arXiv:2611.33158v1 [cs.RO] 24 Nov 2026

一个策略，一百万个机械臂

Xiang Fan[1*] Yuheng Wang[1*] Jiayuan Yan[1] Bohan Fang[1] Wisdom Oluchi Ikezogwo[1]
Ainaz Eftekhar[1] Max Argus[3] Zhongzheng Ren[1,2] Jiafei Duan[4] Ranjay Krishna[11]华盛顿大学 [2]北卡罗来纳大学教堂山分校[3]艾伦人工智能研究所 [4]新加坡国立大学

*共同贡献
xiangfan@cs.washington.edu
https://omniact.github.io/



图 1. **OmniAct 是一种具身原生策略**，旨在采用机器人特定运动策略，而不进行与具身无关的规划。我们大规模生成（1）多样化的具身形态和（2）多样化的模拟轨迹，并（3）训练单一策略 OMNIACT。（4/5）在仿真和真实环境中，OMNIACT 在已见和未见的具身形态上均优于所有基线（零样本）。

摘要

能够控制多种机器人的单一策略，不仅必须在不同具身形态之间迁移任务知识，还必须根据每种机器人的能力调整自身行为。现有方法通过共享动作表征促进迁移，但这些表征未必能捕捉机器人形态差异应如何影响操作策略。我们提出 OmniAct，这是一种跨-

具身形态 操作策略可直接根据特定场景中每台机器人的运动学条件生成适应该具身形态的行为。该运动学图表示机器人的关节位置、运动轴及其与视觉观测的连接关系，使策略能够考虑其具身形态与场景中对象和表面的关系。流匹配策略为每个节点预测相对关节运动，从而能够控制自由度和运动学配置各不相同的机器人。为了学习

物理结构如何大规模塑造行为，我们通过程序化方法设计并筛选出一百万种物理上可行的机器人具身形态；接着，开发算法，在仿真中为所有具身形态生成机器人轨迹；最后，在多样化的仿真操作任务上进行训练。在 *2,000* 个未用于训练的具身形态—场景评估回合中，OMNIACT的成功率达到 *63.0%*，而 $\pi_{0.5}$ 为 *17.8%*，最强基线为 *57.9%*；在 *GPT-6 Astra* 接受评估的任务类别上，其成功率达到 *64.7%*，而对比方法为 *41.1%*。在此前未见过的真实世界机器人上，其成功率接近次优方法 *MolmoAct2* 的两倍。除了任务成功率之外，我们还观察到，OMNIACT 能够学习到独特的操作策略，充分利用不同形态结构的能力并适应其局限性。

1. 引言

一种能够控制具备多种具身形态的机器人的统一策略，必须解决两个相关问题：在不同机器人之间迁移任务知识，以及根据每种机器人的物理能力调整行为。现有方法在解决前一个问题上取得了显著进展，方法是寻找可在不同具身形态间共享的表征和接口。通用视觉-语言-动作模型采用共享的末端执行器指令、统一的动作分词器或学习得到的动作潜变量 [4, 9, 21, 27, 36]。以对象为中心的方法则对期望的对象运动或交互进行表征，使其能够在不同机器人之间迁移 [3, 18, 25, 44]。这些方法利用不同具身形态之间的共性来促进迁移。然而，迁移任务知识并不一定能实现对每种机器人的有效控制。由于运动学、工作空间和灵巧性各不相同，即使两个机器人追求相同目标，也可能需要采用不同的抓取方式、轨迹或接触策略。

学习具身形态特定行为面临着几项挑战。首先，策略必须结合观测到的场景，对机器人的几何结构进行推理。关节构型是否有用，取决于其关节和运动轴相对于需要与之交互的对象和障碍物的位置。现有策略会纳入 URDF文件或运动学图等描述，但往往将机器人独立于场景进行编码 [7, 32, 35]。其次，策略必须适应不同的运动学结构，包括关节类型、自由度和连杆拓扑的差异，同时支持通用的控制接口。现有方法已在相对较小规模的机械臂集合中展示了迁移能力 [28, 39, 47]。第三，策略必须学习形态变化如何影响行为，以便控制训练期间未见过的具身形态。这需要在足够多样的物理结构上积累经验。以往将学习扩展到数百种形态结构的工作，主要聚焦于运动 [1, 15]，而非操作。

