarun Chiruvolu, Deven-dra Chaplot, and Ruslan Salakhutdinov. Plan-seq-learn: Language model guided rl for solving long horizon robotics tasks. arXiv preprint arXiv:2405.01534, 2024. 1, 3

Murtaza Dalal, Tarun Chiruvolu, Devendra Chaplot, Ruslan Salakhutdinov. Plan-seq-learn: 语言模型引导的强化学习用于解决长时域机器人任务. arXiv 预印本 arXiv:2405.01534, 2024. 1, 3

[Dosovitskiy et al., 2020] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Min-derer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In International Conference on Learning Representations, 2020. 4

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly 等. 一张图像相当于 16x16 个词: 大规模图像识别的 Transformer. 国际学习表征会议, 2020. 4

[Feichtenhofer et al., 2022] Christoph Feichtenhofer, Yang-hao Li, Kaiming He, et al. Masked autoencoders as spatiotemporal learners. NeurIPS, 35:35946-35958, 2022. 4

Christoph Feichtenhofer, Yanghao Li, Kaiming He 等. 掩码自编码器作为时空学习者. NeurIPS, 35:35946-35958, 2022. 4

[Garrett et al., 2020] Caelan Reed Garrett, Chris Paxton, Tomás Lozano-Pérez, Leslie Pack Kaelbling, and Dieter Fox. Online replanning in belief space for partially observable task and motion problems. In ICRA, pages 5678-5684. IEEE, 2020. 3

Caelan Reed Garrett, Chris Paxton, Tomás Lozano-Pérez, Leslie Pack Kaelbling, Dieter Fox. 在信念空间中进行在线重新规划以解决部分可观测的任务与运动问题. ICRA 会议, 页 5678-5684. IEEE, 2020. 3

[Garrett et al., 2021] Caelan Reed Garrett, Rohan Chitnis, Rachel Holladay, Beomjoon Kim, Tom Silver, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. Integrated task and motion planning. Annual review of control, robotics, and autonomous systems, 4(1):265-293, 2021. 3

Caelan Reed Garrett, Rohan Chitnis, Rachel Holladay, Beomjoon Kim, Tom Silver, Leslie Pack Kaelbling, Tomás Lozano-Pérez. 集成任务与运动规划. 控制、机器人与自主系统年评, 4(1):265-293, 2021. 3

[Goyal et al., 2023] Ankit Goyal, Jie Xu, Yijie Guo, Valts Blukis, Yu-Wei Chao, and Dieter Fox. Rvt: Robotic view transformer for 3 d object manipulation. In *CoRL* , pages 694-710. PMLR, 2023. 2, 4

Ankit Goyal, Jie Xu, Yijie Guo, Valts Blukis, Yu-Wei Chao, Dieter Fox. RVT: 用于 3 d 物体操作的机器人视图变换器. 见 *CoRL* , 页 694-710. PMLR, 2023. 2, 4

[Hafner et al., 2019] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In ICLR, pages 2555-2565. PMLR, 2019. 4

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, James Davidson. 从像素学习潜在动力学以进行规划. ICLR 会议, 页 2555-2565. PMLR, 2019. 4

[Hafner et al., 2021] Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. In International Conference on Learning Representations, 2021. 5

Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, Jimmy Ba. 使用离散世界模型掌握 Atari 游戏. 国际学习表征会议, 2021. 5

[He et al., 2021] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. arXiv preprint arXiv:2111.06377, 2021. 2, 6

何凯明, 陈新磊, 谢赛宁, 李阳昊, Piotr Dollár, Ross Girshick. 掩码自编码器是可扩展的视觉学习者. arXiv 预印本 arXiv:2111.06377, 2021. 2, 6

[Heo et al., 2023] Minho Heo, Youngwoon Lee, Doohyun Lee, and Joseph J. Lim. Furniturebench: Reproducible real-world benchmark for long-horizon complex manipulation. In Robotics: Science and Systems, 2023. 2, 5, 13

Minho Heo, Youngwoon Lee, Doohyun Lee 和 Joseph J. Lim。Furniturebench: 可复现的现实世界长时复杂操作基准。在《机器人学: 科学与系统》，2023 年。2, 5, 13

[Hsu et al., 2022] Kyle Hsu, Moo Jin Kim, Rafael Rafailov, Jiajun Wu, and Chelsea Finn. Vision-based manipulators need to also see from their hands. arXiv preprint arXiv:2203.12677, 2022. 3

Kyle Hsu, Moo Jin Kim, Rafael Rafailov, Jiajun Wu 和 Chelsea Finn。基于视觉的机械臂也需要从手部视角观察。arXiv 预印本 arXiv:2203.12677, 2022 年。3

[Huang et al., 2022] Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In ICLR, pages 9118-9147. PMLR, 2022. 2

Wenlong Huang, Pieter Abbeel, Deepak Pathak 和 Igor Mordatch。语言模型作为零样本规划器: 为具身智能体提取可执行知识。在 ICLR，页 9118-9147。PMLR, 2022 年。2

[Huang et al., 2023] Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models. In CoRL, pages 540-562. PMLR, 2023. 2

Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu 和 Li Fei-Fei。Voxposer: 用于机器人操作的可组合三维价值图与语言模型。在 CoRL，页 540-562。PMLR, 2023 年。2

[James and Davison, 2022] Stephen James and Andrew J Davison. Q-attention: Enabling efficient learning for vision-based robotic manipulation. IEEE Robotics and Automation Letters, 7(2):1612-1619, 2022. 2, 4, 5

Stephen James 和 Andrew J Davison。Q-attention: 实现基于视觉的机器人操作的高效学习。IEEE 机器人与自动化快报，7(2):1612-1619, 2022 年。2, 4, 5

[James et al.,2020] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J Davison. Rl-bench: The robot learning benchmark & learning environment. IEEE Robotics and Automation Letters, 5(2):3019-3026, 2020. 2, 5, 13

Stephen James, Zicong Ma, David Rovick Arrojo 和 Andrew J Davison。RL-bench: 机器人学习基准与学习环境。IEEE 机器人与自动化快报，5(2):3019-3026, 2020 年。2, 5, 13

[Kaelbling and Lozano-Pérez, 2013] Leslie Pack Kaelbling and Tomás Lozano-Pérez. Integrated task and motion planning in belief space. The International Journal of Robotics Research, 32(9-10):1194-1227, 2013. 3

Leslie Pack Kaelbling 和 Tomás Lozano-Pérez。信念空间中的集成任务与运动规划。《国际机器人研究杂志》，32(9-10):1194-1227, 2013 年。3

[Ke et al., 2024] Tsung-Wei Ke, Nikolaos Gkanatsios, and Katerina Fragkiadaki. 3d diffuser actor: Policy diffusion with 3d scene representations. arXiv preprint arXiv:2402.10885, 2024. 2, 3, 4

Tsung-Wei Ke, Nikolaos Gkanatsios 和 Katerina Fragkiadaki。3D 扩散演员: 基于三维场景表示的策略扩散。arXiv 预印本 arXiv:2402.10885, 2024 年。2, 3, 4

[Kingma and Welling, 2014] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In International Conference on Learning Representations, 2014. 5

> Diederik P Kingma 和 Max Welling。自动编码变分贝叶斯。在国际学习表征会议，2014 年。5

[Liang et al., 2023] Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. In ICRA, pages 9493-9500. IEEE, 2023. 2

> Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence 和 Andy Zeng。代码即策略: 用于具身控制的语言模型程序。在 ICRA，页 9493-9500。IEEE, 2023 年。2

