arXiv:2301.01382v1 [cs.RO] 3 Jan 2023

任务序列模拟器：面向机器人操作的机器学习与执行仿真集成

Kazuhiro Sasabuchi[1], Daichi Saito[1], Atsushi Kanehira[1],Naoki Wake[1],Jun Takamatsu[1],Katsushi Ikeuchi[1]

摘要——本文介绍一种用于机器人操作的任务序列模拟器，旨在集成学习仿真与执行仿真。不同于现有的机器学习仿真采用不可分解的仿真来模拟训练场景，任务序列模拟器使用构建模块运行组合式仿真。通过这种方式，学习仿真的结构与多步骤执行仿真相似。为组合学习场景和执行场景，本文提出并使用一种统一的、可训练且可组合的模块描述形式，称为概念模型。借助模拟器设计和概念模型，实现并展示了一种可重复使用的模拟器，可用于学习不同任务，并为从学习到执行以及从仿真到现实提供通用基础。

I. 引言

模拟器在机器人学中十分重要。与现实世界相比，模拟器可以持续运行、确保安全、消除随机性，并提供真实数据。在机器人学中使用模拟器的一种方式，是在真实机器人执行前帮助检查碰撞并确保安全执行。另一种方式是将模拟器用作机器学习工具。如今，适用于上述任一用途的模拟器已经问世；然而，若能拥有一款兼具两种用途的模拟器，将更有助于整合学习与执行，尤其是在多步骤操作领域。

在多步骤操作中，必须对依次发生的一系列任务进行仿真。一个按顺序执行的任务序列示例是搬运物体，该任务由以下步骤组成：抓取、拾取、搬运。执行仿真器必须能够触发这些不同的任务，并将它们连接成序列化仿真。当模拟器用于检查机器人执行情况时，仿真就是组合已编程或训练好的构建模块（例如，先运行“抓取”，再运行“拾取”，然后运行“搬运”）。

相比之下，机器学习模拟器通常忽略任务序列组合，其结构旨在训练特定问题或基准测试（例如，未分解的“抓取与放置”仿真）[1][2]。这种结构差异导致执行仿真与
学习仿真。学习所得的结果会局限于训练场景，这与用于执行的仿真相矛盾，因为后者的场景并非固定。

相反，可以设计一个类似于执行仿真器的机器学习仿真器。“抓取与放置”场景可以分解为按序进行的仿真，即“先抓取，再拾取，再搬运，再放置，最后释放”。与执行仿真器的区别在于，

[1]所有作者均来自美国华盛顿州雷德蒙德的微软
Kazuhiro.Sasabuchi@microsoft.com



图 1. 所提出的任务序列模拟器，可为机器人操作中的学习和执行构建场景。

其中一些构建模块处于“训练中”，并会随着数据的收集而更新。更新完成后，训练好的构建模块可以组合起来，用于不同的场景，因此，学习仿真可以直接过渡到执行仿真。此外，这种设计还能基于已编程或预先训练的构建模块，学习新的操作技能。例如，可以通过“先抓取再拾取”这一序列来训练“抓取”，其中“拾取”是一项已编程任务，用于提供监督信号（即，如果“拾取”成功，就说明“抓取”成功）。

本文介绍了一种任务序列模拟器结构，可实现从学习到执行的一体化模拟。该模拟器的核心采用一种称为“概念模型”的统一模块设计，这是本文提出的设计，用于定义训练任务、收集训练任务以及运行任务以组成序列所需的描述。

本文其余部分安排如下：第二节介绍现有机器人仿真器的背景。第三节解释实现从机器学习到机器人执行仿真的整体模拟器结构。第四节介绍模拟器的概念模型核心组件，第五节则提供该模型的一些详细实现示例。第六节展示该

模拟器在机器学习到机器人执行方面的能力，随后第七节给出结论。

II. 背景

虽然现有的机器人学模拟器种类繁多，但由于以下原因之一，它们可能无法实现从学习到执行的一体化多步骤操作目的：1）模拟器面向操作以外的领域；2）模拟器可用于操作，但缺少用于学习的仿真能力；3）模拟器可用于操作，但缺少执行仿真能力；4）模拟器可用于操作，也能满足学习和执行的需求，但并非专门用于从学习到执行的场景。

