arXiv:2611.31894v2 [cs.RO] 25 Nov 2026

RESET：一次示范，一个经验证的世界，跨具身形态的自我改进技能

Zheng Sun[1*] Zhuo Li[1*] Zhihao Li[1*] Heng Yi Sim[1]Ge Wang[2] Ziyue Zhu[3] Mingkai Jia[4] Fei Chen[1]

项目主页： zainzh.github.io/RESET

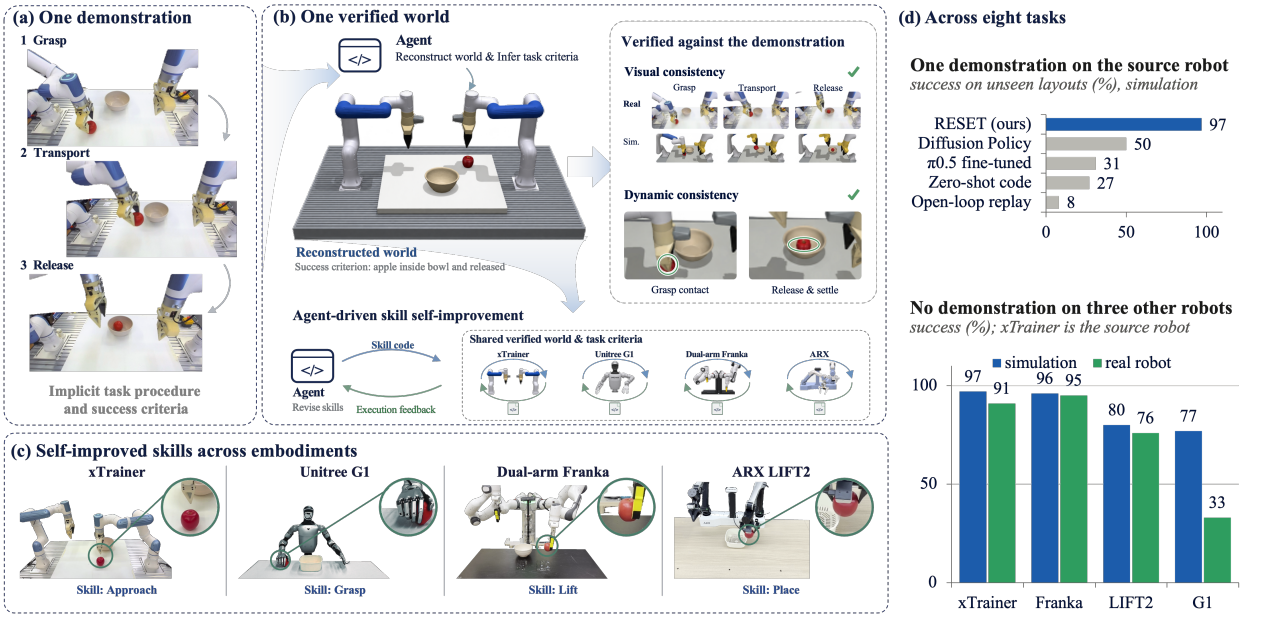


图 1. **RESET 概览**。(a) 一次真实示范隐含地指定了任务流程和终止成功条件。(b) 一个经验证的世界：RESET 重建一个丰富接触世界和只读判据（重建），并根据示范对其进行检查（检查），并根据执行反馈改进白盒控制代码（自我改进）。(c) 跨本体迁移：共享的经验证世界和判据使异构机器人无需新的示范即可获得改编技能（迁移），并在真实机器人上执行这些技能（执行）。(d) 八项任务的结果：源机器人与基线方法的比较，以及全部四个机器人在仿真和真实机器人上的表现。

摘要

大语言模型（*LLM*）编码代理能够在仿真基准测试中迭代式地自我改进机器人操作技能，然而，现实世界中的任务既没有真实成功评分器，也没有可重置练习环境。我们的关键观察是，单个真实示范能够同时描述场景几何、定义终止成功判定标准，并作为用于验证世界保真度的物理参考轨迹——而这一切都与源机器人运动学相解耦。我们提出了***RESET***：我们重建了丰富接触世界，并从一次示范中推导出冻结的只读判据（***R***重建）；通过开环重放和轮廓一致性，根据示范检查该世界（***E***检查）；根据逐节点失败证据，演化模块化白盒任务图和技能代码（***S***自我改进）；在真实机器人上部署冻结代码，无需运行时大语言模型调用（***E***执行）；无需新的示范，即可将经验证的世界和判据共享给未见机器人（***T***迁移）。在 *8* 项操作任务中，每项任务都由一次真实示范重建而成，*RESET* 在仿真中的未见布局上取得了 *97%* 的平均成功率，而零样本代码达到 *27%*，按任务微调的 $\pi_{0.5}$ 策略达到 *31%*，扩散策略达到 *50%*；注入重建缺陷表明，实践能够容忍许多重建误差，但几何误差仍可能使其停滞或产生误导，

*共同贡献。[1]香港中文大学。[2]香港中文大学（深圳）。[3]南开大学。[4]香港科技大学。

并且该检查能够标记其中大多数缺陷，而物理随机化则覆盖其余缺陷。在没有任何目标示范的情况下，*3* 个异构机器人在共享世界中获得了相同的任务技能：具有相似机械臂的机器人能够从源机器人复制 *95%* 的技能函数且无需修改，而具有不同机械臂和机械手的机器人则自行编写技能函数。在真实机器人实验中，所得程序在 *4* 个机器人上运行，并在 *280* 次试验中取得了 *80%* 的成功率。

1. 引言

大语言模型驱动的编码智能体正逐渐成为获取机器人操作技能的一种引人注目的范式。与手工设计或一次性合成的控制程序相比 [10]，后者难以扩展到接触密集型任务和多种机器人；闭环递归自我改进（RSI）使智能体能够检查失败证据、修改控制代码并保留经过验证的改进，从而持续提升其在 RoboDojo 等仿真基准测试中的表现 [2, 6, 9, 15, 20, 24]。然而，这一范式依赖一个强有力的隐含前提：人类工程师已经构建好了仿真场景、重置机制和真实成功判定器。物理任务不具备其中任何一项。近期尝试直接在真实机器人上进行自我改进的工作 [22] 依赖手动或启发式自检以及定制的重置夹具，每项任务都需要数小时乃至数十小时的物理试错，而且无法跨具身形态泛化。因此，要将代码自我改进扩展到真实世界，关键在于一个基础性问题：在没有手工设计的基准测试的情况下，可靠的成功判据和可信的练习世界从何而来？

将真实场景重建到仿真中，在其中进行练习，再迁移回真实世界（Real-to-Sim-to-Real），提供了一条自然路径 [4, 5, 17],但现有流程在三个方面存在不足。首先，重建世界缺乏预训练验证：以往方法从场景重建直接进入策略学习，将世界保真度留待事后通过真实世界的失败才发现 [12]。其次，成功标准依赖人工奖励设计，并容易引发奖励破解：在多轮代码搜索下，智能体很容易利用宽松的判定标准实施非预期行为 [17]。第三，示范仍与源机器人运动学纠缠在一起：模仿记录的轨迹会将任务知识绑定到单一机器人形态上，因此一旦自由度（DoF）或末端执行器发生变化，该示范便失效 [4, 17]。

我们的核心洞见是：**一次成功的真实世界示范，同时提供了世界的蓝图、成功判据，以及用于验证该世界的物理参照——这一切均独立于源机器人运动学**。初始 RGB-D帧捕捉场景几何，而终止状态则规定精确的几何完成关系。关键在于，由于这段示范是一条已知成功的物理轨迹，准确的世界模型应能在对记录的关节命令进行开环接触回放时，重现物体抓取与运动，并使渲染出的物体轮廓与观测掩码对齐。此外，智能体无需模仿示范轨迹，而是将其提炼为具身无关规范（世界 +冻结的判定标准 +局部接触先验），从而能够自主编写模块化控制代码。这既解决了手动编写技能代码的可扩展性瓶颈，也利用代码技能的白盒特性——明确的几何约束、阶段级验证和确定性执行——实现长时程稳定性。

受这一洞见启发，我们提出**RESET** (**R**ebuild, **E**xamine，**S**elf-improve，**E**xecute，**T**ransfer；图1）。该名称体现了这一方法为真实任务恢复的内容：一个可以随时重置并进行练习的世界。Rebuild 从一份 RGB-D 示范中重建可交互的 MuJoCo 场景，并提取只读的终端评分器。Examine 在练习前通过动态回放和轮廓对齐，根据示范检查重建世界，并将无法辨识的质量和摩擦力留给随机化练习。Self-improve 在诊断/留出门控下，根据逐节点失败证据演化以对象为中心的任务图和白盒技能库。Execute 使用冻结代码在线运行，无需大语言模型调用，成本低且支持节点级调试。Transfer 将经验证的世界和判定标准原样交给目标机器人，目标机器人仅调整一层薄硬件层。

我们在 4 种不同的具身形态和 8 项真实操作任务上评估 RESET。仅凭源机器人的一次示范，RESET 在仿真中的未见布局上取得了 97% 的成功率；零样本代码和按任务微调的 $\pi_{0.5}$ 策略成功率均徘徊在 30% 左右，而扩散策略达到 50%。受控缺陷注入表明，练习能够容忍许多重建误差。然而，几何误差可能使任务无法完成，或在练习期间掩盖失败，并改变模拟成功率的含义；Examine 能标记出大多数此类误差，而物理随机化则涵盖质量和摩擦力。无需任何目标示范，双臂 Franka、ARX LIFT2 和 Unitree G1 都能在共享世界中习得各自的技能。最终程序在 4 个实体平台上进行的 280 次真实机器人试验中，成功率达到 80%。

我们的主要贡献有三点：

- **从一次示范到跨具身形态技能的框架**。RESET 将一次真实示范转化为冻结且可部署的代码技能，既适用于接受示范的机器人，也适用于从未接受过示范的机器人。
- **示范取代基准测试**。一次示范就能提供智能体自我改进原本需要通过手工构建基准测试获得的内容：一个练习世界、一项成功判据，以及用于检查该世界的参照。因此，智能体自我改进可以在真实任务上进行，无需手工构建场景、奖励或重置机制。