[Mahler et al., 2016] Jeffrey Mahler, Florian T Pokorny, Brian Hou, Melrose Roderick, Michael Laskey, Mathieu Aubry, Kai Kohlhoff, Torsten Kröger, James Kuffner, and Ken Goldberg. Dex-net 1.0: A cloud-based network of 3d objects for robust grasp planning using a multi-armed bandit model with correlated rewards. In ICRA, pages 1957-1964. IEEE, 2016. 3

> Jeffrey Mahler, Florian T Pokorny, Brian Hou, Melrose Roderick, Michael Laskey, Mathieu Aubry, Kai Kohlhoff, Torsten Kröger, James Kuffner 和 Ken Goldberg。Dex-net 1.0: 基于云的三维物体网络，利用带相关奖励的多臂赌博机模型实现鲁棒抓取规划。在 ICRA，页 1957-1964。IEEE, 2016 年。3

[Marton, 1984] J. Marton. Scientific fundamentals of robotics 1. dynamics of manipulation robots: Theory and application: Edited by miomir vukobratovic and veljko potkonjak. Autom., 20(2):265-266, 1984. 1, 3

> J. Marton。机器人学科学基础 1: 操作机器人动力学——理论与应用。Miomir Vukobratovic 和 Veljko Potkonjak 编著。自动化，20(2):265-266, 1984 年。1, 3

[Mason, 2001] Matthew T Mason. Mechanics of robotic manipulation. MIT press, 2001. 3

> Matthew T Mason。机器人操作力学。麻省理工学院出版社，2001 年。3

[Miller and Allen, 2004] Andrew T Miller and Peter K Allen. Graspit! a versatile simulator for robotic grasping. IEEE Robotics & Automation Magazine, 11(4):110-122, 2004. 3

> Andrew T Miller 和 Peter K Allen。GraspIt!: 多功能机器人抓取仿真器。IEEE 机器人与自动化杂志，11(4):110-122, 2004 年。3

[Murphy, 2019] Robin R Murphy. Introduction to AI robotics. MIT press, 2019. 1, 3

> Robin R Murphy。人工智能机器人学导论。麻省理工学院出版社，2019 年。1, 3

[Paul, 1981] Richard P Paul. Robot manipulators: mathematics, programming, and control: the computer control of robot manipulators. Richard Paul, 1981. 1, 3

> Richard P Paul。机器人机械臂: 数学、编程与控制——机器人机械臂的计算机控制。Richard Paul, 1981 年。1, 3

[Radford et al., 2021] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In $ICLR$, 2021.6

> Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark 等。通过自然语言监督学习可迁移的视觉模型。发表于 $ICLR$, 2021.6

[Schulman et al., 2015] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. arXiv preprint arXiv:1506.02438, 2015. 5

> John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan 和 Pieter Abbeel。使用广义优势估计的高维连续控制。arXiv 预印本 arXiv:1506.02438, 2015. 5

[Seo et al., 2023a] Younggyo Seo, Danijar Hafner, Hao Liu, Fangchen Liu, Stephen James, Kimin Lee, and Pieter Abbeel. Masked world models for visual control. In CoRL, pages 1332-1344. PMLR, 2023. 4, 5, 6

> Younggyo Seo, Danijar Hafner, Hao Liu, Fangchen Liu, Stephen James, Kimin Lee 和 Pieter Abbeel。用于视觉控制的掩码世界模型。发表于 CoRL, 页码 1332-1344。PMLR, 2023. 4, 5, 6

[Seo et al., 2023b] Younggyo Seo, Junsu Kim, Stephen James, Kimin Lee, Jinwoo Shin, and Pieter Abbeel. Multiview masked world models for visual robotic manipulation. In ICLR, pages 30613-30632. PMLR, 2023. 2, 3, 4, 5, 6, 13

> Younggyo Seo, Junsu Kim, Stephen James, Kimin Lee, Jinwoo Shin 和 Pieter Abbeel。用于视觉机器人操作的多视角掩码世界模型。发表于 ICLR, 页码 30613-30632。PMLR, 2023. 2, 3, 4, 5, 6, 13

[Sermanet et al., 2018] Pierre Sermanet, Corey Lynch, Yev-gen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, and Sergey Levine. Time-contrastive networks: Self-supervised learning from video. In $ICRA$, 2018.3, 6

> Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal 和 Sergey Levine。时间对比网络: 基于视频的自监督学习。发表于 $ICRA$, 2018.3, 6

[Shridhar et al., 2023] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In CoRL, pages 785-799. PMLR, 2023.2,3,4

> Mohit Shridhar, Lucas Manuelli 和 Dieter Fox。Perceiver-actor: 用于机器人操作的多任务变换器。发表于 CoRL, 页码 785-799。PMLR, 2023.2,3,4

[Snell et al., 2022] Charlie Snell, Mengjiao Yang, Justin Fu, Yi Su, and Sergey Levine. Context-aware language modeling for goal-oriented dialogue systems. arXiv preprint arXiv:2204.10198, 2022. 2

> Charlie Snell, Mengjiao Yang, Justin Fu, Yi Su 和 Sergey Levine。面向目标对话系统的上下文感知语言建模。arXiv 预印本 arXiv:2204.10198, 2022. 2

[Sundermeyer et al., 2021] Martin Sundermeyer, Arsalan Mousavian, Rudolph Triebel, and Dieter Fox. Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes. In ICRA, pages 13438-13444. IEEE, 2021. 3

Martin Sundermeyer, Arsalan Mousavian, Rudolph Triebel 和 Dieter Fox。Contact-graspnet: 在杂乱场景中高效生成六自由度抓取。发表于 ICRA, 页码 13438-13444。IEEE, 2021. 3

[Sutton et al., 1999] Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. NeurIPS, 12, 1999. 3

Richard S Sutton, David McAllester, Satinder Singh 和 Yishay Mansour。带函数逼近的强化学习策略梯度方法。NeurIPS, 12, 1999. 3

[Taylor et al., 1987] Russ H Taylor, Matthew T Mason, and Kenneth Y Goldberg. Sensor-based manipulation planning as a game with nature. In Fourth International Symposium on Robotics Research, pages 421-429, 1987. 3

Russ H Taylor, Matthew T Mason 和 Kenneth Y Goldberg。基于传感器的操作规划作为与自然的博弈。发表于第四届国际机器人研究研讨会, 页码 421-429, 1987. 3

[Tong et al., 2022] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. NeurIPS, 35:10078-10093, 2022. 4

Zhan Tong, Yibing Song, Jue Wang 和 Limin Wang。VideoMAE: 掩码自编码器是高效的数据驱动自监督视频预训练方法。NeurIPS, 35:10078-10093, 2022. 4

[Whitney, 2004] Daniel E Whitney. Mechanical assemblies: their design, manufacture, and role in product development, volume 1. Oxford university press New York, 2004. 3

Daniel E Whitney。机械组件: 设计、制造及其在产品开发中的作用，第一卷。牛津大学出版社，纽约，2004. 3

[Yamada et al., 2021] Jun Yamada, Youngwoon Lee, Gautam Salhotra, Karl Pertsch, Max Pflueger, Gaurav Sukhatme, Joseph Lim, and Peter Englert. Motion planner augmented reinforcement learning for robot manipulation in obstructed environments. In CoRL, pages 589-603. PMLR, 2021. 1