我们推出了**OMNIACT**，这是一种在一百万种具身形态上训练的策略，能够针对每种具身形态及其在特定场景中所能实现的能力调整行为。我们的方法通过场景 *grounded* 运动学图表示每个机器人，将其关节几何结构表达在各观测相机的三维坐标系中。每个关节和指尖都表示为一个节点，其中包含其相机坐标系位置、运动轴、关节类型、本体感受状态和运动学连接关系。这使策略能够将机器人的物理结构与相应的视觉观测联系起来。我们将这种表示与可变自由度动作表示相结合，利用流匹配策略联合预测相对关节运动和末端执行器位姿更新。通过在所有具身形态间统一索引方案，我们的模型学会在同一个网络中支持数量和构型各异的关节。这两个组成部分使策略能够直接根据每个机器人的运动学结构及其与场景的关系生成动作。

训练策略以利用这些差异，需要机器人形态具有相当大的多样性。仅在少数几种具身形态上训练的策略，或许能为这些机器人学会有效的操作策略，却未必能学会如何将其调整以适用于新机器人。我们开发了一套流程，用于**实例化一百万种独特的机器人具身形态，并创建能够在仿真中自动为每种形态生成轨迹的算法**。我们的数据生成管线通过改变关节配置、连杆几何形状和运动学拓扑来生成不同具身形态；我们采用逆运动学筛选和基于物理的验收，确保每条轨迹都可行。由此形成的训练分布让策略接触到广泛的机器人配置，并使我们能够在未见过的具身形态上进行测试。

我们在训练时未见过的具身形态和场景中评估了 OMNIACT。在 OMNIACT-BENCH上，它在 500 步内的回合成功率为 63.0%，而最强基线为 57.9%，$\pi_{0.5}$为 17.8%；在 GPT-6 Astra 进行评估的任务类别中，它的成功率达到 64.7%，高于 GPT-6 Astra 在相同步数预算下的 41.1%，也高于其在两倍步数预算下的 54.8%（表3）。对于训练时未见过、配备细尖端夹爪的真实 OpenYAM 机械臂，OMNIACT完成了 20 次抓取与放置试验中的 13 次。唯一取得成功的基线是 MolmoAct2-BimanualYAM；由于该机械臂属于其训练分布，它完成了 20 次试验中的 7 次。其他所有基线均未能完成任何试验（表5）。我们观察到，在这款机械臂上，该策略有时会利用这种未见过的具身形态特有的长末端连杆：先将其水平伸出，再旋转夹爪以够到远处的目标（第3.4节）。这种行为在仅使用 YAM 训练的 MolmoAct2 中并未出现。这些结果表明，以机器人的运动学为基础的策略有望实现更好的跨具身形态泛化。

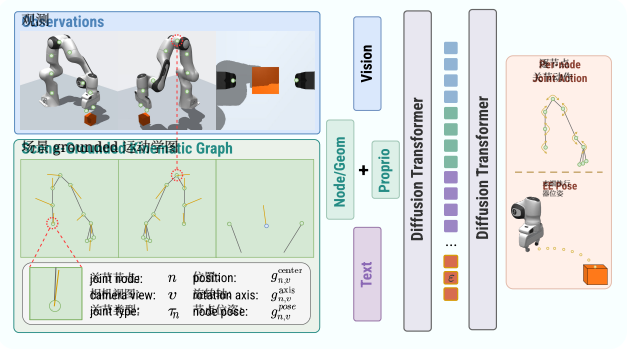


图2. **OMNIACT 架构**。多视角 RGB 观测、语言指令和相机坐标系下的场景锚定运动学图共同作为共享流匹配 Transformer 的条件输入。每个节点表示机器人上的一个关节或标注点，并编码其在各个相机坐标系中的几何状态和本体感受状态。该策略通过可变自由度动作表示联合预测相对关节运动和末端执行器位姿。

