DeCo: Task Decomposition and Skill Composition for Zero-Shot Generalization in Long-Horizon 3D Manipulation

DeCo: 用于长时序三维操作中零样本泛化的任务分解与技能组合

Zixuan Chen [1,*] Junhui Yin [1,*] Yangtao Chen [1] Jing Huo [1,†]

陈子轩 [1,*] 尹俊辉 [1,*] 陈阳涛 [1] 霍晶 [1,†]

Pinzhuo Tian [2] Jieqi Shi [1] Yiwen Hou [3] Yinchuan Li [4] Yang Gao [1] [1] Nanjing University [2] Shanghai University

田品卓 [2] 石杰琦 [1] 侯奕文 [3] 李银川 [4] 高扬 [1] [1] 南京大学 [2] 上海大学

[3] National University of Singapore [4] Huawei

[3] 新加坡国立大学 [4] 华为

Abstract: Generalizing language-conditioned multi-task imitation learning (IL) models to novel long-horizon 3D manipulation tasks remains a significant challenge. To address this, we propose DeCo (Task Decomposition and Skill Composition), a model-agnostic framework compatible with various multi-task IL models, designed to enhance their zero-shot generalization to novel, compositional, long-horizon 3D manipulation tasks. DeCo first decomposes IL demonstrations into a set of modular atomic tasks based on the physical interaction between the gripper and objects, and constructs an atomic training dataset that enables models to learn a diverse set of reusable atomic skills during imitation learning. At inference time, DeCo leverages a vision-language model (VLM) to parse high-level instructions for novel long-horizon tasks, retrieve the relevant atomic skills, and dynamically schedule their execution; a spatially-aware skill-chaining module then ensures smooth, collision-free transitions between sequential skills. We evaluate DeCo in simulation using DeCoBench, a benchmark specifically designed to assess zero-shot generalization of multi-task IL models in compositional long-horizon 3D manipulation. Across three representative multitask IL models—RVT-2, 3DDA, and ARP—DeCo achieves success rate improvements of 66.67%, 21.53% , and 57.92% , respectively, on 12 novel compositional tasks. Moreover, in real-world experiments, a DeCo-enhanced model trained on only 6 atomic tasks successfully completes 9 novel long-horizon tasks, yielding an average success rate improvement of 53.33% over the base multi-task IL model. Video demonstrations are available at: https://deco226.github.io

摘要: 将语言条件下的多任务模仿学习 (IL) 模型泛化到新颖的长时序三维操作任务仍然是一个重大挑战。为此，我们提出了 DeCo(任务分解与技能组合)，这是一种与多种多任务 IL 模型兼容的模型无关框架，旨在提升其对新颖、组合性强的长时序三维操作任务的零样本泛化能力。DeCo 首先基于夹爪与物体之间的物理交互，将 IL 示范分解为一组模块化的原子任务，并构建原子训练数据集，使模型在模仿学习过程中能够学习多样且可复用的原子技能。在推理阶段，DeCo 利用视觉-语言模型 (VLM) 解析新颖长时序任务的高层指令，检索相关原子技能并动态调度其执行；空间感知的技能链模块确保连续技能之间的平滑且无碰撞的过渡。我们在模拟环境中使用专门设计用于评估多任务 IL 模型在组合性长时序三维操作中零样本泛化能力的基准 DeCoBench 进行了评测。在三种代表性多任务 IL 模型——RVT-2、3DDA 和 ARP 上，DeCo 在 12 个新颖组合任务中分别实现了 66.67%, 21.53% 和 57.92% 的成功率提升。此外，在真实环境实验中，基于仅 6 个原子任务训练的 DeCo 增强模型成功完成了 9 个新颖长时序任务，平均成功率较基础多任务 IL 模型提升了 53.33%。视频演示见:https://deco226.github.io

Keywords: Multi-task Imitation Learning, Task Decomposition, Skill Composition, Long-Horizon 3D Manipulation, Generalization

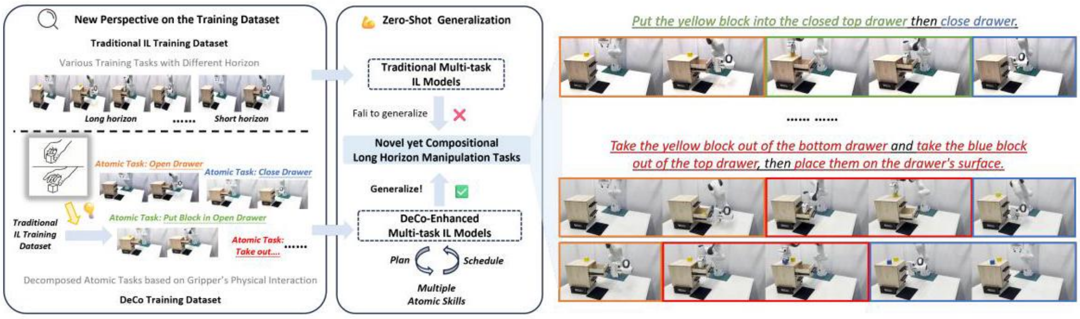
关键词: 多任务模仿学习，任务分解，技能组合，长时序三维操作，泛化

1 Introduction

1 引言

In recent years, imitation learning (IL) has emerged as a mainstream approach for robotic manipulation. By leveraging visual demonstrations and language instructions, IL trains language-conditioned multi-task control policies, enabling robots to acquire diverse skills and perform complex tasks in unstructured 3D environments. However, current multi-task IL models still suffer from limited generalization [1, 2, 3, 4, 5] , particularly when facing novel long-horizon 3D manipulation tasks [6]- even when such tasks are merely sequential compositions of previously learned skills. For instance, a model may have learned to follow individual instructions such as ”open drawer”, ”put block in opened drawer”, and ”close drawer”, yet still fail to execute the composed instruction ”put block into the closed drawer and then close drawer”. This failure stems from the model’s inability to decompose novel tasks and to retrieve, schedule, and perform the correct composition of its learned skills-failing to recognize that the task can be completed by sequentially executing three known skills: opening the drawer, placing the block, and then closing the drawer. Such limitations in task decomposition and skill composition severely undermine the real-world applicability and scalability of current multi-task IL models. Although vision-language models (VLMs) have been used to generate subtasks for long-horizon tasks via instruction plans [7, 8, 6] , executable code [9, 10] , spatial keypoints [11], or affordance maps [12, 13], they often fail to align high-level semantic plans with low-level execution. Low-level tasks are typically limited to simple motion planning or pretrained skills, and the semantic decomposition does not directly map to the physical skill space. This gap limits the effective composition of low-level skills, ultimately hindering zero-shot performance on long-horizon 3D manipulation tasks [14, 15, 16].

近年来，模仿学习 (IL) 已成为机器人操作的主流方法。通过利用视觉示范和语言指令，IL 训练语言条件下的多任务控制策略，使机器人能够掌握多样技能并在非结构化三维环境中执行复杂任务。然而，当前多任务 IL 模型仍存在泛化能力有限的问题，尤其是在面对新颖的长时序三维操作任务时 [6]——即使这些任务仅是先前学习技能的顺序组合。例如，模型可能已学会执行“打开抽屉”、“将积木放入已打开的抽屉”和“关闭抽屉”等单独指令，但仍无法执行组合指令“将积木放入关闭的抽屉然后关闭抽屉”。这种失败源于模型无法分解新任务，也无法检索、调度并执行正确的技能组合——未能识别该任务可通过依次执行三项已知技能完成: 打开抽屉、放置积木、关闭抽屉。任务分解与技能组合能力的不足严重制约了当前多任务 IL 模型在现实中的适用性和扩展性。尽管视觉-语言模型 (VLM) 已被用于通过指令计划 [7, 8, 6] 、可执行代码 [9, 10] 、空间关键点 [11] 或可供性图 [12, 13] 生成长时序任务的子任务，但它们常常难以将高层语义计划与低层执行对齐。低层任务通常仅限于简单的运动规划或预训练技能，语义分解并未直接映射到物理技能空间。这一差距限制了低层技能的有效组合，最终阻碍了长时序三维操作任务的零样本表现 [14, 15, 16]。


New Perspective on the Training Dataset
Traditional IL Training Dataset
Various Training Tasks with Different Horizon
Long horizon
......
Short horizon
Atomic Task: Open Drawer
Atomic Task: Close Drawer
Atomic Task: Put Block in Open Drawer
Traditional IL Training Dataset
Atomic Task: Take out....
......
Decomposed Atomic Tasks based on Gripper's Physical Interaction
DeCo Training Dataset
Zero-Shot Generalization
Traditional Multi-task IL Models
Fali to generalize
Novel yet Compositional Long Horizon Manipulation Tasks
Generalize!
DeCo-Enhanced Multi-task IL Models
Plan
Schedule
Multiple Atomic Skills
Put the yellow block into the closed top drawer then close drawer.
....
Take the yellow block out of the bottom drawer and take the blue block out of the top drawer, then place them on the drawer's surface.

(a) Insight and Performance of DeCo Framework

(b) DeCo-Enhanced Models Zero-Shot Execution of Real-World Long-Horizon Tasks

Figure 1: We present DeCo, a model-agnostic framework that enables diverse multi-task IL models to zero-shot generalize to novel yet compositional long-horizon 3D manipulation tasks.

图 1: 我们提出了 DeCo，一种模型无关的框架，使多任务模仿学习 (IL) 模型能够零样本泛化到新颖且具有组合性的长时域三维操作任务。

In this paper, we aim to enhance the zero-shot generalization of multi-task IL models to novel, compositional long-horizon 3D manipulation tasks—tasks that are unseen during training but can be solved by composing previously learned skills through semantic reasoning. To achieve this, we propose a model-agnostic framework for skill learning and composition, compatible with a wide range of multi-task IL models. This framework enables models to decompose novel tasks into reusable atomic skills, flexibly schedule them, and execute without additional training. The core question we address is: How can long-horizon tasks be decomposed into learned skills such that multi-task IL models can interpret their structure, plan accordingly, and successfully complete the overall task in

*: denotes equal contribution

*: 表示同等贡献

† :Correspondence to: huojing@nju.edu.cn

† : 通讯作者:huojing@nju.edu.cn

a 3D environment? To this end, we propose DeCo (Task Decomposition and Skill Compositon), a model-agnostic framework compatible with various multi-task IL models, enabling zero-shot generalization to novel yet compositional long-horizon 3D manipulation tasks, as illustrated in Figure 1. DeCo consists of three key components: First, inspired by how humans decompose long-horizon tasks through hand-object interactions, DeCo proposes a new perspective on training datasets for multi-task imitation learning, based on prior methods of subtask discovery [13, 17]. It preprocesses original IL demonstrations by analyzing the physical interactions between the gripper and objects, decomposing them into a set of modular and reusable atomic tasks. Each task is paired with a natural language instruction and a goal pose, forming an atomic training dataset for training a multi-task IL model to acquire diverse skills. Second, during testing, DeCo uses VLMs to parse the novel language instructions and visual inputs, retrieve relevant atomic instructions from the atomic training dataset, and generate an execution plan. The multi-task IL model sequentially executes the skills, while DeCo monitors task progress via gripper interactions, enabling dynamic scheduling and flexible skill composition. Finally, to ensure smooth transitions between skills, DeCo builds a spatially aware cost map for the scene to calculate collision-free chaining poses, guiding the robotic arm between sequential skills and ensuring motion continuity and safety.

本文旨在提升多任务模仿学习模型对新颖、组合性长时域三维操作任务的零样本泛化能力——这些任务在训练时未见过，但可以通过语义推理将先前学得的技能组合起来解决。为此，我们提出了一种适用于多种多任务模仿学习模型的模型无关技能学习与组合框架。该框架使模型能够将新任务分解为可复用的原子技能，灵活调度并执行，无需额外训练。我们关注的核心问题是: 如何将长时域任务分解为已学技能，使多任务模仿学习模型能够理解其结构、据此规划，并在三维环境中成功完成整体任务? 为此，我们提出了 DeCo(任务分解与技能组合)，一种兼容多种多任务模仿学习模型的模型无关框架，实现对新颖且组合性的长时域三维操作任务的零样本泛化，如图 1 所示。DeCo 包含三个关键组成部分: 首先，受人类通过手-物体交互分解长时域任务的启发，DeCo 基于先前的子任务发现方法 [13,17]，提出了多任务模仿学习训练数据集的新视角。它通过分析夹爪与物体的物理交互，对原始模仿学习示范进行预处理，将其分解为一组模块化且可复用的原子任务。每个任务配有自然语言指令和目标姿态，形成用于训练多任务模仿学习模型以习得多样技能的原子训练数据集。其次，在测试阶段，DeCo 利用视觉语言模型 (VLMs) 解析新颖的语言指令和视觉输入，从原子训练数据集中检索相关原子指令，并生成执行计划。多任务模仿学习模型按序执行技能，DeCo 通过夹爪交互监控任务进展，实现动态调度和灵活技能组合。最后，为确保技能间平滑过渡，DeCo 构建了场景的空间感知代价图，用于计算无碰撞的连接姿态，引导机械臂在连续技能间运动，保证动作连续性和安全性。

We carry out extensive evaluations in both simulated and real-world settings. In simulation, we introduce DeCoBench, a new benchmark built upon RLBench [18], designed to systematically evaluate the zero-shot generalization capabilities of multi-task IL models on novel yet compositional long-horizon 3D manipulation tasks. We equip DeCo with three representative multi-task IL models—RVT-2 [3], 3DDA [19], and ARP [5]—and evaluate their performance on DeCoBench. Experiments show that DeCo significantly boosts the generalization performance across all three models. Beyond simulation, we design 6 atomic tasks grounded in physical interaction to train a multi-task IL model, and evaluate it on 9 novel yet compositional long-horizon tasks. The DeCo-enhanced model demonstrates strong zero-shot generalization, validating the practicality of DeCo.

我们在模拟和真实环境中进行了广泛评估。在模拟中，我们基于 RLBench [18] 构建了 DeCoBench，一个系统评估多任务模仿学习模型在新颖且组合性的长时域三维操作任务上零样本泛化能力的新基准。我们为 DeCo 配备了三种代表性多任务模仿学习模型——RVT-2 [3]、3DDA [19] 和 ARP [5]，并在 DeCoBench 上评估其性能。实验结果表明，DeCo 显著提升了这三种模型的泛化表现。除模拟外，我们设计了 6 个基于物理交互的原子任务训练多任务模仿学习模型，并在 9 个新颖且组合性的长时域任务上进行评估。DeCo 增强的模型展现出强大的零样本泛化能力，验证了 DeCo 的实用性。

Our main contributions are follows: (1) We introduce DeCo, a model-agnostic framework that equips diverse multi-task IL models with zero-shot generalization capabilities for novel yet compositional long-horizon 3D manipulation tasks. (2) We introduce DeCoBench, a benchmark for systematically evaluating zero-shot generalization in multi-task IL models on compositional long-horizon 3D manipulation tasks. Extensive experiments on DeCoBench show that DeCo significantly improves the generalization of three representative multi-task IL models, validating its effectiveness. (3) We validate DeCo in real-world settings by constructing 6 atomic tasks and 9 novel yet composi-tonal long-horizon tasks. Results demonstrate that DeCo effectively enables multi-task IL models to achieve zero-shot generalization in novel long-horizon tasks, underscoring its practical applicability.

我们的主要贡献如下:(1) 提出了 DeCo，一种模型无关框架，使多样的多任务模仿学习模型具备对新颖且组合性的长时域三维操作任务的零样本泛化能力。(2) 引入了 DeCoBench，一个用于系统评估多任务模仿学习模型在组合性长时域三维操作任务上零样本泛化能力的基准。大量 DeCoBench 实验表明，DeCo 显著提升了三种代表性多任务模仿学习模型的泛化能力，验证了其有效性。(3) 通过构建 6 个原子任务和 9 个新颖且组合性的长时域任务，在真实环境中验证了 DeCo。结果表明，DeCo 有效使多任务模仿学习模型在新颖长时域任务中实现零样本泛化，凸显其实用价值。

2 Related Work

2 相关工作

Learning Manipulation Policies from Demonstrations Learning manipulation policies from offline visual demonstrations has garnered significant attention, fueled by advances in visual perception [20, 21]. Early 2D-based approaches [22, 23, 24, 25, 26, 14, 27, 28] have demonstrated success in simple pick-and-place tasks, benefiting from fast training, low hardware requirements, and modest computational demands. However, their reliance on pretrained image encoders and limited spatial understanding makes them less effective for tasks requiring high-precision and robust 3D interactions. To address this, works such as C2F-ARM [29] and PerAct [1] extend learning to 6-DoF actions in 3D environments, but they still require training separate task-specific policies. More recent efforts [2, 30, 31, 32, 13, 3, 19, 5, 6] aim to develop unified multi-task imitation learning (IL) models that can perform diverse tasks from heterogeneous demonstrations. This shift is crucial for building general-purpose robotic agents. However, most of these models are limited to tasks observed during training, and particularly struggle to generalize to novel long-horizon scenarios, which hinders their deployment in real-world applications [6]. To address this limitation, we propose a model-agnostic framework that is compatible with existing representative multi-task imitation learning (IL) models, enabling them to achieve zero-shot generalization to novel long-horizon 3D manipulation tasks.

从示范中学习操作策略从离线视觉示范中学习操作策略因视觉感知 (visual perception)[20, 21] 的进步而受到广泛关注。早期基于二维 (2D) 的方法 [22, 23, 24, 25, 26, 14, 27, 28] 在简单的抓取与放置任务中取得了成功，得益于训练速度快、硬件需求低和计算负担适中。然而，这些方法依赖于预训练的图像编码器且空间理解能力有限，因而在需要高精度和鲁棒三维 (3D) 交互的任务中效果不佳。为此，诸如 C2F-ARM [29] 和 PerAct [1] 等工作将学习扩展到三维环境中的六自由度 (6-DoF) 动作，但仍需训练独立的任务特定策略。近期的研究 [2, 30, 31, 32, 13, 3, 19, 5, 6] 致力于开发统一的多任务模仿学习 (imitation learning, IL) 模型，能够从异构示范中执行多样任务。这一转变对于构建通用机器人代理至关重要。然而，大多数模型仅限于训练时观察到的任务，尤其难以推广到新颖的长时序场景，限制了其在实际应用中的部署 [6]。为解决此限制，我们提出了一个模型无关的框架，兼容现有代表性多任务模仿学习模型，使其能够实现对新颖长时序三维操作任务的零样本泛化。

Methods for Long-Horizon Manipulation A common strategy for long-horizon manipulation is to decompose complex tasks into sequential subtasks using predefined action primitives (e.g., grasp, place, pull) [33, 34, 35] or environment-specific cues [14, 36, 37, 38, 39, 40]. While effective in structured settings, these methods lack compositional flexibility and generalization, making them fragile to goal shifts and environmental changes, and limiting scalability in multi-task IL. Recent work leverages Vision-Language Models (VLMs) to enhance subtask generation by exploiting their semantic understanding and decomposition capabilities. VLMs facilitate high-level planning via natural language, executable code, spatial keypoints, or affordance maps [7,8,9,10,12,13]. However, they often fail to align high-level semantic plans with low-level execution. Low-level behaviors remain constrained to motion primitives or pretrained skills, and semantic decomposition seldom maps directly to the physical skill space. This misalignment limits skill composition and hinders generalization in long-horizon 3D manipulation [14, 15, 16]. To address these issues, we propose a new atomic task construction with modularity and reusability, enabling consistent decomposition across diverse scenarios and compatibility with diverse multi-task IL models. We also introduce a spatially-aware skill chaining module with collision avoidance. Combined with VLM-guided planning, our framework improves generalization and robustness in compositional long-horizon tasks.

长时序操作方法长时序操作的常用策略是利用预定义的动作原语 (如抓取、放置、拉动)[33, 34, 35] 或环境特定线索 [14, 36, 37, 38, 39, 40] 将复杂任务分解为顺序子任务。尽管在结构化环境中有效，这些方法缺乏组合灵活性和泛化能力，对目标变化和环境变动较为脆弱，限制了多任务模仿学习的可扩展性。近期工作利用视觉-语言模型 (Vision-Language Models, VLMs) 通过其语义理解和分解能力增强子任务生成。VLMs 支持通过自然语言、可执行代码、空间关键点或可供性图 (affordance maps) 进行高层规划 [7,8,9,10,12,13]。然而，它们常常难以将高层语义规划与低层执行对齐。低层行为仍受限于运动原语或预训练技能，语义分解很少能直接映射到物理技能空间。这种不匹配限制了技能组合，阻碍了长时序三维操作的泛化 [14, 15, 16]。为解决这些问题，我们提出了一种具有模块化和可重用性的原子任务构建方法，实现跨多样场景的一致分解，并兼容多种多任务模仿学习模型。我们还引入了具备避碰功能的空间感知技能链模块。结合 VLM 引导的规划，我们的框架提升了组合长时序任务的泛化性和鲁棒性。

3 Method

3 方法

In this section, we introduce DeCo (Task Decomposition and Skill Composition), a model-agnostic framework compatible with diverse existing multi-task IL models, enabling zero-shot generalization

本节介绍 DeCo(任务分解与技能组合)，一个模型无关的框架，兼容多种现有多任务模仿学习模型，实现零样本泛化

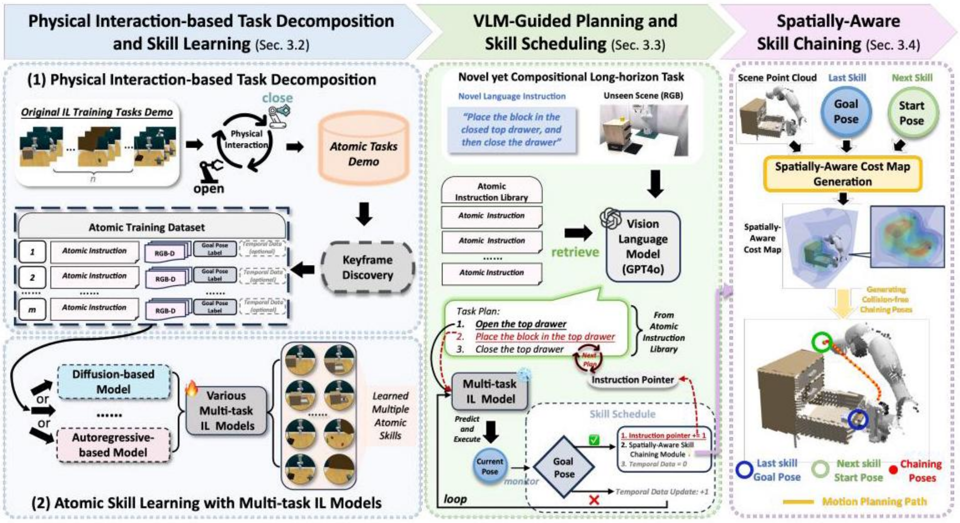

Physical Interaction-based Task Decomposition and Skill Learning (Sec. 3.2)
VLM-Guided Planning and Skill Scheduling (Sec. 3.3)
Spatially-Aware Skill Chaining (Sec. 3.4)
(1) Physical Interaction-based Task Decomposition
Original IL Training Tasks Demo
n
close
Physical Interaction
open
Atomic Tasks Demo
Keyframe Discovery
Atomic Training Dataset
1 Atomic Instruction RGB-D Goal Pose Label Temporal Data (optional)
2 Atomic Instruction RGB-D Goal Pose Label Temporal Data (optional)
m Atomic Instruction RGB-D Goal Pose Label Temporal Data (optional)
Diffusion-based Model
or
or
Autoregressive-based Model
Various Multi-task IL Models
Learned Multiple Atomic Skills
(2) Atomic Skill Learning with Multi-task IL Models
Novel yet Compositional Long-horizon Task
Novel Language Instruction
"Place the block in the closed top drawer, and then close the drawer"
Unseen Scene (RGB)
Atomic Instruction Library
Atomic Instruction
Atomic Instruction
Atomic Instruction
retrieve
Vision Language Model (GPT4o)
Task Plan:
1. Open the top drawer
2. Place the block in the top drawer
3. Close the top drawer
From Atomic Instruction Library
Next Plan
Instruction Pointer
Multi-task IL Model
Predict and Execute
Current Pose
monitor
Goal Pose
Skill Schedule
1. Instruction pointer += 1
2. Spatially-Aware Skill Chaining Module
3. Temporal Data = 0
Temporal Data Update: +1
loop
Scene Point Cloud
Last Skill
Goal Pose
Next Skill
Start Pose
Spatially-Aware Cost Map Generation
Spatially-Aware Cost Map
Generating Collision-free Chaining Poses
Last skill Goal Pose
Next skill Start Pose
Chaining Poses
Motion Planning Path

Figure 2: An overview of DeCo framework.