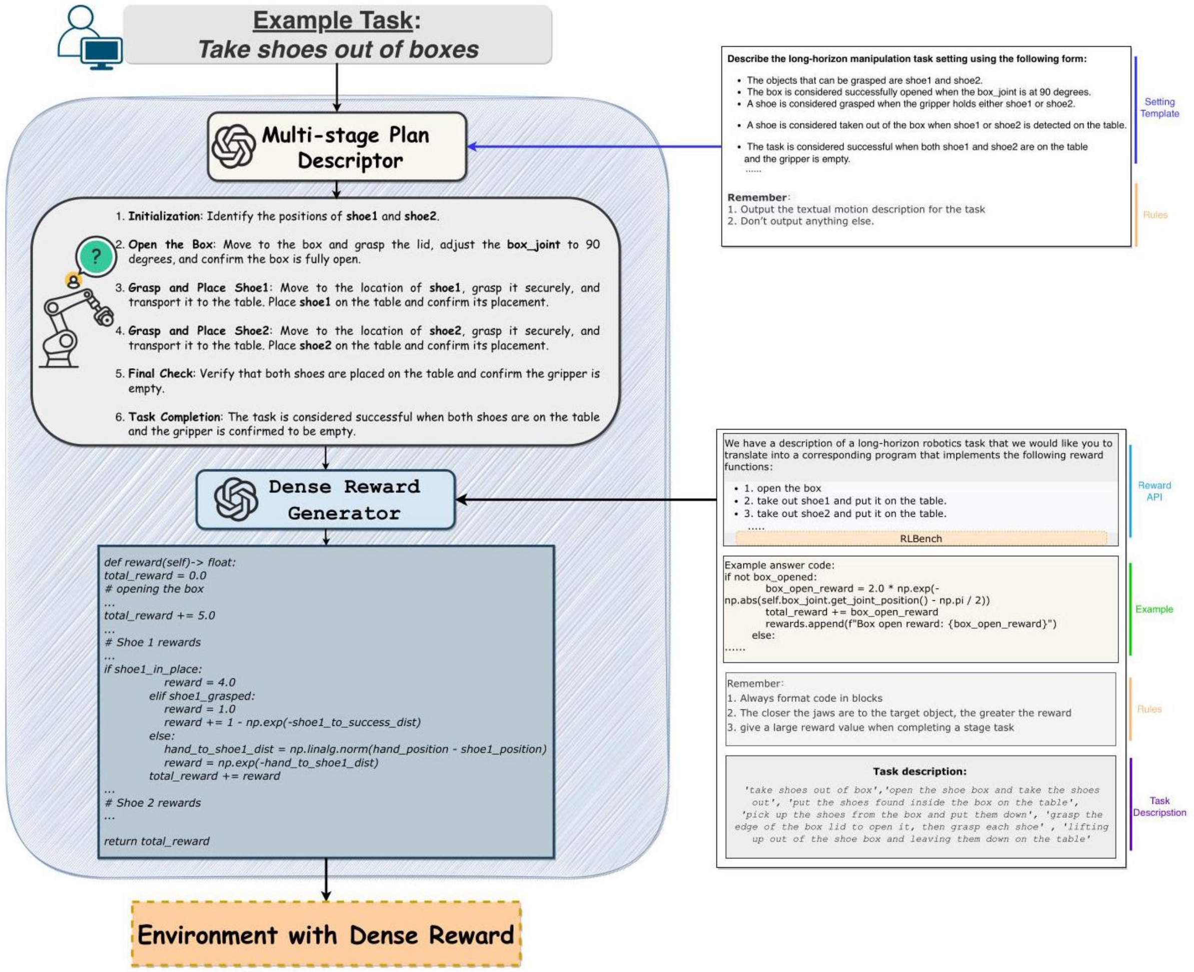
Jun Yamada, Youngwoon Lee, Gautam Salhotra, Karl Pertsch, Max Pflueger, Gaurav Sukhatme, Joseph Lim 和 Peter Englert。用于障碍环境中机器人操作的运动规划增强强化学习。发表于 CoRL, 页码 589-603。PMLR, 2021. 1

[Yu et al., 2023] Wenhao Yu, Nimrod Gileadi, Chuyuan Fu, Sean Kirmani, Kuang-Huei Lee, Montserrat Gonzalez Arenas, Hao-Tien Lewis Chiang, Tom Erez, Leonard Hasen-clever, Jan Humplik, et al. Language to rewards for robotic skill synthesis. In CoRL, pages 374-404. PMLR, 2023. 2, 4

Wenhao Yu, Nimrod Gileadi, Chuyuan Fu, Sean Kirmani, Kuang-Huei Lee, Montserrat Gonzalez Arenas, Hao-Tien Lewis Chiang, Tom Erez, Leonard Hasenclever, Jan Humplik 等。用于机器人技能合成的语言到奖励。发表于 CoRL, 页码 374-404。PMLR, 2023. 2, 4

[Zeng et al., 2022] Andy Zeng, Maria Attarian, Brian Ichter, Krzysztof Choromanski, Adrian Wong, Stefan Welker, Federico Tombari, Aveek Purohit, Michael Ryoo, Vikas Sindhwani, et al. Socratic models: Composing zero-shot multimodal reasoning with language. arXiv preprint arXiv:2204.00598, 2022. 2

Andy Zeng, Maria Attarian, Brian Ichter, Krzysztof Choromanski, Adrian Wong, Stefan Welker, Federico Tombari, Aveek Purohit, Michael Ryoo, Vikas Sindhwani 等. Socratic 模型: 利用语言组合零样本多模态推理。arXiv 预印本 arXiv:2204.00598, 2022. 2



1. **Initialization**: Identify the positions of **shoe1** and **shoe2**.
2. **Open the Box**: Move to the box and grasp the lid, adjust the **box_joint** to 90 degrees, and confirm the box is fully open.
3. **Grasp and Place Shoe1**: Move to the location of **shoe1**, grasp it securely, and transport it to the table. Place **shoe1** on the table and confirm its placement.
4. **Grasp and Place Shoe2**: Move to the location of **shoe2**, grasp it securely, and transport it to the table. Place **shoe2** on the table and confirm its placement.
5. **Final Check**: Verify that both shoes are placed on the table and confirm the gripper is empty.
6. **Task Completion**: The task is considered successful when both shoes are on the table and the gripper is confirmed to be empty.

```
def reward(self)-> float:
total_reward = 0.0
# opening the box
...
total_reward += 5.0
...
# Shoe 1 rewards
...
if shoe1_in_place:
            reward = 4.0
        elif shoe1_grasped:
            reward = 1.0
            reward += 1 - np.exp(-shoe1_to_success_dist)
        else:
            hand_to_shoe1_dist = np.linalg.norm(hand_position - shoe1_position)
            reward = np.exp(-hand_to_shoe1_dist)
        total_reward += reward
...
# Shoe 2 rewards
...

return total_reward
```

**Describe the long-horizon manipulation task setting using the following form:**

- The objects that can be grasped are shoe1 and shoe2.
- The box is considered successfully opened when the box_joint is at 90 degrees.
- A shoe is considered grasped when the gripper holds either shoe1 or shoe2.
- A shoe is considered taken out of the box when shoe1 or shoe2 is detected on the table.
- The task is considered successful when both shoe1 and shoe2 are on the table and the gripper is empty.

……

**Remember**:
1. Output the textual motion description for the task
2. Don't output anything else.

We have a description of a long-horizon robotics task that we would like you to translate into a corresponding program that implements the following reward functions:

- 1. open the box
- 2. take out shoe1 and put it on the table.
- 3. take out shoe2 and put it on the table.

.....