常见的机器人学模拟器[3][4]包括 Gazebo[5], MuJoCo[6],CoppeliaSim[7]，CARLA[8]，AirSim[9], 和 Webots[10]。Gazebo 的优势在于能够利用集成的 ROS 传感器和执行器来模拟执行，但由于仿真性能较慢，且物理仿真不一致，因此并非数据采集和机器学习的最佳选择。因此，Gazebo 属于第二类。另一方面，MuJoCo 等引擎适用于机器学习中的稳定物理仿真，但缺少一些机器人仿真能力，例如逆运动学和视觉反馈（真实感渲染）。它们侧重于物理仿真，而非用于机器人执行的一体化模拟器，因此属于第三类。CARLA 和 AirSim 主要面向汽车，例如无人机和轿车，因此缺少操作所需的一些重要特性，例如运动学，属于第一类。

CoppeliaSim 是一款集成式模拟器，包含运动学和物理引擎；PyRep 工具包[11] 可与该模拟器配合用于机器学习。WeBots 也是一款集成式模拟器，而 DeepBots[12]等框架则有助于将该模拟器用于机器学习。这些模拟器的机器学习功能是由社区开发的外部功能。虽然这些模拟器既可用于执行，也可用于学习，但它们并非为集成式的“从学习到执行”而设计，而是分别用于其中一种用途。也就是说，这些模拟器并未设计为连接学习与执行；学习和执行是彼此独立的用例：前者使用社区提供的机器学习包装器，后者则使用集成功能来模拟机器人系统的执行。

与现有模拟器相比，任务序列模拟器旨在连接学习仿真与执行仿真。该模拟器采用概念模型，可组合预训练、编程或训练得到的任务，这一强大特性有助于实现从机器学习到真实机器人执行的过渡。（例如，接入机器学习平台后，再连接 ROS 以执行任务。）更重要的是，这种紧密集成支持使用前序任务和后序任务执行进行训练等功能，同时还能通过训练收集可复用执行模块。

III. 任务序列模拟器概述

任务序列模拟器包含两层：用于“动作决策”的概念接口，以及用于“状态观测”的环境引擎流水线（图 1）。然而，与典型的学习模拟器不同，后者针对的问题具有不可分解的结构，且随着问题数据不断收集，动作决策由单一策略更新；任务序列模拟器则在动作决策中增加了一层抽象，使问题由一系列任务组成（即在一组任务之间切换，每项任务都运行一个独立策略）。这样，在训练完成后，学得的任务策略便可成为用于执行的策略集合中的一部分。下文将进一步介绍每一层。

A. 概念接口

在每个仿真时间步，机器人都会根据当前世界状态决定下一步动作。这一决策称为策略。当状态、动作与下一状态之间的关系（系统动力学）完全已知时，可以直接编程实现该策略。当系统动力学未知时，则需要通过数据采集来学习策略或系统动力学。若仅针对未知动力学采集数据，并直接计算已知动力学，数据采集会更加高效。因此，最好将机器人的执行分解为一系列任务，每项任务执行各自的策略，该策略针对其所涵盖的系统动力学进行了优化。此外，将机器人的执行分解为多个任务，可以提高各项任务策略的可复用性，并能利用任务构件组合出不同的执行场景。

概念接口层会为训练或执行场景选择并切换任务，前提是：（1）已知要仿真的任务序列（实验中展示了如何获知）；（2）每项任务策略都能指示任务何时完成（即具有已学习或已编程的完成信号）。在执行仿真期间，概念接口层会按序列选择相应任务，并在当前策略返回任务完成信号后切换到下一项任务。学习仿真也可以采用类似方式进行，但需要评估策略输出的完成信号（例如，根据后续任务是否成功来评估）。这种可互换的结构实现了从学习到执行的一体化仿真。为了在任意训练序列下学习终止信号，各项任务采用一种统一的设计，称为“概念模型”，后文将进一步说明。

B. 环境引擎流水线

在一个时间步的控制尺度下，机器人以物理实体进行感知—规划—行动。因此，机器人学中的仿真需要三个重要引擎：运动学、物理（接触动力学）和渲染。

运动学引擎的作用是仿真机器人的动作计划与本体实际运动之间的关系。物理引擎的作用是仿真机器人本体与环境之间的关系。渲染引擎的作用是仿真环境与机器人感知之间的关系。用更专业的术语来说，运动学引擎负责求解笛卡尔空间与构型空间之间的映射，物理引擎负责求解微分代数方程[13]，采用的技术包括基于速度冲量线性互补的时间步进方法[14][15],；渲染引擎负责求解渲染方程[16]，以确定用于像素颜色生成的光照路径。