图 2:DeCo 框架概览。

to novel yet compositional long-horizon 3D manipulation tasks. Figure 2 shows DeCo's workflow within multi-task IL models. Sec. 3.1 introduces the problem formulation based on gripper-object physical interactions. Sec. 3.2 explains how DeCo processes the original training demonstrations of multi-task imitation learning (IL) models into atomic tasks and constructs a paired atomic instruction library and training dataset to achieve atomic skill learning. Sec. 3.3 details how DeCo utilizes vision-language models (VLMs) to plan novel task instructions and schedule atomic skills accordingly. Finally, Sec. 3.4 presents the spatially-aware skill chaining module, ensuring collision-free execution of long-horizon sequence.

针对新颖且具组合性的长时序三维操作任务。图 2 展示了 DeCo 在多任务模仿学习模型中的工作流程。3.1 节基于夹爪-物体物理交互介绍问题定义。3.2 节说明 DeCo 如何将多任务模仿学习模型的原始训练示范处理为原子任务，并构建配对的原子指令库和训练数据集以实现原子技能学习。3.3 节详述 DeCo 如何利用视觉-语言模型 (VLMs) 规划新任务指令并相应调度原子技能。最后，3.4 节介绍空间感知技能链模块，确保长时序序列的无碰撞执行。

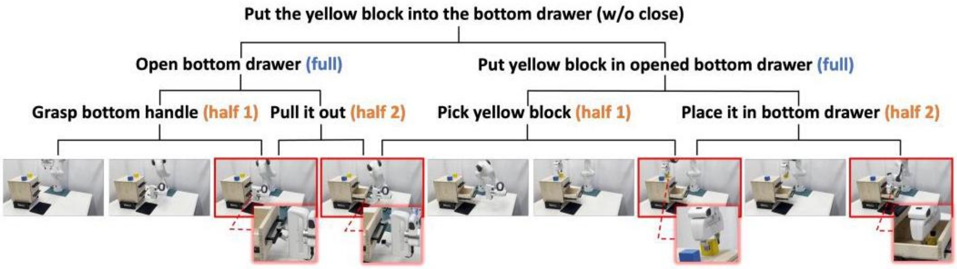

Put the yellow block into the bottom drawer (w/o close)
Open bottom drawer (full)
Put yellow block in opened bottom drawer (full)
Grasp bottom handle (half 1)
Pull it out (half 2)
Pick yellow block (half 1)
Place it in bottom drawer (half 2)

Figure 3: Visual example of full and half interactions.

图 3: 完整与半交互的视觉示例。

3.1 Problem Formulation

3.1 问题定义

We define the physical interaction as the contact event between a robotic gripper and an object, identified by changes in the gripper's openness. A single change in the gripper (from open to closed or vice versa) is a cycle. A full interaction, denoted as p^{full} , consists of two cycles: open → closed → open. A half interaction, p^{half} , represents a single change (one cycle) from open to closed or vice versa: $p^{\text{full}} = p^{\text{half}}_{o\rightarrow c} + p^{\text{half}}_{c\rightarrow o}$, where $p^{\text{half}}_{o\rightarrow c}$ and $p^{\text{half}}_{c\rightarrow o}$ represent the two sub-phases of the gripper transition: from open to closed, and from closed to open, respectively. A visual illustration of the definitions and relationship between full and half interactions is shown in Figure 3. Further differences are discussed in Appendix A. We assume access to a training task set $\mathcal{T}^o = \{T^o_1, T^o_2, \ldots, T^o_n\}$, each paired with a natural language instruction ℓ^o_i . However, the physical interaction phases within each T^o_i are often inconsistent. By decomposing tasks using predefined interaction boundaries, we construct an atomic task set $\mathcal{T}^a = \{T^a_1, T^a_2, \ldots, T^a_m\}$ and a corresponding instruction library $\mathcal{L}^a = \{\ell^a_1, \ell^a_2, \ldots, \ell^a_m\}$. Each atomic task contains a consistent interaction cycle, either p^{full} or p^{half} . We frame 3D manipulation as keypose prediction [41,29,2,1,13]. A language-conditioned multi-task IL model $\mathcal{M}$ takes as input the observation o_t (RGB-D) and instruction ℓ , and predicts a 7- DoF action a_t : a 6-DoF end-effector pose and a 1-DoF gripper state $[1, 2]$. Trained on $\mathcal{T}^a$, the model $\mathcal{M}$ learns a policy π to solve any $T^a_i \in \mathcal{T}^a$. Our objective is for the enhanced model $\mathcal{M}+$ DeCo to generalize zero-shot to a novel yet compositional long-horizon task T^{new} , decomposable into atomic steps: $T^{\text{new}} = T^a_x + T^a_y + \cdots + T^a_z$. At inference time, $\mathcal{M} +$ DeCo retrieves, schedules, and executes the relevant atomic skills based on the retrieved instructions $\{\ell^a_x, \ell^a_y, \ldots, \ell^a_z\}$, enabling zero-shot generalization to new compositions.

我们将物理交互定义为机器人夹持器与物体之间的接触事件，通过夹持器开合状态的变化来识别。夹持器状态的一次变化 (从打开到关闭或反之) 称为一个周期。完整交互，记作 p^{full} ，包含两个周期: 开 → 闭 → 开。半交互, p^{half} , 表示从开到闭或反之的单次变化 (一个周期): $p^{\text{full}} = p^{\text{half}}_{o\rightarrow c} + p^{\text{half}}_{c\rightarrow o}$, 其中 $p^{\text{half}}_{o\rightarrow c}$ 和 $p^{\text{half}}_{c\rightarrow o}$ 分别代表夹持器状态转换的两个子阶段: 从开到闭和从闭到开。定义及完整与半交互关系的示意图见图 3。更多差异讨论见附录 A。我们假设可访问训练任务集 $\mathcal{T}^o = \{T^o_1, T^o_2, \ldots, T^o_n\}$, 每个任务配有自然语言指令 ℓ^o_i 。然而，每个 T^o_i 中的物理交互阶段常不一致。通过使用预定义的交互边界分解任务, 我们构建了原子任务集 $\mathcal{T}^a = \{T^a_1, T^a_2, \ldots, T^a_m\}$ 及对应的指令库 $\mathcal{L}^a = \{\ell^a_1, \ell^a_2, \ldots, \ell^a_m\}$ 。每个原子任务包含一致的交互周期, 即 p^{full} 或 p^{half} 。我们将三维操作视为关键姿态预测 [41,29,2,1,13]。一个基于语言条件的多任务模仿学习模型 $\mathcal{M}$ 以观测 o_t (RGB-D) 和指令 ℓ 为输入，预测 7 自由度动作 a_t :6 自由度末端执行器姿态和 1 自由度夹持器状态 $[1, 2]$ 。模型 $\mathcal{M}$ 在 $\mathcal{T}^a$ 上训练，学习策略 π 以解决任意 $T^a_i \in \mathcal{T}^a$ 。我们的目标是使增强模型 $\mathcal{M}+$ DeCo 实现对新颖但可组合的长时序任务 T^{new} 的零样本泛化，该任务可分解为原子步骤: $T^{\text{new}} = T^a_x + T^a_y + \cdots + T^a_z$ 。推理时，$\mathcal{M} +$ DeCo 基于检索到的指令 $\{\ell^a_x, \ell^a_y, \ldots, \ell^a_z\}$ 检索、调度并执行相关原子技能，实现对新组合的零样本泛化。

3.2 Physical Interaction-based Task Decomposition and Skill Learning

3.2 基于物理交互的任务分解与技能学习

To construct modular and reusable atomic skills, DeCo proposes a novel task decomposition strategy inspired by human hand-object interactions and prior work $[13, 17]$. This decomposition is based on the physical interactions of the robotic gripper, as described in Sec. 3.1. For the original demonstrations $\mathcal{T}^o$ used to train the multi-task IL model $\mathcal{M}$, DeCo decomposes tasks based on full physical interactions $p^{\text{ full}}$. For instance, a demonstration for the instruction ”put item in a closed drawer without closing the drawer” can be divided into two atomic tasks: ”open drawer” and ”place item into open drawer”, each aligned with a full gripper interaction. After decomposition, DeCo reformats the atomic demonstrations for skill learning. Each atomic task T_i^a is paired with a language instruction ℓ_i^a , forming the instruction library $\mathcal{L}^a$. Demonstrations are processed using a keyframe discovery method [41] that identifies keyframes based on gripper state transitions or near-zero joint velocities. Each demonstration concludes with a full physical interaction $p^{\text{ full}}$, and the end-effector pose in the final keyframe is marked as the goal pose. Optionally, demonstrations may include temporal data (e.g., time steps) to support task progression modeling. Finally, $\mathcal{M} +$ DeCo is trained with these physically consistent atomic datasets, enabling it to effectively acquire multiple atomic skills. The objective is to learn a language-conditioned policy π_θ^a that maps observation-instruction pairs to actions: $\pi_\theta^a = \arg\min_\theta \mathbb{E}_{i,(o,a)} \left[\mathcal{L}_{\text{ MT-IL}} \left(\pi_\theta^a \left(o, \ell_i^a\right), a\right)\right]$, where $\pi_\theta^a \left(o, \ell_i^a\right) = \mathcal{M}_\theta \left(o, \ell_i^a\right)$. Unless otherwise stated, all atomic training datasets and experimental results of $\mathcal{M} +$ DeCo presented in the main paper are based on $p^{\text{ full}}$. To explore the suitable granularity of physical interaction, we also implement a DeCo variant based on $p^{\text{ half}}$. Ablation study results are discussed in Sec. 5.3, with additional comparison experiments detailed in subsection C.2.

为了构建模块化且可复用的原子技能，DeCo 提出了一种受人类手-物体交互及先前工作启发的新颖任务分解策略。该分解基于机器人夹持器的物理交互，如第 3.1 节所述。对于用于训练多任务模仿学习 (IL) 模型的原始示范，DeCo 基于完整的物理交互对任务进行分解。例如，指令“将物品放入未关闭的抽屉”对应的示范可分为两个原子任务:“打开抽屉”和“将物品放入打开的抽屉”，每个任务均对应一次完整的夹持器交互。分解后，DeCo 重新格式化原子示范以进行技能学习。每个原子任务配有语言指令，形成指令库。示范通过关键帧发现方法 [41] 处理，该方法基于夹持器状态转换或接近零的关节速度识别关键帧。每个示范以一次完整的物理交互结束，最终关键帧中的末端执行器位姿被标记为目标位姿。示范可选地包含时间数据 (如时间步) 以支持任务进展建模。最后，使用这些物理一致的原子数据集训练模型，使其能够有效掌握多种原子技能。目标是学习一个语言条件策略，将观察-指令对映射到动作；除非另有说明，本文中所有原子训练数据集和实验结果均基于完整物理交互。为探索物理交互的合适粒度，我们还实现了基于另一种交互的 DeCo 变体。消融研究结果见第 5.3 节，更多对比实验详见附录 C.2 节。

3.3 VLM-Guided Planning and Skill Scheduling

3.3 基于视觉语言模型的规划与技能调度

After a multi-task IL model $\mathcal{M}$ has mastered the skills of multiple atomic tasks, with the support of DeCo, $\mathcal{M}$ can further enhance its ability to tackle novel long-horizon tasks. When faced with a novel yet compositional task $T^{\text{ new}}$, $\mathcal{M}$ + DeCo first utilizes the state-of-the-art vision-language model (VLM) GPT-4o [42], inputting the task’s language instructions, observed RGB images, and the pre-obtained atomic instruction library. Through the powerful reasoning and planning capabilities of the VLM, it retrieves relevant instructions from the atomic instruction library and forms a planning sequence for the current new task, outlining each atomic task required to achieve the task’s objectives. Subsequently, $\mathcal{M} +$ DeCo initiates the execution of the first atomic skill based on the first atomic instruction in the planning sequence. During execution, the system continuously monitors the robot’s pose to determine if it matches the goal pose specified in the corresponding skill. This real-time feedback loop is crucial for ensuring the correct and efficient execution of each skill. If the current pose matches the goal pose, it indicates that the atomic skill has been successfully completed, meaning the gripper has completed a full physical interaction cycle (from open to close and back to open), allowing the system to proceed to the next atomic skill's instruction. Conversely, if the pose does not match, the system continues executing the current skill until the desired pose is achieved. Details of this process, including the prompts input to the VLM, can be found in Appendix B.

在多任务模仿学习 (IL) 模型 $\mathcal{M}$ 掌握了多个原子任务技能后，借助 DeCo 的支持，$\mathcal{M}$ 能够进一步提升其应对新颖长时序任务的能力。当面对一个新颖且具有组合性的任务时，T^{new} , $\mathcal{M}+$ DeCo 首先利用最先进的视觉语言模型 (VLM)GPT-4o [42]，输入任务的语言指令、观察到的 RGB 图像以及预先获得的原子指令库。通过 VLM 强大的推理和规划能力，它从原子指令库中检索相关指令，并为当前新任务形成规划序列，勾勒出实现任务目标所需的每个原子任务。随后，$\mathcal{M}+$ DeCo 根据规划序列中的第一个原子指令启动第一个原子技能的执行。在执行过程中，系统持续监控机器人的姿态，以判断其是否与对应技能中指定的目标姿态匹配。该实时反馈环路对于确保每个技能的正确且高效执行至关重要。如果当前姿态与目标姿态匹配，表明该原子技能已成功完成，即夹爪完成了一个完整的物理交互周期 (从打开到关闭再回到打开)，系统即可进入下一个原子技能的指令。反之，若姿态不匹配，系统将继续执行当前技能，直至达到期望姿态。该过程的详细信息，包括输入给 VLM 的提示，见附录 B。

3.4 Spatially-Aware Skill Chaining for Long-Horizon 3D Manipulation

3.4 面向空间感知的技能链用于长时序三维操作

Although $\mathcal{M}+$ DeCo is capable of semantically combining atomic skills to accomplish long-horizon tasks via VLM-guided planning and skill scheduling (see Sec. 3.3), challenges remain in executing these skills sequentially. A primary issue lies in achieving smooth transitions between atomic skills in 3D space. Each atomic skill acquired by $\mathcal{M}$ is associated with distinct start and goal poses, often resulting in considerable spatial discontinuities between successive skills. Traditional motion planners, lacking spatial awareness, may trigger collisions during these transitions, ultimately causing task failure. To address this limitation, DeCo introduces a spatially-aware skill chaining module that enables seamless transitions without modifying the pose distributions. Specifically, once the current skill completes-i.e., the robot's pose matches the goal pose-the system schedules the next atomic instruction and predicts its start pose. The goal pose of the current skill, the predicted start pose of the next skill, and the scene point cloud are then passed to a spatially-aware cost map generation module, adapted from the foundation model Voxposer [12]. This module produces a set of collision-free chaining poses that bridge the gap between the current skill's goal pose and the next skill's start pose, as show in the third part of Figure 2. The robot then performs RRT-based [43] motion planning over these chaining poses to ensure safe transitions. This skill chaining module operates during the handoff between atomic skills in $\mathcal{M}+$ DeCo , enabling smooth composition of sequential skills and reliable execution of long-horizon 3D manipulation tasks.

尽管 $\mathcal{M}+$ DeCo 能够通过 VLM 引导的规划和技能调度 (见第 3.3 节) 语义组合原子技能以完成长时序任务，但在顺序执行这些技能时仍面临挑战。主要问题在于实现原子技能之间在三维空间中的平滑过渡。$\mathcal{M}$ 获得的每个原子技能都关联有不同的起始和目标姿态，常导致连续技能间存在显著的空间不连续性。传统的运动规划器缺乏空间感知，可能在这些过渡过程中引发碰撞，最终导致任务失败。为解决此限制，DeCo 引入了一个空间感知的技能链模块，使得在不修改姿态分布的情况下实现无缝过渡。具体而言，一旦当前技能完成——即机器人姿态匹配目标姿态——系统便调度下一条原子指令并预测其起始姿态。当前技能的目标姿态、下一技能的预测起始姿态及场景点云随后被传入一个空间感知的代价地图生成模块，该模块基于基础模型 Voxposer [12] 改编。该模块生成一组无碰撞的链路姿态，桥接当前技能目标姿态与下一技能起始姿态之间的空隙，如图 2 第三部分所示。机器人随后基于 RRT [43] 的运动规划在这些链路姿态上进行规划，以确保安全过渡。该技能链模块在 $\mathcal{M}+$ DeCo 的原子技能交接过程中运行，实现了顺序技能的平滑组合和长时序三维操作任务的可靠执行。

4 DeCoBench : Benchmarking Multi-task IL Generalization on Compositional Long-Horizon 3D Manipulation Tasks

4 DeCoBench: 基于组合性长时序三维操作任务的多任务模仿学习泛化基准

To better evaluate DeCo's performance in simulation, we introduce DeCoBench-a benchmark built on the physical interaction-based task decomposition described in Sec. 3.2 (overview in Figure 7, details in Appendix D). It includes 22 tabletop manipulation tasks: 10 atomic tasks (with 24 variations) for training, and 12 compositional long-horizon tasks (with 36 variations) for zero-shot evaluation. DeCoBench is designed to assess the zero-shot generalization of multi-task IL models on novel yet compositional 3D manipulation tasks. The benchmark spans three domains: Object Rearrangement with Drawer, Object Rearrangement with Cupboard, and Rubbish Cleanup. In the Drawer domain, 2 original IL tasks are decomposed into 5 atomic skills based on full physical interaction, yielding tasks with 4, 6, and 10 cycles. The Cupboard domain includes 3 atomic tasks and a 4-cycle long-horizon task, while the Cleanup domain includes 2 atomic tasks forming another 4-cycle task. DeCoBench also includes two cross-domain tasks designed to evaluate models' cross-task generalization capabilities: Transfer Box (Drawer + Cupboard) and Retrieve and Sweep Rubbish (Cupboard + Cleanup). See Appendix D for further details about the DeCoBench tasks.

为更好地评估 DeCo 在仿真中的表现，我们引入了 DeCoBench——一个基于第 3.2 节所述物理交互任务分解构建的基准 (概览见图 7，详情见附录 D)。该基准包含 22 个桌面操作任务:10 个原子任务 (含 24 个变体) 用于训练，12 个组合性长时序任务 (含 36 个变体) 用于零样本评估。DeCoBench 旨在评估多任务模仿学习模型在新颖且具有组合性的三维操作任务上的零样本泛化能力。该基准涵盖三个领域: 带抽屉的物体重排、带橱柜的物体重排和垃圾清理。在抽屉领域，2 个原始 IL 任务基于完整物理交互分解为 5 个原子技能，形成包含 4、6 和 10 个周期的任务。橱柜领域包括 3 个原子任务和一个 4 周期的长时序任务，清理领域包含 2 个原子任务组成的另一个 4 周期任务。DeCoBench 还包括两个跨领域任务，用于评估模型的跨任务泛化能力: 转移箱子 (抽屉 + 橱柜) 和取回并清扫垃圾 (橱柜 + 清理)。有关 DeCoBench 任务的更多细节，请参见附录 D。

5 Experiments

5 实验

We study DeCo in both simulated and real-world environments. Specifically, we aims to answer the following research questions: 1) How well does DeCo enhance the generalization of multitask IL models on long-horizon 3D manipulation tasks (Sec. 5.2)? 2) How do heuristic settings in DeCo influence its generalization performance (Sec. 5.3)? 3) How well does DeCo enhance the generalization of multi-task IL model on real-world long-horizon manipulation tasks (Sec. 5.4)?

我们在模拟和真实环境中研究 DeCo。具体来说，我们旨在回答以下研究问题:1)DeCo 在长时域三维操作任务上对多任务模仿学习 (IL) 模型的泛化能力提升效果如何 (第 5.2 节)? 2)DeCo 中的启发式设置如何影响其泛化性能 (第 5.3 节)? 3)DeCo 在真实世界长时域操作任务上对多任务 IL 模型的泛化能力提升效果如何 (第 5.4 节)?

5.1 Experimental Setup

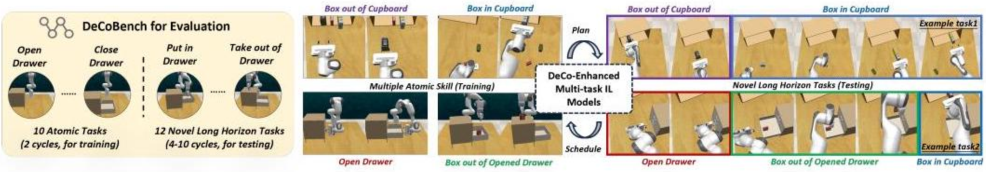
5.1 实验设置

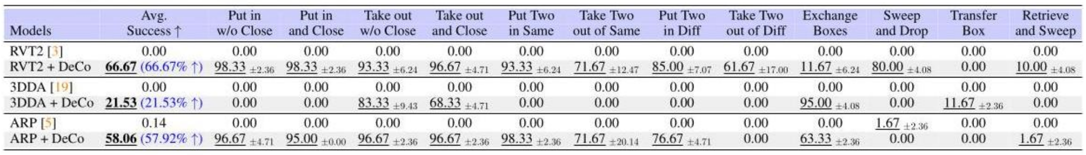
Baseline Multi-task IL Models. We apply DeCo to three representative multi-task IL models—RVT-2 [3], 3DDA [19], and ARP [5]—to validate its model-agnostic design and demonstrate its generalization benefits. RVT-2 is a multi-view robotic transformer that follows a coarse-to-fine strategy on constructed point clouds to predict the next-best action heatmap. 3DDA combines 3D scene representations with a diffusion-based policy for robotic manipulation. ARP leverages a Chunking Causal Transformer [5] to autoregressively generate heterogeneous action sequences for manipulation tasks. All three models have demonstrated competitive multi-task IL performance on the RLBench benchmark [1] .

基线多任务 IL 模型。我们将 DeCo 应用于三种代表性多任务 IL 模型——RVT-2 [3]、3DDA [19] 和 ARP [5]——以验证其模型无关设计并展示其泛化优势。RVT-2 是一种多视角机器人变换器，采用粗到细策略在构建的点云上预测下一步最佳动作热图。3DDA 结合了三维场景表示与基于扩散的机器人操作策略。ARP 利用分块因果变换器 (Chunking Causal Transformer)[5] 自回归生成异构动作序列以完成操作任务。三种模型均在 RLBench 基准测试中展现了竞争力的多任务 IL 性能 [1] 。

Simulation Setup. We conduct simulation experiments using our proposed DeCoBench benchmark suite. Observations are collected from four RGB-D cameras positioned at the front, left shoulder, right shoulder, and wrist. RVT-2 and ARP use 128×128 image inputs, while 3DDA uses 256×256 , following their original settings. Each baseline and its DeCo-enhanced variant is trained on atomic tasks (50 demonstrations per task) and evaluated on compositional tasks (20 test demonstrations per task). All policies are evaluated with three random seeds, and standard deviations are reported.