RLBench

```
Example answer code:
if not box_opened:
            box_open_reward = 2.0 * np.exp(-
np.abs(self.box_joint.get_joint_position() - np.pi / 2))
            total_reward += box_open_reward
            rewards.append(f"Box open reward: {box_open_reward}")
        else:
......
```

Remember:
1. Always format code in blocks
2. The closer the jaws are to the target object, the greater the reward
3. give a large reward value when completing a stage task

**Task description:**

'take shoes out of box','open the shoe box and take the shoes out', 'put the shoes found inside the box on the table', 'pick up the shoes from the box and put them down', 'grasp the edge of the box lid to open it, then grasp each shoe' , 'lifting up out of the shoe box and leaving them down on the table'

Figure 7: Detailed data flow of the LLM-assisted Reward Generation, the take shoes out of box task as the illustration example.

图 7:LLM 辅助奖励生成的详细数据流，以“从盒子中取出鞋子”任务为示例说明。

# A Technical Appendix

# 技术附录

In this technical appendix, we first provide a detailed example illustration of the data flow in the LLM-assisted Reward Generation (the Recognize part) and the two-stage (Multi-stage Plan Descriptor and Dense Reward Generator) prompts examples. Then, we present a detailed introduction to the RLBench and FurnitureBench benchmark used in the experiments, along with the related tasks.

在本技术附录中，我们首先详细展示 LLM 辅助奖励生成 (识别部分) 和两阶段 (多阶段计划描述器与密集奖励生成器) 提示示例中的数据流示意。随后，介绍实验中使用的 RLBench 和 FurnitureBench 基准及相关任务。

## A.1 Detailed LLM-Assisted Reward Generation

## A.1 详细的 LLM 辅助奖励生成

As shown in Fig. 7, the LLM-assisted Reward Generation process is divided into two stages. We use the long-horizon task ”take shoes out of the box” from RLBench as a demonstration to illustrate the two stages.

如图 7 所示，LLM 辅助奖励生成过程分为两个阶段。我们以 RLBench 中的长时序任务“从盒子中取出鞋子”为示例，说明这两个阶段。

In Stage 1, we employ a pre-trained LLM as the Multi-stage Plan Descriptor that interprets and expands user input into detailed language descriptions of the required robot motions using predefined templates. To enable the multi-stage plan descriptor to generate a coherent structure for long-horizon tasks, we create a prompt template that outlines the current robot task settin: This leverages the pre-trained LLM’s internal knowledge of motion planning to produce detailed motion descriptions.

第一阶段，我们采用预训练的大型语言模型 (LLM) 作为多阶段计划描述器，利用预定义模板将用户输入解读并扩展为详细的机器人动作语言描述。为使多阶段计划描述器能生成连贯的长时序任务结构，我们设计了一个提示模板，概述当前机器人任务环境: 此举利用了预训练 LLM 对运动规划的内在知识，生成详尽的动作描述。

In Stage 2, we introduce another LLM as the Dense Reward Generator, which translates motion descriptions into corresponding dense reward structures. Specifically, for each sub-task stage, the model generates smaller stepwise rewards (e.g., rewards for each action during the grasping of shoe1) and larger staged rewards (e.g., rewards for successfully grasping shoe1). This process is treated as a coding task, leveraging the pre-trained LLM’s deep understanding of code and its structure. The Dense Reward Generator is guided by four types of prompts to generate reward code: i) task stage descriptions based on the task environment interface, ii) examples of expected outputs from the reward generator, iii) constraints and rules for the reward encoder, and iv) specific task descriptions.

第二阶段，我们引入另一 LLM 作为密集奖励生成器，将动作描述转化为对应的密集奖励结构。具体而言，对于每个子任务阶段，模型生成较小的逐步奖励 (例如抓取鞋子 1 时每个动作的奖励) 和较大的阶段奖励 (例如成功抓取鞋子 1 的奖励)。该过程视为编码任务，利用预训练 LLM 对代码及其结构的深刻理解。密集奖励生成器由四类提示引导生成奖励代码:i) 基于任务环境接口的任务阶段描述，ii) 奖励生成器预期输出示例，iii) 奖励编码器的约束与规则，iv) 具体任务描述。

We provide a detailed two-stage data flow for the ”take shoes out of the box” task, along with the related prompts, the plans generated in Stage 1 and the reward codes generated in Stage 2. Data Flow for Stage 1

我们提供“从盒子中取出鞋子”任务的详细两阶段数据流，以及相关提示、第一阶段生成的计划和第二阶段生成的奖励代码。第一阶段数据流

Prompt: Describe the long-horizon manipulation task ”take shoes out of the box” using the following format:

提示: 使用以下格式描述长时序操作任务“从盒子中取出鞋子”:

- Graspable objects are shoe1 and shoe2.

  - 可抓取的物体为鞋子 1 和鞋子 2。

- The box is considered successfully opened when the box_joint is at 90 degrees.

  - 当盒子关节 (box_joint) 达到 90 度时，视为盒子成功打开。

- A shoe is considered grasped when the gripper holds either shoe1 or shoe2.

  - 当夹爪抓住鞋子 1 或鞋子 2 时，视为鞋子被抓取。

- A shoe is considered out of the box when it is detected on the table.

  - 当鞋子被检测到放置在桌面上时，视为鞋子已取出盒子。

- The task is considered successful when both shoe1 and shoe2 are placed on the table.

  - 当鞋子 1 和鞋子 2 均放置在桌面上时，任务视为成功完成。

Remember to:

请记住:

- Provide only the motion description for the task.

  - 仅提供任务的动作描述。

- Exclude any other information.

  - 排除任何其他信息。

Plan: [Motion Description for the Task]

计划:[任务的动作描述]

1. Initialize: Identify shoel and shoe2.

1. 初始化: 识别鞋子 1 和鞋子 2。

2. Open the Box: Move towards the box and adjust the box-joint to 90 degrees. Confirm the box is open.

2. 打开盒子: 移动到盒子旁，将盒子连接处调整到 90 度。确认盒子已打开。

3. Grasp and Place Shoe1: Move towards shoe1, grasp it, move to the table, and place shoe1 on the table. Confirm shoe1 is on the table.

3. 抓取并放置鞋子 1: 移动到鞋子 1 处，抓取它，移动到桌子上，将鞋子 1 放在桌子上。确认鞋子 1 已放置在桌子上。

4. Grasp and Place Shoe2: Move towards shoe2, grasp it, move to the table, and place shoe2 on the table. Confirm shoe2 is on the table.

4. 抓取并放置鞋子 2: 移动到鞋子 2 处，抓取它，移动到桌子上，将鞋子 2 放在桌子上。确认鞋子 2 已放置在桌子上。

5. Final Check: Ensure both shoes are on the table and the gripper is empty.

5. 最终检查: 确保两只鞋都在桌子上，夹持器为空。

6. Task Completion: The task is successful when both shoes are placed on the table and the gripper is empty.

6. 任务完成: 当两只鞋都放置在桌子上且夹持器为空时，任务成功。

## Data Flow for Stage 2

## 第二阶段数据流

Prompt: We have a description of a long-horizon robotics task ”take shoes out of the box” that we would like you to translate into a corresponding code that implements the following reward functions:

提示: 我们有一个关于“将鞋子从盒子中取出”的长周期机器人任务描述，希望你将其翻译成实现以下奖励函数的相应代码:

- Open the Box: Move towards the box and adjust the box joint to 90 degrees. Confirm the box is open.

- 打开盒子: 移动到盒子旁，将盒子连接处调整到 90 度。确认盒子已打开。

- Grasp and Place Shoe1: Move towards shoe1, grasp it, move to the table, and place shoe1 on the table. Confirm shoe1 is on the table.