渲染引擎通过生成图像来模拟机器人的感知，但感知与感知理解（从生成的图像中提取有意义的状态）之间存在差距。与其直接学习从感知到规划的映射，有时先进行感知再规划、提取视觉特征会更高效。此外，在机器人学中，结合视觉和力反馈十分重要；视觉特征有助于压缩反馈，使视觉与力具有对齐的状态维度。因此，所提出的模拟器在渲染引擎之外，又加入了第四种“后处理引擎”。

这些承担不同角色的引擎按照有序的流水线依次触发，以计算仿真世界的当前状态。模拟器通常会将特定引擎打包在一起，以生成单个仿真世界，但一般而言，这些引擎也可以独立运行，并组合或协调多个仿真世界，以实现更好的仿真。例如，可以使用 ROS MoveIt 进行精确的逆运动学仿真，使用 PyBullet 进行可复现的物理仿真，并使用 Unreal Engine 进行照片级真实感的光线追踪渲染。只要每个引擎都能够加载相同的机器人和物体模型，并能在各引擎之间共享机器人和物体的状态（可通过 TCP连接等方式实现），就可以组合不同的引擎。此外，也可以不使用仿真引擎，而是连接“真实”引擎，用真实机器人的扭矩传感器替代模拟物理，并用来自机器人相机的真实图像替代模拟渲染。

IV. 概念模型

单个任务块在某种系统动力学下运行，并从初始状态达到目标状态。因此，
任务可以使用系统的执行者、初始状态、目标（结束）状态以及系统动力学参数来描述（图2）。这种任务描述可称为“任务模型”[17]。如果系统动力学完全已知，这种描述足以执行任务。初始状态通常是前一个任务的结束状态。然而，当系统动力学尚未完全已知时，就需要学习；在学习过程中，初始状态必须随机化。

与其完整描述任务，不如使用执行器配置、初始状态、根据可观测系统状态描述的必要目标状态，以及根据不可观测

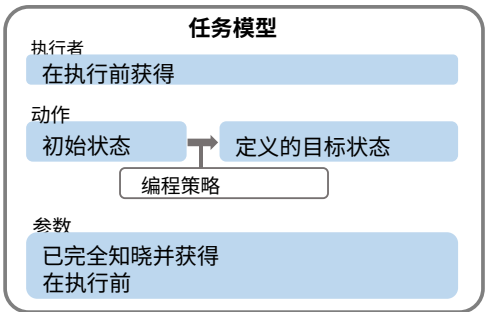


图 2. 任务模型示意图。

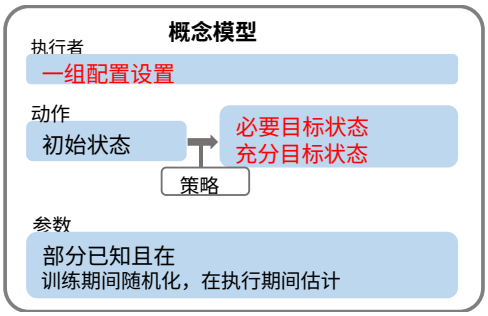


图 3. 概念模型示意图。

系统状态描述的充分目标状态和系统动力学的部分已知参数来描述任务（图3）。这种任务描述称为“概念模型”。与任务模型相比，它可能不够具体，无法用于执行，但描述了任务的概念，以便学习如何执行任务。在系统动力学完全已知且任务可编程的特殊情况下，概念模型的描述与任务模型相同。

通过提供结构化的描述格式，概念接口层可以在训练序列中交替访问这些模块，因为它们的结构相同，仅细节有所不同。

A. 学习中的概念模型使用

概念模型描述用于学习任务完成信号，以及用于处理未知系统动力学的动作。必要目标状态和充分目标状态描述用于评估任务完成信号和动作是否恰当。评估通过最小化当前状态到目标状态的代价来完成。请注意，每次动作决策后都会评估到必要目标状态的代价，而到充分目标状态的代价仅在训练期间选定任务完成信号时评估一次。

执行器配置描述了用于训练任务的可能环境（包括机器人和任何操作目标的世界）。如果训练时不涉及前序任务，则使用其中一个执行器配置来定义任务的初始状态。否则，前一任务的结束状态就是初始状态。与初始状态不同，执行器配置独立于前一任务的状态，因此可以进行配置，并用于随机化训练状态。