• **在八项真实任务和四种具身形态上的证据**。在模拟中，RESET 在未见布局上的成功率达到 97%，而零样本代码仅为 27%；三个目标机器人无需新的示范，就能通过在共享世界中练习而非复制源机器人的代码来学会这些任务；练习能够容忍大量重建误差；冻结后的技能可在 4 个实体平台上运行（280 次真实机器人试验的成功率为 80%）。

2. 相关工作

仿真基准中的代码技能自我改进。Codeas Policies [10] 和 Voyager [18] 率先利用大语言模型合成控制程序和技能库。近期的闭环 RSI 框架——包括 RATs [24],同期提出的 RPG [20], EmbodiedEvo [9], PhysEvo [15],具身图灵机 [6], PyRUA-Lean [13], 和 ReSkill [14]——基于 RoboDojo 等基准测试上的执行反馈，迭代改进任务图和技能代码 [2]。然而，这些方法都假设仿真场景、重置机制和真实标签评分器均由人工设计。RESET 摆脱了对基准测试的依赖：它在一个根据单次真实示范重建并经验证的世界中进行自我改进，同时支持在不同机器人形态之间迁移。

实体机器人上的自我改进。 为绕过仿真基准测试，ENPIRE [22]直接在真实机器人上探索并修订策略。尽管这证明了在现实世界中实现 RSI 的可行性，但实体探索需要数小时的机器人磨损、定制重置机制以及针对特定机器人的验证逻辑。RESET 将反复试错限制在经过验证的仿真世界中，让真实机器人仅用于执行最终代码，并使该世界可在不同具身形态间复用。

真实到仿真再到真实与场景验证。RialTo [17], Re3Sim [5], 和 RoboSplat [23] 重建场景，用于训练神经策略或合成数据；X-Sim [4] 从模拟物体运动中推导跨本体稠密奖励；EmbodiRSI [8] 和 D-World [21] 则通过单张图像或在线推演更新数字孪生。然而，以往的流程将重建视为开环预处理步骤，训练前不检查重建世界 [12]。RESET 将示范本身用作动态和视觉验证标准，刻画重建误差对代码策略的影响，并输出可检查的白盒程序。

从单次示范中学习。Demo2Code [19]将示范归纳为代码规范，Ro- boSplat [23] 通过新视角合成扩充单次示范，Human2Sim2Robot [11] 则通过从单段视频进行强化学习，弥合人类与机器人的形态差异。不同于模仿示范轨迹的方法，RESET 仅利用示范来构建并验证环境和终止判据；运行时控制器绝不重放示范，从而实现零示范迁移。

3. 方法

3.1. 动机

RESET 的设计由三个问题塑造。每个问题都从一个观察出发：如果没有相应组件，会发生什么（第4节）。

为什么代码需要练习，又能在哪里练习？ 只编写一次代码是不够的。在具备 RESET 所用的相同接口和接触提示的情况下，智能体的初始程序在未见过的模拟布局上的平均成功率为 27%，不同样本间差异很大（某项任务中的成功率分别为 70%、0% 和 36%），且在最难的四项任务中从未成功（表 1）。程序只有经过运行、失败和修改才能改进，而这需要一个可随意重置的环境和一个无需人工参与的评分。真实任务两者皆不具备，而在真实机器人上练习，每项任务都要花费数小时进行监督试验 [22]。然而，一次成功的示范已同时包含这两者：其第一帧展示场景，最终状态则定义成功，这启发了*Rebuild*。

为什么要用示范来检查重建世界？ 重建不可能尽善尽美，而在错误世界中测得的成功率，对真实世界几乎没有参考价值。物体尺度偏差 10%，或桌子 低了 2 厘米时，在正确世界中表现完美的技能，在若干任务上的成功率会降至 0% （表 2 ）；而判定标准措辞含糊时，把纸箱撞倒的程序也会得到 100% 的评分。示范可以同时检验这两点。它是已知能够成功的实际执行，因此可用的世界必须能够复现它；成功与否则由最终状态定义，而不是由文本描述决定。至于单次示范无法确定的质量和摩擦力，我们不会声称能够验证，而是在练习过程中对其进行随机化。这正是*Examine* 的动机。

为什么使用代码？为什么共享任务，而非动作？ 示范中的动作本身很难迁移：以开环方式重放时，在新布局中的成功率仅为 8%；每项任务使用 100 条轨迹微调的视觉-语言-动作策略成功率达到 31%，而用相同轨迹训练的扩散策略则达到 50%。程序有三个不同之处。它会指出哪一步失败了，让智能体知道该具体修改什么（第 4.4 节）。它可以在真实机器人上运行，且循环中不涉及模型。此外，它将任务要求——世界和判定标准——与机器人如何满足这些要求区分开来，因此，配备不同机械臂或机械手的机器人也能在同一世界中练习（第 4.5 节）。这正是*Self-improve*、*Execute* 和 *Transfer* 的动机。

3.2. 问题形式化与概述

令 $\mathcal{D} = \{(I_t、\ D_t、\ q_t、\ g_t)\}_{t=0}^{T}$ 表示在源机器人 $\mathcal{E}_s$上完成的一次成功真实演示，其中包含 RGB图像 I_t、深度图D_t、关节角度 q_t 和夹爪指令 g_t。场景具有可观测几何形状 θ_{obs} （物体尺度、桌子高度、相机位姿），其 $\mathcal{D}$ 构成约束，而潜在物理属性 θ_{lat} （质量、

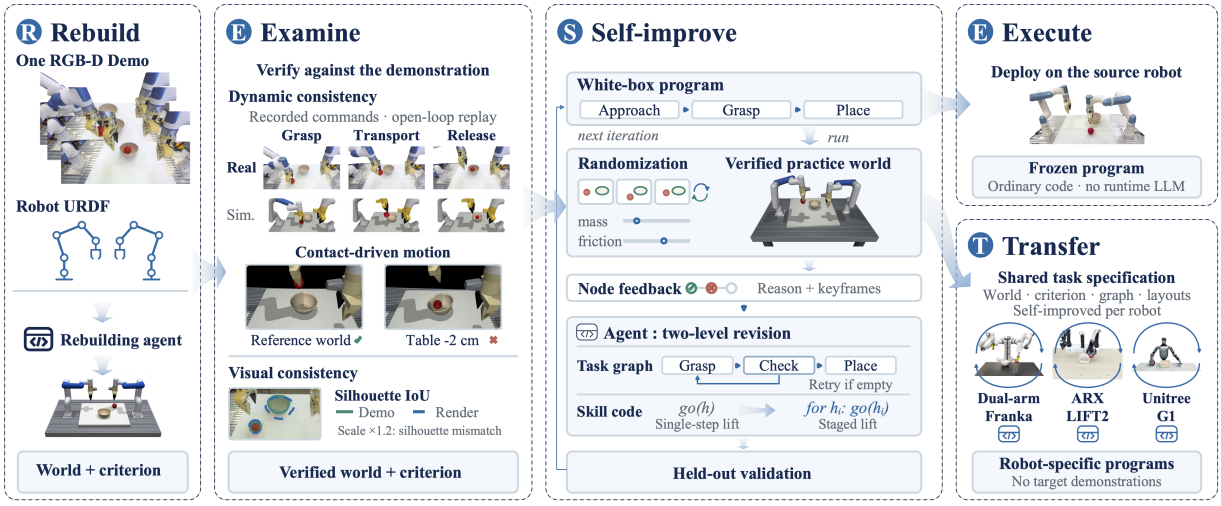


图 2。 **RESET的五个步骤，以 苹果 任务为例进行说明。** 重建将一次 RGB-D 示范和机器人的 URDF 转化为交互式世界和成功判据。Examine 通过开环重放和渲染轮廓，将该世界与示范进行对照检查。自我改进根据节点级失败修订白盒程序，并且只有当修订版本在留出布局上的表现更好时才予以保留。执行在没有大语言模型调用的情况下，在真实机器人上运行最终程序。迁移让其他机器人拥有相同的世界和判定标准，以便在其中进行练习，而无需各自提供示范。

摩擦力），则无法通过一次成功执行加以识别。对于没有演示的目标机器人 $\{\mathcal{E}_k\}$，我们合成控制程序 P_s 和 $\{P_k\}$，使其无需手写奖励，即可在未见物体布局下完成任务。RESET 分五个阶段实现这一点（图2）：重建、检查、自我改进、执行和迁移。

3.3. 重建：世界与只读判据

接触丰富的世界重建。 我们通过将机器人的运动学与示范过程中各帧图像进行对齐来恢复相机位姿，并使用预训练的可提示分割模型，对第一帧中的任务物体（I_0，D_0）进行分割。多模态编码代理以迭代方式将每个物体建模为网格，同时保留堆叠、插入和抽屉拉动所依赖的腔体与关节部件。随后将这些网格配准并缩放到第一帧的深度点云，拟合桌面平面，并将场景连同机器人的 URDF 一起导出到物理模拟器中。θ_{lat}接收标称值。

终止判据与只读评分器。成功率判定标准从演示的最终状态中读取，由一组几何关系（包含、支撑、插入、悬挂、直立方向）以及释放后的静止条件共同构成。世界重建完成并通过 Examine 后，由人员确认一次该判定标准。此后，评分器将设为只读，并对控制器隐藏，因此练习过程无法利用评分器谋取优势。

3.4. Examine：双通道世界验证

传感器噪声、相机误差和尺度误差会使每个重建世界都偏离真实场景。由于 $\mathcal{D}$ 已知能够成功，Examine 通过两个互补通道检查重建世界是否能够复现它。动态通道将物体放置在第一帧位姿上，并以开环方式重放记录的关节和夹爪指令；物体从不与夹爪相连，因此抓取和放置必须源于接触。该通道会测量演示中的抓取有多少得以复现，以及物体偏离其跟踪运动的程度，从而约束桌子高度和接触支撑。视觉通道会在接触前按照记录的关节角度渲染世界，并测量每个任务物体相对于演示掩码的轮廓交并比，从而约束相机位姿和物体尺度。这些测量用于比较相同场景的重建结果，并舍弃那些明显较差地复现演示的结果；边界会在留出评估前，基于开发任务固定下来。Examine 是对世界可用性的检查，而不是对保真度的保证：质量和摩擦力几乎不会改变两个通道的结果，其影响通过随机化练习来覆盖。