- 抓取并放置鞋子 1: 移动到鞋子 1 处，抓取它，移动到桌子上，将鞋子 1 放在桌子上。确认鞋子 1 已放置在桌子上。

- Grasp and Place Shoe2: Move towards shoe2, grasp it, move to the table, and place shoe2 on the table. Confirm shoe2 is on the table.

- 抓取并放置鞋子 2: 移动到鞋子 2 处，抓取它，移动到桌子上，将鞋子 2 放在桌子上。确认鞋子 2 已放置在桌子上。

- Final Check: Ensure both shoes are on the table and the gripper is empty.

- 最终检查: 确保两只鞋都在桌子上，夹持器为空。

- Task Completion: The task is successful when both shoes are placed on the table and the gripper is empty.

- 任务完成: 当两只鞋都放置在桌子上且夹持器为空时，任务成功。

## Example answer code:

## 示例答案代码:

---

```
if not box_opened:
```

如果盒子未打开:

```
    box_open_reward = 2.0 * np.exp(-np.abs(self.box_joint_get_joint_position( ) - np.pi / 2))
```

```
  box_open_reward = 2.0 * np.exp(-np.abs(self.box_joint_get_joint_position( ) - np.pi / 2))
```

```
else:
```

```
else:
```

```
    total_reward + = 5.0 # Box opened reward
```

```
  total_reward + = 5.0 # 盒子开启奖励
```

```
    rewards.append(”Box opened reward: 5.0”)
```

```
  rewards.append(” 盒子开启奖励: 5.0”)
```

---

## Remember:

## 记住:

- Always format code in blocks

- 始终将代码格式化为代码块

---

- The closer the jaws are to the target object, the greater the reward

- 夹爪越接近目标物体，奖励越高

---

- Give a large reward value when completing a stage task Task description:

- 完成阶段任务时给予较大奖励任务描述:

’take shoes out of box’, ’open the shoe box and take the shoes out’, ’put the shoes found inside the box on the table’, ’pick up the shoes from the box and put them down’, ’grasp the edge of the box lid to open it, then grasp each shoe’, ’lift them up out of the shoe box and leave them down on the table’ Generated Rewards:

’ 从盒子里取出鞋子’, ’ 打开鞋盒并取出鞋子’, ’ 将盒子内找到的鞋子放到桌子上’, ’ 从盒子里拿起鞋子并放下’, ’ 抓住盒盖边缘打开盒子，然后抓住每只鞋子’, ’ 将鞋子从鞋盒中提起并放到桌子上’ 生成的奖励:

---

```
def reward(self) -> float:
```

def reward(self) -> float:

```
    rewards = []
```

rewards = []

```
    total_reward = 0.0
```

total_reward = 0.0

```
    box_opened = self._opened_cond_box.condition_met( )[0]
```

box_opened = self._opened_cond_box.condition_met( )[0]

```
    #Reward for opening the box
```

# 开盒奖励

```
    if not box_opened:
```

if not box_opened:

```
        box_open_reward = 2.0 * np.exp(-np.abs(self.box_joint.get_joint_position( ) - np.pi / 2))
```

box_open_reward = 2.0 * np.exp(-np.abs(self.box_joint.get_joint_position( ) - np.pi / 2))

```
        total_reward += box_open_reward
```

```
    total_reward += box_open_reward

      rewards.append(f”Box open reward: {box_open_reward}”)

    rewards.append(f” 开箱奖励: {box_open_reward}”)

   else:

  else:

      total_reward + = 5.0 # Box opened reward

    total_reward + = 5.0 # 开箱奖励

      rewards.append(”Box opened reward: 5.0”)

    rewards.append(” 开箱奖励: 5.0”)

   #Shoe 1 rewards

  # 鞋子 1 奖励

   shoe1_grasped = self._grasped_cond_shoe1.condition_met( )[0]

  shoe1_grasped = self._grasped_cond_shoe1.condition_met( )[0]

   shoel_in_place = self._detected_cond_shoel.condition_met( )[0]

  shoel_in_place = self._detected_cond_shoel.condition_met( )[0]

   hand_position = self.robot.arm.get_tip( ).get_position( )

  hand_position = self.robot.arm.get_tip( ).get_position( )

   shoel_to_success_dist = np.linalg.norm(shoel_position - self.success_sensor.get_position( ))

  shoel_to_success_dist = np.linalg.norm(shoel_position - self.success_sensor.get_position( ))

   if shoe1_in_place:

  if shoe1_in_place:

      reward = 4.0

    reward = 4.0

      rewards.append(f”Shoe1 in place reward: 4.0”)
```

```
        rewards.append(f’ 鞋子 1 到位奖励: 4.0”)
    elif shoe1_grasped:
    elif shoe1_grasped:
        reward = 1.0
        reward = 1.0
        rewards.append(f’Shoe1 grasped reward:  1.0”)
        rewards.append(f’ 鞋子 1 抓取奖励: 1.0”)
        reward += 1 - np.exp(-shoel_to_success_dist)
        reward += 1 - np.exp(-shoel_to_success_dist)
        rewards.append(f’Shoe1 moving towards target reward:  {1 - np.exp(-shoe1_to_success_dist)}”)
        rewards.append(f’ 鞋子 1 向目标移动奖励: {1 - np.exp(-shoe1_to_success_dist)}”)
        hand_to_shoe1_dist = np.linalg.norm(hand_position - shoe1_position)
        hand_to_shoe1_dist = np.linalg.norm(hand_position - shoe1_position)
        reward = np.exp(-hand_to_shoe1_dist)
        reward = np.exp(-hand_to_shoe1_dist)
        rewards.append(f’Shoe1 approach reward:  {np.exp(-hand_to_shoe1_dist)}”)
        rewards.append(f’ 鞋子 1 接近奖励: {np.exp(-hand_to_shoe1_dist)}”)
    total_reward += reward
    total_reward += reward
    #Shoe 2 rewards
    # 鞋子 2 奖励
    shoe2_grasped = self._grasped_cond_shoe2.condition_met( )[0]
    shoe2_grasped = self._grasped_cond_shoe2.condition_met( )[0]
    shoe2_in_place = self._detected_cond_shoe2.condition_met( )[0]
```

```
shoe2_in_place = self._detected_cond_shoe2.condition_met( )[0]
  shoe2_position = self.shoe2.get_position( )
shoe2_position = self.shoe2.get_position( )
  shoe2_to_success_dist = np.linalg.norm(shoe2_position - self.success_sensor.get_position( ))
shoe2_to_success_dist = np.linalg.norm(shoe2_position - self.success_sensor.get_position( ))
  if shoe2_in_place:
if shoe2_in_place:
      reward = 4.0
    reward = 4.0
      rewards.append(f’Shoe2 in place reward: 4.0”)
    rewards.append(f’鞋子2到位奖励: 4.0”)
  elif shoe2_grasped:
elif shoe2_grasped:
      reward = 1.0
    reward = 1.0
      rewards.append(f’Shoe2 grasped reward: 1.0”)
    rewards.append(f’Shoe2抓取奖励: 1.0”)
      reward += 1 - np.exp(-shoe2_to_success_dist)
    reward += 1 - np.exp(-shoe2_to_success_dist)
      rewards.append(f’Shoe2 moving towards target reward: {1 - np.exp(-shoe2_to_success_dist)}”)
    rewards.append(f’Shoe2向目标移动奖励: {1 - np.exp(-shoe2_to_success_dist)}”)
  else:
else:
      hand_to_shoe2_dist = np.linalg.norm(hand_position - shoe2_position)
```

```
        hand_to_shoe2_dist = np.linalg.norm(hand_position - shoe2_position)
```

```
        rewards.append(f’Shoe2 approach reward: {np.exp(-hand_to_shoe2_dist)}”)
```

```
        rewards.append(f’Shoe2 接近奖励: {np.exp(-hand_to_shoe2_dist)}”)
```

```
    total_reward += reward
```

```
    total_reward += reward
```

```
    return total_reward
```

```
    return total_reward
```

---

## A.2 Experimental Details

## A.2 实验细节

Simulation environment We use the RLBench [James et al., 2020] and FurnitureBench [Heo et al., 2023] simulator. In the RLBench environment, we conduct experiments using a 7-DoF Franka Panda robot arm equipped with a parallel gripper on 4 short-horizon and 6 long-horizon visual manipulation tasks. In the FurnitureBench environment, we perform experiments with the same robot configuration on 3 long-horizon furniture assembly tasks.