模拟设置。我们使用提出的 DeCoBench 基准套件进行模拟实验。观察数据由位于前方、左肩、右肩和手腕的四个 RGB-D 摄像头采集。RVT-2 和 ARP 使用 128 ×128 图像输入，3DDA 使用 256 × 256 ，均遵循其原始设置。每个基线模型及其 DeCo 增强版本均在原子任务 (每任务 50 个示范) 上训练，并在组合任务 (每任务 20 个测试示范) 上评估。所有策略均使用三个随机种子进行评估，并报告标准差。





Models	Avg. Success ↑	Put in w/o Close	Put in and Close	Take out w/o Close	Take out and Close	Put Two in Same	Take Two out of Same	Put Two in Diff	Take Two out of Diff	Exchange Boxes	Sweep and Drop	Transfer Box	Retrieve and Sweep
RVT2 [3]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
RVT2 + DeCo	**66.67** (66.67% ↑)	98.33 ±2.36	98.33 ±2.36	93.33 ±6.24	96.67 ±4.71	93.33 ±6.24	71.67 ±12.47	85.00 ±7.07	61.67 ±17.00	11.67 ±6.24	80.00 ±4.08	0.00	10.00 ±4.08
3DDA [19]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3DDA + DeCo	**21.53** (21.53% ↑)	0.00	0.00	83.33 ±9.43	68.33 ±4.71	0.00	0.00	0.00	0.00	95.00 ±4.08	0.00	11.67 ±2.36	0.00
ARP [5]	0.14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.67 ±2.36	0.00	0.00
ARP + DeCo	**58.06** (57.92% ↑)	96.67 ±4.71	95.00 ±0.00	96.67 ±2.36	96.67 ±2.36	98.33 ±2.36	71.67 ±20.14	76.67 ±4.71	0.00	63.33 ±2.36	0.00	0.00	1.67 ±2.36

Table 1: Generalization Performance on DeCoBench Long-horizon tasks. Above is a visualization illustrating how DeCo enables zero-shot generalization on two long-horizon tasks from DeCoBench.

表 1:DeCoBench 长时域任务的泛化性能。上图展示了 DeCo 如何实现对 DeCoBench 中两个长时域任务的零样本泛化能力。

Real-robot Setup. We validate DeCo on a Franka Emika Panda robot equipped with an exocentric Intel RealSense D435i camera. We compare RVT-2 and RVT-2+DeCo on an object rearrangement task involving a drawer. Training uses 6 atomic tasks (16 variations), while evaluation covers 9 long-horizon tasks (30 variations) for zero-shot generalization. The test set includes 3 tasks with 4 cycles, 2 with 6, 2 with 12, and 2 with 16 cycles. see Appendix E for more details. Each task is executed 10 times with randomized initial object placements to compute average success rates.

真实机器人设置。我们在配备外视 Intel RealSense D435i 摄像头的 Franka Emika Panda 机器人上验证 DeCo。我们比较了 RVT-2 与 RVT-2+DeCo 在涉及抽屉的物体重排任务上的表现。训练使用 6 个原子任务 (16 种变体)，评估涵盖 9 个长时域任务 (30 种变体) 以实现零样本泛化。测试集包括 3 个任务各 4 个循环，2 个任务各 6 个循环，2 个任务各 12 个循环，2 个任务各 16 个循环。详见附录 E。每个任务执行 10 次，初始物体位置随机，以计算平均成功率。

5.2 Generalization Performance on DeCoBench

5.2 DeCoBench 上的泛化性能

[1] https://paperswithcode.com/sota/robot-manipulation-on-rlbench

[1] https://paperswithcode.com/sota/robot-manipulation-on-rlbench

Table 1 presents the generalization performance on 12 long-horizon tasks in DeCoBench for RVT-2, 3DDA, and ARP models trained on 10 atomic tasks, along with their DeCo-enhanced counterparts.Although RVT- 2, 3DDA, and ARP perform well in atomic tasks (see subsection C. 1 for their performance on atomic tasks), they almost completely fail on 12 long-horizon tasks, indicating that base models struggle to generalize atomic skills to long-horizon scenarios. In contrast, DeCo substantially enhances their performance-yielding a 66.67% gain for RVT-2+DeCo, 21.53% for 3DDA+DeCo, and 57.92% for ARP+DeCo. These results demonstrate DeCo’s model-agnostic design and its effectiveness for zero-shot generalization in multi-task IL models on novel long-horizon tasks through compositional reuse of learned atomic skills.

表 1 展示了在 DeCoBench 中 12 个长时域任务上, 基于 10 个原子任务训练的 RVT-2、3DDA 和 ARP 模型及其 DeCo 增强版本的泛化性能。尽管 RVT-2、3DDA 和 ARP 在原子任务上表现良好 (原子任务性能详见附录 C.1), 但它们在 12 个长时域任务上几乎完全失败, 表明基础模型难以将原子技能泛化到长时域场景。相比之下, DeCo 显著提升了它们的性能——RVT-2+DeCo 提升 66.67%, 3DDA+DeCo 提升 21.53%, ARP+DeCo 提升 57.92%。这些结果证明了 DeCo 的模型无关设计及其通过组合复用已学原子技能, 实现多任务 IL 模型在新颖长时域任务上的零样本泛化的有效性。

Task	RVT-2+DeCo	RVT-2 (6 Long training)
6 Novel	83.89% (53.89% ↑)	30.00%
12 All	66.67% (14.31% T)	52.36%

任务	RVT-2+DeCo	RVT-2(6 次长时间训练)
6 新颖	83.89%(提升 53.89%)	30.00%
12 全部	66.67% (14.31% T)	52.36%

Table 2: Impact of atomic task design.

表 2: 原子任务设计的影响。

To further assess the impact of atomic task design, we train RVT-2 on 6 long-horizon tasks (6 Long) from DeCoBench, comprising 4 original IL tasks and 2 cross-domain tasks (see Appendix D for details). We then evaluate the model on the remaining 6 novel long-horizon tasks not seen during training (6 Novel), as well as on all 12 long-horizon tasks (12 All). As shown in Table 2, compared to RVT-2 trained directly on the 6 long-horizon tasks (RVT-2 (6 Long training)), the DeCo-based variant (RVT-2 + DeCo) achieves substantially better zero-shot generalization on the unseen 6 Novel tasks, improving the success rate by 53.89%. Even when evaluated across all 12 tasks, including those seen by RVT-2 (6 Long training), the DeCo-based model still yields an overall improvement of 14.31%. These results indicate that DeCo’s atomic task training set more effectively supports skill acquisition in multi-task IL models and enhances generalization to novel compositional tasks. Additional analysis on impact of atomic task design are provided in subsection C.3.

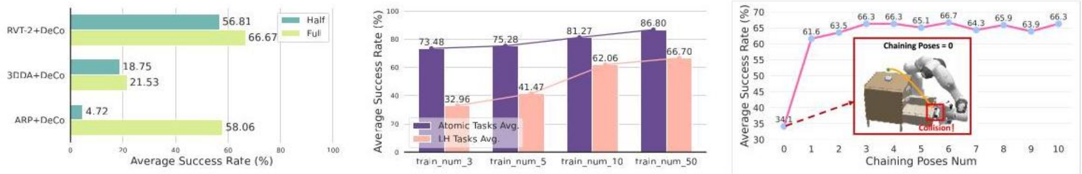
为了进一步评估原子任务设计的影响，我们在 DeCoBench 中的 6 个长时序任务 (6 Long) 上训练 RVT-2 模型，这些任务包括 4 个原始的模仿学习 (IL) 任务和 2 个跨领域任务 (详见附录 D)。随后，我们在训练时未见过的另外 6 个新颖长时序任务 (6 Novel) 以及全部 12 个长时序任务 (12 All) 上对模型进行评估。如表 2 所示，与直接在 6 个长时序任务上训练的 RVT-2(6 Long training) 相比，基于 DeCo 的变体 (RVT-2 + DeCo) 在未见过的 6 个新颖任务上实现了显著更好的零样本泛化，成功率提升了 53.89%。即使在包括训练时见过的 6 个任务的全部 12 个任务上评估，基于 DeCo 的模型仍然整体提升了 14.31%。这些结果表明，DeCo 的原子任务训练集更有效地支持了多任务模仿学习模型的技能习得，并增强了对新颖组合任务的泛化能力。关于原子任务设计影响的更多分析见子章节 C.3。

5.3 Ablation Studies

5.3 消融研究

Figure 4 summarizes three ablation studies on DeCo's heuristic settings. Figure 4a compares the generalization performance of three models using DeCo under half and full interaction settings. The results show that DeCo improves model generalization in both settings, but different models have varying sensitivity to physical interactions. The full interaction (open → closed → open)

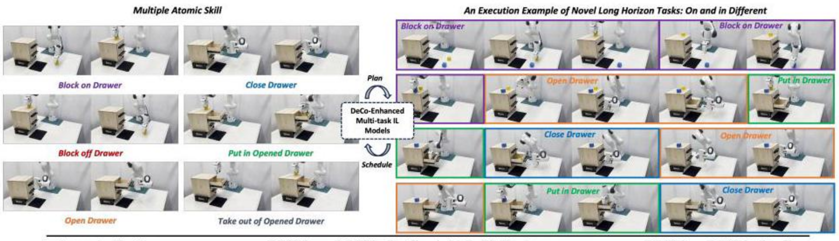
图 4 总结了关于 DeCo 启发式设置的三项消融研究。图 4a 比较了三种模型在半交互和全交互设置下使用 DeCo 的泛化性能。结果显示，DeCo 在两种设置下均提升了模型的泛化能力，但不同模型对物理交互的敏感度存在差异。全交互 (开 → 关 → 开)

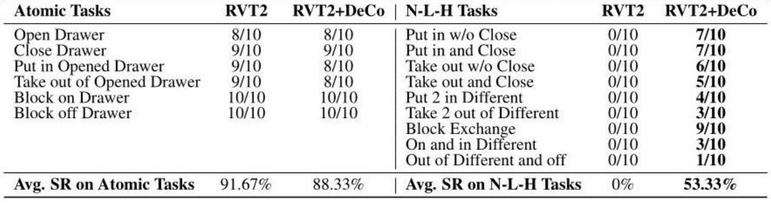


(a) Half vs. Full Interaction (b) Effect of Training Demo Num (c) Effect of Chaining Poses Num

Figure 4: Ablation study of heuristic settings in DeCo. (b) and (c) are based on the RVT-2+DeCo model.

图 4:DeCo 启发式设置的消融研究。(b) 和 (c) 基于 RVT-2+DeCo 模型。





Atomic Tasks	RVT2	RVT2+DeCo	N-L-H Tasks	RVT2	RVT2+DeCo
Open Drawer	8/10	8/10	Put in w/o Close	0/10	**7/10**
Close Drawer	9/10	9/10	Put in and Close	0/10	**7/10**
Put in Opened Drawer	9/10	8/10	Take out w/o Close	0/10	**6/10**
Take out of Opened Drawer	9/10	8/10	Take out and Close	0/10	**5/10**
Block on Drawer	10/10	10/10	Put 2 in Different	0/10	**4/10**
Block off Drawer	10/10	10/10	Take 2 out of Different	0/10	**3/10**
			Block Exchange	0/10	**9/10**
			On and in Different	0/10	**3/10**
			Out of Different and off	0/10	**1/10**
Avg. SR on Atomic Tasks	91.67%	88.33%	**Avg. SR on N-L-H Tasks**	0%	**53.33%**

Table 3: Real-world results. Each entry represents the successful trials out of 10. Above is a visualization example showing how DeCo performs zero-shot generalization in one of the longest-horizon tasks (16 cycles).

表 3: 真实环境结果。每项数据代表 10 次试验中的成功次数。上方为一个可视化示例，展示 DeCo 在最长时序任务之一 (16 个周期) 中的零样本泛化表现。

enhances DeCo's ability to provide better compositional generalization. Notably, ARP+DeCo is most affected under the half interaction, while RVT-2+DeCo and 3DDA+DeCo show only minor changes. For detailed difference of DeCo based on p^{half} and p^{full} , and further analyses of the experimental results, please refer to Appendix A and subsection C.2, respectively.

增强了 DeCo 提供更好组合泛化的能力。值得注意的是，ARP+DeCo 在半交互下受影响最大，而 RVT-2+DeCo 和 3DDA+DeCo 仅有轻微变化。关于基于 p^{half} 和 p^{full} 的 DeCo 差异及实验结果的进一步分析，请参见附录 A 和子章节 C.2。

We also vary the number of atomic task demonstrations for RVT-2+DeCo and report the average success rates over 10 atomic tasks and 12 long-horizon tasks. As shown in Figure 4b, more demonstrations indeed help improve the performance of IL model. As long as the RVT-2 model learns atomic skills to a certain extent, DeCo effectively composes these skills to achieve a degree of zero-shot generalization on long-horizon tasks. Moreover, the better RVT-2 performs on atomic tasks, the better generalization performance DeCo achieves. Figure 4c shows that disabling the spatially-aware skill chaining module (Chaining Poses Num = 0) leads to a significant drop in performance. We visualize a failure case of Put in and Close task, where the robot collides with the drawer after opening it due to a poor transition plan, preventing it from successfully picking up the item. In contrast, regardless of the number of poses, enabling our spatially-aware skill chaining module consistently enhances task success significantly. In DeCo, we heuristically set the chaining poses num to 6 .

我们还改变了 RVT-2+DeCo 的原子任务示范数量，并报告了 10 个原子任务和 12 个长时序任务的平均成功率。如图 4b 所示，更多示范确实有助于提升模仿学习模型的性能。只要 RVT-2 模型在一定程度上学会了原子技能，DeCo 就能有效地组合这些技能，实现对长时序任务的零样本泛化。此外，RVT-2 在原子任务上的表现越好，DeCo 的泛化性能也越佳。图 4c 显示，禁用空间感知技能链模块 (Chaining Poses Num = 0) 会导致性能显著下降。我们可视化了"放入并关闭"任务的失败案例，机器人在打开抽屉后因过渡计划不佳与抽屉碰撞，导致无法成功拾取物品。相比之下，无论姿态数量多少，启用我们的空间感知技能链模块均显著提升了任务成功率。在 DeCo 中，我们启发式地将链式姿态数量设置为 6。

5.4 Real-robot Evaluations

5.4 真实机器人评估

We conduct extensive experiments on a real-world robotic platform to further validate the practical effectiveness of the DeCo framework. As shown in Table 3 compares the success rates of RVT- 2 and RVT-2+DeCo, both trained on 6 atomic tasks, when tested on 9 novel long-horizon (N-L-H) tasks. These tasks are unseen during training but are composable via atomic skills. RVT-2 performs well on atomic tasks but fails to generalize to N-L-H tasks, achieving 0% success. In contrast, RVT-2+DeCo, though slightly less stable on atomic tasks due to vision-language-based decomposition, achieves strong zero-shot generalization, with a 53.33% average success rate on N-L-H tasks. These results confirm DeCo's practicality and generalization ability in real-world robotic settings. Experiment videos are available on the project website.

我们在真实机器人平台上进行了大量实验，以进一步验证 DeCo 框架的实际有效性。如表 3 所示，比较了在 6 个原子任务上训练的 RVT-2 和 RVT-2+DeCo 在 9 个新颖长时序 (N-L-H) 任务上的成功率。这些任务在训练时未见过，但可通过原子技能组合完成。RVT-2 在原子任务上表现良好，但无法泛化到 N-L-H 任务，成功率为 0%。相比之下，尽管由于基于视觉语言的分解，RVT-2+DeCo 在原子任务上稳定性略有下降，但其在 N-L-H 任务上实现了强大的零样本泛化，平均成功率达 53.33%。这些结果证实了 DeCo 在真实机器人环境中的实用性和泛化能力。实验视频可在项目网站观看。

6 Conclusion

6 结论

In this work, we introduce DeCo, a model-agnostic framework that enables multi-task IL models to generalize zero-shot to novel yet compositional long-horizon 3D manipulation tasks. DeCo decomposes IL demonstrations into modular, reusable atomic tasks via physical interaction analysis. At test time, it leverages VLMs to interpret high-level instructions and retrieve relevant skills, enabling flexible planning and scheduling for novel, compositional tasks. A spatially-aware chaining module ensures collision-free skill transitions, addressing temporal and spatial discontinuities in long-horizon execution. Extensive evaluations in both simulation and real-world settings demonstrate that DeCo significantly enhances the generalization of three representative multi-task IL models.

在本研究中，我们提出了 DeCo，一种模型无关的框架，使多任务模仿学习 (IL) 模型能够零样本泛化到新颖且具有组合性的长时序三维操作任务。DeCo 通过物理交互分析将模仿学习示范分解为模块化、可复用的原子任务。在测试阶段，它利用视觉语言模型 (VLMs) 来理解高级指令并检索相关技能，从而实现对新颖组合任务的灵活规划与调度。一个具备空间感知能力的链式模块确保技能切换时无碰撞，解决了长时序执行中的时间和空间不连续性问题。在仿真和真实环境中的大量评估表明，DeCo 显著提升了三种代表性多任务模仿学习模型的泛化能力。

7 Limitations and Discussion

7 限制与讨论

7.1 Limitations

7.1 限制

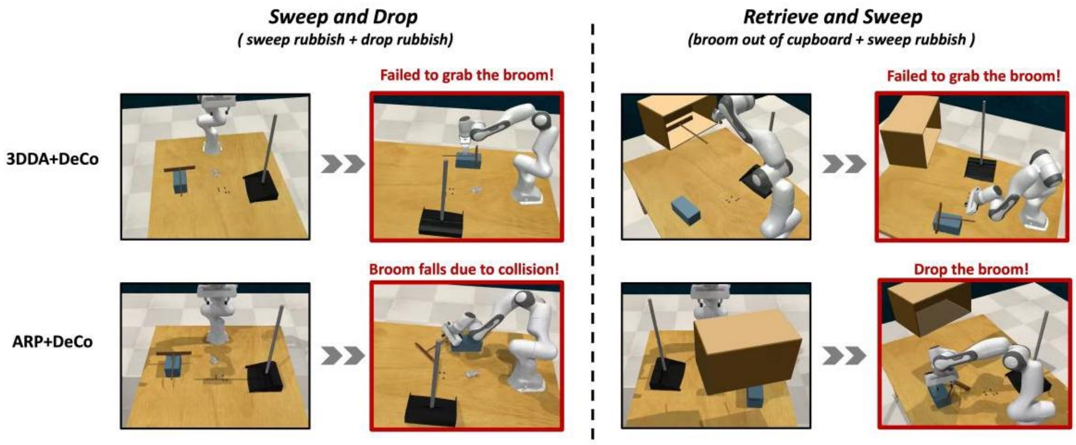

Sweep and Drop
(sweep rubbish + drop rubbish)
Retrieve and Sweep
(broom out of cupboard + sweep rubbish)
Failed to grab the broom!
Failed to grab the broom!
3DDA+DeCo
Broom falls due to collision!
Drop the broom!
ARP+DeCo

Figure 5: Two visual failure cases of 3DDA+DeCo and ARP+DeCo.

图 5:3DDA+DeCo 和 ARP+DeCo 的两个视觉失败案例。

While DeCo demonstrates strong zero-shot generalization for novel yet compositional long-horizon 3D manipulation tasks across three representative multi-task imitation learning (IL) models (RVT- 2 [3], 3DDA [19] and ARP [5]), some limitations remain:

虽然 DeCo 在三个具有代表性的多任务模仿学习 (IL) 模型 (RVT-2 [3]、3DDA [19] 和 ARP [5]) 中，对新颖但具有组合性的长时序三维操作任务表现出强大的零样本泛化能力，但仍存在一些局限性:

- Dependency on Base Model: The ability of the base multi-task IL model to learn atomic skills is crucial for DeCo's generalization performance in long-horizon tasks (see Figure 4b. However, the model's generalization capability relies not only on its learning of atomic skills but also on other factors. Visual robustness is a key factor: different models have varying levels of visual robustness when confronted with previously unseen combinations in scenarios. If the base multi-task IL model's visual capability cannot effectively handle these variations, it will directly impact DeCo's ability to generalize in multi-task IL models when handling combined long-horizon tasks. We provide visual failure cases of 3DDA+DeCo and ARP+DeCo in Figure 5 to further illustrate this limitation. Although 3DDA+DeCo and ARP+DeCo excel in learning atomic tasks (see Table 5), they encounter failures when facing the compositional long-horizon tasks Sweep and Drop (sweep rubbish + drop rubbish) and Retrieve and Sweep (broom out of cupboard + sweep rubbish). Even though DeCo can plan and schedule the corresponding atomic skills, both 3DDA and ARP struggle with visual processing in unseen combination scenarios. As a result, 3DDA+DeCo and ARP+DeCo fail to execute the atomic skills, which prevents them from completing the entire long-horizon tasks. A future direction to address this issue could involve adding a visual enhancement module [44, 45] to the base multi-task IL model in DeCo to improve its visual robustness in unseen scenarios.

- 对基础模型的依赖: 基础多任务模仿学习 (IL) 模型学习原子技能的能力对于 DeCo 在长时序任务中的泛化性能至关重要 (见图 4b)。然而，模型的泛化能力不仅依赖于其对原子技能的学习, 还受其他因素影响。视觉鲁棒性是关键因素: 不同模型在面对先前未见过的场景组合时, 视觉鲁棒性存在差异。如果基础多任务 IL 模型的视觉能力无法有效应对这些变化，将直接影响 DeCo 在处理组合长时序任务时的多任务 IL 模型泛化能力。我们在图 5 中提供了 3DDA+DeCo 和 ARP+DeCo 的视觉失败案例，以进一步说明这一限制。尽管 3DDA+DeCo 和 ARP+DeCo 在学习原子任务方面表现出色 (见表 5)，但在面对组合长时序任务 Sweep 和 Drop(清扫垃圾) 时仍会出现失败情况

+ 扔垃圾) 和取回及扫地 (从橱柜取出扫帚 + 扫垃圾)。尽管 DeCo 能够规划和调度相应的原子技能，但 3DDA 和 ARP 在未见过的组合场景中都难以处理视觉信息。因此，3DDA+DeCo 和 ARP+DeCo 无法执行原子技能，导致它们无法完成整个长时任务。未来解决该问题的方向可能是在 DeCo 的基础多任务模仿学习 (IL) 模型中添加视觉增强模块 [44, 45]，以提升其在未见场景中的视觉鲁棒性。

- Dependency of Task Planning on VLM Capabilities: The effectiveness of DeCo in task planning is closely tied to the capabilities of visual language models (VLMs). Therefore, if the VLM falls short in accurately understanding the spatial relationships and contextual combinations of task language instructions in scenarios, DeCo's planning performance may be negatively impacted (see subsection C.1). This dependency could hinder DeCo's ability to generalize effectively in diverse and dynamic environments, especially when confronted with novel long-horizon tasks that require complex and advanced integration of visual information and instructions across multiple task domains. To address this limitation, a Human-in-the-Loop feedback mechanism [46] can be implemented, enabling human operators to provide corrections and insights during task execution. This interaction can refine the VLM's understanding and enhance DeCo's overall performance in complex scenarios. Additionally, DeCo currently relies on GPT-4o as its VLM; exploring alternative models such as Gemini [47], DeepSeek-VL [48], or LLaMA [49] in the future may provide deeper insights into how VLM model selection affects DeCo's performance and generalization

- 任务规划对视觉语言模型 (VLM) 能力的依赖:DeCo 在任务规划中的有效性与视觉语言模型 (VLM) 的能力密切相关。因此，如果 VLM 在准确理解场景中任务语言指令的空间关系和上下文组合方面存在不足，DeCo 的规划性能可能会受到负面影响 (参见子章节 C.1)。这种依赖性可能限制 DeCo 在多样且动态环境中的有效泛化能力，尤其是在面对需要跨多个任务领域复杂且高级视觉信息与指令整合的新颖长时任务时。为解决此限制，可引入人机交互反馈机制 [46]，使人工操作员在任务执行过程中提供修正和见解。此类交互能够优化 VLM 的理解能力，提升 DeCo 在复杂场景中的整体表现。此外，DeCo 目前依赖 GPT-4o 作为其 VLM；未来探索如 Gemini[47]、DeepSeek-VL[48] 或 LLaMA[49] 等替代模型，或能深入揭示 VLM 模型选择对 DeCo 性能及泛化能力的影响。

- Atomic Task Horizon: The atomic tasks in DeCo are currently constrained to short horizons-either 1 cycle (representing half interaction) or 2 cycles (representing a full interaction). It remains an open question how DeCo would perform with longer-horizon atomic tasks, such as those with intermediate durations ($1 < \text{cycles} < 2$) or extended sequences ($\text{cycles} > 2$), and whether such variations would facilitate or hinder generalization when composing more complex tasks from these primitives.