3.5. 自我改进：双层代码演化

经验证的世界、随机无碰撞布局和只读评分器共同构成了一个会自动重置的特定任务基准。在其中，控制器是一个白盒程序$P = (G,\ \Pi,\ \Lambda)$：一个以对象为中心的任务图 G，其节点调用带有入口与出口检查及重试预算的技能；一个技能库 Π （位姿估计、逆运动学，以及抓取、放置、扫地和交接等基本操作）；以及连接它们的绑定关系Λ。 P 只能读取实体机器人所提供的信息：相机数据流、标定数据、关节编码器，以及在演示过程中观测到的夹爪与物体的接触偏移

表 1. **源具身化 xTrainer 在每项任务的 50 种未见布局（测试集0–49）上的主要盲测成功率（%），在仿真中。** 零样本代码报告 3 个独立样本的平均值和最佳值。 $\pi_{0.5}$ 和扩散策略各自针对每项任务训练一个专用模型，使用相同的100 条成功的 RESET 轨迹($\pi_{0.5}$ 经过微调；扩散策略从头开始训练）。

方法	苹果	方块堆叠	叠杯子	杯子搭塔	马克杯-盒子	打开抽屉	扫地	抬起盒子	Mean
开环演示回放	4%	0%	12%	0%	38%	0%	0%	10%	8%
零样本代码（三次运行平均值）	85%	35%	55%	39%	0%	0%	0%	0%	27%
零样本代码（三次运行最佳结果）	**100%**	70%	**96%**	76%	0%	0%	0%	0%	43%
$\pi_{0.5}$ 微调（按任务） [7]	4%	0%	76%	6%	72%	4%	32%	52%	31%
扩散策略（按任务）	20%	0%	94%	6%	94%	26%	70%	88%	50%
RESET（我们的方法）	**100%**	**98%**	**96%**	**100%**	**100%**	**100%**	**88%**	**94%**	**97%**

量。它从不读取模拟器状态或评分器，因此同一个程序无需修改即可在实体机器人上运行。

在每一轮中，编码代理会在 P 诊断集上运行，读取每个节点的状态和失败原因，以及相应的关键帧，定位最早出错的节点，并在两个层级上修订程序。任务级修订会改变图（例如，在一次空抓取后重试，或在机械臂之间进行交接）；技能级修订会改变技能内部的几何形状（例如抓取位置）。为避免针对特定失败进行拟合，只有当修订后的程序在另一组留出布局上取得更高的成功率时，才会替换当前程序。当练习收敛或轮次预算用尽时， P 会被固定下来，并在封闭布局上评估一次。

3.6. 执行与迁移

最终程序以普通代码的形式运行在实体机器人上，无需大语言模型调用，也无需人工干预。在每个节点之前，程序会根据分割和深度估计物体位姿，然后执行技能。所有推理都在线下完成；由于程序在练习后被冻结，真实机器人上运行的代码与仿真中评估的代码完全一致，因此任何失败都可以追溯到具体节点。

目标机器人 $\mathcal{E}_k$ 会原样接收所有与机器人无关的内容：经验证的世界、判定标准、布局分布，以及源机器人的任务图和其中关于物理陷阱的备注。它会为自己的末端执行器提供 URDF、相机标定和接触偏移量，并在共享世界中运行自我改进，以获得 P_k。源机器人的任何轨迹或技能代码都不是必需的：目标机器人会在源函数适用于自身机械臂或机械手时加以复用，否则便自行编写（第4.5节）。因此，示范只被使用一次，用于构建并检查世界，而不会被当作需要模仿的动作。

4. 实验

我们评估五个核心问题——Q1：布局泛化能力与基线方法的比较；Q2：对重建误差的容忍度；Q3：自我改进组件消融；Q4：零示范跨本体迁移；Q5：物理部署与回合级仿真到现实一致性。

4.1. 实验设置

具身形态与任务。我们评估了 4 个运动学、相机和末端执行器各不相同的平台（图1c）。双臂 6 自由度 xTrainer（平行夹爪）是唯一的源具身化，并为每项任务提供一次示范。其余三个目标具身形态不接收示范：ARX LIFT2（安装在升降台上的 6 自由度机械臂）、Unitree G1（五指手；仿真中固定下半身）和 7 自由度双臂 Franka。我们测试了 8 项桌面任务（图3）：苹果、立方体、叠杯子、搭建塔、马克杯-盒子（双手交接）、打开抽屉（关节式拉动 +放置）、扫地（工具扫动）以及抬起盒子（保持直立方向）。

实现。 重建使用 GPT-6 Astra（在 Blender 中创建物体资产）、SAM3 [1]（分割）和 MuJoCo [16]；自我改进使用配备 Claude Opus 5.5 的 Claude Code。每个任务由一名人员确认一次成功判据；这仅对抬起盒子任务改变了判定标准。

协议与沙箱隔离。 Q1–Q4 在仿真中的重建世界内进行评估。 各布局使用互不相交的种子：诊断集（DEV 0–19）、留出采用集（DEV 20–39）以及封存的盲测集（测试集 0–49，在开发结束后评估一次，使用最终的只读评分器）。每个布局都会将每个任务物体沿桌子的各个轴移动最多 ±3–6 厘米，并使其绕垂直轴旋转最多 ±10°–180°，围绕其示范姿态；存在碰撞或物体超出可触及范围的布局将被舍弃。 消融实验和重复运行使用相互隔离的智能体，它们只能看到初始代码、任务文本以及自身的日志。

4.2. 源具身化上的主要结果（Q1）

基线方法与定量比较。表1比较了 RESET 与开环演示回放、零样本代码（每项任务使用相同接口和接触先验进行 3 次独立采样），以及按任务 $\pi_{0.5}$ 策略 [7]（在 100 条来自不相交布局的 RESET 轨迹上微调 6,000 步）。开环重放的平均成功率为 8%：这些布局无法通过记忆轨迹来解决。零样本代码的平均成功率为 26.8%，且方差极大（立方体：70/0/36%；堆叠杯子：96/18/50%），并且在马克杯-盒子、打开抽屉、扫地和抬起盒子这几项任务的全部 3 次采样中，评分均为 0 %。按任务 $\pi_{0.5}$ 平均值

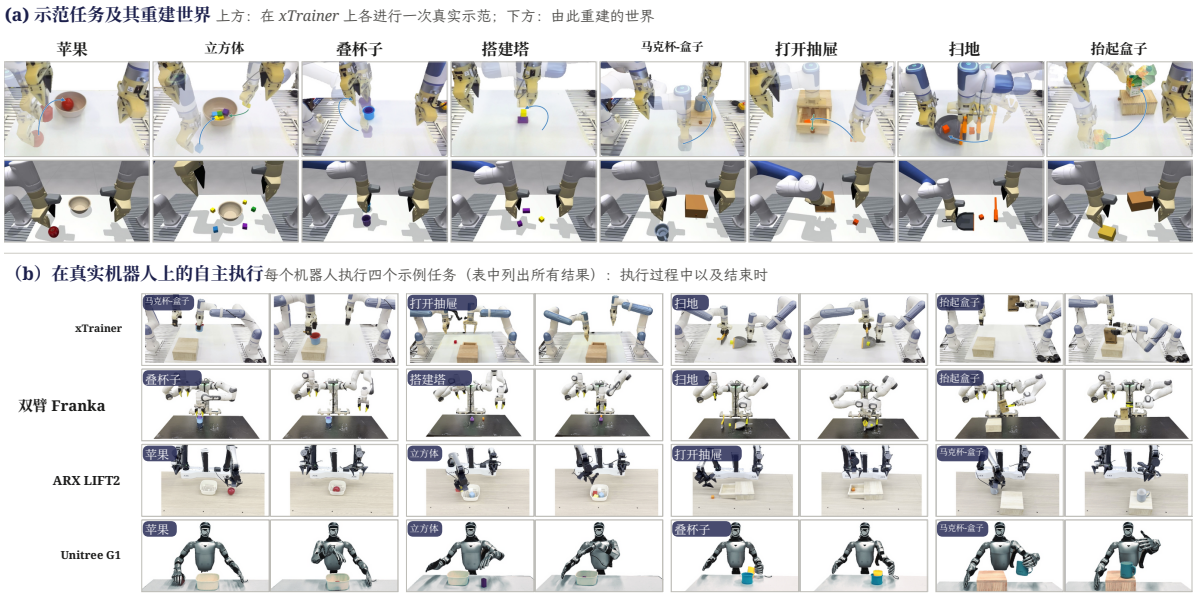


图3. **示范任务、对应的重建世界以及自主真实机器人执行。** (a) 上：八项任务，每项均由源机器人 xTrainer 示范一次。下：根据每项示范重建的世界，由演示摄像机视角渲染。(b) 最终程序在四个机器人上运行，期间无需人工干预。我们展示了每个机器人执行的四项任务（所有结果见表 5），分别展示执行过程和结束状态。Franka、LIFT2 和 G1 均未收到示范。

为 30.8%，而扩散策略 [3] 在相同轨迹上从头训练，成功率为 49.8%：它在叠杯子、马克杯-盒子和抬起盒子上接近 RESET，但在苹果、立方体、搭建塔和打开抽屉上的成功率均不超过 26%，这些任务需要精确抓取或放置。两种策略均使用默认设置；对于一个规模如此大的模型而言，100 条轨迹不足以进行微调 $\pi_{0.5}$，这或许可以解释其较低的评分。RESET 的总体成功率达到 97.0%，而在零样本代码得分为 0% 的四项任务上，成功率为 88%–100%。