2. 方法

我们提出了 OMNIACT，这是一种单一策略，能够在运动学结构各异的机器人之间学习跨具身形态操作。我们的方法包含三个部分。首先，我们通过场景 grounded 运动学图表示每种机器人，该图在视觉观测的坐标系中描述其几何结构。其次，我们开发了一种可变自由度动作表示，可针对不同的机器人配置联合预测关节运动和末端执行器位姿。第三，我们生成多样化的具身形态，并在仿真中收集成功的操作轨迹，以训练该策略，使其覆盖一百万台机器人。

策略构建。我们研究不同机器人之间的操作任务，这些机器人的关节数量、连杆几何形状和运动学结构各不相同。在每个时间步 t，策略接收语言指令 ℓ、多视角 RGB 观测 $\mathcal{I}_t = \{I_{t,v}\}_{v=1}^{V}$，以及机器人的当前构型和本体感受状态。机器人的具身形态规定了其运动学拓扑、关节类型、限位和几何参数。

策略预测一个包含相对关节运动和末端执行器（EE）位姿更新的动作块。训练期间会对这两类预测同时进行监督，并在执行期间用于控制。执行预测动作块的前缀部分后，再获取更新后的观测并重新规划。所有具身形态共享相同的策略参数。

我们使用 SigLIP v2 [38] 对 RGB 观测进行编码，并使用 Qwen 2. 对语言指令进行编码。5 [2] 机器人的几何结构、配置和本体感知通过运动学节点标记来表示。

2.1. 场景 grounded 运动学图

机器人能够执行的动作取决于其运动学结构，以及该结构相对于视觉观测所处的位置。

运动学结构。我们将每个机器人分解为多个节点，这些节点表示机械臂关节、夹爪驱动器、夹爪尖端以及工具中心点（TCP）等标注点。每个节点 n 都有一个类型 $\tau_n \in${终端、旋转、移动、点}，以及一个驱动指示器，用于区分驱动关节、终端关节和标注点。运动学拓扑通过节点之间的最短路径距离进行编码，并作为注意力偏置 [43]输入 Transformer。节点按照其运动学结构排序，并在左侧填充至支持的最大节点数，同时将末端执行器节点置于序列的最后位置。这在保留不同具身形态关节数量可变性的同时，为其提供了一致的索引方案。

关节本体感受。不同具身形态中的关节状态，其物理单位、限位和范围各不相同。我们将每个关节坐标以其限位中点为中心，并转换为统一的数值尺度：$p_n = \kappa_n(q_n - c_n)$，其中 q_n 是当前关节位置，c_n 是其限位范围的中点，κ_n 是依关节类型而定的单位因子。旋转关节以弧度表示，移动关节以分米表示。随后，将中心化后的标量扩展为傅里叶特征，并通过多层感知机进行编码。

相机坐标系几何定位。对于每个节点 n和相机 v，我们利用正向运动学和已知的相机外参计算其当前几何结构。该几何结构通过三个字段以度量三维相机坐标表示：$G_{n,v} = \left(g_{n,v}^{\text{center}}, g_{n,v}^{\text{axis}}, g_{n,v}^{\text{pose}}\right)$，其中$g_{n,v}^{\text{center}} \in \mathbb{R}^3$表示节点中心，$g_{n,v}^{\text{axis}} \in \mathbb{R}^3$ 是沿其运动轴方向距节点一个单位的点，而$g_{n,v}^{\text{pose}} \in \mathbb{R}^9$ 则使用连续的 6D 旋转表示法 [48]来表示平移和朝向。运动轴表示旋转关节的旋转轴，或移动关节的平移方向。

为所有节点提供中心点，为受驱动节点提供运动轴，为标注节点提供位姿。编码前，无法获取的字段会置零。在单机械臂操作中，工具中心点被表示为腕部的标注后代节点，因此其位姿可以纳入同一表示中。