- 原子任务视界:DeCo 中的原子任务目前被限制在较短的视界——要么为 1 个周期 (表示半次交互)、要么为 2 个周期 (表示一次完整交互)。对于 DeCo 在更长视界下的原子任务，例如具有中间持续时长 ($1 < \text{cycles} < 2$) 或扩展序列 (周期 > 2) 的任务，其表现如何，以及此类变化在从这些原语组合更复杂任务时是促进还是阻碍泛化，仍是一个悬而未决的问题。

- Definition of Atomic Tasks and Detection of Skill Completion: The current definition of atomic tasks in DeCo is derived from the gripper's open/close state. This simplistic criterion may be insufficient for detecting meaningful task boundaries in more complex scenarios, such as multi-stage tool manipulation, deformable object handling, or non-prehensile manipulation. Furthermore, reliable detection of atomic skill completion under such conditions remains an unresolved challenge, particularly when transitioning to more versatile or dexterous robot hardware. To address this limitation, incorporating tactile or force modalities [50, 51] in the future can provide real-time information about the state of objects and the environment, helping DeCo to flexibly divide atomic tasks and more accurately assess task completion.

- 原子任务的定义及技能完成的检测:DeCo 中当前对原子任务的定义基于夹爪的开合状态。这一简单的标准在更复杂的场景中，如多阶段工具操作、可变形物体处理或非抓取式操作，可能不足以检测有意义的任务边界。此外，在此类条件下可靠检测原子技能完成仍是一个未解决的难题，尤其是在向更通用或灵巧的机器人硬件过渡时。为解决此限制，未来引入触觉或力觉模态 [50, 51]，可提供关于物体和环境状态的实时信息，帮助 DeCo 灵活划分原子任务并更准确地评估任务完成情况。

7.2 Future Work

7.2 未来工作

We plan to extend DeCo along several key directions. First, we will explore the flexibility of its physically grounded task decomposition by incorporating tactile and force-based modalities, enabling richer manipulation capabilities such as dexterous control, deformable object handling, and multi-step tool use. Second, we aim to integrate DeCo with advanced vision-language-action (VLA) models [52, 53, 54] and deploy it across a broader range of robotic platforms, including mobile manipulators and dexterous hands. A key challenge lies in ensuring robust generalization across complex tasks, which requires improving atomic skill completion detection, enhancing transition reliability, and addressing the scalability limits of a growing skill library. While expanding the skill set increases expressiveness and task coverage, it also raises demands on data, training, and inference. Balancing these trade-offs will be critical to realizing DeCo's full potential in real-world, long-horizon manipulation. References

> 我们计划沿若干关键方向扩展 DeCo。首先，将通过引入触觉和基于力的模态，探索其物理基础任务分解的灵活性，从而实现更丰富的操作能力，如灵巧控制、可变形物体处理和多步骤工具使用。其次，旨在将 DeCo 与先进的视觉-语言-动作 (VLA) 模型 [52, 53, 54] 集成，并部署于更广泛的机器人平台，包括移动操作臂和灵巧机械手。一个关键挑战是确保在复杂任务中的鲁棒泛化，这需要提升原子技能完成检测、增强转换可靠性，并解决技能库扩展的可扩展性限制。虽然扩展技能集提升了表达力和任务覆盖，但也增加了对数据、训练和推理的需求。平衡这些权衡对于实现 DeCo 在现实世界长时域操作中的全部潜力至关重要。参考文献

[1] M. Shridhar, L. Manuelli, and D. Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In Conference on Robot Learning, pages 785-799. PMLR, 2023.

M. Shridhar, L. Manuelli, and D. Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In Conference on Robot Learning, pages 785-799. PMLR, 2023.

[2] A. Goyal, J. Xu, Y. Guo, V. Blukis, Y.-W. Chao, and D. Fox. Rvt: Robotic view transformer for 3d object manipulation. In Conference on Robot Learning, pages 694-710. PMLR, 2023.

A. Goyal, J. Xu, Y. Guo, V. Blukis, Y.-W. Chao, and D. Fox. Rvt: Robotic view transformer for 3d object manipulation. In Conference on Robot Learning, pages 694-710. PMLR, 2023.

[3] A. Goyal, V. Blukis, J. Xu, Y. Guo, Y.-W. Chao, and D. Fox. Rvt2: Learning precise manipulation from few demonstrations. RSS, 2024.

A. Goyal, V. Blukis, J. Xu, Y. Guo, Y.-W. Chao, and D. Fox. Rvt2: Learning precise manipulation from few demonstrations. RSS, 2024.

[4] J. Jiang, X. Wu, Y. He, L. Zeng, Y. Wei, D. Zhang, and W. Zheng. Rethinking bimanual robotic manipulation: Learning with decoupled interaction framework, 2025.

J. Jiang, X. Wu, Y. He, L. Zeng, Y. Wei, D. Zhang, and W. Zheng. Rethinking bimanual robotic manipulation: Learning with decoupled interaction framework, 2025.

[5] X. Zhang, Y. Liu, H. Chang, L. Schramm, and A. Boularias. Autoregressive action sequence learning for robotic manipulation. IEEE Robotics and Automation Letters, 2025.

X. Zhang, Y. Liu, H. Chang, L. Schramm, and A. Boularias. Autoregressive action sequence learning for robotic manipulation. IEEE Robotics and Automation Letters, 2025.

[6] R. Garcia, S. Chen, and C. Schmid. Towards generalizable vision-language robotic manipulation: A benchmark and llm-guided 3d policy. arXiv preprint arXiv:2410.01345, 2024.

> R. Garcia, S. Chen, and C. Schmid. Towards generalizable vision-language robotic manipulation: A benchmark and llm-guided 3d policy. arXiv preprint arXiv:2410.01345, 2024.

[7] V. Myers, C. Zheng, O. Mees, K. Fang, and S. Levine. Policy adaptation via language optimization: Decomposing tasks for few-shot imitation. In 8th Annual Conference on Robot Learning, 2024.

> V. Myers, C. Zheng, O. Mees, K. Fang, and S. Levine. Policy adaptation via language optimization: Decomposing tasks for few-shot imitation. In 8th Annual Conference on Robot Learning, 2024.

[8] A. Curtis, N. Kumar, J. Cao, T. Lozano-Pérez, and L. P. Kaelbling. Trust the proc3s: Solving long-horizon robotics problems with llms and constraint satisfaction. In 8th Annual Conference on Robot Learning, 2024.

> A. Curtis, N. Kumar, J. Cao, T. Lozano-Pérez, and L. P. Kaelbling. Trust the proc3s: Solving long-horizon robotics problems with llms and constraint satisfaction. In 8th Annual Conference on Robot Learning, 2024.

[9] J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. Florence, and A. Zeng. Code as policies: Language model programs for embodied control. In 2023 IEEE International Conference on Robotics and Automation (ICRA), pages 9493-9500. IEEE, 2023.

> J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. Florence, and A. Zeng. Code as policies: Language model programs for embodied control. In 2023 IEEE International Conference on Robotics and Automation (ICRA), pages 9493-9500. IEEE, 2023.

[10] Z. Chen, J. Huo, Y. Chen, and Y. Gao. Robohorizon: An llm-assisted multi-view world model for long-horizon robotic manipulation. arXiv preprint arXiv:2501.06605, 2025.

> Z. Chen, J. Huo, Y. Chen, and Y. Gao. Robohorizon: An llm-assisted multi-view world model for long-horizon robotic manipulation. arXiv preprint arXiv:2501.06605, 2025.

[11] W. Huang, C. Wang, Y. Li, R. Zhang, and L. Fei-Fei. Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation. arXiv preprint arXiv:2409.01652, 2024.

> W. Huang, C. Wang, Y. Li, R. Zhang, 和 L. Fei-Fei. Rekep: 用于机器人操作的关系关键点约束的时空推理。arXiv 预印本 arXiv:2409.01652, 2024。

[12] W. Huang, C. Wang, R. Zhang, Y. Li, J. Wu, and L. Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models. arXiv preprint arXiv:2307.05973, 2023.

> W. Huang, C. Wang, R. Zhang, Y. Li, J. Wu, 和 L. Fei-Fei. Voxposer: 用于机器人操作的可组合三维价值图与语言模型。arXiv 预印本 arXiv:2307.05973, 2023。

[13] Y. Chen, Z. Chen, J. Yin, J. Huo, P. Tian, J. Shi, and Y. Gao. Gravmad: Grounded spatial value maps guided action diffusion for generalized 3 d manipulation. arXiv preprint arXiv:2409.20154, 2024.

> Y. Chen, Z. Chen, J. Yin, J. Huo, P. Tian, J. Shi, 和 Y. Gao. Gravmad: 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。

[14] Z. Chen, Z. Ji, J. Huo, and Y. Gao. Scar: Refining skill chaining for long-horizon robotic manipulation via dual regularization. Advances in Neural Information Processing Systems, 37: 111679-111714, 2024.

Z. Chen, Z. Ji, J. Huo, 和 Y. Gao. Scar: 通过双重正则化优化长时域机器人操作的技能链。神经信息处理系统进展，37: 111679-111714, 2024。

[15] Y. Chen, C. Wang, L. Fei-Fei, and C. K. Liu. Sequential dexterity: Chaining dexterous policies for long-horizon manipulation. arXiv preprint arXiv:2309.00987, 2023.

Y. Chen, C. Wang, L. Fei-Fei, 和 C. K. Liu. 顺序灵巧性: 为长时域操作串联灵巧策略。arXiv 预印本 arXiv:2309.00987, 2023。

[16] G. Tziafas and H. Kasaei. Lifelong robot library learning: Bootstrapping composable and generalizable skills for embodied control with language models. In 2024 IEEE International Conference on Robotics and Automation (ICRA), pages 515-522. IEEE, 2024.

G. Tziafas 和 H. Kasaei. 终身机器人库学习: 利用语言模型引导的可组合且可泛化技能的自举，用于具身控制。2024 年 IEEE 国际机器人与自动化会议 (ICRA)，页 515-522。IEEE, 2024。

[17] N. Saito, J. Moura, T. Ogata, M. Y. Aoyama, S. Murata, S. Sugano, and S. Vijayakumar. Structured motion generation with predictive learning: Proposing subgoal for long-horizon manipulation. In 2023 IEEE International Conference on Robotics and Automation (ICRA), pages 9566-9572. IEEE, 2023.

N. Saito, J. Moura, T. Ogata, M. Y. Aoyama, S. Murata, S. Sugano, 和 S. Vijayakumar. 结构化运动生成与预测学习: 为长时域操作提出子目标。2023 年 IEEE 国际机器人与自动化会议 (ICRA), 页 9566-9572。IEEE, 2023。

[18] S. James, Z. Ma, D. R. Arrojo, and A. J. Davison. Rlbench: The robot learning benchmark & learning environment. IEEE Robotics and Automation Letters, 5(2):3019-3026, 2020.

S. James, Z. Ma, D. R. Arrojo, 和 A. J. Davison. Rlbench: 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。

[19] T.-W. Ke, N. Gkanatsios, and K. Fragkiadaki. 3d diffuser actor: Policy diffusion with 3d scene representations. arXiv preprint arXiv:2402.10885, 2024.

T.-W. Ke, N. Gkanatsios, 和 K. Fragkiadaki. 3d diffuser actor: 基于三维场景表示的策略扩散。arXiv 预印本 arXiv:2402.10885, 2024。

[20] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. De-hghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929, 2020.

A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. De-hghani, M. Minderer, G. Heigold, S. Gelly, 等. 一张图像胜过 16x16 个词: 大规模图像识别的 Transformer。arXiv 预印本 arXiv:2010.11929, 2020。

[21] J. Liang, B. Wen, K. E. Bekris, and A. Boularias. Learning sensorimotor primitives of sequential manipulation tasks from visual demonstrations. In Proceedings of the 2022 International Conference on Robotics and Automation (ICRA), 2022.

J. Liang, B. Wen, K. E. Bekris, 和 A. Boularias. 从视觉示范中学习顺序操作任务的传感器运动原语。2022 年国际机器人与自动化会议 (ICRA) 论文集，2022。

[22] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn. Learning fine-grained bimanual manipulation with low-cost hardware. arXiv preprint arXiv:2304.13705, 2023.

T. Z. Zhao, V. Kumar, S. Levine, 和 C. Finn. 使用低成本硬件学习细粒度双手操作。arXiv 预印本 arXiv:2304.13705, 2023。

[23] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. The International Journal of Robotics Research, page 02783649241273668, 2023.

C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, 和 S. Song. 扩散策略: 通过动作扩散进行视觉运动策略学习。国际机器人研究杂志，页 02783649241273668, 2023。

[24] A. Zeng, P. Florence, J. Tompson, S. Welker, J. Chien, M. Attarian, T. Armstrong, I. Krasin, D. Duong, V. Sindhwani, et al. Transporter networks: Rearranging the visual world for robotic manipulation. In Conference on Robot Learning, pages 726-747. PMLR, 2021.

A. Zeng, P. Florence, J. Tompson, S. Welker, J. Chien, M. Attarian, T. Armstrong, I. Krasin, D. Duong, V. Sindhwani, 等. 运输者网络: 为机器人操作重新排列视觉世界。机器人学习会议论文集，页 726-747。PMLR, 2021。

[25] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Haus-man, A. Herzog, J. Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. arXiv preprint arXiv:2212.06817, 2022.

A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Haus-man, A. Herzog, J. Hsu, 等. Rt-1: 面向大规模现实世界控制的机器人变换器 (Robotics Transformer)。arXiv 预印本 arXiv:2212.06817, 2022.

[26] M. Shridhar, L. Manuelli, and D. Fox. Cliport: What and where pathways for robotic manipulation. In Conference on robot learning, pages 894-906. PMLR, 2022.

M. Shridhar, L. Manuelli, 和 D. Fox. Cliport: 机器人操作的“什么”和“哪里”路径。在机器人学习会议 (Conference on Robot Learning)，第 894-906 页。PMLR, 2022.

[27] Z. Chen, Z. Ji, S. Liu, J. Huo, Y. Chen, and Y. Gao. Casil: Cognizing and imitating skills via a dual cognition-action architecture. arXiv preprint arXiv:2309.16299, 2023.

Z. Chen, Z. Ji, S. Liu, J. Huo, Y. Chen, 和 Y. Gao. Casil: 通过双重认知-动作架构实现技能认知与模仿。arXiv 预印本 arXiv:2309.16299, 2023.

[28] Z. Chen, W. Li, Y. Gao, and Y. Chen. Tild: Third-person imitation learning by estimating domain cognitive differences of visual demonstrations. In Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems, pages 2421-2423, 2023.

Z. Chen, W. Li, Y. Gao, 和 Y. 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。在 2023 年国际自主代理与多智能体系统会议论文集，第 2421-2423 页，2023.

[29] S. James, K. Wada, T. Laidlow, and A. J. Davison. Coarse-to-fine q-attention: Efficient learning for visual robotic manipulation via discretisation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 13739-13748, 2022.

S. James, K. Wada, T. Laidlow, 和 A. J. Davison. 粗到细的 q-注意力: 通过离散化实现视觉机器人操作的高效学习。在 IEEE/CVF 计算机视觉与模式识别会议 (CVPR) 论文集，第 13739-13748 页，2022.

[30] P.-L. Guhur, S. Chen, R. G. Pinel, M. Tapaswi, I. Laptev, and C. Schmid. Instruction-driven history-aware policies for robotic manipulations. In Conference on Robot Learning, pages 175-187. PMLR, 2023.

P.-L. Guhur, S. Chen, R. G. Pinel, M. Tapaswi, I. Laptev, 和 C. Schmid. 基于指令驱动的历史感知策略用于机器人操作。在机器人学习会议 (Conference on Robot Learning)，第 175-187 页。PMLR, 2023.

[31] T. Gervet, Z. Xian, N. Gkanatsios, and K. Fragkiadaki. Act3d: Infinite resolution action detection transformer for robotic manipulation. arXiv preprint arXiv:2306.17817, 2023.

T. Gervet, Z. Xian, N. Gkanatsios, 和 K. Fragkiadaki. Act3d: 用于机器人操作的无限分辨率动作检测变换器。arXiv 预印本 arXiv:2306.17817, 2023.

[32] Z. Xian, N. Gkanatsios, T. Gervet, T.-W. Ke, and K. Fragkiadaki. Chaineddiffuser: Unifying trajectory diffusion and keypose prediction for robotic manipulation. In 7th Annual Conference on Robot Learning, 2023.

Z. Xian, N. Gkanatsios, T. Gervet, T.-W. Ke, 和 K. Fragkiadaki. Chaineddiffuser: 统一轨迹扩散与关键姿态预测用于机器人操作。在第七届机器人学习年会，2023.

[33] T. Gao, S. Nasiriany, H. Liu, Q. Yang, and Y. Zhu. Prime: Scaffolding manipulation tasks with behavior primitives for data-efficient imitation learning. IEEE Robotics and Automation Letters, 2024.

T. Gao, S. Nasiriany, H. Liu, Q. Yang, 和 Y. Zhu. Prime: 利用行为原语搭建操作任务框架，实现数据高效的模仿学习。IEEE 机器人与自动化快报，2024.

[34] U. A. Mishra, S. Xue, Y. Chen, and D. Xu. Generative skill chaining: Long-horizon skill planning with diffusion models. In Conference on Robot Learning, pages 2905-2925. PMLR, 2023.

> U. A. Mishra, S. Xue, Y. Chen, 和 D. Xu. 生成式技能链: 基于扩散模型的长时域技能规划。在机器人学习会议 (Conference on Robot Learning)，第 2905-2925 页。PMLR, 2023.

[35] C. Agia, T. Migimatsu, J. Wu, and J. Bohg. Stap: Sequencing task-agnostic policies. In 2023 IEEE International Conference on Robotics and Automation (ICRA), pages 7951-7958. IEEE, 2023.

> C. Agia, T. Migimatsu, J. Wu, 和 J. Bohg. Stap: 任务无关策略的序列化。在 2023 年 IEEE 国际机器人与自动化会议 (ICRA)，第 7951-7958 页。IEEE, 2023.

[36] Y. Hou, J. Ma, H. Sun, and F. Wu. Effective offline robot learning with structured task graph. IEEE Robotics and Automation Letters, 9(4):3633-3640, 2024.

> Y. Hou, J. Ma, H. Sun, 和 F. Wu. 结构化任务图下的高效离线机器人学习。IEEE 机器人与自动化快报，9(4):3633-3640, 2024.

[37] K. Zentner, R. Julian, B. Ichter, and G. S. Sukhatme. Conditionally combining robot skills using large language models. In 2024 IEEE International Conference on Robotics and Automation (ICRA), pages 14046-14053. IEEE, 2024.

> K. Zentner, R. Julian, B. Ichter, 和 G. S. Sukhatme. 利用大型语言模型有条件地组合机器人技能。在 2024 年 IEEE 国际机器人与自动化会议 (ICRA)，第 14046-14053 页。IEEE, 2024.

[38] Z. Zhang, Y. Li, O. Bastani, A. Gupta, D. Jayaraman, Y. J. Ma, and L. Weihs. Universal visual decomposer: Long-horizon manipulation made easy. In 2024 IEEE International Conference on Robotics and Automation (ICRA), pages 6973-6980. IEEE, 2024.

> Z. Zhang, Y. Li, O. Bastani, A. Gupta, D. Jayaraman, Y. J. Ma, 和 L. Weihs. 通用视觉分解器: 简化长时域操作。2024 年 IEEE 国际机器人与自动化会议 (ICRA)，第 6973-6980 页。IEEE, 2024.

[39] C. Zhao, S. Yuan, C. Jiang, J. Cai, H. Yu, M. Y. Wang, and Q. Chen. Erra: An embodied representation and reasoning architecture for long-horizon language-conditioned manipulation tasks. IEEE Robotics and Automation Letters, 8(6):3230-3237, 2023.

> C. Zhao, S. Yuan, C. Jiang, J. Cai, H. Yu, M. Y. Wang, 和 Q. Chen. Erra: 面向长时域语言条件操作任务的具身表示与推理架构。IEEE 机器人与自动化快报，8(6):3230-3237, 2023.

[40] S. Cheng and D. Xu. League: Guided skill learning and abstraction for long-horizon manipulation. IEEE Robotics and Automation Letters, 8(10):6451-6458, 2023.

> S. Cheng 和 D. Xu. League: 用于长时域操作的引导技能学习与抽象。IEEE 机器人与自动化快报，8(10):6451-6458，2023 年。

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

S. James 和 A. 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。

[42] A. Hurst, A. Lerer, A. P. Goucher, A. Perelman, A. Ramesh, A. Clark, A. Ostrow, A. Welihinda, A. Hayes, A. Radford, et al. Gpt-4o system card. arXiv preprint arXiv:2410.21276, 2024.

A. Hurst, A. Lerer, A. P. Goucher, A. Perelman, A. Ramesh, A. Clark, A. Ostrow, A. Welihinda, A. Hayes, A. Radford 等. Gpt-4o 系统说明。arXiv 预印本 arXiv:2410.21276，2024 年。

[43] S. Karaman, M. R. Walter, A. Perez, E. Frazzoli, and S. Teller. Anytime motion planning using the rrt. In 2011 IEEE international conference on robotics and automation, pages 1478-1483. IEEE, 2011.

S. Karaman, M. R. Walter, A. Perez, E. Frazzoli 和 S. Teller. 使用 RRT 的任意时刻运动规划。载于 2011 年 IEEE 国际机器人与自动化会议，页 1478-1483。IEEE，2011 年。

[44] C. Yuan, S. Joshi, S. Zhu, H. Su, H. Zhao, and Y. Gao. Roboengine: Plug-and-play robot data augmentation with semantic robot segmentation and background generation. arXiv preprint arXiv:2503.18738, 2025.

C. Yuan, S. Joshi, S. Zhu, H. Su, H. Zhao 和 Y. Gao. Roboengine: 基于语义机器人分割和背景生成的即插即用机器人数据增强。arXiv 预印本 arXiv:2503.18738，2025 年。

[45] B. Wen, M. Trepte, J. Aribido, J. Kautz, O. Gallo, and S. Birchfield. Foundationstereo: Zero-shot stereo matching. arXiv preprint arXiv:2501.09898, 2025.

B. Wen, M. Trepte, J. Aribido, J. Kautz, O. Gallo 和 S. Birchfield. Foundationstereo: 零样本立体匹配。arXiv 预印本 arXiv:2501.09898，2025 年。

[46] Y. Dai, J. Lee, N. Fazeli, and J. Chai. Racer: Rich language-guided failure recovery policies for imitation learning. arXiv preprint arXiv:2409.14674, 2024.

Y. Dai, J. Lee, N. Fazeli 和 J. Chai. Racer: 基于丰富语言指导的模仿学习失败恢复策略。arXiv 预印本 arXiv:2409.14674，2024 年。

[47] G. Team, R. Anil, S. Borgeaud, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, K. Millican, et al. Gemini: a family of highly capable multimodal models. arXiv preprint arXiv:2312.11805, 2023.

G. Team, R. Anil, S. Borgeaud, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, K. Millican 等. Gemini: 一系列高性能多模态模型。arXiv 预印本 arXiv:2312.11805，2023 年。

[48] H. Lu, W. Liu, B. Zhang, B. Wang, K. Dong, B. Liu, J. Sun, T. Ren, Z. Li, H. Yang, et al. Deepseek-vl: towards real-world vision-language understanding. arXiv preprint arXiv:2403.05525, 2024.

H. Lu, W. Liu, B. Zhang, B. Wang, K. Dong, B. Liu, J. Sun, T. Ren, Z. Li, H. Yang 等. Deepseek-vl: 迈向真实世界的视觉-语言理解。arXiv 预印本 arXiv:2403.05525，2024 年。

[49] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E.

Hambro, F. Azhar, et al. Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971, 2023.

> H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar 等. Llama: 开放且高效的基础语言模型。arXiv 预印本 arXiv:2302.13971，2023 年。

[50] H. Xue, J. Ren, W. Chen, G. Zhang, Y. Fang, G. Gu, H. Xu, and C. Lu. Reactive diffusion policy: Slow-fast visual-tactile policy learning for contact-rich manipulation. arXiv preprint arXiv:2503.02881, 2025.