可重复性与判定标准精度。 从零样本起始状态独立重新运行，在搭建塔任务上达到100%，在打开抽屉任务上达到98%，在扫地任务上达到86%。在抬起盒子任务中，宽泛的文本翻译判定标准（“纸箱在盒子上”）使零样本程序在将纸箱拍平时获得100%的评分；而从示范中得出的直立判定标准则使其得分为0%。

4.3. RESET 能容忍多大的重建误差？（Q2）

重建不可能完美，因此我们每次向重建世界中注入一个缺陷：几何误差（物体尺度 $\times 0.8$–$\times 1.2$、低了 2 厘米的桌子、发生偏移的摄像头）以及潜在物理误差（摩擦力 $\times 0.3$、质量 $\times 3$）。我们要考察的是：在这样的世界中进行练习，是否仍能获得可在正确世界中运行的技能；在那里测得的成功率是否可信；以及 Examine 是否会标记该世界。

发现 1：练习可以容忍大多数重建误差；错误世界会在掩盖失败时造成损害。 我们在有缺陷的世界中运行完整的自我改进流程，并在正确世界中测试结果（表。 3）。在大多数情况下，结果是

表2。 **错误世界：它们破坏技能的频率，以及 Examine 标记它们的频率。** 在正确世界中学会的技能会在带有一个注入缺陷的世界中运行；条目统计六项任务中的任务–缺陷案例。破坏：成功率至多为 50%。最后一列：被 Examine 标记的破坏案例。

注入缺陷	技能破坏	Examine 标记	破坏，已标记
无（正确世界）	0/6	0/6	–
缩放 $\times 0.8, \times 0.9$	1/12	11/12	1/1
缩放 $\times 1.1$–$\times 1.2$	12/18	11/18	9/12
桌子 -2厘米	3/6	6/6	3/3
相机移动	0/6	6/6	–
摩擦力 $\times 0.3$	0/6	0/6	–
质量 $\times 3$	2/6	0/6	0/2

与在正确世界中练习后的结果相同：马克杯-盒子在三个有缺陷的世界中均达到100%，搭建塔达到98%–100%，苹果 在桌子低了2厘米时达到94%，而叠杯子 在六次运行中的四次里、面对放大的物体时达到82%–100%。智能体所做的修订会根据其感知到的内容重新锚定技能（例如，根据感知到的边缘确定抓取高度，而不是使用演示中的绝对高度），而且这些修订能够迁移到其他情境。两个案例失败了。几何误差可能使任务在练习世界中无法解决，因此程序停留在起始状态（放大物体时的苹果 ）。更微妙的是，几何误差可能掩盖失败：当桌子低了2厘米时，零样本叠杯子 程序碰巧在正确的高度释放，并在98%的练习回合中成功，因此练习找不到任何需要修复的问题。在三个相互独立的运行中，所得程序在

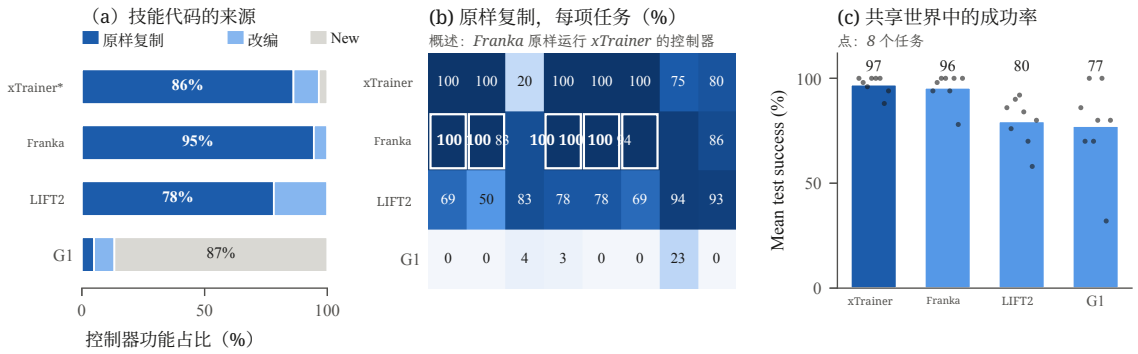


图 4。 **跨本体技能复用（AST 分析）与任务表现。** (a) 从另一种本体未经改动复用、经过改编或从头编写的技能所占比例。(b) 按任务统计的未经改动技能复用情况，突出显示双臂 Franka 使用 xTrainer 的控制器直接通过的 5 个任务。(c) 各本体间的盲测成功率分布。 *对于源机器人，其函数中也以未经改动的形式出现在另一台机器人代码中的函数所占比例。

表 3。 **先在有缺陷的世界中练习，再在正确世界中测试。** 在出现给定缺陷的世界中运行自我改进后，在正确世界中的成功率。起始状态：零样本程序。多个数值：独立运行结果。叠杯子 在 ×1.1 和 ×1.2：判定标准随世界缩放。

		练习世界			
Task	起始状态	正确 缩放	×1.1 缩放	×1.2 桌子	−2厘米
苹果	78%	100%	78%	78%	94%
搭建塔 42%		100%	100%	98%	100%
马克杯-盒子	0%	100%	100%	100%	100%
叠杯子	18%	96%	18/98/98%	8/82/100%	18/20/18%

正确世界中仅达到18%–20%，而在那里练习后可达到96%。

发现2：在错误世界中测得的成功率不可信。反方向则不具备容错性。在正确世界中表现完美的技能，在物体放大10%–20%的18个世界中有12个会失效；在桌子降低2厘米的6个世界中有3个会失效，而相机偏移和摩擦变化则可由在线定位吸收。因此，即使几何误差不会阻止学习，也会改变模拟成功率的含义；这正是我们要根据示范检查世界，并验证其是否能够预测真实机器人结果的原因（第4.6节）。

发现3：Examine 标记大多数几何误差；随机化覆盖物理学。Examine 比较相同场景的不同版本，并标记那些演示效果明显差于最佳版本的版本。它没有标记任何正确世界，并标记了导致技能失效的16个几何缺陷中的13个，包括每个降低的桌子（表 2），以及表 3 中练习始终失败的每个世界。它漏掉了三个扩大后的世界，因此它是一项检查，而非保证。Examine 无法检测质量和摩擦力，而通过在4个质量和摩擦力随机化的世界中进行练习可以覆盖这些因素：在马克杯-盒子上，这使未见质量条件下的成功率 ×3 从10%提升至100%（表 4）；而 3× 繁重的打开抽屉 仍未解决。

表4。**在随机化物理学环境中进行练习。**在正确世界以及未见过的物理缺陷条件下的成功率（%），仅针对物理缺陷会破坏的两个任务（表2）。

Task	练习	正确质量	×3 摩擦力.	×0.3 摩擦力.	×3
马克杯-盒子	仅名义值	100%	10%	100%	100%
	随机化	100%	**100%**	100%	100%
打开抽屉	仅名义值	98%	0%	100%	95–100%
	随机化	98%	0%	100%	100%

4.4. 自我改进消融实验（Q3）

在马克杯-盒子 和打开抽屉上，从得分为0%的零样本程序开始（表 6），移除节点级失败日志和关键帧（仅结果反馈）后，经过8轮数值仍为12%和0%；而完全重置在3–5轮内收敛至100%和98%：节点级失败证据至关重要。移除留出门控——即只有当修订版本在留出布局上的表现更好时才采用该修订的规则——仍能解决 马克杯-盒子（100%），但在打开抽屉 经过全部8轮后仍为0%。

4.5. 跨具身迁移（Q4）

跨形态的迁移性能。我们将 xTrainer 的 8 个已验证世界及其标准交给三个目标机器人（见表 5）。双臂 Franka 和 ARX LIFT2 分别达到 95.5% 和 79.5%；Unitree G1 使用五指手而非平行夹爪，达到 77.3%（在 扫地任务中为 32%，因为其手指在扁平刷柄上缺乏横向刚度）。

这是共享世界，还是仅仅复制了代码？为区分二者，我们将源机器人的最终程序原样运行在每个目标机器人上，不进行练习。在机械臂与 xTrainer 接近的双臂 Franka 上，五个程序无需修改即可运行（94%–100%）；另外三个程序的成功率分别为 10%、24% 和 0%（叠杯子、扫地、抬起盒子），在共享世界中练习后分别提升至 100%、78% 和 100%。在 ARX LIFT2 上

表 5. **不同具身形态下的仿真和真实机器人结果**。 仿真：每项任务在 50 个未见测试布局（测试集0–49）上的成功率（%）。真实：在真实机器人上运行同一冻结程序的成功次数/试验次数，未进行代码修改或人工干预（–：未评估）。三个目标机器人均未收到示范。

机器人		苹果	立方体	叠杯子	搭建塔	马克杯-盒子	打开抽屉	扫地	抬起盒子	Mean
xTrainer（源）	Sim.	100%	98%	96%	100%	100%	100%	88%	94%	97%
	Real	10/10	10/10	9/10	8/10	10/10	10/10	6/10	10/10	73/80
双臂 Franka	Sim.	94%	98%	100%	100%	94%	100%	78%	100%	96%
	Real	10/10	10/10	10/10	10/10	10/10	8/10	8/10	10/10	76/80
ARX LIFT2	Sim.	86%	76%	90%	92%	84%	70%	58%	80%	80%
	Real	7/10	8/10	9/10	7/10	9/10	9/10	6/10	6/10	61/80
Unitree G1	Sim.	86%	70%	100%	70%	80%	100%	32%	80%	77%
	Real	5/10	2/10	3/10	–	3/10	–	–	–	13/40

表6。 **自我改进的消融实验**，每项实验均从相同的零样本程序开始，该程序的评分为 0。成功率：测试成功率（%）。轮数：使用的自我改进轮数，预算上限为 8 轮。

	马克杯-盒子		打开抽屉	
变体	成功率	轮数	成功率	轮数
完全重置	**100%**	3	**98%**	5
不使用留出门控	**100%**	3	0%	8
仅结果反馈	12%	8	0%	8