相机坐标系坐标通过正弦特征和学习投影进行编码，然后与本体感知和离散节点属性相结合：

$$z_{n,v}^{G} = f_G\left(g_{n,v}^{\text{center}}, g_{n,v}^{\text{axis}}, g_{n,v}^{\text{pose}}\right) + f_P(p_n) + e_{j(n)}^{\text{slot}} + e_{\tau_n}^{\text{type}} + e_v^{\text{view}} + e^{\text{node}}, \quad (1)$$

其中，f_G 和 f_P 编码几何结构和本体感知，$e^{\text{slot}}_{j(n)}$ 标识节点填充后的槽位，$e^{\text{type}}_{\tau_n}$ 标识其类型。嵌入 e^{node} 用于区分节点词元与其他模态。我们为每个节点和相机视图构建一个词元。每个词元与来自相应相机的图像词元共享其视图嵌入 e^{view}_v，使注意力机制能够将机器人的几何结构与其视觉观测关联起来，并整合不同视图中的信息。

几何定位在相机的三维坐标系中进行。我们不会显式地将关节位置投影到图像像素上，也不会在投影位置采样视觉特征。共享的视点和视图嵌入为几何标记与视觉标记提供对应关系，而 Transformer 则学习它们之间的交互。

2.2. 可变自由度动作预测

不同机器人的关节数量、关节类型和可用运动方向各不相同。我们采用节点对齐动作表示，使输出遵循与输入运动学图相同的填充后顺序。这样，单个网络便可为不同具身形态预测动作，同时保持每个节点的几何结构与其控制信号之间的对应关系。

每个输出节点都由两个预测头处理：相对关节运动预测头和相对位姿预测头。关节运动预测头预测机器人关节坐标的变化，而位姿预测头则预测与相应末端执行器节点关联的相对末端执行器位姿。这两个预测头均通过流匹配进行训练，并且两者的预测结果都会用于机器人执行的控制。

我们使用与左侧填充节点槽位相关联的经验统计量对关节动作进行归一化；事实证明，这对于表示多样化的具身形态十分有效。动作输出会根据每个机器人的有效节点和动作量进行掩码处理。

架构与训练。视觉词元、语言词元、接地运动学节点和噪声动作词元由 Transformer 联合处理。模型使用流匹配进行训练 [24]。令 y 表示一个经过归一化的目标动作块，其中包含关节运动和末端执行器位姿更新。在流时间 $s \in [0,1]$，我们构造

$$x_s = (1-s)\epsilon + sy, \qquad \epsilon \sim \mathcal{N}(0, I), \tag{2}$$

其中，高斯噪声是在有效动作维度上采样的。

网络根据语言指令、视觉观测和场景 grounded 运动学图，预测该插值的速度。我们使用流匹配目标，在有效动作维度上进行训练：

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{s,\epsilon,y}\left[\left\| M \odot \left(v_\theta(x_s, s; \ell, \mathcal{I}_t, G_t) - (y - \epsilon)\right)\right\|_2^2\right], \tag{3}$$

其中，M 用于屏蔽无效或填充的动作维度，G_t 表示 grounded 运动学表示。模型联合学习关节和末端执行器位姿预测。在推理时，预测的动作块会同时向控制接口提供关节级运动和末端执行器位姿更新。执行动作块的前缀后，策略会根据更新后的观测重新规划。

2.3. 扩展具身形态

具身形态生成。在少量机器人上训练的策略可能会学到特定于这些运动学结构的操作策略，从而限制其适应陌生具身形态的能力。因此，我们以程序化方式生成一百万种机器人具身，涵盖不同的运动学结构、关节配置和连杆几何结构。每种具身形态均根据一个随机种子确定性生成，包括其形态、关节限位、执行器动力学、连杆几何形状、惯性和外观。这样，无需单独存储每个机器人的完整描述，即可确保其可复现。

我们会筛除生成的候选具身形态中存在自碰撞或不具备抓取可行性的个体。我们通过对随机采样物体上具有代表性的目标抓取姿态求解逆运动学，来评估抓取可行性。尽管这些具身形态是从旨在生成逼真机器人分布中采样得到的，所有生成的具身形态中仍有 84.4% 未通过自碰撞检查，而剩余具身形态中有 97.5% 未通过抓取检查。6自由度、7自由度和8自由度具身形态的总体通过率分别为 0.016%、0.299% 和 0.653%。筛除这些候选项，可以避免为无法执行简单操作任务的具身形态进行成本高昂的轨迹采集。