> H. Xue, J. Ren, W. Chen, G. Zhang, Y. Fang, G. Gu, H. Xu 和 C. Lu. Reactive diffusion policy: 用于接触丰富操作的慢快视觉-触觉策略学习。arXiv 预印本 arXiv:2503.02881，2025 年。

[51] A. Agarwal, A. Ajith, C. Wen, V. Stryzheus, B. Miller, M. Chen, M. K. Johnson, J. L. S. Rincon, J. Rosca, and W. Yuan. Robotic defect inspection with visual and tactile perception for large-scale components. In 2023 IEEE/RSJ Ineternational Conference on Intelligent Robots and Systems (IROS), pages 10110-10116. IEEE, 2023.

> A. Agarwal, A. Ajith, C. Wen, V. Stryzheus, B. Miller, M. Chen, M. K. Johnson, J. L. S. Rincon, J. Rosca 和 W. Yuan. 基于视觉和触觉感知的大型零件机器人缺陷检测。载于 2023 年 IEEE/RSJ 国际智能机器人与系统会议 (IROS)，页 10110-10116。IEEE，2023 年。

[52] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. P. Foster, P. R. Sanketi, Q. Vuong, T. Kollar, B. Burchfiel, R. Tedrake, D. Sadigh, S. Levine, P. Liang, and C. Finn. Openvla: An open-source vision-language-action model. In Conference on Robot Learning, volume 270, pages 2679-2713. PMLR, 2024.

> M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. P. Foster, P. R. Sanketi, Q. Vuong, T. Kollar, B. Burchfiel, R. Tedrake, D. Sadigh, S. Levine, P. Liang 和 C. Finn. Openvla: 一个开源视觉-语言-动作模型。载于机器人学习会议，卷 270，页 2679-2713。PMLR，2024 年。

[53] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Haus-man, B. Ichter, S. Jakubczak, T. Jones, L. Ke, S. Levine, A. Li-Bell, M. Mothukuri, S. Nair, K. Pertsch, L. X. Shi, J. Tanner, Q. Vuong, A. Walling, H. Wang, and U. Zhilinsky. $\pi0$: A vision-language-action flow model for general robot control. ArXiv, abs/2410.24164, 2024.

> K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Haus-man, B. Ichter, S. Jakubczak, T. Jones, L. Ke, S. Levine, A. Li-Bell, M. Mothukuri, S. Nair, K. Pertsch, L. X. Shi, J. Tanner, Q. Vuong, A. Walling, H. Wang, and U. Zhilinsky. $\pi0$: 一种用于通用机器人控制的视觉-语言-动作流模型。ArXiv, abs/2410.24164, 2024。

[54] S. Liu, L. Wu, B. Li, H. Tan, H. Chen, Z. Wang, K. Xu, H. Su, and J. Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. ArXiv, abs/2410.07864, 2024.

> S. Liu, L. Wu, B. Li, H. Tan, H. Chen, Z. Wang, K. Xu, H. Su, 和 J. Zhu。Rdt-1b: 一种用于双手操作的扩散基础模型。ArXiv, abs/2410.07864, 2024。

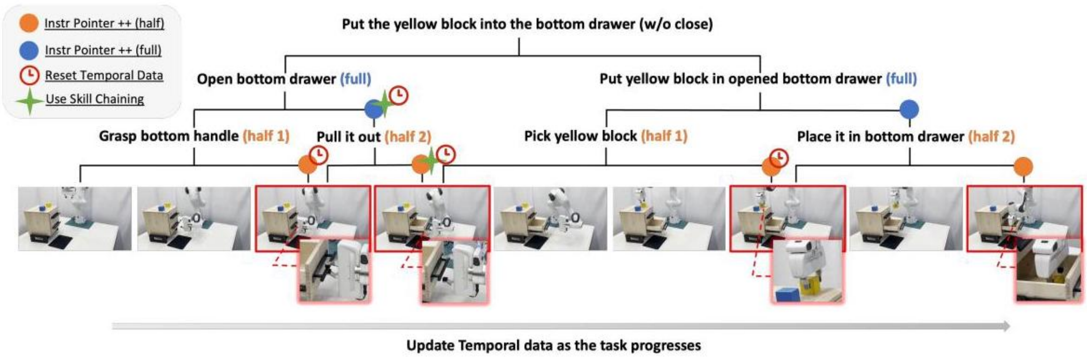

Instr Pointer ++ (half)
Instr Pointer ++ (full)
Reset Temporal Data
Use Skill Chaining
Put the yellow block into the bottom drawer (w/o close)
Open bottom drawer (full)
Put yellow block in opened bottom drawer (full)
Grasp bottom handle (half 1)
Pull it out (half 2)
Pick yellow block (half 1)
Place it in bottom drawer (half 2)
Update Temporal data as the task progresses

Figure 6: Visualizing Differences in the DeCo Framework under Half and Full Interactions

图 6: 在半交互和全交互下 DeCo 框架差异的可视化

To further compare the impact of the definitions of half and full interactions from Section 3.1 on the performance of DeCo , we detail the differences between half and full interactions in the four key stages of DeCo: task decomposition, instruction alignment, temporal modeling, and skill chaining. Figure Figure 6 visualizes these differences using the real-robot task ”put the yellow block into the bottom drawer.”

为了进一步比较第 3.1 节中半交互和全交互定义对 DeCo 性能的影响，我们详细说明了 DeCo 四个关键阶段中半交互和全交互的差异: 任务分解、指令对齐、时间建模和技能链。图 6 通过真实机器人任务“将黄色积木放入底部抽屉”直观展示了这些差异。

In the task decomposition phase, the half interaction treats each change in gripper state (e.g., open or close) as an atomic unit, leading to finer-grained subgoals and 2-cycle tasks. The full interaction mode combines an open-close pair into a single, semantically complete unit, resulting in coarser 1-cycle skills.

在任务分解阶段，半交互将夹爪状态的每次变化 (如打开或关闭) 视为原子单元，导致更细粒度的子目标和两周期任务。全交互模式则将开-关配对合并为一个语义完整的单元，产生更粗粒度的一周期技能。

For instruction alignment, the instruction pointer in the half interaction setting advances after each gripper action, allowing for fine-grained alignment with low-level steps. In the full interaction setting, it updates only after completing a full 2-cycle interaction, aligning more naturally with the structure of natural language instructions.

在指令对齐方面，半交互设置中指令指针在每次夹爪动作后前进，允许与低级步骤进行细粒度对齐。全交互设置中，指令指针仅在完成完整的两周期交互后更新，更自然地与自然语言指令结构对齐。

For temporally conditioned models like RVT-2, the two modes adopt distinct temporal data scheduling strategies. In the half interaction case, temporal data are reset after each gripper event, resulting in a finer decomposition of skill phases. In the full interaction setting, the reset occurs only after an entire atomic skill (i.e., 2 cycles) is executed, ensuring that temporal dynamics align with higher-level semantic units and reducing unnecessary fluctuations.

对于如 RVT-2 这类时间条件模型，两种模式采用不同的时间数据调度策略。半交互情况下，时间数据在每次夹爪事件后重置，导致技能阶段的更细分解。全交互情况下，重置仅在执行完整原子技能 (即两周期) 后进行，确保时间动态与更高级语义单元对齐，减少不必要的波动。

For skill chaining, both modes adopt the same triggering condition: a new skill is activated only after completing a full open-close cycle, ensuring consistent and coherent transitions between atomic skills.

在技能链方面，两种模式采用相同的触发条件: 仅在完成完整的开-关周期后激活新技能，确保原子技能之间的转换一致且连贯。

These design choices significantly influence how atomic skills are decomposed, supervised, and executed, directly affecting DeCo's temporal and semantic alignment capabilities. Ultimately, the granularity of interaction shapes both the structure of training data and the behavior of the model during inference, impacting its ability to generalize and successfully perform novel long-horizon tasks.

这些设计选择显著影响原子技能的分解、监督和执行方式，直接影响 DeCo 的时间和语义对齐能力。最终，交互粒度决定了训练数据结构和模型推理时的行为，影响其泛化能力及成功执行新颖长时任务的能力。

B VLM Prompts in DeCo

B DeCo 中的视觉语言模型提示

The DeCo framework provides two prompt templates for VLM-guided planning: Full Interaction and Half Interaction. Table 4 shows the details of both prompt templates. More detailed prompts can be found at https://deco226.github.io.

DeCo 框架为视觉语言模型 (VLM) 引导规划提供了两种提示模板: 全交互和半交互。表 4 展示了两种提示模板的详细信息。更多详细提示可见 https://deco226.github.io。

Prompt Engineering Template: Full Interaction vs Half Interaction

提示工程模板: 全交互与半交互

Full	**Half**
Background and Role Definition You are a robotic task planner. Translate high-level instructions and environment states into steps based on predefined skills.	**Background and Role Definition** Same as Full.
Scene Description and Constraints `<scene: Description of the current environment, interactive objects, and their initial states.>`	**Scene Description and Constraints** Same as Full.
Robotic Arm Task Goal The robotic arm interprets instructions and interacts with the environment to efficiently complete tasks. `<goal: Summary of the robotic arm's main task objective.>`	**Robotic Arm Task Goal** Same as Full.
User Goal Description Generate optimized step-by-step commands using predefined skills, eliminating redundancies.	**User Goal Description** Same as Full.
Task Decomposition Requirements 1. Break down instructions. 2. Align with environment. 3. `<decomposition rule of full>`	**Task Decomposition Requirements** 1. Break down instructions. 2. Align with environment. 3. `<decomposition rule of half: Typically emphasizes outputting an even number of steps.>`
Domain Skills List `<list-full: Enumerate skills clearly.>`	**Domain Skills List** `<list-half>`
Skill Usage Rules `<usage of skill in full: Usage defines how skills should be correctly applied during planning.>`	**Skill Usage Rules** `<usage of skill in half>`
Input Format 1. High-Level Semantic Instruction 2. Environment Image	**Input Format** Same as Full.
Output Format and Rules - Only Atomic-Level Commands (lowercase, numbered) Rules: Logical, efficient, minimal.`<example>`	**Output Format and Rules** - High-Level Plan + Atomic-Level Commands Rules: Logical, efficient, minimal. The High-Level Plan should provide a clear overview, while the Atomic-Level Commands should detail executable actions.`<example>`
User Input 1. High-Level Instruction `<instruction>` 2. Environment Image `<details>`, `<image>`	**User Input** 1. High-Level Instruction `<instruction>` 2. Environment Image `<details: Scene-related visual descriptions.>`, `<image>`

Table 4: Prompt Templates for Full and Half Interaction Versions in the DeCo Framework for VLM-Guided Planning.

表 4:DeCo 框架中用于视觉语言模型引导规划的全交互和半交互版本提示模板。

	Avg.	Open	Close	Put in Opened	Take Out of	Box Out of	Box in	Box Out	Broom Out	Sweep to	Rubbish in
Models	Success ↑	Drawer	Drawer	Drawer	Opened Drawer	Opened Drawer	Cupboard	Cupboard	Cupboard	Dustpan	Dustpan
RVT-2 [3]	91.83	98.33 ± 2.36	96.67 ± 2.36	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	$35.00_{\pm 4.08}$	98.33 ± 2.36	98.33 ±2.36	91.67 ± 6.24	100.00 ± 0.00
RVT-2+DeCo	86.80	98.33 ± 2.36	100.00 ± 0.00	88.33 ± 6.24	100.00 ± 0.00	100.00 ± 0.00	48.33 ± 2.36	$85.00_{\pm 7.07}$	$65.00_{\pm 0.00}$	$83.00_{\pm 6.24}$	100.00 ± 0.00
3DDA [19]	98.00	98.33 ± 2.36	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	98.33 ± 2.36	91.67 ± 4.71	98.33 ±2.36	96.67 ± 2.36	98.33 ±2.36	100.00 ± 0.00
3DDA+DeCo	96.00	$95.00_{\pm 0.00}$	100.00 ± 0.00	100.00 ± 0.00	100.00	100.00 ± 0.00	88.33 ± 2.36	$100.00_{\pm 0.00}$	98.33 ± 2.36	78.33 ± 2.36	100.00 ±0.00
ARP [5]	94.67	100.00 ± 0.00	$95.00_{\pm 0.00}$	100.00 ± 0.00	100.00	100.00 ± 0.00	$65.00_{\pm 7.07}$	$95.00_{\pm 0.00}$	100.00 ± 0.00	93.33 ±2.36	98.33 ±2.36
ARP + DeCo	91.67	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00	100.00 ± 0.00	30.00 ± 7.07	$95.00_{\pm 0.00}$	96.67 ± 2.36	96.67 ±2.36	98.33 ±2.36

	平均	打开	关闭	放入已打开	取出	盒子取出	盒子放入	盒子取出	扫出	扫到	垃圾放入
模型	成功 ↑	抽屉	抽屉	抽屉	打开的抽屉	打开的抽屉	橱柜	橱柜	橱柜	簸箕	簸箕
RVT-2 [3]	91.83	98.33 ± 2.36	96.67 ± 2.36	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	$35.00_{\pm 4.08}$	98.33 ± 2.36	98.33 ±2.36	91.67 ± 6.24	100.00 ± 0.00
RVT-2+DeCo	86.80	98.33 ± 2.36	100.00 ± 0.00	88.33 ± 6.24	100.00 ± 0.00	100.00 ± 0.00	48.33 ± 2.36	$85.00_{\pm 7.07}$	$65.00_{\pm 0.00}$	$83.00_{\pm 6.24}$	100.00 ± 0.00
3DDA [19]	98.00	98.33 ± 2.36	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	98.33 ± 2.36	91.67 ± 4.71	98.33 ±2.36	96.67 ± 2.36	98.33 ±2.36	100.00 ± 0.00
3DDA+DeCo	96.00	$95.00_{\pm 0.00}$	100.00 ± 0.00	100.00 ± 0.00	100.00	100.00 ± 0.00	88.33 ± 2.36	$100.00_{\pm 0.00}$	98.33 ± 2.36	78.33 ± 2.36	100.00 ±0.00
ARP [5]	94.67	100.00 ± 0.00	$95.00_{\pm 0.00}$	100.00 ± 0.00	100.00	100.00 ± 0.00	$65.00_{\pm 7.07}$	$95.00_{\pm 0.00}$	100.00 ± 0.00	93.33 ±2.36	98.33 ±2.36
ARP + DeCo	91.67	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00	100.00 ± 0.00	30.00 ± 7.07	$95.00_{\pm 0.00}$	96.67 ± 2.36	96.67 ±2.36	98.33 ±2.36

Table 5: Test Performance on 10 atomic tasks in DeCoBench. Evaluations on 10 atomic tasks are conducted using 3 seeds, with 20 test episodes per task, utilizing the final checkpoints from training on 10 atomic tasks. Performance is reported for different model variants across specific task categories.

表 5:DeCoBench 中 10 个原子任务的测试表现。对 10 个原子任务的评估使用 3 个随机种子，每个任务进行 20 个测试回合，采用在 10 个原子任务上训练的最终检查点。报告了不同模型变体在特定任务类别上的表现。

	train_set_50		train_set_10		train_set_5		train_set_3	
	half	full	half	full	half	full	half	full
Average (Atomic Tasks)	81.24	86.50	77.50	81.27	70	75.28	68.50	73.48
Average (Compositional Tasks)	56.81	66.70	53.58	62.06	41.47	32.50	23.75	32.96

	训练集 _50		训练集 _10		训练集 _5		训练集 _3	
	一半	全部	一半	全部	一半	全部	一半	全部
平均 (原子任务)	81.24	86.50	77.50	81.27	70	75.28	68.50	73.48
平均 (组合任务)	56.81	66.70	53.58	62.06	41.47	32.50	23.75	32.96

Table 6: Average success rates for RVT-2+DeCo, defined through half and full interactions, after training with varying numbers of atomic task demos, on 10 atomic tasks and 12 combinatorial long-sequence tasks in DeCoBench.

表 6: 在 DeCoBench 中，经过不同数量原子任务示范训练后，RVT-2+DeCo 通过半交互和全交互定义的平均成功率，涵盖 10 个原子任务和 12 个组合长序列任务。

C More Experimental Results

C 更多实验结果

C.1 Test Performance on Atomic Tasks

C.1 原子任务的测试表现

Table 5 presents the performance of various models on 10 atomic tasks in the DeCoBench benchmark. Among them, the 3DDA model achieves the highest overall success rate, while ARP and RVT-2 also demonstrate stable and competitive performance. However, integrating DeCo into these models results in a noticeable drop in success rates for certain tasks. This degradation may stem from the instability of the vision-language model (VLM) used by

DeCo during atomic task planning, which affects the instruction-following accuracy of multi-task imitation learning models. In contrast, the three baseline models are evaluated using ground-truth task instructions-those they have already encountered during training-which enables them to achieve strong test performance. These results further support the argument in Sec. 7: although 3DDA and ARP perform well in acquiring atomic skills, their limited robustness in visual processing constrains their ability to handle complex, long-horizon tasks that require compositional visual and semantic reasoning. While DeCo significantly improves the generalization ability of multi-task IL models on novel compositional long-horizon tasks (as shown in Table 1), the performance of 3DDA+DeCo and ARP+DeCo still falls short of RVT-2+DeCo, which adopts a coarse-to-fine visual feature extraction strategy to handle such tasks more reliably.

表 5 展示了各模型在 DeCoBench 基准中 10 个原子任务上的表现。其中，3DDA 模型取得了最高的整体成功率，而 ARP 和 RVT-2 也表现出稳定且具有竞争力的性能。然而，将 DeCo 集成到这些模型中，某些任务的成功率明显下降。这种性能下降可能源于 DeCo 在原子任务规划中使用的视觉语言模型 (VLM) 不稳定，影响了多任务模仿学习模型的指令遵循准确性。相比之下，三个基线模型使用的是训练时已见过的真实任务指令，因此能够取得较强的测试表现。这些结果进一步支持第 7 节的论点: 尽管 3DDA 和 ARP 在获取原子技能方面表现良好，但其视觉处理的鲁棒性有限，限制了它们处理需要组合视觉和语义推理的复杂长时序任务的能力。虽然 DeCo 显著提升了多任务模仿学习模型在新颖组合长时序任务上的泛化能力 (如表 1 所示)，但 3DDA+DeCo 和 ARP+DeCo 的表现仍不及采用粗到细视觉特征提取策略以更可靠处理此类任务的 RVT-2+DeCo。

C.2 Impact of Half and Full Interactions on DeCo's Generalization Performance

C.2 半交互与全交互对 DeCo 泛化性能的影响

Experimental results in Table 6 show that using full interaction as the unit of task decomposition consistently yields higher success rates across different training set sizes compared to half interaction in RVT-2+DeCo. This indicates that using full interaction decomposition in DeCo provides a more effective granularity for structuring demonstrations in language-conditioned multi-task imitation learning.

表 6 的实验结果表明，在 RVT-2+DeCo 中，使用全交互作为任务分解单元，在不同训练集规模下均持续获得比半交互更高的成功率。这表明在 DeCo 中采用全交互分解为语言条件多任务模仿学习的示范结构提供了更有效的粒度。

Half interaction decomposition tends to produce overly fine-grained subtasks that often lack semantic or operational coherence. This fragmentation weakens the learning signal, increases task interference, and makes policy learning more difficult. In contrast, full interactions typically correspond to semantically meaningful subgoals, providing richer and more stable inputs that support generalization. Furthermore, full interactions align more naturally with language instructions. Since each full segment usually maps to a complete command or interpretable subgoal, it facilitates better alignment between language and perception/action modalities—an essential factor in DeCo.

半交互分解往往产生过于细粒度的子任务，这些子任务常缺乏语义或操作上的连贯性。这种碎片化削弱了学习信号，增加了任务干扰，使策略学习更加困难。相比之下，全交互通常对应语义上有意义的子目标，提供更丰富且稳定的输入，有助于泛化。此外，全交互更自然地与语言指令对齐。由于每个完整片段通常映射到一个完整的命令或可解释的子目标，它促进了语言与感知/动作模态之间更好的对齐——这是 DeCo 的关键因素。

Task	RVT-2+DeCo	RVT-2 (6 Long training)
10 Atomic	86.80% (30.63% % % % % % %	56.17%
12 All	66.67% (14.31% ↑)	52.36%

任务	RVT-2+DeCo	RVT-2(6 次长时间训练)
10 原子	86.80% (30.63% % % % % % %	56.17%
12 全部	66.67%(提升 14.31%)	52.36%

Table 7: Additional Results on the Impact of Atomic Task Design in DeCo.

表 7:DeCo 中原子任务设计影响的额外结果。

While finer decomposition may offer more compositional flexibility, it also expands the task space and increases the risk of generating incoherent or inconsistent combinations. Full interaction decomposition strikes a more balanced trade-off between flexibility and contextual consistency, improving generalization in long-horizon settings. This level of granularity also mirrors how humans plan and describe multi-step tasks. High-level instructions such as ”open drawer” or ”pick up the item” naturally correspond to full interactions, making this decomposition more compatible with downstream applications such as long-term task planning and skill transfer.

虽然更细粒度的分解可能提供更大的组合灵活性，但也扩大了任务空间，增加了生成不连贯或不一致组合的风险。全交互分解在灵活性和上下文一致性之间实现了更平衡的权衡，提升了长时序任务中的泛化能力。这种粒度水平也反映了人类规划和描述多步骤任务的方式。诸如“打开抽屉”或“拿起物品”等高级指令自然对应于完整交互，使得这种分解更适合下游应用，如长期任务规划和技能迁移。

Overall, full interaction decomposition not only improves empirical performance but also provides a stronger semantic structure and practical relevance, making it a more effective strategy for language-conditioned multi-task imitation learning.

总体而言，全交互分解不仅提升了经验性能，还提供了更强的语义结构和实际相关性，使其成为语言条件多任务模仿学习中更有效的策略。

C.3 Additional Analysis on Impact of Atomic Task Design

C.3 原子任务设计影响的额外分析

To further investigate the necessity of atomic task decomposition, we conduct an extended evaluation of the RVT-2 model under two training strategies: (1) training directly on demonstrations from six long-horizon tasks (including four original imitation learning tasks and two cross-domain tasks), denoted as RVT-2 (6 Long training); and (2) training on 10 atomic tasks in DeCoBench using the DeCo framework, denoted as RVT-2 + DeCo. We evaluate the models on two subsets: the 10 atomic tasks and the full set of 12 long-horizon tasks (12 All).

为进一步探讨原子任务分解的必要性，我们对 RVT-2 模型在两种训练策略下进行了扩展评估:(1) 直接在六个长时序任务的示范上训练 (包括四个原始模仿学习任务和两个跨域任务)，记为 RVT-2(6 长训练)；(2) 使用 DeCo 框架在 DeCoBench 的 10 个原子任务上训练，记为 RVT-2 + DeCo。我们在两个子集上评估模型:10 个原子任务和全部 12 个长时序任务 (12 全部)。

The six long-horizon tasks implicitly include the 10 atomic skills. Therefore, the baseline model trained on these six tasks (RVT-2 (6 Long training)) is expected to learn and execute the atomic skills and generalize to tasks that can be completed by composing them. However, the baseline model shows severe overfitting to the six tasks. When evaluated on atomic tasks or on unseen long-horizon tasks composed of those atomic skills, the model does not demonstrate true understanding of the execution process and tends to succeed in a more ”coincidental” manner.

这六个长时序任务隐含包含了 10 个原子技能。因此，在这六个任务上训练的基线模型 (RVT-2(6 长训练)) 预期能够学习并执行原子技能，并泛化到可通过组合这些技能完成的任务。然而，基线模型表现出对这六个任务的严重过拟合。在原子任务或由这些原子技能组成的未见长时序任务上评估时，模型未能展现对执行过程的真正理解，成功更多表现为“偶然”。

As shown in Table 7, the baseline performs significantly worse than our model, both in executing atomic skills and in completing long-horizon tasks. These improvements are not merely due to more data, but rather to better task structure. By decomposing complex tasks into reusable and semantically meaningful sub-goals, DeCo promotes structured and transferable policy representations, reduces trajectory overfitting, and improves policy composability. Maintaining consistency in physical interaction and ensuring the reusability of skills in atomic task design are key to enhancing generalization in multi-task imitation learning models.