B. 概念模型在执行中的使用

当系统动力学完全已知时，概念模型与任务模型的作用相同。任务状态根据系统动力学和执行者，通过编程策略所提供的动作从初始状态发生变化。执行期间，任务的初始状态是前一任务的结束状态；达到目标状态后，任务即告结束。

当动力学信息并不完全已知时，任务的可观测系统状态会在学习策略的作用下发生变化。与已编程的情况类似，初始状态是前一任务结束时的可观测状态。然而，由于目标状态的一部分不可观测，因此不能仅凭模型描述来确定任务是否结束。相反，我们使用概念模型描述中的学习任务完成信号来识别任务的结束。

V. 模型实现

遵循概念模型结构，任务的实现方式既便于训练，也便于随后将其收集为执行用的构建模块。本文采用基于螺旋理论[18]的动力学分离[17]，以将一个任务与另一个任务区分开来。也就是说，一旦初始化了操作目标与机器人的末端执行器之间的关系，任务就会打破或维持目标与环境之间的接触状态。通过对接触点的不等式方程模式进行分类，可得到七种纯平移任务和七种纯旋转任务。图4展示了其中一些任务作为概念模型的实现示例。下文将介绍图中部分示例的细节。请注意，由于任务分类仅取决于末端执行器、操作目标与环境之间的关系，因此可以忽略机械臂的运动（构型空间）[19]，任务只关注末端执行器的运动（笛卡尔空间）。

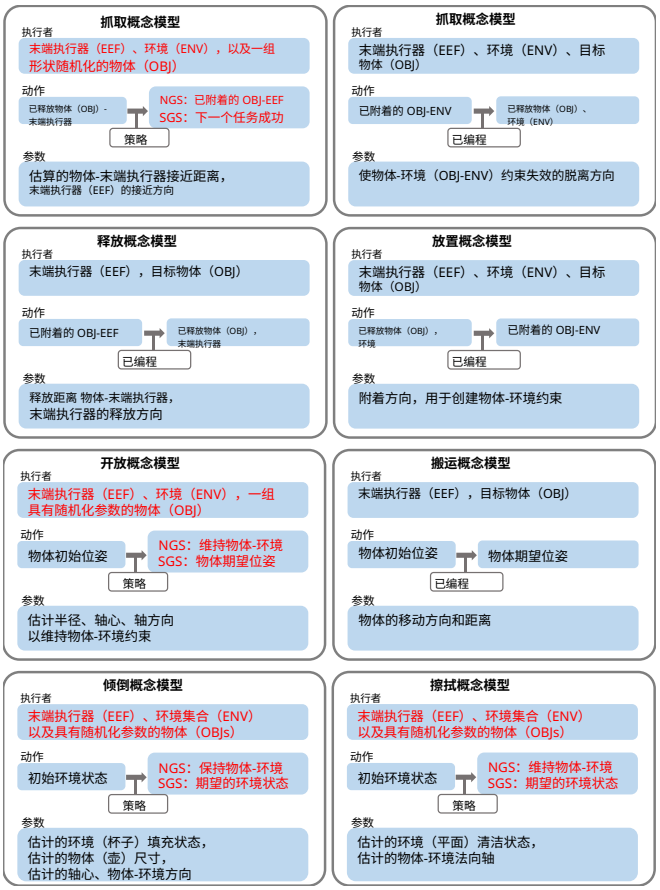


图 4. 基于螺旋理论的分类中八种不同任务的概念模型示例。

A. 抓取

抓取任务建立操作目标与机器人的末端执行器之间的关系。执行者包括目标物体、环境（例如桌子）和末端执行器。初始状态是目标物体附着于环境，但未附着于末端执行器的状态（包括手指关节的形状）。目标状态是目标物体也附着于末端执行器，并施加足够的力以执行后续任务的状态。参数是.....之间的距离

目标和末端执行器，以及末端执行器的接近方向。

当目标物体的细节完全已知时，可以根据上述细节定义并编程任务模型。然而，在现实世界中，物体形状和与目标之间的距离都存在不确定性；由于真实接触传感器精度不足，或缺乏检测滑移的传感器，难以判断施加的力是否足够。如果未能针对物体属性的不确定性谨慎设计策略，在接近过程中可能会因手指与物体碰撞而导致抓取失败。

相反，可以将一组执行器配置定义为来自随机范围的物体形状；将必要目标状态定义为末端执行器与目标物体的适当表面接触（可在真实机器人上通过手指配置以及带阈值的手指力矩传感器获取，以判断接触或未接触的二元状态）；将充分目标状态定义为成功完成后续任务，并将估算的距离用于参数。利用定义的目标状态构建奖励（剩余代价），以供策略学习。