源程序在任何任务上都未成功；在 Unitree G1 上仅在苹果任务上成功（18%）。练习后，这两个机器人平均达到 80% 和 77%。源机器人的任务图和备注同样不是必需的：仅给定共享世界和判定标准，并从头开始，LIFT2 在马克杯-盒子和打开抽屉上的成功率分别达到 94% 和 84%，而 G1 分别达到 74% 和 92%；这些结果与继承任务图和备注时获得的 84%、70%、80% 和 100% 相当。

技能层级复用分析。比较最终技能库中的函数级语法树（图4），Dual-机械臂 Franka 的技能函数中有 94.5% 与另一台机器人相同，LIFT2 的复用率为 78.2%；而 Unitree G1 有 86.7% 的技能是从头编写的，但平均成绩仍达到 77%。跨具身形态迁移的是经验证的世界和判定标准，而非代码。

4.6. 真实世界部署与仿真到现实一致性（Q5）

我们在四台实体机器人上部署冻结程序，不进行任何代码修改，也不需要人工干预（每项任务试验 10 次，物体由人工放置在不同位置）。源 xTrainer 以及零示范的双臂 Franka 和 ARX LIFT2 均运行全部 8 项任务，而 Unitree G1 运行其中 4 项（表5，图 3）。总计 280 次真实机器人试验中有 223 次成功（80%）；在三台固定基座机器人上，240 次中有 210 次成功（88%）。xTrainer 在 80 次试验中成功 73 次（91%），双臂 Franka 在 80 次中成功 76 次（95%），接近其仿真成功率（97% 和 96%）；两者的机械臂均具有较大的工作空间，并能定位场景中的物体。ARX LIFT2 达到 80 次中 61 次成功（76%，仿真中为 80%）：其机械臂的可达工作空间较小，因此某些布局会将物体放置在接近其可达范围边界的位置。Unitree G1 在其四项任务中 40 次试验成功 13 次（33%，而在相同任务的仿真中为 84%）。其全身控制器会在机械臂移动时带动基座移动，而开环程序不会补偿这种位移（第5节）。

经验证的世界能否预测真实结果？ 对于每次真实试验，我们都会注册第一帧相机帧，以还原实际物体布局，在经验证的世界中重现该布局，并在那里运行相同的冻结程序。在 xTrainer 上的 80 次试验中，仿真与真实机器人有 95% 的结果相同；在 ARX LIFT2 上则有 90% 的结果相同。ARX LIFT2 是一款未参与构建或检验该世界的机器人。**成本**。自我-

改进在 2–8 轮内收敛，每项任务和每台机器人使用 100–320 个仿真回合，以及约 3000 万个大语言模型词元。部署时不使用大语言模型词元。

5. 局限性与结论

局限性。RESET 目前面向刚性物体、可关节运动物体以及准静态扫掠任务，不适用于可变形布料或动态投掷。需要彻底改变接触拓扑的极端潜在参数跳变（例如 3× 重型抽屉）仍超出参数随机化的能力范围。最后，当前的练习尚未对人形机器人的全身控制器所造成的影响进行建模：在 Unitree G1 上，控制器会在机械臂移动时带动基座偏移，因此真实机器人的成功率远低于固定基座机械臂（表5）；将 RESET 适配至真实人形机器人留待未来工作。

结论。 我们提出了 RESET，将由智能体驱动的代码自我改进从手工构建的基准测试推进到真实任务。一段真实示范即可提供练习世界、成功判据以及用于检查该世界的参照，因此，任务的要求与特定机器人执行任务的方式得以分离。重建世界不必完美：练习可以容忍大多数重建误差，而

将世界与示范进行比对，可以标记出会造成误导的误差。更广泛地说，RESET 提供了一条一次性、跨本体获取机器人技能的途径，其基础是智能体对世界和代码的通用知识：用户只需示范一次任务，不同机器人便能获得可检查、可直接运行的技能。

参考文献

[1] Nicolas Carion、Laura Gustafson、Yuan-Ting Hu、Shoubhik Debnath、Ronghang Hu、Didac Suris Coll-Vinent、Chaitanya Ryali、Kalyan Vasudev Alwala、Haitham Khedr、Andrew Huang 等。Sam 3：基于概念的万物分割。载于《国际学习表征会议》，第 138846–138923 页，2026 年。5[2] Tianxing Chen、Yue Chen、Zixuan Li、Junyuan Tang、Kailun Su、Haoran Lu、Weijie Wan、Baijun Chen、Songling Liu、Haowen Yan 等。RoboDojo：用于全面评估通用机器人操作策略的统一仿真与真实环境基准。arXiv 预印本 arXiv:2607.04434，2026 年。2, 3[3] Cheng Chi、Zhenjia Xu、Siyuan Feng、Eric Cousineau、Yilun Du、Benjamin Burchfiel、Russ Tedrake 和 Shuran 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》，44 (10-11):1684–1704，2025 年。6[4] Prithwish Dan、Kushal Kedia、Angela Chao、Edward Weiyi Duan、Maximus Adrian Pace、Wei-Chiu Ma 和 Sanjiban Choudhury。X-Sim：通过真实到仿真再到真实实现跨本体学习。arXiv 预印本 arXiv:2505.07096，2025 年。2, 3[5] Xiaoshen Han、Junqiu Yu、Minghuan Liu、Yilun Chen、Xiaoyang Lyu、Yang Tian、Bolun Wang、Weinan Zhang 和 Jiangmiao Pang。Re 3 sim：通过三维照片级真实到仿真生成高保真仿真数据，用于机器人操作。载于 2026 IEEE国际机器人与自动化会议（ICRA），第 20781–20788 页。IEEE，2026 年。2, 3[6] Kairui Hu、Siyuan Hu、Fangzhou Hong、Zhaoxi Chen 和 Ziwei Liu。具身图灵机：用于机器人递归自我改进的有状态代码。arXiv 预印本 arXiv:2610.12369，2026 年。2, 3[7] Physical Intelligence、Kevin Black、Noah Brown、James Darpinian、Karan Dhabalia、Danny Driess、Adnan Esmail、Michael Equi、Chelsea Finn、Niccolo Fusai 等。 $\pi_{0.5}$：具备开放世界泛化能力的视觉-语言-动作模型。arXiv 预印本 arXiv:2504.16054，2025 年。5[8] Haoran Lang、Haotao Lu、Shiyu Sang、Haoyang Luo、Guo Chen、Qun Li、Jingyi Yu、Ye Shi 和 Jingya Wang。Embodirsi：用于数据高效的机器人适应的递归自我改进。arXiv 预印本 arXiv:2609.38905，2026 年。3[9] Wenyi Li、Yizhe Chi、Youjie Zheng、Chunyu Liu、Xiaoqiu Wang、Mingju Gao、Zheng Jiang、Deyao Hong、Calvin Xiao、Kaisen Yang 等。EmbodiedEvo：面向物理智能的自我演化编码代理。arXiv 预印本 arXiv:2610 18867，2026 年。2, 3。[10] Jacky Liang、Wenlong Huang、Fei Xia、Peng Xu、Karol Hausman、Brian Ichter、Pete Florence 和 Andy Zeng。代码即策略：用于具身控制的语言模型程序。载于 2023 年 IEEE国际机器人与自动化会议（ICRA），第 9493–9500 页。IEEE，2023 年。2, 3[11] Tyler Ga Wei Lum、Olivia Y Lee、C Karen Liu 和 Jeannette Bohg。利用单个人类示范，通过仿真到现实强化学习弥合人机具身鸿沟。arXiv 预印本 arXiv:2504.12609，2025 年。3[12] Hiroki Nishizawa、Takehiko Ohkawa、Nicola Scianca、Tatsuya Matsushima、Shigeo Morishima 和 Kei Ota。Yubi-real2sim：用于可靠策略评估的 3DGS 场景验证。arXiv 预印本 arXiv:2610.17521，2026 年。2, 3[13] Ruiyang Si、Jianxin Bi、Shunyu Yang、Rui Ni、Wenbo Huang、Qiang Wang、Shulong Jiang、Duomin Wang、Xiuyu Li、Haiwen Feng 等。更少令牌，更佳动作：GPT-6 Astra 机器人智能体在令牌减少 65% 的同时，成功率提高 14%。arXiv 预印本 arXiv:2610.01939，2026 年。3[14] Zheng Sun、Zhihao Li、Zhuo Li、Wenrui Liu、Xin Wang 和 Fei Chen。ReSkill：面向机器人操作的选择性自我改进。arXiv 预印本 arXiv:2611.10510，2026 年。3[15] Wenqing Tian、Zeyu Zhang、Zhaocheng Liu、Fengwei Liu、Qiang Liu 和 Liang Wang。PhysEvo：Astra 能行动，就让它行动。arXiv 预印本 arXiv:2610.08995，2026 年。2, 3[16] Emanuel Todorov、Tom Erez 和 Yuval Tassa。MuJoCo：用于基于模型的控制的物理引擎。载于 2012 年 IEEE/RSJ国际智能机器人与系统会议，第 5026–5033 页。IEEE，2012 年。5[17] Marcel Torne、Anthony Simeonov、Zechu Li、April Chan、Tao Chen、Abhishek Gupta 和 Pulkit Agrawal。通过仿真调和现实：一种用于鲁棒操作的真实到仿真再到真实方法。arXiv 预印本 arXiv:2403.03949，2024 年。2, 3[18] Guanzhi Wang、Yuqi Xie、Yunfan Jiang、Ajay Mandlekar、Chaowei Xiao、Yuke Zhu、Linxi Fan 和 Anima Anandkumar。Voyager：一种采用大语言模型的开放式具身智能体。arXiv 预印本 arXiv:2305.16291，2023 年。3[19] Yuki Wang、Gonzalo Gonzalez-Pumariega、Yash Sharma 和 Sanjiban Choudhury。Demo2Code：从概述示范到通过扩展思维链合成代码。神经信息处理系统进展，36:14848–14956，2023 年。3[20] Yen-Jen Wang、Haozhe Jiang、Shuying Deng、Haoru Xue、Weirui Ye、Rocky Duan、Nika Haghtalab、S Shankar Sastry、Pieter Abbeel 和 Haozhi Qi。重建、练习、实战：具身智能体的引导式自我改进。arXiv 预印本 arXiv: *2610 02204*, 2026 2, 3 。。Xuan Wen 和 Baichao Wang。[21] D-World：一种在双重控制下进行递归自我改进的显式世界-语言-动作模型。arXiv 预印本 arXiv:2610.17114，2026 年。3[22] Wenli Xiao、Jia Xie、Tonghe Zhang、Haotian Lin、Letian Fu、Haoru Xue、Jalen Lu、Yi Yang、Cunxi Dai、Zi Wang 等。ENPIRE：真实世界中的智能体机器人策略自我改进。arXiv 预印本 arXiv:2606.19980，2026 年。2, 3[23] Sizhe Yang、Wenye Yu、Jia Zeng、Jun Lv、Kerui Ren、Cewu Lu、Dahua Lin 和 Jiangmiao Pang。利用高斯泼溅生成新颖示范，实现鲁棒的单次操作。arXiv 预印本 arXiv:2504.13175，2025 年。3[24] Junyi Zhang、Jiaxin Ge、Hanjun Yoo、Letian Fu、Zihan Yang、Yaowei Liu、Raj Saravanan、Shaofeng Yin、Justin Yu、Dantong