仿真环境我们使用 RLBench [James et al., 2020] 和 FurnitureBench [Heo et al., 2023] 仿真器。在 RLBench 环境中，我们使用配备平行夹爪的 7 自由度 Franka Panda 机械臂，在 4 个短期视觉操作任务和 6 个长期视觉操作任务上进行实验。在 FurnitureBench 环境中，我们使用相同的机器人配置，在 3 个长期家具组装任务上进行实验。

Data collection To achieve key-horizon multi-view representation learning and policy learning that combines reinforcement learning with behavior cloning, we first collect expert data across both types of simulation tasks. For demonstration data collection in RLBench, we double the maximum velocity of the Franka Panda robot arm in PyRep [?], which reduces the duration of the demonstrations without significantly compromising their quality. For each short-horizon task, we use RLBench’s dataset generator to collect 50 demonstration trajectories per camera view, and for each long-horizon task, we collect 100 demonstration trajectories per camera view. For data collection in the FurnitureBench tasks, we utilize the automated furniture assembly scripts provided by the platform to automate the data collection process. Similarly, for each long-horizon furniture assembly task, we collect 100 demonstration trajectories per camera view.

数据收集为实现关键时域多视角表示学习及结合强化学习与行为克隆的策略学习，我们首先收集两类仿真任务的专家数据。对于 RLBench 的示范数据收集，我们在 PyRep [?] 中将 Franka Panda 机械臂的最大速度加倍，从而缩短示范时长且不显著降低示范质量。对于每个短期任务，我们使用 RLBench 的数据集生成器收集每个摄像头视角下 50 条示范轨迹，长期任务则收集每个视角 100 条示范轨迹。对于 FurnitureBench 任务，我们利用平台提供的自动家具组装脚本实现数据收集自动化。同样地，对于每个长期家具组装任务，我们收集每个摄像头视角下 100 条示范轨迹。

Implementation Our implementation is built on the official MV-MWM [Seo et al., 2023b] framework, and unless otherwise specified, the implementation details remain the same. To expedite training and mitigate the bottleneck caused by a slow simulator, we run 8 parallel simulators. Our autoencoder is composed of an 8-layer ViT encoder and a 6-layer ViT decoder, with an embedding dimension set to 256 . We maintain a consistent set of hyperparameters across all experiments.

实现我们的实现基于官方 MV-MWM [Seo et al., 2023b] 框架，除非另有说明，其他实现细节保持不变。为加快训练并缓解仿真器速度慢的瓶颈，我们运行 8 个并行仿真器。我们的自编码器由 8 层 ViT 编码器和 6 层 ViT 解码器组成，嵌入维度设为 256。所有实验中保持一致的超参数设置。

Computing hardware For all RLBench experiments, we use a single NVIDIA GeForce RTX 4090 GPU with 24GB VRAM and it takes 12 hours for training MV-RoboWM and 16 hours for training MV-MWM.

计算硬件对于所有 RLBench 实验, 我们使用单块配备 24GB 显存的 NVIDIA GeForce RTX 4090 GPU, 训练 MV-RoboWM 需 12 小时，训练 MV-MWM 需 16 小时。

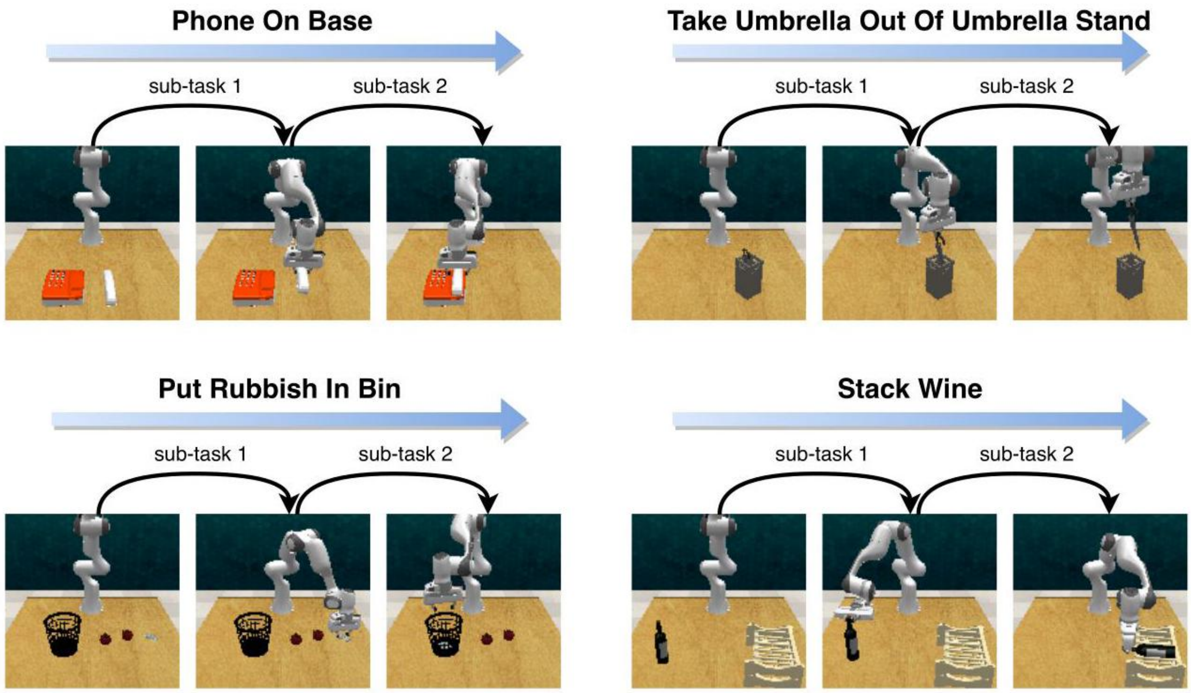


Figure 8: Visualization of the 4 short-horizon RLBench tasks in the experiment.

图 8: 实验中 4 个短期 RLBench 任务的可视化。

## A.3 RLBench Tasks

## A.3 RLBench 任务

We select 10 tasks from the 100 available in RLBench for our simulation experiments, including 4 short-horizon tasks (as shown in Fig. 8) and 6 long-horizon tasks (as shown in Fig. 9). To reduce training time with limited resources, we only use variation0. In the following sections, we describe each of these 10 tasks in detail, including any modifications made to the original codebase.

我们从 RLBench 中 100 个任务中选取 10 个用于仿真实验，包括 4 个短期任务 (见图 8) 和 6 个长期任务 (见图 9)。为减少有限资源下的训练时间，我们仅使用 variation0。以下章节详细介绍这 10 个任务，包括对原始代码库的任何修改。

### Phone On Base

### 电话放置在底座上

Task: grasp the phone and put it on the base.

任务: 抓住电话并将其放在底座上。

filename: phone_on_base.py

文件名:phone_on_base.py

Modified: Rewards are defined for each step based on the LLM-assisted reward generation module.