如表 7 所示，基线在执行原子技能和完成长时序任务方面均显著不及我们的模型。这些提升不仅仅是由于更多数据，而是更优的任务结构。通过将复杂任务分解为可复用且语义明确的子目标，DeCo 促进了结构化且可迁移的策略表示，减少了轨迹过拟合，提升了策略的可组合性。在原子任务设计中保持物理交互的一致性并确保技能的可复用性，是增强多任务模仿学习模型泛化能力的关键。

D DeCoBench: Benchmark Overview

D DeCoBench: 基准概述

We introduce DeCoBench, illustrated in Fig. 7, a benchmark designed to evaluate the zero-shot generalization capabilities of multi-task imitation learning (IL) models on novel yet compositional long-horizon 3D manipulation tasks. DeCoBench encompasses three task domains: Object Rearrangement with Drawer, Object Rearrangement with Cupboard, and Rubbish Cleanup.

我们介绍 DeCoBench，如图 7 所示，该基准旨在评估多任务模仿学习 (IL) 模型在新颖且组合性的长时序三维操作任务上的零样本泛化能力。DeCoBench 涵盖三个任务领域: 带抽屉的物体重排、带橱柜的物体重排和垃圾清理。

In the Object Rearrangement with Drawer domain, the original IL tasks-put item in drawer without close and take item out of drawer and close-are decomposed based on the gripper's complete physical interaction cycle (i.e., open → closed → open), resulting in four atomic tasks: open drawer, close drawer, put item in drawer, and take item out of drawer. In addition, a variant atomic task, take box out of drawer, is constructed via object substitution. This domain includes two 4-cycle compositional tasks (put item in without close, take item out without close), five 6-cycle tasks (put item in and close, take item out and close, put two items in same, take two items out of same), and two 10-cycle tasks (put two items in two different, take two items out of two different).

在带抽屉的物体重排领域，原始 IL 任务——将物品放入抽屉 (不关闭) 和从抽屉中取出物品并关闭——基于夹持器的完整物理交互周期 (即打开 → 关闭 → 打开) 进行分解，产生四个原子任务: 打开抽屉、关闭抽屉、将物品放入抽屉和从抽屉中取出物品。此外，通过物体替换构建了一个变体原子任务: 从抽屉中取出盒子。该领域包括两个 4 周期组合任务 (放入物品不关闭、取出物品不关闭)、五个 6 周期任务 (放入物品并关闭、取出物品并关闭、将两个物品放入同一抽屉、从同一抽屉取出两个物品) 和两个 10 周期任务 (将两个物品放入两个不同抽屉、从两个不同抽屉取出两个物品)。

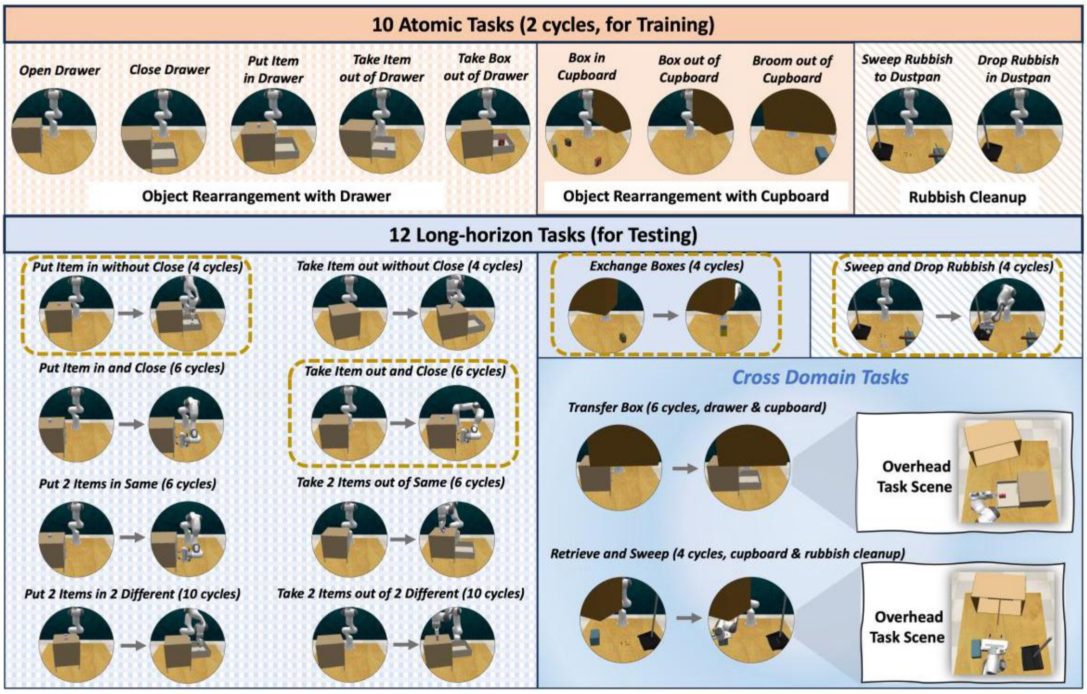

10 Atomic Tasks (2 cycles, for Training)
Open Drawer
Close Drawer
Put Item in Drawer
Take Item out of Drawer
Take Box out of Drawer
Box in Cupboard
Box out of Cupboard
Broom out of Cupboard
Sweep Rubbish to Dustpan
Drop Rubbish in Dustpan
Object Rearrangement with Drawer
Object Rearrangement with Cupboard
Rubbish Cleanup
12 Long-horizon Tasks (for Testing)
Put Item in without Close (4 cycles)
Take Item out without Close (4 cycles)
Exchange Boxes (4 cycles)
Sweep and Drop Rubbish (4 cycles)
Put Item in and Close (6 cycles)
Take Item out and Close (6 cycles)
Cross Domain Tasks
Transfer Box (6 cycles, drawer & cupboard)
Overhead Task Scene
Put 2 Items in Same (6 cycles)
Take 2 Items out of Same (6 cycles)
Retrieve and Sweep (4 cycles, cupboard & rubbish cleanup)
Put 2 Items in 2 Different (10 cycles)
Take 2 Items out of 2 Different (10 cycles)
Overhead Task Scene

Figure 7: DeCoBench Overview: The benchmark covers three task domains. Based on the physical interaction-based task decomposition principle proposed in Sec. 3.2, 4 original IL tasks (marked with yellow dashed lines) are decomposed into 10 atomic tasks—including 2 constructed by replacing objects within the scene-for training multi-task IL models to acquire atomic skills. By semantically composing these atomic skills, we construct 12 long-horizon 3D manipulation tasks-including both original and cross-domain tasks-for evaluating models' zero-shot generalization capabilities.

图 7:DeCoBench 概览: 该基准覆盖三个任务领域。基于第 3.2 节提出的基于物理交互的任务分解原则，4 个原始 IL 任务 (以黄色虚线标记) 被分解为 10 个原子任务——其中 2 个通过场景内物体替换构建——用于训练多任务 IL 模型以掌握原子技能。通过语义组合这些原子技能，我们构建了 12 个长时序三维操作任务——包括原始任务和跨域任务——用于评估模型的零样本泛化能力。

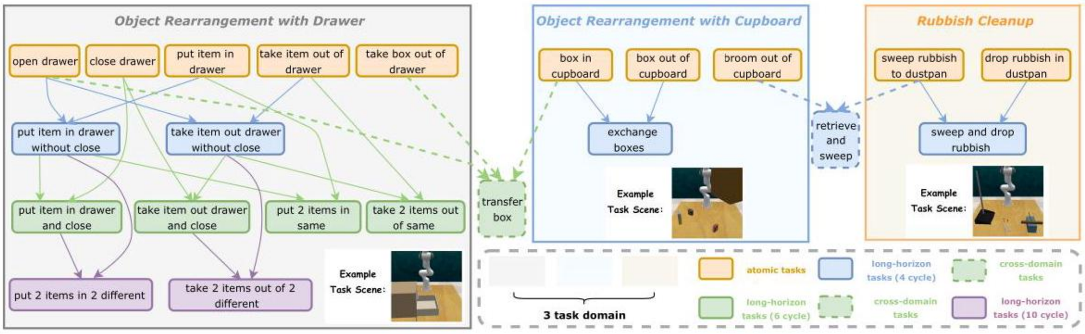


Figure 8: The compositional relationships between the atomic tasks and the compositonal tasks as well as cross-domain tasks in DeCoBench are represented using arrows.

图 8:DeCoBench 中原子任务与组合任务及跨域任务之间的组合关系通过箭头表示。

In the Object Rearrangement with Cupboard domain, the original IL task exchange boxes is decomposed into two atomic tasks: box in cupboard and box out of cupboard. Another atomic task, take broom out of cupboard, is introduced by substituting the manipulated object. The exchange boxes task serves as a 4-cycle compositional long-horizon task for evaluation.

在带有橱柜的物体重新排列领域，原始的 IL 任务“交换盒子”被分解为两个原子任务: 盒子放入橱柜和盒子取出橱柜。通过替换操作对象，引入了另一个原子任务“从橱柜中取出扫帚”。交换盒子任务作为一个四周期的组合长时任务用于评估。

Task name	Language Template	#Items	#Variations	Variation Type
(a) open_drawer	”Open the <top middle=”” bottom=””> drawer.”</top>	1	3	category, placement
(b) close_drawer	”Close the <top middle=”” bottom=””> drawer.”</top>	1	3	category, placement
(c) put_in_opened_drawer	”Put the block in the <top middle=”” bottom=””> drawer.”</top>	1	3	category, placement
(d) take_out_of_opened_drawer	”Take the block out of the <top middle=”” bottom=””> drawer and place it on the drawer’s surface.”</top>	1	3	category, placement
(e) box_out_of_opened_drawer	”Take the strawberry jello box out of the <top middle=”” bottom=””> drawer and place it on the drawer’s surface.”</top>	1	3	category, placement
(f) box_in_cupboard	”Put the <strawberry jello=”” spam=”” sugar=””> in the cupboard.”</strawberry>	1	3	category, placement
(g) box_out_of_cupboard	”Put the <strawberry jello=”” spam=”” sugar=””> on the table.”</strawberry>	1	3	category, placement
(h) broom_out_of_cupboard	”Take the broom out of the cupboard and place it on the table.”	1	1	placement
(i) sweep_to_dustpan	”Sweep dirt to dustpan.”	1	1	placement
(j) rubbish_in_dustpan	”Drop the rubbish into the dustpan.”	1	1	placement

任务名称	语言模板	物品数量	变体数量	变体类型
(a) 打开抽屉	“打开 <top middle=”” bottom=””> 抽屉。” </top>	1	3	类别，位置
(b) 关闭抽屉	“关闭 <top middle=”” bottom=””> 抽屉。” </top>	1	3	类别，位置
(c) 放入已打开的抽屉	“将积木放入 <top middle=”” bottom=””> 抽屉。” </top>	1	3	类别，位置
(d) 从已打开的抽屉取出	“从 <top middle=”” bottom=””> 抽屉中取出积木，放在抽屉表面。” </top>	1	3	类别，位置
(e) 从已打开的抽屉取出盒子	“从 <top middle=”” bottom=””> 抽屉中取出草莓果冻盒，放在抽屉表面。” </top>	1	3	类别，位置
(f) 盒子放入橱柜	“将 <strawberry jello=”” spam=”” sugar=””> 放入橱柜。” </strawberry>	1	3	类别，位置
(g) 盒子取出橱柜	“将 <strawberry jello=”” spam=”” sugar=””> 放在桌子上。” </strawberry>	1	3	类别，位置
(h) 从橱柜取出扫帚	“从橱柜取出扫帚，放在桌子上。”	1	1	放置
(i) 扫向簸箕	“将灰尘扫入簸箕。”	1	1	放置
(j) 垃圾放入簸箕	“将垃圾倒入簸箕。”	1	1	放置

Table 8: Properties of the atomic tasks, in DeCoBench. We report on language template, the number of items that the robot can interact with, the task variations and variation type.

表 8:DeCoBench 中原子任务的属性。我们报告语言模板、机器人可交互的物品数量、任务变体及变体类型。

In the Rubbish Cleanup domain, the original IL task sweep and drop rubbish is decomposed into two atomic tasks: sweep rubbish to dustpan and drop rubbish in dustpan, with the original task used as a 4-cycle compositional long-horizon evaluation task.

在垃圾清理领域，原始的 IL 任务“扫地并丢垃圾”被分解为两个原子任务: 将垃圾扫到簸箕中和将垃圾丢入簸箕，原始任务作为一个 4 周期的组合长时任务进行评估。

Moreover, DeCoBench includes two cross-domain compositional tasks: transfer box (6 cycles), spanning the Drawer and Cupboard domains, and retrieve and sweep (4 cycles), spanning the Cupboard and Cleanup domains.

此外，DeCoBench 包含两个跨领域组合任务: 转移箱子 (6 周期)，跨越抽屉和橱柜领域，以及取回并扫地 (4 周期)，跨越橱柜和清理领域。

Execution videos demonstrating the performance of different models on all 12 compositional tasks in DeCoBench are available on the project website: https://deco226.github.io.The full dataset for all DeCoBench tasks is provided in the supplementary materials.

展示不同模型在 DeCoBench 全部 12 个组合任务上表现的执行视频可在项目网站观看:https://deco226.github.io。所有 DeCoBench 任务的完整数据集见补充材料。

In the following we provide details of the DeCoBench tasks. A summary of the 10 atomic tasks is provided in Table 8. A summary of the 12 compositional long-horizon tasks is provided in Table 9.

以下我们提供 DeCoBench 任务的详细信息。表 8 总结了 10 个原子任务，表 9 总结了 12 个组合长时任务。

D.1 Atomic Tasks in DeCoBench

D.1 DeCoBench 中的原子任务

(a) open_drawer

(a) open_drawer

Task Description: The robot must open a specified drawer (top, middle, or bottom) by pulling its handle.

任务描述: 机器人必须通过拉动把手打开指定的抽屉 (上、中或下)。

Success Metric: The task is considered successful when the specified drawer is opened by at least 0.15 meters.

成功标准: 当指定抽屉被打开至少 0.15 米时，任务视为成功。

Objects: A cabinet with three drawers (top, middle, bottom), each with its own joint and handle.

物体: 一个带有三个抽屉 (上、中、下) 的柜子，每个抽屉有独立的关节和把手。

Language Instructions for Full Interaction: Open the <top/middle/bottom> drawer.

完整交互的语言指令: 打开 <top/middle/bottom> 抽屉。

Language Instructions for Half Interactions: Grasp the <top/middle/bottom> drawer handle; Pull the <top/middle/bottom> drawer open.

半交互的语言指令: 抓住 <top/middle/bottom> 抽屉把手；拉开 <top/middle/bottom> 抽屉。

Variation Number: 3

变体数量:3

(b) close_drawer

(b) close_drawer

Task Description: The robot must close a specified drawer (top, middle, or bottom) by pushing it shut.

任务描述: 机器人必须通过推压关闭指定的抽屉 (上、中或下)。

Success Metric: The task is considered successful when the target drawer is pushed to within 0.03 meters of its fully closed position.

成功标准: 当目标抽屉被推至距离完全关闭位置 0.03 米以内时，任务视为成功。

Objects: A cabinet with three drawers (top, middle, bottom), each with its own joint and handle.

物体: 一个带有三个抽屉 (上、中、下) 的柜子，每个抽屉都有自己的接头和把手。

Language Instructions for Full Interaction: Close the <top/middle/bottom> drawer.

完整交互的语言指令: 关闭 <b0></b0> 上/中/下 <b1></b1> 抽屉。

Language Instructions for Half Interactions: Move close to the <top/middle/bottom> drawer handle; Push the <top/middle/bottom> drawer shut.

半交互的语言指令: 靠近 <b0></b0> 上/中/下 <b1></b1> 抽屉把手; 推 <b0></b0> 上/中/下 <b1></b1> 抽屉关闭。

Variation Number: 3

变体编号:3

(c) put_in_opened_drawer

(c) 放入已打开的抽屉

Task Description: The robot must pick up a block and place it into a specified drawer (top, middle, or bottom) that is already open.

任务描述: 机器人必须拾起一个积木，并将其放入指定的已打开的抽屉 (上、中或下)。

Success Metric: The task is considered successful when the block is detected by the proximity sensor placed inside the target drawer.

成功标准: 当目标抽屉内的接近传感器检测到积木时，任务视为成功。

Objects: A cabinet with three drawers (top, middle, bottom), and one block placed near the drawer.

物体: 一个带有三个抽屉 (上、中、下) 的柜子，和一个放置在抽屉附近的积木。

Language Instructions for Full Interaction: Put the block in the <top/middle/bottom> drawer.

完整交互的语言指令: 将积木放入 <b0></b0> 上/中/下 <b1></b1> 抽屉。

Language Instructions for Half Interactions: Pick up the block on the drawer's surface; Place the block in the <top/middle/bottom> drawer.

半交互的语言指令: 拾起放在抽屉表面的积木；将积木放入 <b0></b0> 上/中/下 <b1></b1> 抽屉。

Variation Number: 3

变体编号:3

(d) take_out_of_opened_drawer

(d) 从已打开的抽屉取出

Task Description: The robot must take a block out of a specified drawer (top, middle, or bottom) that is already open, and place it on the surface of the drawer.

任务描述: 机器人必须从指定的已打开抽屉 (上、中或下) 中取出一个积木，并将其放置在抽屉表面。

Success Metric: The task is considered successful when the block is detected by the proximity sensor placed on the drawer's surface.

成功标准: 当抽屉表面的接近传感器检测到积木时，任务视为成功。

Objects: A cabinet with three drawers (top, middle, bottom), one block inside the drawer, and a proximity sensor on the drawer's surface.

物体: 一个带有三个抽屉 (上、中、下) 的柜子，一个放在抽屉内的积木，以及放置在抽屉表面的接近传感器。

Language Instructions for Full Interaction: Take the block out of the < top/middle/bottom > drawer and place the block on the drawer's surface.

完整交互的语言指令: 从 < 上/中/下 > 抽屉取出积木，并将积木放在抽屉表面。

Language Instructions for Half Interactions: Pick up the block in the < top/middle/bottom > drawer; Place the block on the drawer's surface.

半交互语言指令: 从 < 上层/中层/下层 > 抽屉中拿起积木；将积木放置在抽屉表面。

Variation Number: 3

变体编号:3

(e) box_out_of_opened_drawer

(e) 从已打开的抽屉中取出盒子

Task Description: The robot must take a specific object (a strawberry jello box) out of a specified drawer (top, middle, or bottom) that is already open, and place it onto the drawer's surface.

任务描述: 机器人必须从指定的已打开抽屉 (上层、中层或下层) 中取出特定物品 (草莓果冻盒)，并将其放置在抽屉表面。

Success Metric: The task is considered successful when the jello box is detected on the drawer's surface, outside of the drawer.

成功标准: 当果冻盒被检测到位于抽屉表面、抽屉外部时，任务视为成功。

Objects: A cabinet with three drawers (top, middle, bottom), and one strawberry jello box placed inside the drawer.

物品: 一个带有三个抽屉 (上层、中层、下层) 的柜子，以及放置在抽屉内的一个草莓果冻盒。

Language Instructions for Full Interaction: Take the strawberry jello box out of the <top/middle/bottom> drawer and place it on the drawer's surface.

全交互语言指令: 将草莓果冻盒从 < 上层/中层/下层 > 抽屉中取出，放置在抽屉表面。

Language Instructions for Half Interactions: Pick up the strawberry jello box in the <top/middle/bottom> drawer; Place it on the drawer's surface.

半交互语言指令: 从 < 上层/中层/下层 > 抽屉中拿起草莓果冻盒；将其放置在抽屉表面。

Variation Number: 3

变体编号:3

(f) box_in_cupboard

(f) 盒子放入橱柜

Task Description: The robot must pick up a specific grocery item (either strawberry jello, spam, or sugar) from a tabletop and place it into a cupboard.

任务描述: 机器人必须从桌面上拿起特定的杂货物品 (草莓果冻、午餐肉或糖)，并将其放入橱柜中。

Success Metric: The task is considered successful when the selected grocery item is detected inside the cupboard by a proximity sensor.

成功标准: 当所选杂货物品被接近传感器检测到放入橱柜内时，任务视为成功。

Objects: A cupboard, three grocery items (strawberry jello, spam, sugar), and a proximity sensor to detect when the items are placed in the cupboard.

物品: 一个橱柜，三种杂货物品 (草莓果冻、午餐肉、糖)，以及用于检测物品放入橱柜的接近传感器。

Language Instructions for Full Interaction: Put the <strawberry jello/spam/sugar> in the cupboard.

全交互语言指令: 将 < 草莓果冻/午餐肉/糖 > 放入橱柜。

Language Instructions for Half Interactions: Pick up the <strawberry jello/spam/sugar> on the table; Place the <strawberry jello/spam/sugar> in the cupboard.

半交互语言指令: 从桌面上拿起 < 草莓果冻/午餐肉/糖 >；将 < 草莓果冻/午餐肉/糖 > 放入橱柜。

Variation Number: 3

变体编号:3

(g) box_out_of_cupboard

(g) 盒子从橱柜中取出

Task Description: The robot must take a specific grocery item (either strawberry jello, spam, or sugar) out of a cupboard and place it onto the table.

任务描述: 机器人必须从橱柜中取出特定的杂货物品 (草莓果冻、午餐肉或糖)，并将其放到桌子上。

Success Metric: The task is considered successful when the selected grocery item is detected by a proximity sensor placed on the table.

成功标准: 当桌子上的接近传感器检测到所选杂货物品时，任务即视为成功。

Objects: A cupboard, three grocery items (strawberry jello, spam, sugar), and a proximity sensor to detect when the items are placed on the table.

物品: 一个橱柜，三种杂货物品 (草莓果冻、午餐肉、糖)，以及一个用于检测物品放置在桌子上的接近传感器。

Language Instructions for Full Interaction: Put the <strawberry jello/spam/sugar> on the table.

完整交互的语言指令: 把 < 草莓果冻/午餐肉/糖 > 放到桌子上。

Language Instructions for Half Interactions: Pick up the <strawberry jello/spam/sugar> in the cupboard; Place the <strawberry jello/spam/sugar> on the table.

半交互的语言指令: 从橱柜里拿起 < 草莓果冻/午餐肉/糖 >；把 < 草莓果冻/午餐肉/糖 > 放到桌子上。

Variation Number: 3

变体编号:3

(r) broom_out_of_cupboard

(r) 扫帚从橱柜中取出

Task Description: The robot must take the broom out of the cupboard and place it on the table.

任务描述: 机器人必须将扫帚从橱柜中取出并放到桌子上。

Success Metric: The task is considered successful when the broom is detected on the table by a proximity sensor.

成功标准: 当接近传感器检测到扫帚放置在桌子上时，任务即视为成功。

Objects: A broom, a broom holder, a cupboard, a table, and a proximity sensor for detecting the broom on the table.

物品: 一把扫帚，一个扫帚架，一个橱柜，一张桌子，以及一个用于检测扫帚放置在桌子上的接近传感器。

Language Instructions for Full Interaction: Take the broom out of the cupboard and place it on the table.

完整交互的语言指令: 把扫帚从橱柜中取出并放到桌子上。

Language Instructions for Half Interactions: Pick up the broom in the cupboard; Place the broom on the table.

半交互的语言指令: 从橱柜里拿起扫帚；把扫帚放到桌子上。

Variation Number: 1

变体编号:1

(h) sweep_to_dustpan

(h) 扫向簸箕

Task Description: The robot must use a broom to sweep dirt into a dustpan.

任务描述: 机器人必须使用扫帚将灰尘扫入簸箕中。

Success Metric: The task is considered successful when all dirt particles are detected in the dustpan by a proximity sensor.

成功标准: 当所有灰尘颗粒被靠近传感器检测到在簸箕中时，任务即视为成功。

Objects: A broom, a dustpan, and multiple dirt particles.