利用该概念模型，可以学习接近策略（围绕末端执行器接近方向调整运动），以及为执行后续任务所需的适当手指闭合量。学习策略根据物体形状选择足够的闭合量；物体形状可以部分通过手指接触物体后末端执行器手指关节的形状推断出来。达到足够的闭合量后，策略会返回终止信号。

B. 开门

开门任务是一项单自由度纯旋转任务。执行者包括目标物体（门）、环境（铰链）和末端执行器（连接到

门把手）。初始状态是末端执行器与目标物体处于连接状态。目标状态是目标物体移动到某个所需方位。参数包括由目标物体和环境定义的旋转半径、旋转轴中心和旋转轴方向。

当目标和环境的细节完全已知时，可以根据上述细节定义并编程任务模型。然而，在现实世界中，环境参数存在不确定性。因此，使用一组执行器配置（随机设定半径）、一个必要目标状态（在每个时间步沿环境约束移动物体；可在真实机器人上通过在腕部使用力传感器并检查是否超过最大应力阈值来获得）、一个充分目标状态（确保门已达到所需朝向），以及参数估计值来描述概念模型。

通过使用腕部仅配备力传感器反馈的末端执行器，该模型能够学习一种策略，在每个时间步更新参数估计值，然后根据更新后的参数生成手部运动。策略在推断出已达到所需朝向后，会返回终止信号。

C. 搬运

搬运任务是一项具有六个自由度的平移和旋转任务。执行者是目标物体和末端执行器。初始状态为末端执行器与目标物体处于附着状态。目标状态为目标物体移动到某个相对位置。参数为移动方向和距离。

由于目标或环境不存在不确定性，因此可以手动指定参数，并直接根据参数指定目标。因此，概念模型与任务模型相同，且可以通过编程实现。

VI. 实验

实验采用了上一节所示的概念模型实现，以及开发的任务序列模拟器。

在学习实验中，要仿真的任务序列是预先定义的。这些实验旨在展示模拟器的有效性，以及其用于训练不同任务时的可复用性。

通过在每个回合开始时运行待训练任务的前置任务，并在奖励返回时运行后续任务，该模拟器兼容常见的强化学习平台，并可使用现成的学习算法。在实验中，模拟器连接到 Bonsai 平台，并使用 PPO 算法。

在执行实验中，任务序列取自人工演示，动作则使用学习到的任务策略生成。这些实验旨在展示模拟器对执行仿真的有效性（不同组合场景的执行）。需要注意的是，本实验使用了与训练时完全相同的模拟器和任务块，展示了模拟器从学习仿真过渡到执行仿真的能力。

对于仿真，将 PyBullet 引擎接入环境引擎流水线以承担物理引擎的功能，并使用 Unreal Engine 进行渲染，从而获取状态。对于真实机器人，则将 ROS（通过 roslibpy 连接）接入环境引擎流水线以获取状态。机械臂运动学采用 ROS MoveIt 软件包。

A. 训练

图5-(A)展示了用于运行抓取训练的任务序列模拟器构型。配置的训练序列“先抓取再拾取”被传递给模拟器。“抓取”任务是待训练的任务，而已编程的“拾取”任务则用作后续任务，以评估充分目标状态。

图5-(B)展示了在真实机器人上执行的已训练抓取结果。概念模型将物体形状参数设为未知。尽管存在这种不确定性，学习策略仍成功抓取了各种不同形状的物体，包括但不限于盒子、圆柱形杯子、椭圆形饭团和菱形糖果盒。

图5-(C)展示了另一种机器人手学习到的抓取策略，该策略使用相同的模拟器和概念模型进行训练，但采用了不同的执行器配置（末端执行器）设置。结果表明，该模拟器可复用于训练具有不同机械结构的机器人（包括多指机器人手和自由度有限的夹爪）。

图5-(D)显示，通过将已配置的训练序列改为“先抓取再打开”，可以使用同一个模拟器训练开门任务。“打开”任务是待训练任务，而训练好的“抓取”任务则被重新用作前置任务，以建立末端执行器与目标门之间的关系。尽管旋转半径、中心和轴方向存在不确定性，真实机器人仍使用学习策略完成了开门。尽管该策略仅使用仿真数据进行训练，但它可以直接应用于真实机器人，因为充分目标状态不要求真实机器人具备可观测性，且策略动作决策仅依赖于仿真到现实差距极小的状态。