Niu 等。趣味智能体机器人学习。*arXiv* 预印本 *arXiv:2606.19419*，2026 年。2，3

附录

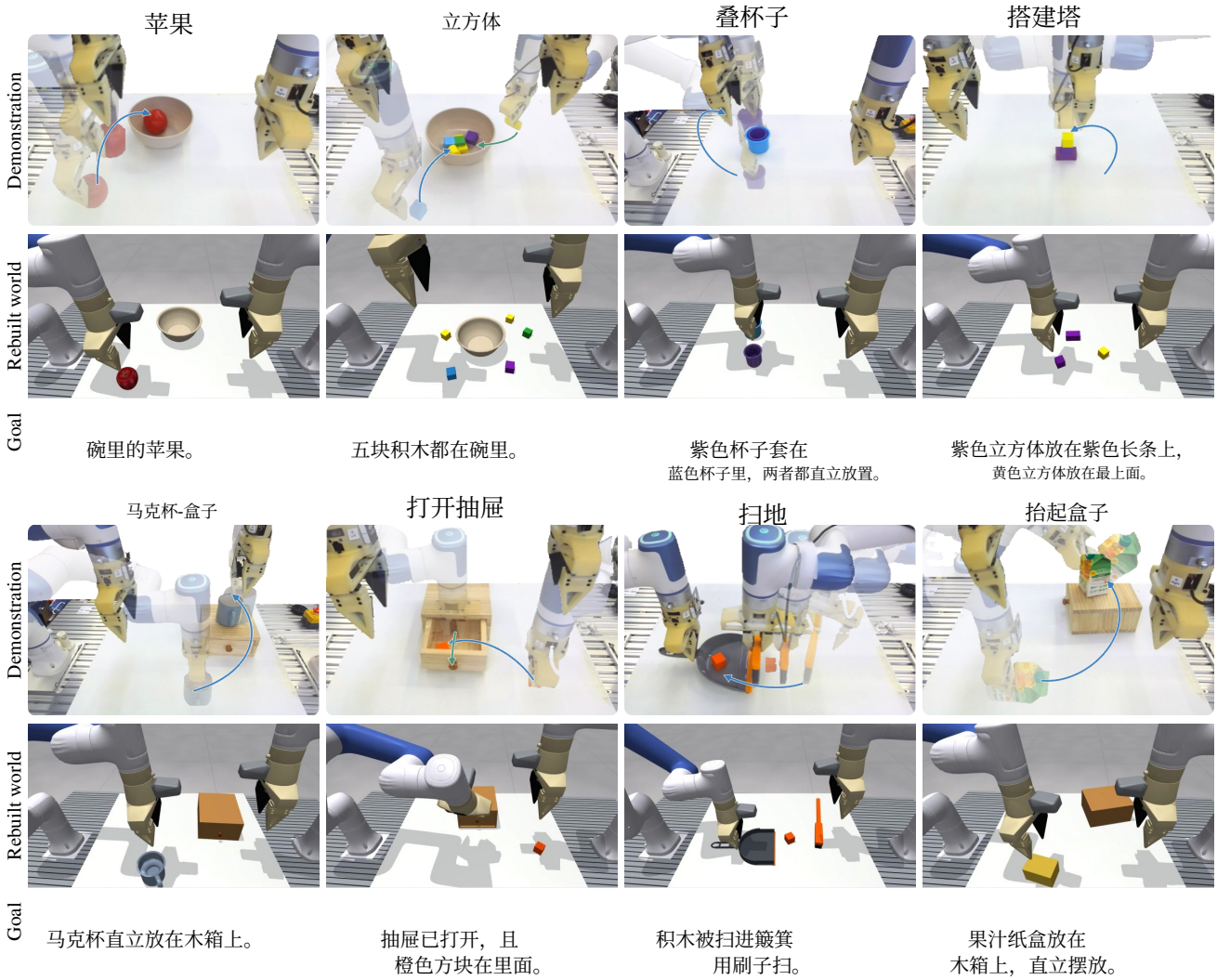


图 S1。**八项任务及其重建世界**。示范：源机器人 xTrainer 上的单次示范，示范中物体的运动以箭头表示。重建世界：根据该示范重建的交互式世界，以演示摄像机视角渲染（图 3(a) 的放大视图）。目标：用通俗语言说明成功判据的要求。

A. 任务与重建世界

八项任务中的每一项都在源机器人 xTrainer 上示范一次（第 4.1 节）。图 S1 展示了每次示范，以及根据该示范重建的世界（第 3.3 节），还展示了成功判据从演示的最终状态中读取的目标。每项判定标准还要求物体在回合结束时已被释放并保持静止。

B. Examine 示例

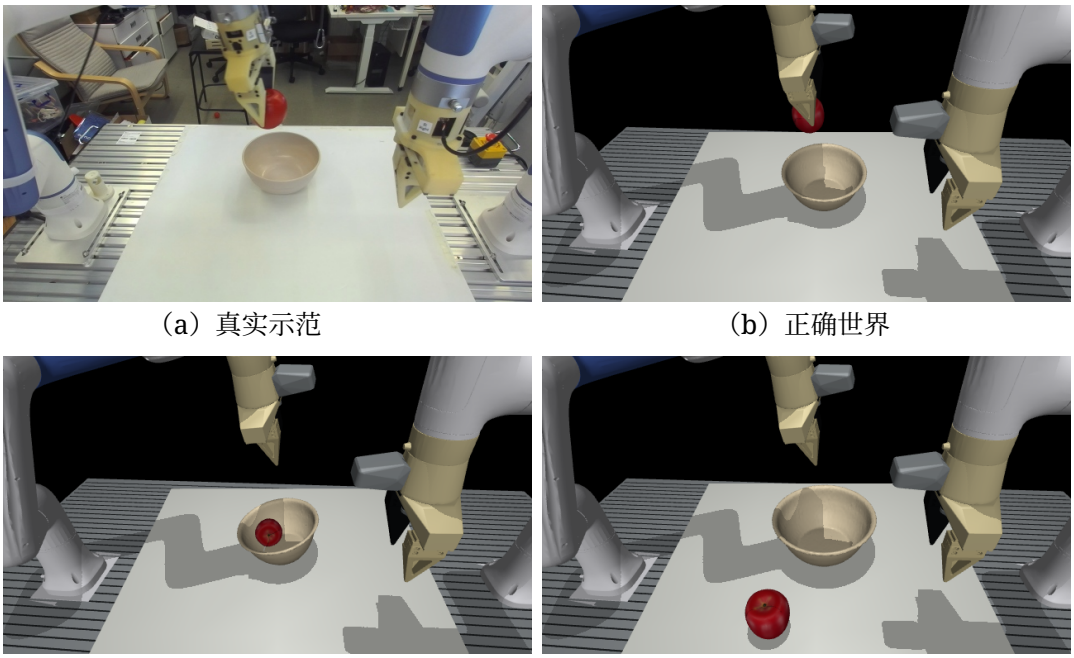
Examine（第 3.4 节）通过两个渠道检查重建世界与演示是否一致。图 S2 和 S3 展示了苹果任务中的两个渠道，分别对应正确世界，以及在第 4.3 节中注入一种几何缺陷的世界：桌子低了 2 厘米，且所有物体均按 $\times 1.2$ 进行缩放。

C. 一轮自我改进

每一轮中，自我改进（第 3.5 节）都会在诊断布局上运行程序，读取每个节点的状态和失败原因，并修订代码。图 S4展示了源机器人 xTrainer 上苹果任务的一次此类修订过程：从第 0 轮程序的一次失败练习回合，到第 1 轮修订后的程序。

D. 四个机器人的仿真执行

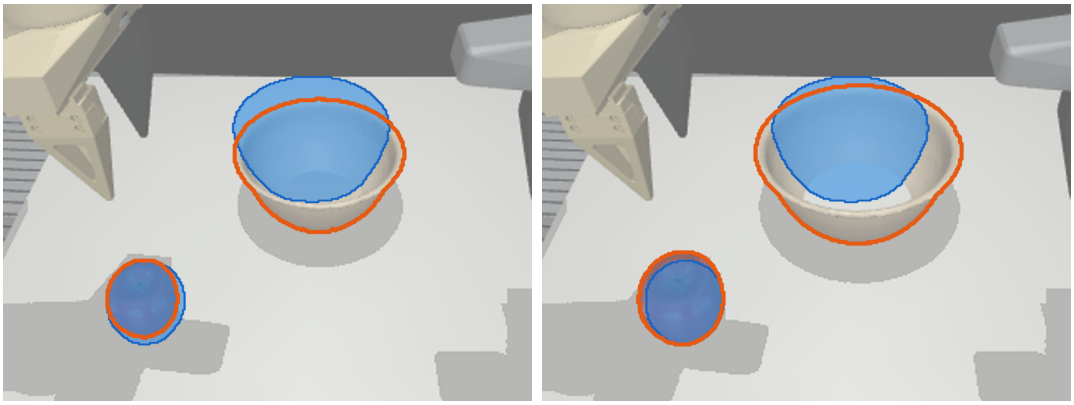
在迁移（第 3.6 节）中，每个目标机器人都在相同的重建世界中按照相同的标准进行练习，无需提供其自身的演示—