物体: 一把扫帚、一只簸箕和多个灰尘颗粒。

Language Instructions for Full Interaction: Sweep dirt to dustpan.

完整交互的语言指令: 将灰尘扫入簸箕。

Task name	Language Template	#Items	#Variations	Variation Type
(j) put_in_without_close	”Open the <top middle=”” bottom=””> drawer; Put the block in the <top middle=”” bottom=””> drawer.”</top></top>	1	3	category, placement
(k) take_out_without_close	”Open the <top middle=”” bottom=””> drawer; Take the block out of the <top middle=”” bottom=””> drawer and place the block on the drawer’s surface.”</top></top>	1	3	category, placement
(l) put_in_and_close	”Open the <top middle=”” bottom=””> drawer; Put the block in the <top middle=”” bottom=””> drawer; Close the <top middle=”” bottom=””> drawer.”</top></top></top>	1	3	category, placement
(m) take_out_and_close	”Open the <top middle=”” bottom=””> drawer; Take the block out of the <top middle=”” bottom=””> drawer and place the block on the drawer’ s surface; Close the <top middle=”” bottom=””> drawer.”</top></top></top>	1	3	category, placement
(n) put_two_in_same	”Open the <top middle=”” bottom=””> drawer; Put the block in the <top middle=”” bottom=””> drawer; Put the block in the < top/middle/bottom > drawer.”</top></top>	2	3	category, placement
(o) take_two_out_of_same	”Open the <top middle=”” bottom=””> drawer; Take the block out of the <top middle=”” bottom=””> drawer and place the block on the drawer’s surface; Take the block out of the <top middle=”” bottom=””> drawer and place the block on the drawer’s surface.”</top></top></top>	2	3	category, placement
(p) put_two_in_different	”Open the <top middle=”” bottom=””> drawer; Put one block in the <top middle=”” bottom=””> drawer; Open the <top middle=”” bottom=””> drawer; Put the other block in the <top middle=”” bottom=””> drawer.”</top></top></top></top>	2	6	category, placement
(q) take_two_out_of_different	”Open the <top middle=”” bottom=””> drawer; Take one block out of the <top middle=”” bottom=””> drawer and place the block on the drawer’s surface; Open the <top middle=”” bottom=””> drawer; Take the other block out of the < top/middle/bottom > drawer and place the block on the drawer’s surface.”</top></top></top>	2	6	category, placement
(r) box_exchange	”Put the sugar on the table and put the spam in the cupboard.”	2	1	placement
(s) sweep_and_drop	”Drop the rubbish into the dustpan and sweep dirt into the dustpan.”	2	1	placement
(t) transfer_box	”Open the < top/middle/bottom > drawer; Take the strawberry jello out of the < top/middle/bottom > drawer and place it in the cupboard.”	2	3	category, placement

任务名称	语言模板	物品数量	变体数量	变体类型
(j) 放入且不关闭	“打开 <b0></b0> 抽屉；将积木放入 <b0></b0> 抽屉。”	1	3	类别，放置
(k) 取出且不关闭	“打开 <b0></b0> 抽屉；将积木从 <b0></b0> 抽屉取出并放置在抽屉表面。”	1	3	类别，放置
(l) 放入并关闭	“打开 <b0></b0> 抽屉；将积木放入 <b0></b0> 抽屉；关闭 <b0></b0> 抽屉。”	1	3	类别，放置
(m) 取出并关闭	“打开 <b0></b0> 抽屉；将积木从 <b0></b0> 抽屉取出并放置在抽屉表面；关闭 <b0></b0> 抽屉。”	1	3	类别，放置
(n) 两个放入同一抽屉	“打开 < 抽屉；将积木放入 < 抽屉；将积木放入 < > 顶部/中部/底部 <b2></b2> 抽屉。”	2	3	类别，放置
(o) 两个取出同一抽屉	“打开 <b0></b0> 抽屉；将积木从 <b0></b0> 抽屉取出并放置在抽屉表面；将积木从 <b0></b0> 抽屉取出并放置在抽屉表面。”	2	3	类别，放置
(p) 两个放入不同抽屉	“打开 <b0></b0> 抽屉；将一个积木放入 <b0></b0> 抽屉；打开 <b0></b0> 抽屉；将另一个积木放入 <b0></b0> 抽屉。”	2	6	类别，放置
(q) 两个取出不同抽屉	“打开 < 抽屉；将一个积木从 < 抽屉取出并放置在抽屉表面；打开 < 抽屉；将另一个积木从 < > 顶部/中部/底部 <b2></b2> 抽屉取出并放置在抽屉表面。”	2	6	类别，放置
(r) 盒子交换	“把糖放在桌子上，把午餐肉放进橱柜里。”	2	1	放置
(s) 扫除并倒入	“把垃圾倒进簸箕，把灰尘扫进簸箕。”	2	1	放置
(t) 转移盒子	“打开 < > 顶部/中部/底部 <b2></b2> 抽屉；将草莓果冻从 < > 顶部/中部/底部 <b2></b2> 抽屉取出并放入橱柜。”	2	3	类别，放置

Table 9: Properties of the compositional long-horizon tasks in DeCoBench. We report on language template, the number of items that the robot can interact with, the task variations and variation type.

表 9:DeCoBench 中组合式长时任务的属性。我们报告了语言模板、机器人可交互物品数量、任务变体及变体类型。

Language Instructions for Half Interactions: Pick up the broom on the table; Sweep dirt to dustpan.

半交互语言指令: 拿起桌上的扫帚；将灰尘扫入簸箕。

Variation Number: 1

变体编号:1

(i) rubbish_in_dustpan

(i) 垃圾入簸箕

Task Description: The robot must drop a piece of rubbish into a dustpan.

任务描述: 机器人必须将一块垃圾放入簸箕中。

Success Metric: The task is considered successful when the rubbish is detected inside the dustpan by a proximity sensor.

成功标准: 当接近传感器检测到垃圾已放入簸箕内时，任务视为成功。

Objects: A rubbish object and a dustpan with a proximity sensor to detect when the rubbish is placed inside.

物体: 一件垃圾物体和一个带有接近传感器的簸箕，用于检测垃圾是否放入其中。

Language Instructions for Full Interaction: Drop the rubbish into the dustpan.

全交互语言指令: 将垃圾放入簸箕。

Language Instructions for Half Interactions: Pick up the rubbish on the table; Drop the rubbish into the dustpan.

半交互语言指令: 拿起桌上的垃圾；将垃圾放入簸箕。

Variation Number: 1

变体编号:1

D.2 Compositional Long-horizon Tasks in DeCoBench

D.2 DeCoBench 中的组合式长时任务

(j) put_in_without_close

(j) 放入但不关闭

Task Description: The robot must place a block into a specified drawer (top, middle, or bottom) without closing the drawer afterward.

任务描述: 机器人必须将一个积木放入指定的抽屉 (上、中、下)，但之后不关闭抽屉。

Success Metric: The task is considered successful when the block is detected inside the specified drawer by a proximity sensor.

成功标准: 当接近传感器检测到积木已放入指定抽屉内时，任务视为成功。

Objects: A cabinet with three drawers (top, middle, bottom), a block, and a proximity sensor for detecting the block's presence in the drawer.

物体: 一个带有三个抽屉 (上、中、下) 的柜子，一个积木，以及用于检测积木是否在抽屉内的接近传感器。

Language Instructions for Full Interaction: Open the <top/middle/bottom> drawer; Put the block in the <top/middle/bottom> drawer.

全交互语言指令: 打开 <top/middle/bottom> 抽屉；将积木放入 <top/middle/bottom> 抽屉。

Language Instructions for Half Interactions: Grasp the <top/middle/bottom> drawer handle; Pull the <top/middle/bottom> drawer open; Pick up the block on the drawer's surface; Place the block in the <top/middle/bottom> drawer.

半交互语言指令: 抓住 <top/middle/bottom> 抽屉把手；拉开 <top/middle/bottom> 抽屉；拿起抽屉表面的积木；将积木放入 <top/middle/bottom> 抽屉。

(k) take_out_without_close

(k) 不关闭抽屉取出

Task Description: The robot must take a block out of a specified drawer (top, middle, or bottom) and place it on the drawer's surface without closing the drawer afterward.

任务描述: 机器人必须从指定的抽屉 (上、中、下) 取出一个积木，并将其放在抽屉表面，且之后不关闭抽屉。

Success Metric: The task is considered successful when the block is detected on the drawer's surface by a proximity sensor.

成功标准: 当接近传感器检测到积木位于抽屉表面时，任务视为成功。

Objects: A cabinet with three drawers (top, middle, bottom), a block, and a proximity sensor for detecting the block's presence on the drawer's surface.

物体: 一个带有三个抽屉 (上、中、下) 的柜子，一个积木，以及用于检测积木是否在抽屉表面的接近传感器。

Language Instructions for Full Interaction: Open the <top/middle/bottom> drawer; Take the block out of the <top/middle/bottom> drawer and place the block on the drawer's surface.

全交互语言指令: 打开 <top/middle/bottom> 抽屉；从 <top/middle/bottom> 抽屉取出积木并放在抽屉表面。

Language Instructions for Half Interactions: Grasp the <top/middle/bottom> drawer handle; Pull the <top/middle/bottom> drawer open; Pick up the block in the <top/middle/bottom> drawer; Place the block on the drawer's surface.

半交互语言指令: 抓住 <top/middle/bottom> 抽屉把手; 拉开 <top/middle/bottom> 抽屉; 拿起 <top/middle/bottom> 抽屉内的积木；将积木放在抽屉表面。

Variation Number: 3

变体编号:3

(l) put_in_and_close

(l) 放入并关闭

Task Description: The robot must place a block into a specified drawer (top, middle, or bottom) and then close the drawer.

任务描述: 机器人必须将积木放入指定的抽屉 (上、中、下)，然后关闭抽屉。

Success Metric: The task is considered successful when the block is detected inside the specified drawer and the drawer is closed, verified by a proximity sensor.

成功标准: 当接近传感器检测到积木位于指定抽屉内且抽屉已关闭时，任务视为成功。

Objects: A cabinet with three drawers (top, middle, bottom), a block, and a proximity sensor for detecting the block's presence in the drawer.

物体: 一个带有三个抽屉 (上、中、下) 的柜子，一个积木，以及用于检测积木是否在抽屉内的接近传感器。

Language Instructions for Full Interaction: Open the <top/middle/bottom> drawer; Put the block in the <top/middle/bottom> drawer; Close the <top/middle/bottom> drawer.

全交互语言指令: 打开 <top/middle/bottom> 抽屉；将积木放入 <top/middle/bottom> 抽屉；关闭 <top/middle/bottom> 抽屉。

Language Instructions for Half Interactions: Grasp the <top/middle/bottom> drawer handle; Pull the <top/middle/bottom> drawer open; Pick up the block on the drawer's surface; Place the block in the <top/middle/bottom> drawer; Move close to the <top/middle/bottom> drawer handle; Push the <top/middle/bottom> drawer shut.

半交互语言指令: 抓住 <top/middle/bottom> 抽屉把手; 拉开 <top/middle/bottom> 抽屉; 拿起抽屉表面的积木；将积木放入 <top/middle/bottom> 抽屉；靠近 <top/middle/bottom> 抽屉把手；推上 <top/middle/bottom> 抽屉使其关闭。

Variation Number: 3 (m) take_out_and_close

变体编号:3 (m) 取出并关闭

Task Description: The robot must take a block out of a specified drawer (top, middle, or bottom), place it on the drawer's surface, and then close the drawer.

任务描述: 机器人必须从指定的抽屉 (上、中、下) 取出一个积木，放在抽屉表面，然后关闭抽屉。

Success Metric: The task is considered successful when the block is detected on the drawer's surface and the drawer is closed, verified by a proximity sensor.

成功标准: 当块被检测到位于抽屉表面且抽屉关闭，由接近传感器确认时，任务视为成功。

Objects: A cabinet with three drawers (top, middle, bottom), a block, and a proximity sensor for detecting the block's presence on the drawer's surface and confirming the drawer closure.

物体: 一个带有三个抽屉 (上、中、下) 的柜子，一个块，以及用于检测块是否位于抽屉表面并确认抽屉关闭的接近传感器。

Language Instructions for Full Interaction: Open the <top/middle/bottom> drawer; Take the block out of the < top/middle/bottom > drawer and place the block on the drawer’s surface; Close the <top/middle/bottom> drawer.

完整交互语言指令: 打开 < 上/中/下 > 抽屉；将块从 < 上/中/下 > 抽屉中取出并放置在抽屉表面；关闭 < 上/中/下 > 抽屉。

Language Instructions for Half Interactions: Grasp the <top/middle/bottom> drawer handle; Pull the <top/middle/bottom> drawer open; Pick up the block in the <top/middle/bottom> drawer;

半交互语言指令: 抓住 <b0></b0> 上/中/下 <b1></b1> 抽屉把手; 拉开 <b0></b0> 上/中/下 <b1></b1> 抽屉；拿起位于 <b0></b0> 上/中/下 <b1></b1> 抽屉中的块；

Place the block on the drawer’s surface; Move close to the <top/middle/bottom> drawer handle; Push the <top/middle/bottom> drawer shut.

将块放置在抽屉表面；靠近 <b0></b0> 上/中/下 <b1></b1> 抽屉把手；推闭 <b0></b0> 上/中/下 <b1></b1> 抽屉。

Variation Number: 3

变体编号:3

(n) put_two_in_same

(n) put_two_in_same

Task Description: The robot must place two blocks into a specified drawer (top, middle, or bottom).

任务描述: 机器人必须将两个块放入指定的抽屉 (上、中或下)。

Success Metric: The task is considered successful when both blocks are detected inside the specified drawer by a proximity sensor.

成功标准: 当两个块均被接近传感器检测到位于指定抽屉内时，任务视为成功。

Objects: A cabinet with three drawers (top, middle, bottom), two blocks, and a proximity sensor for detecting the blocks’ presence in the drawer.

物体: 一个带有三个抽屉 (上、中、下) 的柜子，两个块，以及用于检测块是否位于抽屉内的接近传感器。

Language Instructions for Full Interaction: Open the <top/middle/bottom> drawer; Put the block in the <top/middle/bottom> drawer; Put the block in the <top/middle/bottom> drawer.

完整交互语言指令: 打开 <b0></b0> 上/中/下 <b1></b1> 抽屉；将块放入 <b0></b0> 上/中/下 <b1></b1> 抽屉；将块放入 <b0></b0> 上/中/下 <b1></b1> 抽屉。

Language Instructions for Half Interactions: Grasp the <top/middle/bottom> drawer handle; Pull the <top/middle/bottom> drawer open; Pick up the block on the drawer's surface; Place the block in the <top/middle/bottom> drawer; Pick up the block on the drawer's surface; Place the block in the <top/middle/bottom> drawer.

半交互语言指令: 抓住 <b0></b0> 上/中/下 <b1></b1> 抽屉把手; 拉开 <b0></b0> 上/中/下 <b1></b1> 抽屉；拿起位于抽屉表面的块；将块放入 <b0></b0> 上/中/下 <b1></b1> 抽屉；拿起位于抽屉表面的块；将块放入 <b0></b0> 上/中/下 <b1></b1> 抽屉。

Variation Number: 3

变体编号:3

(o) take_two_out_of_same

(o) take_two_out_of_same

Task Description: The robot must take two blocks out of a specified drawer (top, middle, or bottom) and place them on the drawer's surface.

任务描述: 机器人必须从指定的抽屉 (上、中或下) 中取出两个块，并将它们放置在抽屉表面。

Success Metric: The task is considered successful when both blocks are detected on the drawer's surface by a proximity sensor.

成功标准: 当两个块均被接近传感器检测到位于抽屉表面时，任务视为成功。

Objects: A cabinet with three drawers (top, middle, bottom), two blocks, and a proximity sensor for detecting the blocks' presence on the drawer's surface.

物体: 一个带有三个抽屉 (上、中、下) 的柜子，两个积木，以及一个用于检测积木是否放置在抽屉表面的接近传感器。

Language Instructions for Full Interaction: Open the <top/middle/bottom> drawer; Take the block out of the < top/middle/bottom > drawer and place the block on the drawer's surface; Take the block out of the <top/middle/bottom> drawer and place the block on the drawer's surface.

完整交互的语言指令: 打开 < 上/中/下 > 抽屉；从 < 上/中/下 > 抽屉中取出积木并将积木放置在抽屉表面；从上/中/下抽屉中取出积木并将积木放置在抽屉表面。

Language Instructions for Half Interactions: Grasp the <top/middle/bottom> drawer handle; Pull the <top/middle/bottom> drawer open; Pick up the block in the <top/middle/bottom> drawer; Place the block on the drawer's surface; Pick up the block in the <top/middle/bottom> drawer; Place the block on the drawer's surface.

半交互的语言指令: 抓住上/中/下抽屉的把手；拉开上/中/下抽屉；拿起上/中/下抽屉中的积木；将积木放置在抽屉表面；拿起上/中/下抽屉中的积木；将积木放置在抽屉表面。

Variation Number: 3

变体编号:3

(p) put_two_in_different

(p) put_two_in_different

Task Description: The robot must place two blocks into two different specified drawers (top, middle, or bottom).

任务描述: 机器人必须将两个积木分别放入两个不同指定的抽屉 (上、中或下)。

Success Metric: The task is considered successful when each block is detected inside its corresponding drawer by a proximity sensor.

成功标准: 当每个积木被对应抽屉内的接近传感器检测到时，任务视为成功。

Objects: A cabinet with three drawers (top, middle, bottom), two blocks, and two proximity sensors for detecting each block's presence in its designated drawer.

物体: 一个带有三个抽屉 (上、中、下) 的柜子，两个积木，以及两个用于检测每个积木是否放置在指定抽屉内的接近传感器。

Language Instructions for Full Interaction: Open the <top/middle/bottom> drawer; Put one block in the <top/middle/bottom> drawer; Close the <top/middle/bottom> drawer; Open the <top/middle/bottom> drawer; Put the other block in the <top/middle/bottom> drawer.

完整交互的语言指令: 打开上/中/下抽屉；将一个积木放入上/中/下抽屉；关闭上/中/下抽屉；打开上/中/下抽屉；将另一个积木放入上/中/下抽屉。

Language Instructions for Half Interactions: Grasp the <top/middle/bottom> drawer handle; Pull the <top/middle/bottom> drawer open; Pick up the block on the drawer's surface; Place the block in the <top/middle/bottom> drawer; Move close to the <top/middle/bottom> drawer handle; Push the <top/middle/bottom> drawer shut; Grasp the <top/middle/bottom> drawer handle; Pull the <top/middle/bottom> drawer open; Pick up the block on the drawer's surface; Place the block in the <top/middle/bottom> drawer.

半交互的语言指令: 抓住上/中/下抽屉的把手；拉开上/中/下抽屉；拿起放在抽屉表面的积木；将积木放入上/中/下抽屉；靠近上/中/下抽屉把手；推上/中/下抽屉关闭；抓住上/中/下抽屉的把手；拉开上/中/下抽屉；拿起放在抽屉表面的积木；将积木放入上/中/下抽屉。

Variation Number: 6

变体编号:6

(q) take_two_out_of_different

(q) take_two_out_of_different

Task Description: The robot must take one block out of a specified drawer (top, middle, or bottom) and take another block out of a different specified drawer, then place both blocks on the drawer's surface.

任务描述: 机器人必须从指定的一个抽屉 (上、中或下) 中取出一个积木，再从另一个不同指定的抽屉中取出另一个积木，然后将两个积木都放置在抽屉表面。

Success Metric: The task is considered successful when both blocks are detected on the drawer's surface by a proximity sensor.

成功标准: 当两个积木都被抽屉表面的接近传感器检测到时，任务视为成功。

Objects: A cabinet with three drawers (top, middle, bottom), two blocks, and a proximity sensor for detecting each block's presence on the drawer's surface.

物体: 一个带有三个抽屉 (上、中、下) 的柜子，两个积木，以及一个用于检测每个积木是否放置在抽屉表面的接近传感器。

Language Instructions for Full Interaction: Open the <top/middle/bottom> drawer; Take one block out of the < top/middle/bottom > drawer and place the block on the drawer's surface; Close the <top/middle/bottom> drawer; Open the <top/middle/bottom> drawer; Take the other block out of the <top/middle/bottom> drawer and place the block on the drawer's surface.

完整交互的语言指令: 打开 < 上/中/下 > 抽屉；从 < 上/中/下 > 抽屉中取出一个积木并将积木放置在抽屉表面；关闭上/中/下抽屉；打开上/中/下抽屉；从上/中/下抽屉中取出另一个积木并将积木放置在抽屉表面。

Language Instructions for Half Interactions: Grasp the <top/middle/bottom> drawer handle; Pull the <top/middle/bottom> drawer open; Pick up the block in the <top/middle/bottom> drawer; Place the block on the drawer's surface; Push the <top/middle/bottom> drawer shut; Grasp the <top/middle/bottom> drawer handle; Pull the <top/middle/bottom> drawer open; Pick up the block in the <top/middle/bottom> drawer; Place the block on the drawer's surface.

半交互语言指令: 抓住 <top/middle/bottom> 抽屉把手; 拉开 <top/middle/bottom> 抽屉; 拿起 <top/middle/bottom> 抽屉中的积木；将积木放在抽屉表面；推上 <top/middle/bottom> 抽屉；抓住 <top/middle/bottom> 抽屉把手；拉开 <top/middle/bottom> 抽屉；拿起 <top/middle/bottom> 抽屉中的积木；将积木放在抽屉表面。

Variation Number: 6

变体编号:6

(s) box_exchange

(s) box_exchange

Task Description: The robot must exchange the positions of two grocery items: the sugar and the spam. Specifically, the robot will place the sugar on the table and put the spam in the cupboard.

任务描述: 机器人必须交换两种杂货物品的位置: 糖和午餐肉。具体来说，机器人将把糖放在桌子上，把午餐肉放进橱柜。

Success Metric: The task is considered successful when the sugar is detected on the table and the spam is detected in the cupboard by their respective proximity sensors.

成功标准: 当糖被其接近传感器检测到放置在桌子上，午餐肉被其接近传感器检测到放置在橱柜内时，任务视为成功。

Objects: Two grocery items (sugar and spam), a table, a cupboard, and proximity sensors for detecting the placement of each item.

物品: 两种杂货物品 (糖和午餐肉)、一张桌子、一只橱柜，以及用于检测各物品放置位置的接近传感器。

Language Instructions for Full Interaction: Put the sugar on the table and put the spam in the cupboard.

全交互语言指令: 将糖放在桌子上，将午餐肉放进橱柜。

Language Instructions for Half Interactions: Pick up the sugar in the cupboard; Place the sugar on the table; Pick up the spam on the table; Place the spam in the cupboard.

半交互语言指令: 从橱柜中拿起糖；将糖放在桌子上；从桌子上拿起午餐肉；将午餐肉放进橱柜。

Variation Number: 1

变体编号:1

(t) sweep_and_drop

(t) sweep_and_drop

Task Description: The robot must clean all dirt and rubbish into a dustpan using a broom.

任务描述: 机器人必须使用扫帚将所有污垢和垃圾扫入簸箕中。

Success Metric: The task is considered successful when all dirt pieces and the rubbish are detected in the dustpan by a proximity sensor.

成功标准: 当所有污垢碎片和垃圾被接近传感器检测到集中在簸箕中时，任务视为成功。

Objects: A broom, a dustpan, several pieces of dirt (5 in total), and a piece of rubbish, along with a proximity sensor for detecting their presence in the dustpan.

物品: 一把扫帚、一只簸箕、若干污垢碎片 (共5个) 和一块垃圾，以及用于检测它们是否在簸箕中的接近传感器。

Language Instructions for Full Interaction: Drop the rubbish into the dustpan and sweep dirt into the dustpan.

全交互语言指令: 将垃圾放入簸箕，扫除污垢至簸箕中。

Language Instructions for Half Interactions: Pick up the rubbish on the table; Drop the rubbish into the dustpan; Pick up the broom on the table; Sweep dirt into the dustpan.