B. 执行

图6-(A)展示了任务序列模拟器如何使用由人类演示的序列。与之前训练实验中已配置的序列不同，此处的序列是通过使用 [20]中所述方法对演示进行分解而自动生成的。这里使用的概念模型与训练实验中的相同，但禁用了策略更新（模拟器未连接任何训练算法，而是使用固定的学习策略，且不进行更新）。

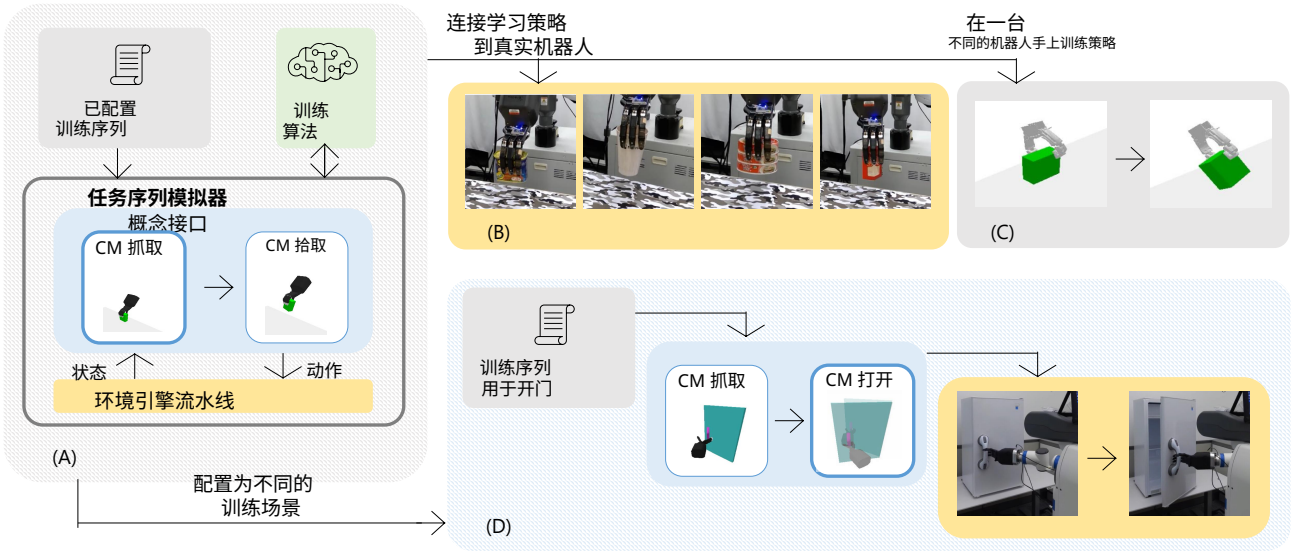


图5. 任务序列模拟器用于学习时的结果。

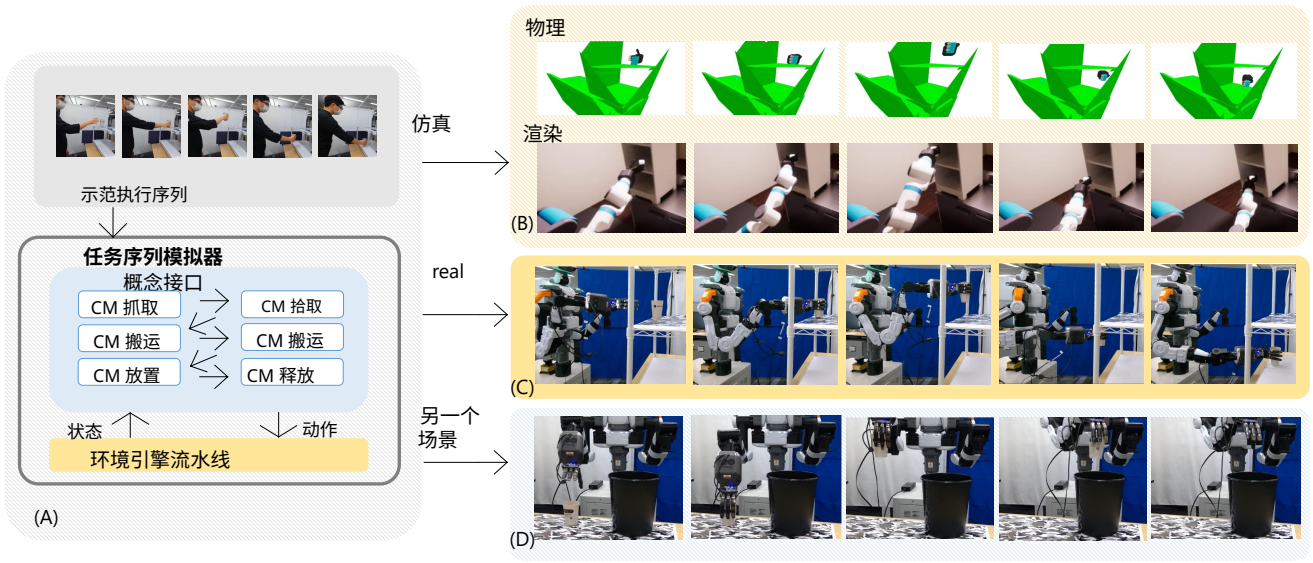


图6. 任务序列模拟器用于执行时的结果。

图6-(B)展示了对演示序列的仿真执行。第一行显示物理引擎的输出，第二行显示渲染引擎的输出。由于训练和执行都运行在同一系统上，学习策略可以轻松用作执行仿真。学习策略本身已是一个构建模块，可以与其他任务组合，以生成诸如“从上层架子上拾取一个杯子并将其重新放到下层架子上”这样的应用。

图6-(C)展示了通过切换环境引擎流水线中的引擎并连接 ROS，对演示序列进行真实机器人执行的过程。这说明模拟器如何通过使用相同的策略连接，仅更改获取状态和执行动作所依托的引擎，便能从仿真机器人执行转为真实机器人执行。通常，从仿真转向现实会引入仿真到现实的差距。然而，场景序列中只有一部分使用了学习策略，并且由于概念模型经过精心设计，将可观测动力学（必要目标状态）的学习与隐藏动力学（充分目标状态）的学习区分开来，因此没有遇到这种差距。

图6-(D)展示了另一种序列的执行：“从桌上拿起一个杯子并把它扔进垃圾桶”。该场景使用相同的概念模型，唯一的区别在于演示输入，这表明借助模拟器和概念模型描述，可以在不同的执行场景中重复使用已学习的策略。如果策略是在完整的“抓取与放置”场景中学习得到的，由于问题动态不同，该策略就难以扩展到“拾取与投掷”场景。