（a）真实示范　（b）正确世界

（c）桌子低了 2 厘米　（d）物体缩放了 $\times 1.2$

图 S2。**Examine 在苹果任务中的动态通道**。四张图像都展示了示范中的同一时刻，即苹果被提起后不久。（b）–（d）在三个重建世界中以开环方式重放记录的关节和夹爪指令；苹果从未附着在夹爪上，因此只能通过接触移动。（b）苹果被夹爪夹住，与示范中相同。（c）夹爪是空的，苹果 леж在碗中。（d）夹爪是空的，苹果仍在桌子上。



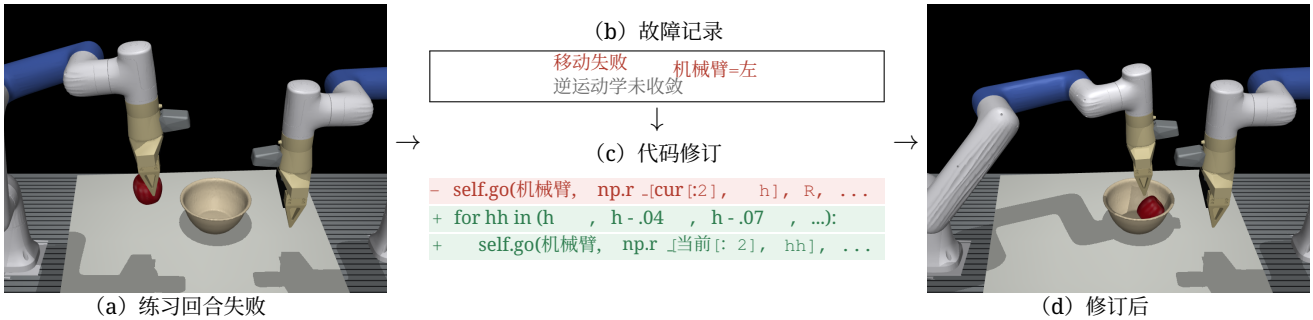
（a）正确世界　(b) 物体缩放 $\times 1.2$

图 S3。**苹果任务中 Examine 的视觉通道**。每个重建世界均按照第一帧记录的关节角度进行渲染，此时尚未发生接触（渲染结果经过淡化并裁剪至物体区域）。蓝色：苹果和碗的演示掩码。橙色：在重建世界中渲染的同两个物体的轮廓。该通道通过计算每个物体相对于其演示掩码的轮廓交并比进行测量。在 (b) 中，两个渲染轮廓均比 (a) 中更大。

。S5图中展示了四个机器人在八项任务中各自最终程序的一个关键帧。成功率见表5。

E. 真实机器人执行

最终程序在真实机器人上作为普通代码运行，无需代码修改、人工干预或大语言模型调用（第 3.6 节）。图S6a–S6d 展示了每种机器人执行每项任务的一次试验中的三个画面：起始状态、关键时刻和最终状态。xTrainer、双臂 Franka 和 ARX LIFT2 在真实机器人上完成了全部八项任务，Unitree G1 则完成了其中四项（苹果、立方体、叠杯子和马克杯-盒子），因此图S6d有两行。图S7 展示了双臂 Franka 的一个程序在三项任务中分别从三种不同物体布局出发运行的情况。



（a）练习回合失败　　（d）修订后

图 S4。**苹果任务上的一轮自我改进**。（a）重建世界中第 0 轮程序的一次练习回合：苹果被拿在碗旁边。（b）该回合的故障记录。（c）编码代理在第1轮中所作的修订（行数以省略号缩短）：机械臂移动到高度 h 的单次动作变为一个遍历递减高度的循环。（d）修订后的程序的一次练习回合：苹果在碗中。

F. 协议和基线方法

真实机器人协议。最终程序在仿真中练习后即被冻结，并作为普通代码在真实机器人上运行，运行时不进行代码修改、无需人工干预，也不调用大语言模型（第 3.6 节）。每项任务进行 10 次试验，物体由人工放置在不同位置。xTrainer、双臂 Franka 和 ARX LIFT2 执行全部 8 项任务；Unitree G1 执行其中 4 项（苹果、立方体、叠杯子和马克杯-盒子）。双臂 Franka、ARX LIFT2 和 Unitree G1 均未接受示范。各任务和机器人的结果见表 5；示例见图3(b)、图S6a–S6d和S7，以及视频中。

基线方法。所有基线方法均在源机器人 xTrainer 上进行仿真评估，使用与 RESET 相同的每项任务 50 个未见测试布局（测试集0–49）（表1）。

- 开环演示回放会重放该示范。
- 零样本代码针对每个任务采样 3 个独立样本，使用与 RESET 相同的接口和接触先验；表1报告这三个样本的平均值和最佳结果。
- $\pi_{0.5}$ 在来自不相交布局的 100 条 RESET 轨迹上进行 6,000 步微调，每个任务使用一个模型。
- 扩散策略基于相同的轨迹从头开始训练，每个任务使用一个模型。

两种策略均使用其默认设置。

G. 补充视频

项目主页上的视频按顺序介绍了问题（为什么自我改进的机器人代码需要一个用于练习的世界）、理念（一次真实示范所能提供的内容）、方法（RESET的五个步骤：重建、Examine、自我改进、执行和迁移）以及仿真结果。随后，视频展示了四个机器人在八项任务上的仿真运行过程，以及最终程序在这四个机器人上的真实机器人运行过程。运行过程均经过加速。

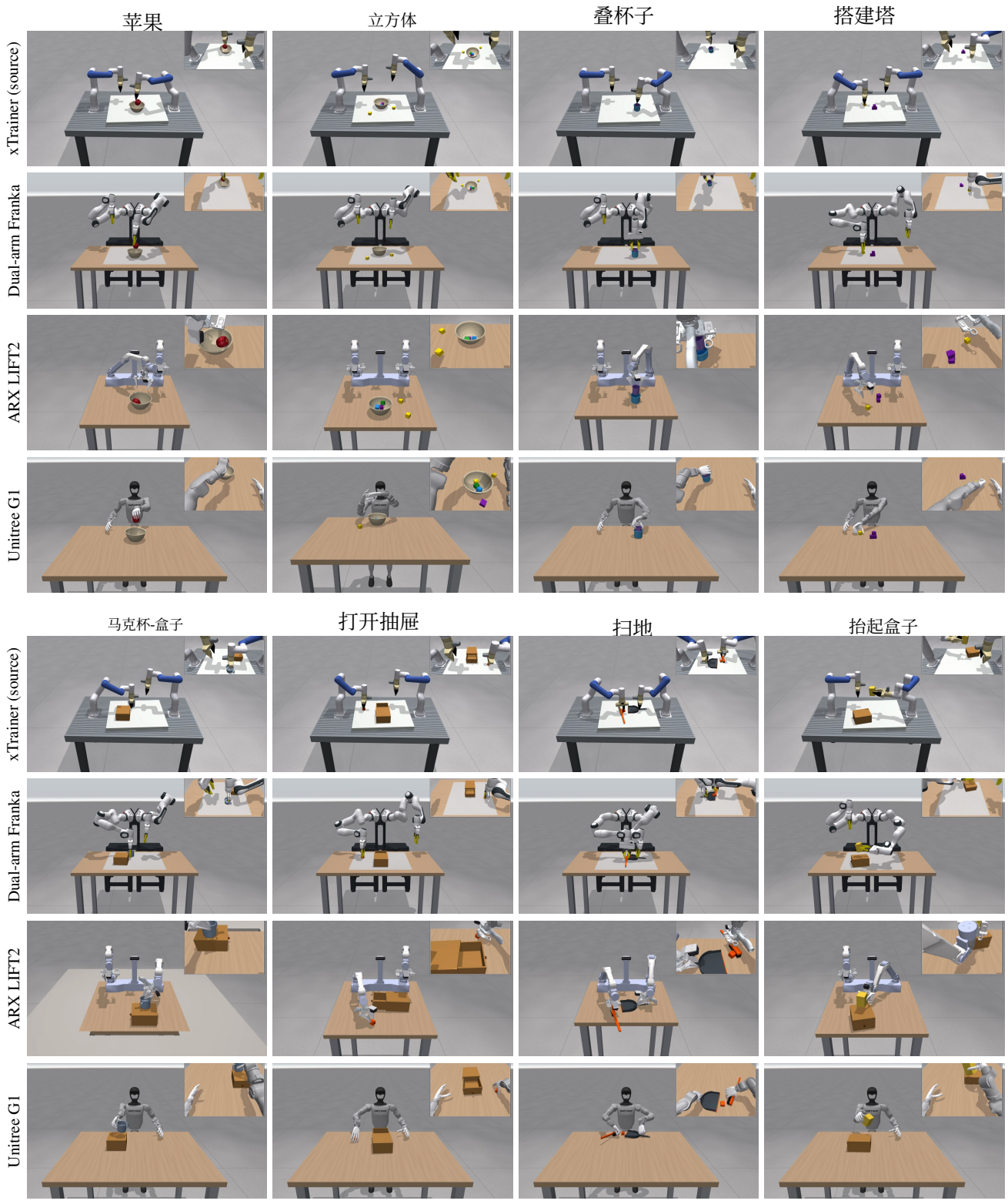


图 S5。**仿真执行，四个机器人执行八项任务**。每个机器人在每项任务的重建世界中运行其最终程序的一个关键帧；右上角的插图显示机器人自身的摄像头视图。双臂 Franka、ARX LIFT2 和 Unitree G1 未接收示范。