按具身形态生成轨迹。我们在 MolmoSpaces [22], 中采集操作轨迹：将每种可行的具身形态置于具有不同任务和初始配置的随机生成场景中。每种具身形态对应两个场景任务实例。

轨迹生成使用规划器，根据当前机器人的运动学约束提出局部可行的运动。在每一步，规划器都会考虑朝向下一个目标姿态的可行即时运动 [22]，而不显式规划特定于具身形态的长时域策略。与 MolmoSpaces 使用基于 Franka机器人假设计算的俯视姿态不同，我们会随机旋转抓取姿态，并允许机器人采用任何运动学上可行的姿态来形成支撑。我们筛选出成功回合，生成适用于各个机器人的可行操作行为演示。6、7 和 8 自由度具身形态的回合成功率分别稳定在 35.9%、34.8% 和 36.4%，而具身形态筛选的通过率则取决于自由度。成功回合会连同足够的仿真状态一并存储，以便重建中间步骤的环境。表格1总结了由此产生的

表1. **OMNIACT-1M 数据集统计**：共包含 1,029,605 条轨迹，涉及 800,196 种具身形态。

Task	回合	帧	平均长度（秒）	总长度（小时）
Pick	295.0k	26.3M	5.8	476
Open	227.7k	38.8M	11.2	707
PnP颜色	141.8k	25.4M	11.7	462
关闭	140.7k	20.2M	9.4	367
抓取与放置	127.7k	21.1M	10.8	385
PnP 相邻	96.6k	18.9M	12.8	343
总计	**1,029.6k**	**150.4M**	**9.6**	**2,740**

具身参数	p5	p50	p95
臂展（米）	0.86	1.26	1.74
分段长度（米）	0.00	0.14	0.41
手臂质量（千克）	3.9	10.2	18.3
连杆质量（千克）	0.06	0.55	4.72
关节活动范围（度）	94	205	322
关节刚度	37	1216	4145
刚度，肩部：手腕	1.8	4.3	45
关节阻尼比	1.1	5.1	14.6
夹爪安装旋转角 (°)	1.0	10.1	159
基座高度 (m)	0.57	0.70	0.84

本体属性	占比
机械臂混合 ≥2 连杆来源	70.5%
带前臂滚转的机械臂	65.1%
移动夹爪驱动器	49.6%
具有偏心限位的关节	49.4%
带万向节的机械臂	44.2%
单向运动关节	28.4%
混合两个机器人连杆的机械臂	24.0%
具有三条平行轴的机械臂	10.0%
可旋转超过一整圈的关节	1.6%

语料库。

使用固定种子的随机数生成器，以可复现的方式留出 5 % 的具身形态，并在每个任务中留出 5% 的场景。只有当一条轨迹的具身形态和场景均未被留出时，才会用于训练。当某个具身形态或场景出现在至少一条被纳入的训练轨迹中时，就称其为已见 。执行此流程后，18.3% 的具身形态和 10.9% 的场景最终会在训练中未见，无论是直接还是间接。所有轨迹中有 14.5% 同时包含未见过的具身形态和未见过的场景。具身形态和场景均未见过的子集称为 OMNIACT-BENCH，用于基准测试。

3. 实验

主要结论如下：(a) 在基线方法见过的机器人上的仿真中，尽管 OMNIACT 未见过机器人本身，但其表现优于或不逊于现有方法；(b) 在未见过的机器人上的仿真中，尽管步数预算更小，OMNIACT 仍优于所有基线方法，包括特定具身形态和不特定具身形态的策略，以及通用视觉语言模型 GPT-6 Astra；(c) 在真实世界部署中，针对现有方法所专门适配的机器人，OMNIACT 的表现优于或不逊于这些现有方法；(d) OMNIACT 会根据具身形态和当前任务调整其运动策略，而不是在不同机体之间重定向同一种规范抓取姿势。