七、结论

本文介绍了任务序列模拟器，它连接了学习仿真与执行仿真。学习用的仿真场景通过任务序列创建。这样，学习仿真与执行仿真便具有相同的结构。模拟器的核心是概念模型，它支持对已编程、已训练和训练中的构建模块进行混合排序。模拟器在集成系统开发方面具有显著优势，同时也为执行仿真和学习仿真提供了新的方向。

从执行角度来看，该模拟器支持组合已编程任务和训练任务，构成任务序列。与仅包含已编程任务的序列不同，混合使用训练任务块的优势在于，部分任务可以包含不确定性，而且任务的目标状态可以通过隐式系统参数来描述（无需直接从真实机器人获取目标状态）。关键在于，通过训练将“观测状态和选定动作是否足以达到目标状态”学习为终止信号。

从学习角度来看，模拟器和概念模型的设计具有以下优势：首先，仿真可复用，并且易于适应场景中的细微变化。只需更改概念模型中的执行器配置，即可学习适用于不同末端执行器的策略。只需更改充分目标状态中的后续任务，即可针对不同场景优化策略。其次，使用概念模型设计来定义学习问题，既能实现分层学习结构，也能构建减少仿真到现实差距的结构。对于使用真实机器人观测时差距不大的状态参数，可用来定义必要目标状态；而对于使用真实机器人观测时差距较大的状态参数，则属于充分目标状态（在仿真中隐式学习，但无需使用真实机器人观测）。由于这种方法只学习存在不确定性的部分，而非学习整个场景序列，因此能够采用这种形式化方法。遵循这一结构化形式化方法，无需额外的真实世界数据收集，即可从仿真迁移到现实，并获得适用于不同执行场景的可复用策略。

致谢

作者感谢 Brice Chung 的团队、Aydan Aksoylar 和 Kartavya Neemafor 在实验所用概念模型的奖励设计和训练方面提供的帮助。