修改: 基于大语言模型 (LLM) 辅助的奖励生成模块，为每一步定义奖励。

Success Metric: The phone is on the base.

成功指标: 电话放置在底座上。

Task Horizon: 2.

任务时长:2。

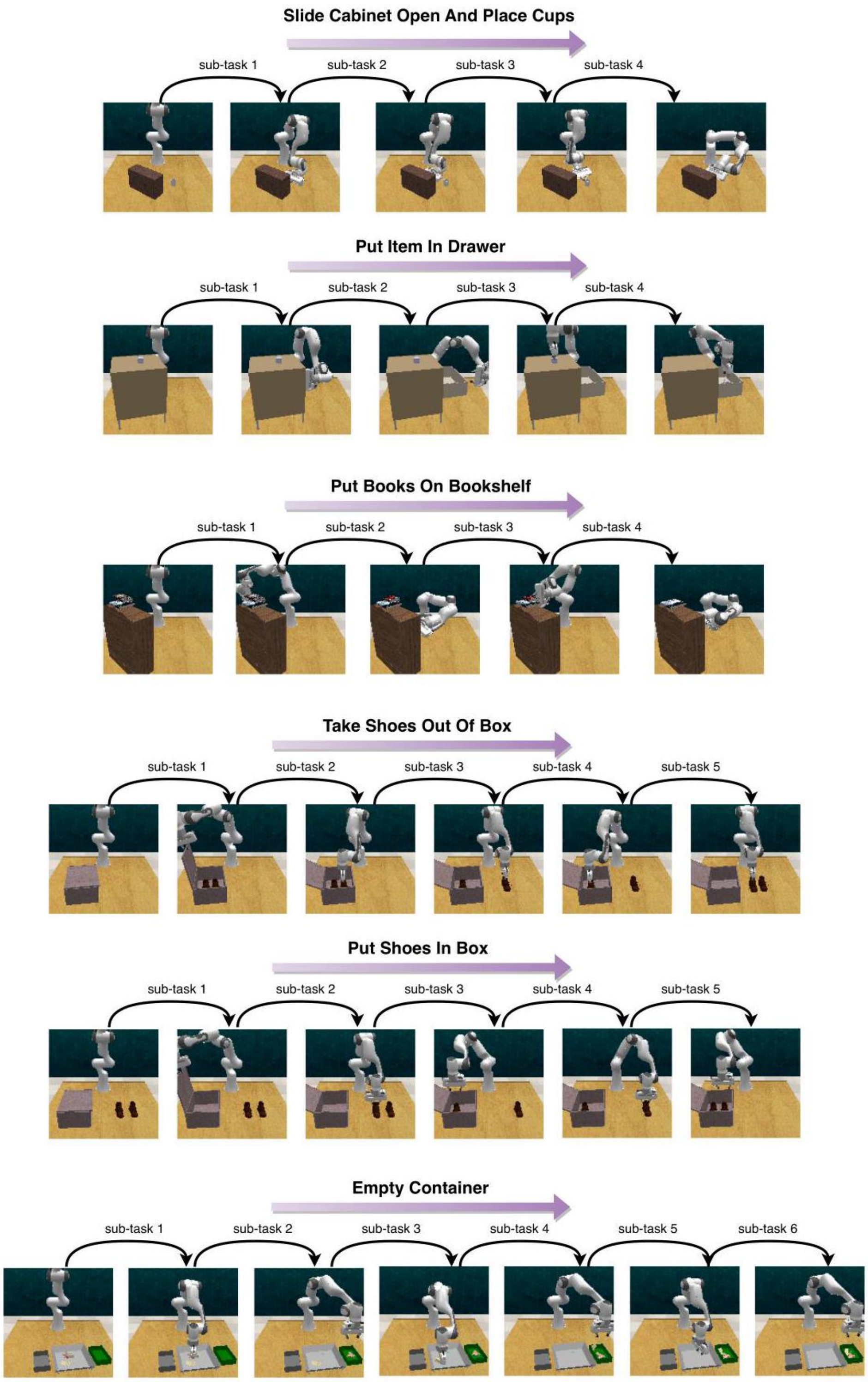


Figure 9: Visualization of the 6 long-horizon RLBench tasks in the experiment.

图 9: 实验中 6 个长时域 RLBench 任务的可视化。

## Take Umbrella Out Of Umbrella Stand

## 从伞架中取出伞

Task: grasp the umbrella by its handle, lift it up and out of the stand.

任务: 抓住伞柄，将伞提起并从伞架中取出。

filename: take_umbrella_out_of_umbrella_stand.py

文件名:take_umbrella_out_of_umbrella_stand.py

Modified: Rewards are defined for each step based on the LLM-assisted reward generation module. Success Metric: The umbrella is taken off the stand.

修改: 基于大语言模型 (LLM) 辅助的奖励生成模块，为每一步定义奖励。成功指标: 伞已从伞架中取出。

Task Horizon: 2.

任务时长:2。

## Put Rubbish In Bin

## 将垃圾放入垃圾桶

Task: pick up the rubbish and leave it in the trash can. filename: put_rubbish_in_bin.py Modified: Rewards are defined for each step based on the LLM-assisted reward generation module. Success Metric: The rubbish is thrown in the bin.

任务: 捡起垃圾并放入垃圾桶。文件名:put_rubbish_in_bin.py 修改: 基于大语言模型 (LLM) 辅助的奖励生成模块，为每一步定义奖励。成功指标: 垃圾已扔进垃圾桶。

Task Horizon: 2.

任务时长:2。

## Stack Wine

## 堆叠酒瓶

Task: place the wine bottle on the wine rack. filename: stack_wine.py Modified: Rewards are defined for each step based on the LLM-assisted reward generation module. Success Metric: The bottle is on the wine rack.

任务: 将酒瓶放置在酒架上。文件名:stack_wine.py 修改内容: 基于大语言模型辅助的奖励生成模块，为每一步定义奖励。成功标准: 酒瓶放置在酒架上。

Task Horizon: 2.

任务时长:2。

### Take Shoes Out Of Box

### 从盒子中取出鞋子

Task: grasp the edge of the box lid to open it, then grasp each shoe, lifting up out of the shoe box and leaving them down on the table. filename: take_shoes_out_of_box.py Modified: Rewards are defined for each step based on the LLM-assisted reward generation module. Success Metric: Both shoes are placed on the table. Task Horizon: 5.

任务: 抓住盒盖边缘打开盒子，然后抓起每只鞋，将其从鞋盒中提起并放置在桌面上。文件名:take_shoes_out_of_box.py 修改内容: 基于大语言模型辅助的奖励生成模块，为每一步定义奖励。成功标准: 两只鞋均放置在桌面上。任务时长:5。

### Put Shoes In Box

### 把鞋子放回盒子

Task: open the box lid and put the shoes inside. filename: put_shoes_in_box.py Modified: Rewards are defined for each step based on the LLM-assisted reward generation module. Success Metric: Both shoes are placed in the box.

任务: 打开盒盖，将鞋子放入盒内。文件名:put_shoes_in_box.py 修改内容: 基于大语言模型辅助的奖励生成模块，为每一步定义奖励。成功标准: 两只鞋均放入盒内。

Task Horizon: 5.

任务时长:5。

### Empty Container

### 清空容器

Task: move all objects from the large container and drop them into the smaller red one.

任务: 将大容器中的所有物品移动并放入较小的红色容器中。

filename: empty_container.py

文件名:empty_container.py

Modified: Rewards are defined for each step based on the LLM-assisted reward generation module. Success Metric: All objects in the large container are placed in the small red container. Task Horizon: 6.