训练数据。 我们在 MolmoSpaces 场景中训练我们的语料库，其中包含 100 万个通过程序生成的机械臂（第 2.3节），执行操控 [22]。

基线。 我们将其与前沿策略进行比较，包括$\pi_{0.5}$ [19]、DreamZero [42], MolmoAct2 [13], GPT-6 Astra（最大思考模式），以及专精于具身形态泛化的模型 UMI [8] 和 UMI-on-Air [16]。为解决以下可能性：与 UMI 的单视角设置相比，多视角设置可能提供更丰富的视觉上下文，我们加入一个采用三视角相机设置的 UMI 变体作为另一条基线。我们使用与训练相同的模拟轨迹训练 UMI，但采用 UMI 自身的设置，以衡量有助于实现特定输入条件设定的架构选择所带来的影响（第 2.1 节）。在真实世界部署中，我们还评估 LAP-3B [45], 这一零样本跨具身形态迁移模型，使用其发布的检查点。

基准测试。 我们在三种设置下评估策略：i）在现有的 MolmoSpaces 基准测试上进行评估，该基准测试使用的是基线策略分布内的机器人；ii）在 OMNIACT-BENCH 上进行评估，这是一个基于留出的训练数据构建的基准测试，因此对所有策略而言都是未见过的；以及 iii）在现实世界任务上进行评估。

实现细节。 OMNIACT 是一个拥有 4 亿参数的流匹配策略，采用固定的 12 槽位节点与动作布局。它观测由冻结的 SigLIP 2 图像编码器编码的 RGB视图 [38], 以及由冻结的 Qwen2.5-VL 文本主干网络编码的语言 [2], , 并使用第 2.1 节所述的逐节点具身表示。训练样本为 32 帧窗口，动作预测时域为 31 步。我们以 128 的有效批量大小进行训练，学习率为10^{-3}，共训练 520K 步。评估时，策略会预测一个包含 31 步的动作块，并在每个动作块中执行前 8 步。

3.1. 在仿真中对已见机器人进行评估

我们在仿真中使用 MolmoSpaces 基准测试评估 OMNIACT，该基准采用 DROID 配置中的 Franka Panda。所有基线在训练期间都见过 Franka机器人，而 OMNIACT 则没有。

OMNIACT 在各模型擅长的具身形态上，表现与具身形态专用模型不相上下。 表 2（完整场景）报告了在相同的 500 步上限下，对 2,622 个留出回合的成功率。OMNIACT 的总体成功率达到 69.1%，MolmoBot 为 63.0%，而经 DROID 训练的策略则低于 30%；它在六个任务类别中的四个类别中领先，在任务对放置位置或接近方式有所限制时优势最大：例如，在将物体放置于参考物体旁边的任务中，领先 MolmoBot +15.5 个百分点。

OMNIACT 能够抵御对象、光照和提示格式的变化，在适应相机和机器人姿态变化方面则与基线的表现大致相当。 $\pi_{0.5}$ 和 MolmoAct2 的得分都低于 MolmoSpaces报告的成功率。为了解这一差异，我们注意到，自发布以来，MolmoSpaces的场景难度有所增加，尤其是涉及定位对象的场景 [11]。因此，我们将基准测试标记为表2（完整场景），并与表 2（简单场景）进行对比。除了加入更难抓取的对象和任务外，变化还包括更多样且随机化的光照、相机、机器人姿态和提示。为了解 $\pi_{0.5}$ 和 MolmoAct2 性能下降的原因，我们逐步改变抓取任务中的各个变量，并

表 2. **在未见过的真实机器人 OMNIACT 上评估与其他针对该机器人训练的模型**。成功率根据 2,622 个留出的 MolmoSpaces 回合计算，采用 500 步上限。（**PnPC** 和 **PnPN** 分别表示带有颜色约束的抓取与放置，以及以“紧邻”为目标的抓取与放置；这两类任务均未出现在 MolmoSpaces 的简单场景评估中。）