参考文献

[1] Matthias Plappert、Marcin Andrychowicz、Alex Ray、Bob McGrew、Bowen Baker、Glenn Powell、Jonas Schneider、Josh Tobin、Maciek Chociej、Peter Welinder 等人。多目标强化学习：具有挑战性的机器人环境与研究需求。*arXiv* 预印本 *arXiv:1802.09464*，2018年。[2] Linxi Fan、Yuke Zhu、Jiren Zhu、Zihua Liu、Orien Zeng、Anchit Gupta、Joan Creus-Costa、Silvio Savarese 和 Li Fei-Fei。Surreal：开源强化学习框架与机器人操作基准测试。载于机器人学习会议，第767–782页。PMLR，2018年。

[3] Jack Collins、Shelvin Chand、Anthony Vanderkop 和 David Howard。机器人应用物理仿真器综述。*IEEE Access*，9:51416–51431，2021年。[4] Wenshuai Zhao、Jorge Pe˜na Queralta 和 Tomi Westerlund。机器人深度强化学习中的仿真到现实迁移：综述。载于*2020*年*IEEE*计算智能研讨会系列（*SSCI*），第737–744页。IEEE，2020年。[5] Nathan Koenig 和 Andrew Howard。Gazebo（开源多机器人仿真器）的设计与使用范式。载于*2004*年*IEEE/RSJ*智能机器人与系统国际会议（*IROS*）（*IEEE*目录号：*04CH37566*），第3卷，第2149–2154页。IEEE，2004年。[6] Emanuel Todorov、Tom Erez 和 Yuval Tassa。MuJoCo：用于基于模型控制的物理引擎。载于*2012*年*IEEE/RSJ*智能机器人与系统国际会议，第5026–5033页。IEEE，2012年。[7] Eric Rohmer、Surya PN Singh 和 Marc Freese。V-REP：一种通用且可扩展的机器人仿真框架。载于*2013*年*IEEE/RSJ*智能机器人与系统国际会议，第1321–1326页。IEEE，2013年。[8] Alexey Dosovitskiy、German Ros、Felipe Codevilla、Antonio Lopez 和 Vladlen Koltun。CARLA：开放式城市驾驶仿真器。载于机器人学习会议，第1–16页。PMLR，2017年。[9] Shital Shah、Debadeepta Dey、Chris Lovett 和 Ashish Kapoor。AirSim：面向自动驾驶车辆的高保真视觉与物理仿真。载于野外与服务机器人，第621–635页。施普林格，2018年。[10] Olivier Michel。Cyberbotics Ltd. WebotsTM：专业移动机器人仿真。《先进机器人系统国际期刊》，1(1):5，2004年。[11] Stephen James、Marc Freese 和 Andrew J Davison。PyRep：将V-REP引入深度机器人学习。*arXiv* 预印本 *arXiv:1906.11176*，2019年。[12] Manos Kirtas、Konstantinos Tsampazis、Nikolaos Passalis 和 Anastasios Tefas。DeepBots：面向机器人学的基于Webots的深度强化学习框架。载于*IFIP*人工智能应用与创新国际会议，第64–75页。施普林格，2020年。[13] Edward J Haug。机械系统计算机辅助运动学与动力学。第*1*卷：基本方法。Allyn & Bacon, Inc.，1989年。[14] David E Stewart 和 Jeffrey C Trinkle。用于具有非弹性碰撞和库仑摩擦的刚体动力学的隐式时间步进格式。《国际工程数值方法期刊》，39(15):2673–2691，1996年。[15] Mihai Anitescu 和 Florian A Potra。将具有摩擦的动态多刚体接触问题表述为可解的线性互补问题。《非线性动力学》，14(3):231–247，1997年。[16] James T Kajiya。渲染方程。载于第*13*届计算机图形学与交互技术年度会议论文集，第143–150页，1986年。[17] Katsushi Ikeuchi、Naoki Wake、Riku Arakawa、Kazuhiro Sasabuchi 和 Jun Takamatsu。用于表征多模态观察学习机器人家务动作所需常识的语义约束。*arXiv* 预印本 *arXiv:2103.02201*，2021年。[18] MS Ohwovoriole 和 B Roth。螺旋理论的扩展。1981年。[19] Kazuhiro Sasabuchi、Naoki Wake 和 Katsushi Ikeuchi。利用身体角色划分，为不同构型的机器人实现面向任务的运动映射。《*IEEE*机器人与自动化快报》，6(2):413–420，2020年。[20] Naoki Wake、Iori Yanokura、Kazuhiro Sasabuchi 和 Katsushi Ikeuchi。用于观察学习的语言注意焦点系统。载于*2021*年*IEEE*机器人与自动化国际会议（*ICRA*），第10377–10384页。IEEE，2021年。