半交互语言指令: 从桌子上拿起垃圾；将垃圾放入簸箕；从桌子上拿起扫帚；将污垢扫入簸箕。

Variation Number: 1

变体编号:1

(u) transfer_box

(u) transfer_box

Task Description: The robot must take the strawberry jello out of a specified drawer (top, middle, or bottom) and place it in the cupboard.

任务描述: 机器人必须将草莓果冻从指定的抽屉 (上层、中层或下层) 取出并放入橱柜中。

Success Metric: The task is considered successful when the strawberry jello is detected inside the cupboard by a proximity sensor.

成功标准: 当接近传感器检测到草莓果冻位于橱柜内时，任务即视为成功。

Objects: A drawer with three compartments (top, middle, bottom), a strawberry jello item, a cupboard, and a proximity sensor for detecting the item’s presence in the cupboard.

物体: 一个有三个隔层 (上层、中层、下层) 的抽屉、一份草莓果冻、一只橱柜，以及用于检测物品是否在橱柜内的接近传感器。

Language Instructions for Full Interaction: Open the <top/middle/bottom> drawer; Take the strawberry jello out of the < top/middle/bottom > drawer and place the strawberry jello on the drawer's surface; Put the strawberry jello in the cupboard.

完整交互的语言指令: 打开 < 上层/中层/下层 > 抽屉；将草莓果冻从 < 上层/中层/下层 > 抽屉中取出，放在抽屉表面；将草莓果冻放入橱柜。

Language Instructions for Half Interactions: Grasp the <top/middle/bottom> drawer handle; Pull the <top/middle/bottom> drawer open; Pick up the strawberry jello in the <top/middle/bottom> drawer; Place the strawberry jello in the cupboard.

半交互的语言指令: 抓住上层/中层/下层抽屉把手；拉开上层/中层/下层抽屉；拿起上层/中层/下层抽屉中的草莓果冻；将草莓果冻放入橱柜。

Variation Number: 3

变体数量:3

(v) retrieve_and_sweep

(v) retrieve_and_sweep

Task Description: The robot must sweep dirt into a dustpan using a broom.

任务描述: 机器人必须使用扫帚将灰尘扫入簸箕中。

Success Metric: The task is considered successful when all dirt pieces are detected in the dustpan by a proximity sensor.

成功标准: 当接近传感器检测到所有灰尘颗粒均在簸箕内时，任务即视为成功。

Objects: A broom, multiple pieces of dirt (5 in total), a dustpan, and a proximity sensor for detecting the presence of dirt in the dustpan.

物体: 一把扫帚、若干灰尘颗粒 (共 5 个)、一只簸箕，以及用于检测灰尘是否在簸箕内的接近传感器。

Language Instructions for Full Interaction: Put the broom on the table and sweep dirt into the dustpan.

完整交互的语言指令: 将扫帚放在桌子上，扫动灰尘至簸箕中。

Language Instructions for Half Interactions: Pick up the broom in the cupboard; Sweep dirt into the dustpan.

半交互的语言指令: 从橱柜中拿起扫帚；将灰尘扫入簸箕中。

Variation Number: 1

变体数量:1

E Real-world Experiments

E 真实世界实验

In the following we provide details of the real-world setup and tasks. Figure 9 illustrates the real-world setup. Figure 10 visualizes the real-world atomic tasks, Figure 11 visualizes the real-world compositional long-horizon tasks. A summary of the 6 atomic tasks is provided in Table 10. A summary of the 9 compositional long-horizon tasks is provided in Table 11. Video demonstrations of the real-world tasks are provided on our website : https://deco226.github.io.

以下内容提供了真实世界设置和任务的详细信息。图 9 展示了真实世界的设置。图 10 展示了真实世界的原子任务，图 11 展示了真实世界的组合长时任务。表 10 总结了 6 个原子任务，表 11 总结了 9 个组合长时任务。真实世界任务的视频演示可在我们的网站观看:https://deco226.github.io。

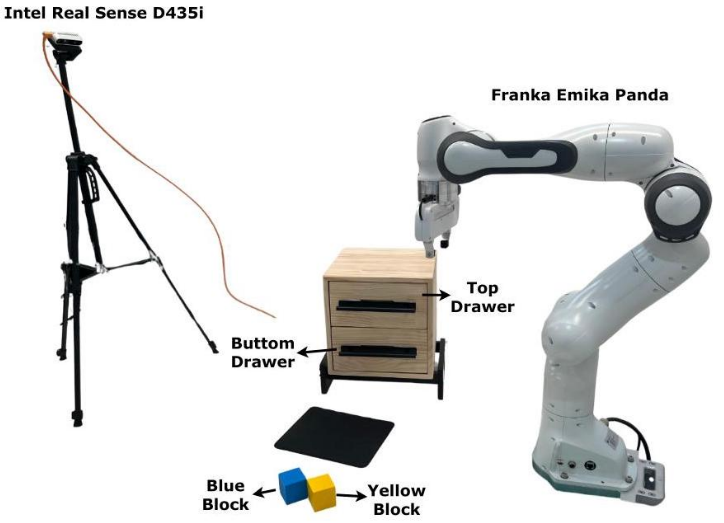


Figure 9: Real-Robot Setup with RealSense D435i and Franka Emika Panda.

图 9: 配备 RealSense D435i 和 Franka Emika Panda 的真实机器人设置。

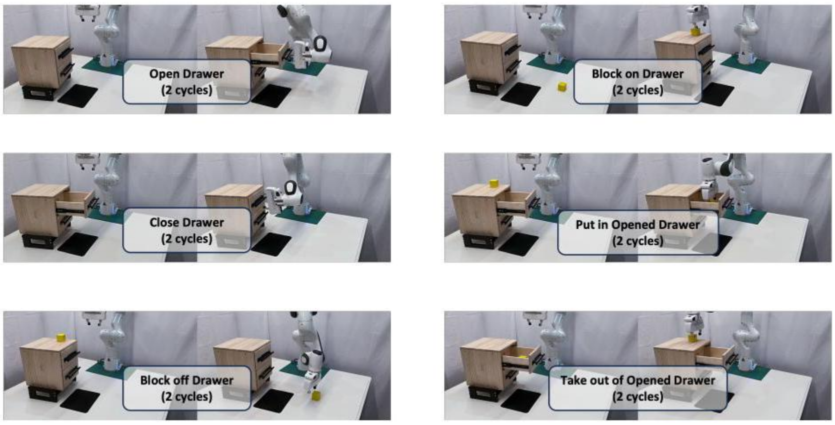


Figure 10: Real-World Atomic Tasks.

图 10: 真实世界的原子任务。

E.1 Real-World Atomic Tasks

E.1 真实世界的原子任务

(a) open_drawer

(a) 打开抽屉

Task Description: The robot must open a specified drawer.

任务描述: 机器人必须打开指定的抽屉。

Success Metric: The task is successful if the target drawer is fully opened.

成功标准: 如果目标抽屉完全打开，则任务成功。

Objects: One drawer.

物体: 一个抽屉。

Language Instructions: Open the <top/bottom> drawer.

语言指令: 打开 <top/bottom> 抽屉。

Variation Number: 2

变体数量:2

(b) close_drawer

(b) 关闭抽屉

Task Description: The robot must close a specified drawer.

任务描述: 机器人必须关闭指定的抽屉。

Task name	Language Template	#Items	#Variations	Variation Type
(a) open_drawer	”Open the < top/bottom > drawer.”	1	2	placement
(b) close_drawer	”Close the <top bottom=””> drawer.”</top>	1	2	placement
(c) put_in_opened_drawer	”Put the <blue yellow=””> block in the <top bottom=””> drawer.”</top></blue>	2	4	color, placement
(d) take_out_of_opened_drawer	”Take the <blue yellow=””> block out of the <top bottom=””> drawer and place the <blue yellow=””> block on the drawer’s surface.”</blue></top></blue>	2	4	color, placement
(e) block_on_drawer	”Put the <blue yellow=””> block on the drawer’s surface.”</blue>	2	2	color
(f) block_off_drawer	”Put the <blue yellow=””> block on the table.”</blue>	3	2	color

任务名称	语言模板	物品数量	变体数量	变体类型
(a) 打开抽屉	“打开 < 上层/下层 > 抽屉。”	1	2	位置
(b) 关闭抽屉	“关闭抽屉。”	1	2	位置
(c) 放入已打开的抽屉	“将 <blue yellow=””> 积木放入 <top bottom=””> 抽屉中。” </top></blue>	2	4	颜色，位置
(d) 从已打开的抽屉取出	“将 <blue yellow=””> 积木从 <top bottom=””> 抽屉中取出，并将 <blue yellow=””> 积木放在抽屉表面。” </blue></top></blue>	2	4	颜色，位置
(e) 积木放在抽屉上	“将 <blue yellow=””> 积木放在抽屉表面。” </blue>	2	2	颜色
(f) 积木放在抽屉外	“将 <blue yellow=””> 积木放在桌子上。” </blue>	3	2	颜色

Table 10: Properties of atomic tasks in real-world experiments.

表 10: 真实环境实验中原子任务的属性。

Success Metric: The task is successful if the target drawer is completely closed.

成功标准: 如果目标抽屉完全关闭，则任务成功。

Objects: One drawer.

物体: 一个抽屉。

Language Instructions: Close the <top/bottom> drawer.

语言指令: 关闭 <top/bottom> 抽屉。

Variation Number: 2

变体数量:2

(c) put_in_opened_drawer

(c) 放入已打开的抽屉

Task Description: The robot places a colored block into an already opened drawer.

任务描述: 机器人将一个彩色积木放入已打开的抽屉中。

Success Metric: The block must be placed inside the correct opened drawer.

成功标准: 积木必须放入正确的已打开抽屉内。

Objects: One colored block and one opened drawer.

物体: 一个彩色积木和一个已打开的抽屉。

Language Instructions: Put the <blue/yellow> block in the <top/bottom> drawer.

语言指令: 将 <blue/yellow> 积木放入 <top/bottom> 抽屉。

Variation Number: 4

变体数量:4

(d) take_out_of_opened_drawer

(d) 从已打开的抽屉中取出

Task Description: The robot retrieves a block from an already opened drawer and places it on the drawer's surface.

任务描述: 机器人从已打开的抽屉中取出一个积木，并将其放置在抽屉表面。

Success Metric: The correct block is taken out and placed on the drawer surface.

成功标准: 正确的积木被取出并放置在抽屉表面。

Objects: One colored block and one opened drawer.

物体: 一个彩色积木和一个已打开的抽屉。

Language Instructions: Take the <blue/yellow> block out of the <top/bottom> drawer and place the <blue/yellow > block on the drawer's surface.

语言指令: 从 <top/bottom> 抽屉中取出 <blue/yellow> 积木，并将 <blue/yellow> 积木放在抽屉表面。

Variation Number: 4

变体编号:4

(e) block_on_drawer

(e) 放置方块于抽屉上

Task Description: The robot places a block on top of the drawer's surface. Success Metric: The block is correctly placed on the surface of the drawer. Objects: One block and one drawer. Language Instructions: Put the <blue/yellow > block on the drawer's surface. Variation Number: 2

任务描述: 机器人将一个方块放置在抽屉表面上。成功标准: 方块正确放置在抽屉表面。物体: 一个方块和一个抽屉。语言指令: 将 <blue/yellow > 方块放在抽屉表面。变体编号:2

(f) block_off_drawer

(f) 放置方块于抽屉外

Task Description: The robot places a block on the table, away from the drawer. Success Metric: The block is placed on the table, not on the drawer.

任务描述: 机器人将一个方块放置在桌子上，远离抽屉。成功标准: 方块放置在桌子上，而非抽屉上。

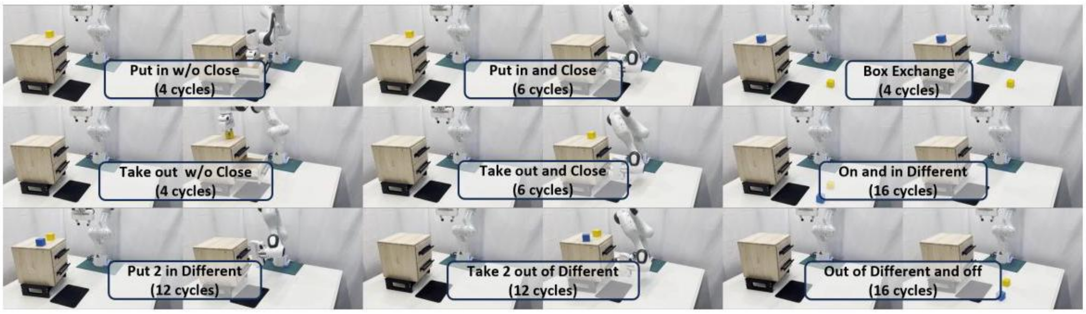


Figure 11: Real-World Compositional Long-Horizon Tasks.

图 11: 现实世界的组合长时任务。

Task name	Language Template	#Items	#Variations	Variation Type
(g) put_in_without_close	”Put the <blue yellow=""> block in the closed <top bottom=""> drawer.”</top></blue>	1	4	color, placement
(h) put_in_and_close	”Put the <blue yellow=""> block in the closed <top bottom=""> drawer and close it.”</top></blue>	1	4	color, placement
(i) take_out_without_close	”Take the <blue yellow=""> block out of the <top bottom=""> closed drawer and place the <blue yellow=""> block on the drawer's surface.”</blue></top></blue>	1	4	color, placement
(j) take_out_and_close	”Take the <blue yellow=""> block out of the <top bottom=""> drawer, place the <blue yellow=""> block on the drawer's surface, and then close the <top bottom=""> drawer.”</top></blue></top></blue>	1	4	color, placement
(k) put_two_in_different	”Put the <blue yellow=""> block in the <top bottom=""> closed drawer and put the <blue yellow=""> block in the <top bottom=""> closed drawer.”</top></blue></top></blue>	2	4	color, placement
(l) take_two_out_of_different	”Take the <blue yellow=""> block out of the <top bottom=""> closed drawer and take the <blue yellow=""> block out of the <top bottom=""> closed drawer, then place them on the drawer's surface.”</top></blue></top></blue>	2	4	color, placement
(m) block_exchange	”Exchange the positions of the two blocks.”	2	2	color, placement
(n) on_and_in_different	”Place both blocks on the drawer's surface, then put the <blue yellow=""> block in the < top/bottom > drawer and the < blue/yellow > block in the < top/bottom > drawer.”</blue>	2	2	color, placement
(o) out_of_different_and_off	”Take the <blue yellow=""> block out of the <top bottom=""> drawer and take the <blue yellow=""> block out of the <top bottom=""> drawer, then put both blocks on the table.”</top></blue></top></blue>	2	2	color, placement

任务名称	语言模板	物品数量	变体数量	变体类型
(g) 放入但不关闭	“将 <blue yellow=""> 积木放入已关闭的 <top bottom=""> 抽屉中。”</top></blue>	1	4	颜色，位置
(h) 放入并关闭	“将 <blue yellow=""> 积木放入已关闭的 <top bottom=""> 抽屉中并关闭它。”</top></blue>	1	4	颜色，位置
(i) 取出但不关闭	“将 <blue yellow=""> 积木从 <top bottom=""> 已关闭的抽屉中取出，并将 <blue yellow=""> 积木放在抽屉表面上。”</blue></top></blue>	1	4	颜色，位置
(j) 取出并关闭	“将 <blue yellow=""> 积木从 <top bottom=""> 抽屉中取出，放在抽屉表面上，然后关闭 <top bottom=""> 抽屉。”</top></top></blue>	1	4	颜色，位置
(k) 放入两个不同的抽屉	“将 <blue yellow=""> 积木放入 <top bottom=""> 已关闭的抽屉中，再将 <blue yellow=""> 积木放入 <top bottom=""> 已关闭的抽屉中。”</top></blue></top></blue>	2	4	颜色，位置
(l) 从两个不同的抽屉取出	“将 <blue yellow=""> 积木从 <top bottom=""> 已关闭的抽屉中取出，再将 <blue yellow=""> 积木从 <top bottom=""> 已关闭的抽屉中取出，然后将它们放在抽屉表面上。”</top></blue></top></blue>	2	4	颜色，位置
(m) 积木交换	“交换两个积木的位置。”	2	2	颜色，位置
(n) 不同位置的放置和放入	“将两个积木都放在抽屉表面，然后将 <blue yellow=""> 积木放入 < 上/下 > 抽屉，将 < 蓝色/黄色 > 积木放入 < 上/下 > 抽屉。”</blue>	2	2	颜色，位置
(o) 从不同抽屉取出并放置桌面	“将 <blue yellow=""> 积木从 <top bottom=""> 抽屉中取出，再将 <blue yellow=""> 积木从 <top bottom=""> 抽屉中取出，然后将两个积木放在桌子上。”</top></blue></top></blue>	2	2	颜色，位置

Table 11: Properties of the real-world compositional long-horizon tasks.

表 11: 现实世界组合长时任务的属性。

Objects: One block, one drawer, and one table.

物体: 一个积木，一个抽屉和一张桌子。

Language Instructions: Put the <blue/yellow > block on the table.

语言指令: 将 <blue/yellow> 积木放在桌子上。

Variation Number: 2

变体数量:2

E.2 Real-World Compositional Long-horizon Tasks

E.2 现实世界组合长时任务

(g) put_in_without_close

(g) 放入但不关闭

Task Description: The robot places a block into a closed drawer without closing it.

任务描述: 机器人将积木放入一个关闭的抽屉中，但不关闭抽屉。

Success Metric: The block is placed in the correct drawer, which remains open.

成功标准: 积木被放置在正确的抽屉中，且抽屉保持打开状态。

Objects: One colored block and one drawer.

物体: 一个彩色积木和一个抽屉。

Language Instructions: Put the <blue/yellow> block in the closed <top/bottom> drawer.

语言指令: 将 <blue/yellow> 积木放入关闭的 <top/bottom> 抽屉中。

Variation Number: 4

变体数量:4

(h) put_in_and_close

(h) 放入并关闭

Task Description: The robot places a block into a closed drawer and then closes the drawer.

任务描述: 机器人将积木放入关闭的抽屉中，然后关闭抽屉。

Success Metric: The block is placed correctly, and the drawer is fully closed.

成功标准: 积木被正确放置，且抽屉完全关闭。

Objects: One colored block and one drawer.

物体: 一个彩色积木和一个抽屉。

Language Instructions: Put the <blue/yellow> block in the closed <top/bottom> drawer and close it.

语言指令: 将 <blue/yellow> 积木放入关闭的 <top/bottom> 抽屉中并关闭它。

Variation Number: 4

变体编号:4

(i) take_out_without_close

(i) 未关闭抽屉取出

Task Description: The robot takes a block out of a closed drawer and places it on the drawer's surface, leaving the drawer open.

任务描述: 机器人从关闭的抽屉中取出一个积木，放置在抽屉表面，保持抽屉打开状态。

Success Metric: The block is retrieved and placed correctly, and the drawer remains open.

成功标准: 积木被正确取出并放置，且抽屉保持打开。

Objects: One colored block and one drawer.

物体: 一个彩色积木和一个抽屉。

Language Instructions: Take the <blue/yellow> block out of the <top/bottom> closed drawer and place the <blue/yellow> block on the drawer's surface.

语言指令: 从 <top/bottom> 关闭的抽屉中取出 <blue/yellow> 积木，并将 <blue/yellow> 积木放在抽屉表面。

Variation Number: 4

变体编号:4

(j) take_out_and_close

(j) 取出并关闭

Task Description: The robot retrieves a block from a closed drawer, places it on the drawer’s surface, and then closes the drawer.

任务描述: 机器人从关闭的抽屉中取出一个积木，放置在抽屉表面，然后关闭抽屉。

Success Metric: The block is placed correctly, and the drawer is closed afterward.

成功标准: 积木被正确放置，且抽屉随后关闭。

Objects: One colored block and one drawer.

物体: 一个彩色积木和一个抽屉。

Language Instructions: Take the <blue/yellow> block out of the <top/bottom> drawer; place the <blue/yellow> block on the drawer’s surface, and then close the <top/bottom> drawer.

语言指令: 从 <top/bottom> 抽屉中取出 <blue/yellow> 积木；将 <blue/yellow> 积木放在抽屉表面，然后关闭 <top/bottom> 抽屉。

Variation Number: 4

变体编号:4

(k) put_two_in_different

(k) 将两个积木放入不同抽屉

Task Description: The robot places two blocks into two different closed drawers. Success Metric: Each block is correctly placed into the specified drawer. Objects: Two colored blocks and two drawers. Language Instructions: Put the <blue/yellow> block in the <top/bottom> closed drawer and put the <blue/yellow> block in the <top/bottom> closed drawer.

任务描述: 机器人将两个积木分别放入两个不同的关闭抽屉。成功标准: 每个积木正确放入指定抽屉。物体: 两个彩色积木和两个抽屉。语言指令: 将 <blue/yellow> 积木放入 <top/bottom> 关闭的抽屉，将 <blue/yellow> 积木放入 <top/bottom> 关闭的抽屉。

Variation Number: 4

变体编号:4

(l) take_two_out_of_different

(l) 从不同的抽屉中取出两个物体

Task Description: The robot retrieves two blocks from two different closed drawers and places them on the drawer surface.

任务描述: 机器人从两个不同的关闭抽屉中取出两个积木，并将它们放在抽屉表面。

Success Metric: Both blocks are correctly retrieved and placed.

成功标准: 两个积木均被正确取出并放置。

Objects: Two colored blocks and two drawers.

物体: 两个彩色积木和两个抽屉。

Language Instructions: Take the <blue/yellow > block out of the <top/bottom> closed drawer and take the <blue/yellow > block out of the <top/bottom> closed drawer, then place them on the drawer’s surface.

语言指令: 从 <top/bottom> 关闭的抽屉中取出 <blue/yellow> 积木，再从 <top/bottom> 关闭的抽屉中取出 <blue/yellow> 积木，然后将它们放在抽屉表面。

Variation Number: 4

变体编号:4

(m) block_exchange

(m) 积木交换

Task Description: The robot swaps the positions of two blocks.

任务描述: 机器人交换两个积木的位置。

Success Metric: The two blocks end up in each other’s original positions.

成功标准: 两个积木最终位于对方的原始位置。

Objects: Two blocks.

物体: 两个积木。

Language Instructions: Exchange the positions of the two blocks.

语言指令: 交换两个积木的位置。

Variation Number: 2

变体编号:2

(n) on_and_in_different

(n) 不同位置的放置与存放

Task Description: The robot places both blocks on the drawer surface, then puts each block into a different drawer.

任务描述: 机器人先将两个积木放在抽屉表面，然后将每个积木放入不同的抽屉中。

Success Metric: Both blocks are correctly placed and then inserted into the correct drawers.

成功标准: 两个积木均被正确放置并插入正确的抽屉。

Objects: Two colored blocks and two drawers.

物体: 两个彩色积木和两个抽屉。

Language Instructions: Place both blocks on the drawer’ s surface, then put the <blue/yellow> block in the <top/bottom> drawer and the <blue/yellow> block in the <top/bottom> drawer.

语言指令: 将两个积木块都放在抽屉表面，然后将 <blue/yellow> 积木块放入 <top/bottom> 抽屉，将 <blue/yellow> 积木块放入 <top/bottom> 抽屉。

Variation Number: 2

变体编号:2

(o) out_of_different_and_off

(o) out_of_different_and_off

Task Description: The robot takes two blocks out of two different drawers and puts them on the table.

任务描述: 机器人从两个不同的抽屉中取出两个积木块并放在桌子上。

Success Metric: Both blocks are retrieved and placed on the table surface.

成功标准: 两个积木块均被取出并放置在桌面上。

Objects: Two colored blocks and two drawers.

物体: 两个彩色积木块和两个抽屉。

Language Instructions: Take the <blue/yellow> block out of the <top/bottom> drawer and take the <blue/yellow> block out of the <top/bottom> drawer, then put both blocks on the table.

语言指令: 从 <top/bottom> 抽屉中取出 <blue/yellow> 积木块，从 <top/bottom> 抽屉中取出 <blue/yellow> 积木块，然后将两个积木块放在桌子上。

Variation Number: 2

变体编号:2