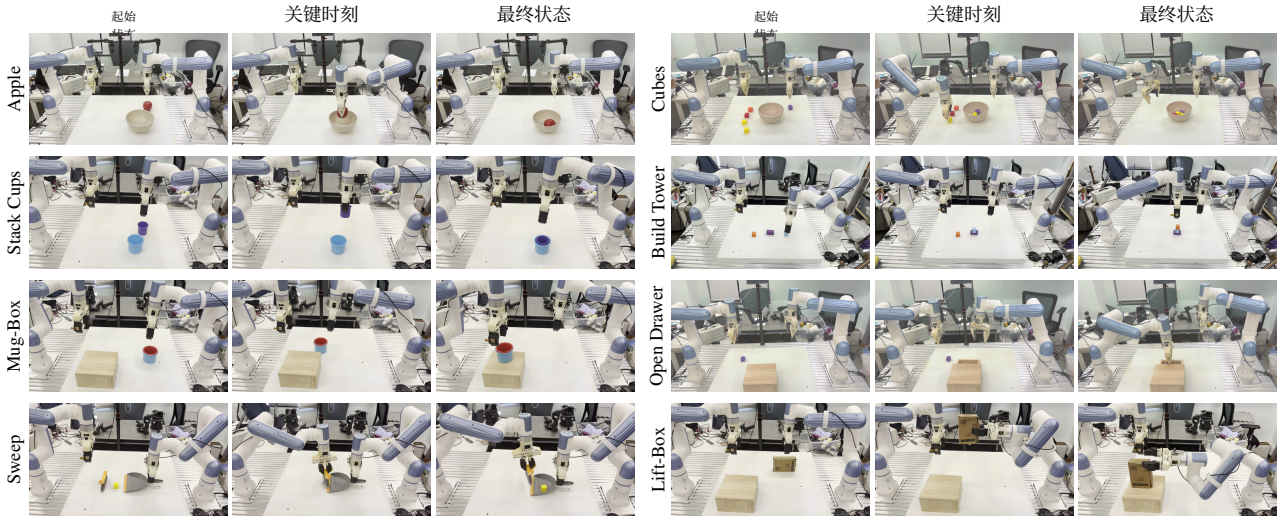


图 S6a。**xTrainer（源机器人）上的真实机器人执行**。每项任务一次试验的起始状态、关键时刻和最终状态。

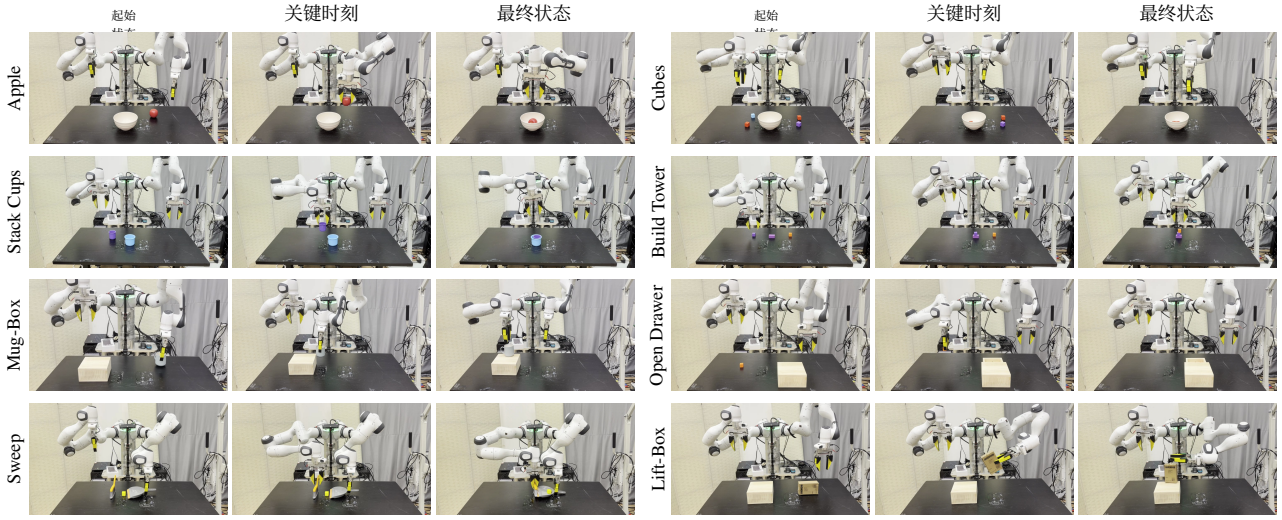


图 S6b。**双臂 Franka 上的真实机器人执行（无示范）**。每项任务的一次试验的起始状态、关键时刻和最终状态。

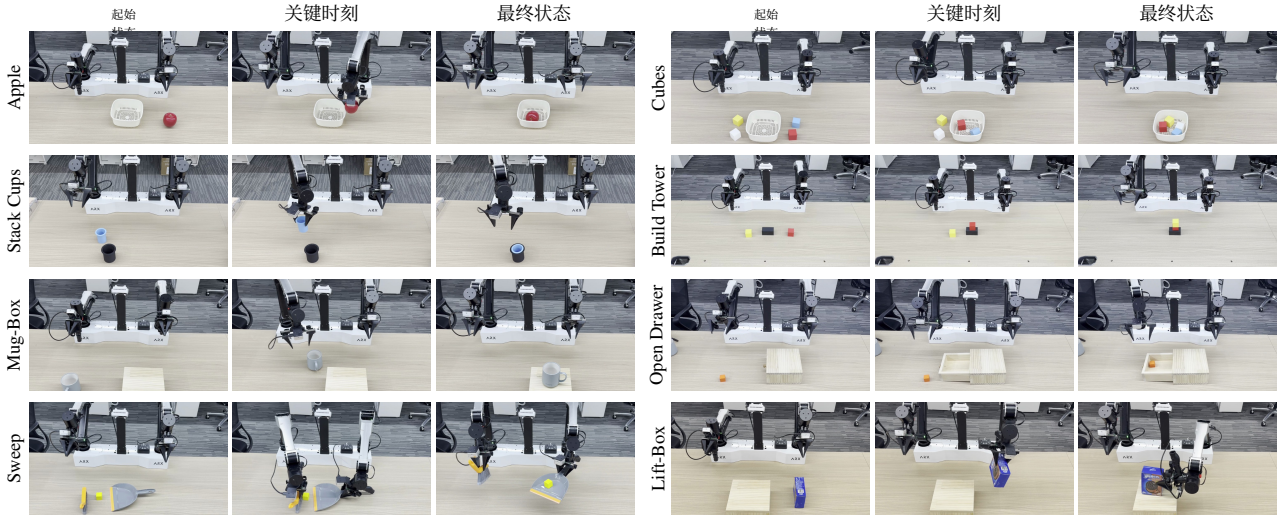


图 S6c. **真实机器人　在 ARX LIFT2 上执行（无示范）**。每项任务一次试验的起始状态、关键时刻和最终状态。

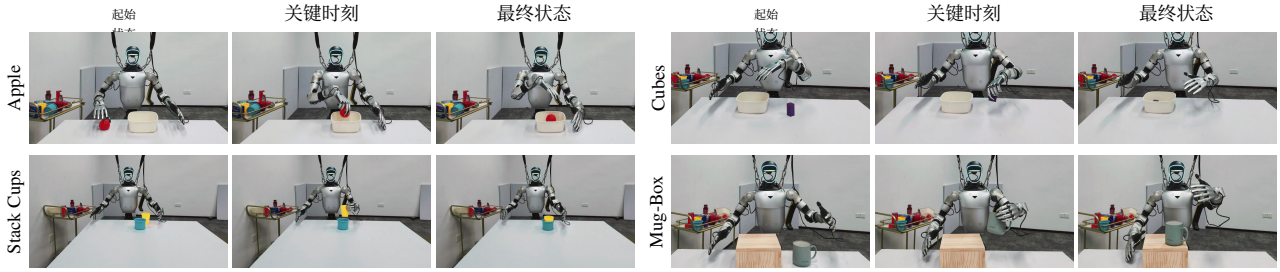


图 S6d。**Unitree G1 上的真实机器人执行（无示范）**。它在真实机器人上执行的四项任务；每项任务的一次试验的起始状态、关键时刻和最终状态。

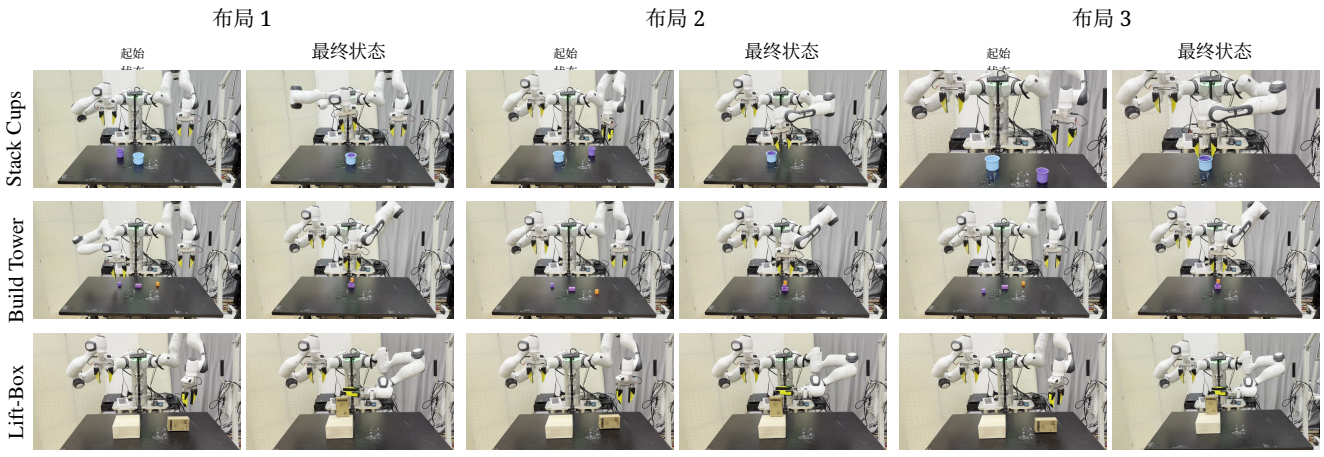


图 S7。**相同程序，不同物体布局**。真实机器人上的双臂 Franka 针对每项任务，从三种不同的手动放置物体布局中分别运行一个程序；每次试验的起始帧和最终帧。