修改内容: 基于大语言模型辅助的奖励生成模块，为每一步定义奖励。成功标准: 大容器中的所有物品均放入小红色容器中。任务时长:6。

## Put Books On Bookshelf

## 把书放到书架上

Task: pick up 2 books and place them on the top shelf.

任务: 拿起两本书，放到最上层的书架上。

filename: put_books_on_bookshelf.py

文件名:put_books_on_bookshelf.py

Modified: Rewards are defined for each step based on the LLM-assisted reward generation module. Success Metric: All the books are placed on top of the shelf. Task Horizon: 4.

修改内容: 基于大语言模型辅助的奖励生成模块，为每一步定义奖励。成功标准: 所有书籍均放置在书架顶层。任务时长:4。

## Put Item In Drawer

## 把物品放进抽屉

Task: open the middle drawer and place the block inside of it.

任务: 打开中间的抽屉并将积木放入其中。

filename: put_item_in_drawer.py

文件名:put_item_in_drawer.py

Modified: Rewards are defined for each step based on the LLM-assisted reward generation module. Success Metric: The block is placed in the middle drawer.

修改: 基于大语言模型 (LLM) 辅助的奖励生成模块，为每一步定义奖励。成功标准: 积木被放置在中间抽屉内。

Task Horizon: 4.

任务时长:4。

## Slide Cabinet Open And Place Cups

## 滑动柜门并放置杯子

Task: grasping the left handle, open the cabinet, then pick up the cup and set it down inside the cabinet. filename: slide_cabinet_open_and_place_cups.py

任务: 抓住左侧把手，打开柜子，然后拿起杯子并将其放入柜内。文件名:slide_cabinet_open_and_place_cups.py

Modified: Rewards are defined for each step based on the LLM-assisted reward generation module. Success Metric: The cup is in the left cabinet. Task Horizon: 5.

修改: 基于大语言模型 (LLM) 辅助的奖励生成模块，为每一步定义奖励。成功标准: 杯子放置在左侧柜子内。任务时长:5。

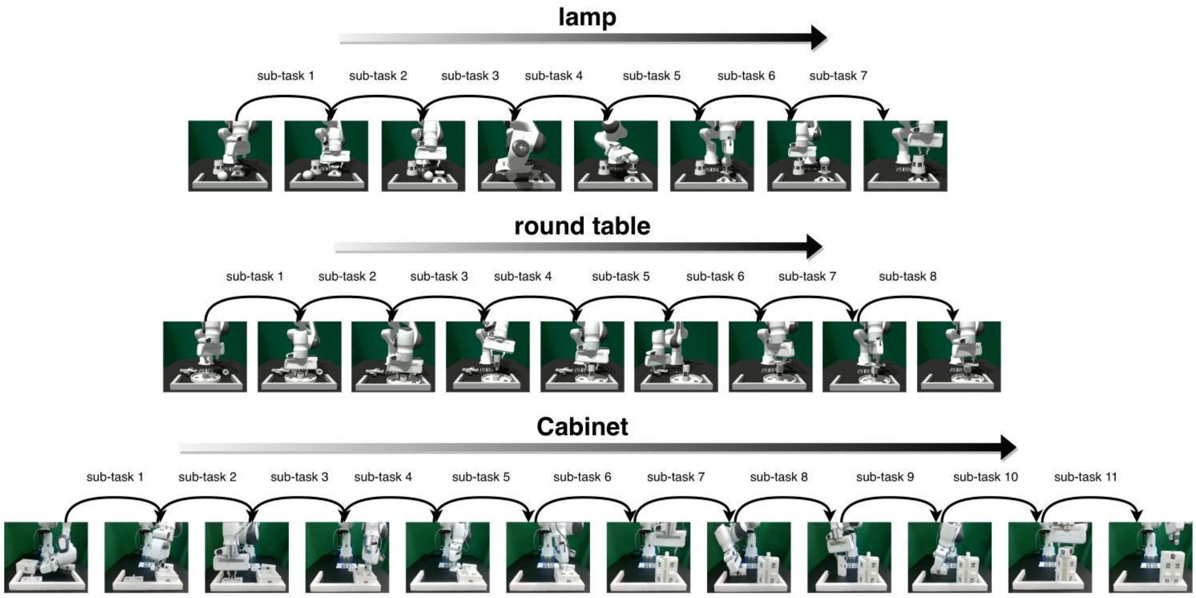


Figure 10: Visualization of the 3 long-horizon FurnitureBench tasks in the experiment, where the assembly of the cabinet is illustrated as the process with a real robot.

图 10: 实验中三个长时任务 FurnitureBench 的可视化，其中柜子组装过程由真实机器人演示。

## A.4 FurnitureBench Tasks

## A.4 FurnitureBench 任务

We select 3 furniture assembly tasks from the 9 available task in FurnitureBench for our experiments, as shown in Fig. 10). In the following parts, we describe each of these 3 tasks in detail, including any modifications made to the original codebase.

我们从 FurnitureBench 中 9 个可用任务中选择了 3 个家具组装任务用于实验，如图 10 所示。以下部分详细描述这 3 个任务，包括对原始代码库的任何修改。

### Lamp

### 台灯

Task: The robot needs to screw in a light bulb and then place a lamp hood on top of the bulb. The robot should perform sophisticated grasping since the bulb can be easily slipped when grasping and screwing due to the rounded shape

任务: 机器人需要旋紧灯泡，然后将灯罩放在灯泡上方。由于灯泡形状圆滑，机器人需进行精细抓取，否则在抓取和旋转时灯泡易滑落。

Modified: Rewards are defined for each step based on the LLM-assisted reward generation module.

修改: 基于大语言模型 (LLM) 辅助的奖励生成模块，为每一步定义奖励。

Success Metric: Three pieces of furniture are assembled into a lamp.

成功标准: 三件家具组装成一盏台灯。

Task Horizon: 7.

任务时长:7。

### Round Table

### 圆桌

Task: The robot should assemble one rounded tabletop, rounded leg, and cross-shaped table base. The robot should handle an easily movable round leg and cross-shaped table base, which requires finding a careful grasping point.

任务: 机器人应组装一个圆形桌面、圆形桌腿和十字形桌子底座。机器人需要处理易于移动的圆形桌腿和十字形桌子底座，这要求找到一个合适的抓取点。

Modified: Rewards are defined for each step based on the LLM-assisted reward generation module.

修改: 基于大语言模型 (LLM) 辅助的奖励生成模块，为每一步定义奖励。

Success Metric: Three pieces of furniture are assembled into an round table. Task Horizon: 8.

成功指标: 三件家具组装成一张圆桌。任务时长:8。

## Cabinet

## 柜子

Task: The robot must first insert two doors into the poles on each side of the body. Next, it must lock the top, so the doors do not slide out. This task requires careful control to align the doors and slide into the pole. Moreover, diverse skills such as flipping the cabinet body, and screwing the top are also needed to accomplish the task

任务: 机器人必须先将两扇门插入机身两侧的支柱中。接着，需要锁定顶部，防止门滑出。该任务要求精确控制以对齐门并滑入支柱。此外，还需多种技能，如翻转柜体和拧紧顶部，才能完成任务。

Modified: Rewards are defined for each step based on the LLM-assisted reward generation module.

修改: 基于大语言模型 (LLM) 辅助的奖励生成模块，为每一步定义奖励。

Success Metric: Four pieces of furniture are assembled into a cabinet.

成功指标: 四件家具组装成一个柜子。

Task Horizon: 11.

任务时长:11。