模型	Pick $n=647$	PnP $n=474$	PnPC $n=394$	PnPN $n=304$	Open $n=551$	近距离 $n=252$	All $n=2,622$
简单场景							
$\pi_{0.5}$-DROID	36.4 ±1.5	13.6 ±1.1	–	–	22.7 ±1.3	65.1 ±1.5	33.8 ±0.7
MolmoAct2-DROID	48。3 ±1.6	23。3 ±1。3	–	–	11.7 ±1.0	71.3 ±1.4	37.9 ±0.8
完整场景							
$\pi_{0.5}$-DROID	29.5 ±1.8	17.2 ±1.7	14.1 ±1.8	16.0 ±2.1	43.1 ±2.1	73.4 ±2.8	28.2 ±0.9
DreamZero-DROID	5.9 ±0.9	2.1 ±0.7	2.5 ±0.8	3.0 ±1.0	20.5 ±1.7	66.3 ±3.0	13.2 ±0.7
MolmoAct2-DROID	27.4 ±1.8	18.6 ±1.8	10.4 ±1.5	12.8 ±1.9	19.0 ±1.7	**76.3** ±2.7	22.5 ±0.8
MolmoBot	**90.4** ±1.2	78.9 ±1.9	78.7 ±2.1	42.4 ±2.8	18.3 ±1.6	61.1 ±3.1	63.0 ±0.9
OMNIACT	89.2 ±1.2	**79.3** ±1.9	**79.2** ±2.0	**57.9** ±2.8	**43.2** ±2.1	52.8 ±3.1	**69.1** ±0.9

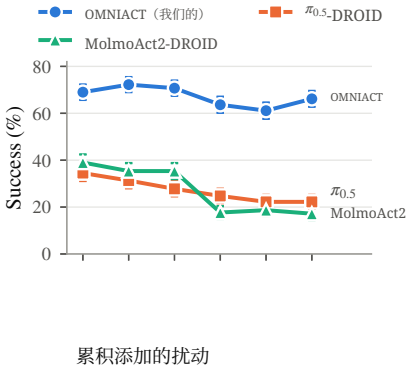


图3。**Molmo- Spaces Pick 上的累积鲁棒性**。随着输入条件中越来越多的方面偏离简单版本，最先进的策略 $\pi_{0.5}$ 和 MolmoAct2 的性能均有所下降。

表3。**在OMNIACT-BENCH 上对未见过的机器人进行评估**。**PnP**、**PnPN**和**PnPC** 分别表示抓取与放置、在参考物体旁抓取与放置，以及按颜色抓取与放置。**Common** 汇总了所有受评估模型均支持的任务类别。**步数** 指每回合步数上限。GPT-6 Astra 采用最高思考等级运行。GPT-6 Astra 每五步重新规划一次，模型调用预算为 200 次，步数上限为 1，000；200 步和 500 步的行会截断这些相同的回合。省略 GPT-6 的部分任务是因为 API 令牌有限。

模型	步数	Pick	PnP	PnPN	PnPC	Open	关闭	常见	All
MolmoAct2-YAM	500	0.2 ±0.2	2.9 ±1.0	0.0 ±0.0	0.8 ±0.6	3.0 ±0.8	20.7 ±2.3	1.8 ±0.4	4.4 ±0.5
DreamZero（DROID）	500	0.7 ±0.4	2.5 ±0.9	1.6 ±0.9	1.6 ±0.8	13.4 ±1.6	40.3 ±2.8	5.5 ±0.6	10.0 ±0.7
MolmoAct2-DROID	500	1.5 ±0.5	2.9 ±1.0	2.6 ±1.2	2.0 ±0.9	19.4 ±1.9	60.0 ±2.8	8.0 ±0.8	14.7 ±0.8
$\pi_{0.5}$-DROID	500	1.9 ±0.6	5.1 ±1.3	6.3 ±1.8	2.7 ±1.0	23。6 ±2.0	69.2 ±2.6	10.1 ±0.9	17.8 ±0.9
UMI	500	23.4 ±1.8	7.6 ±1.6	6.8 ±1.8	6.7 ±1.6	24.5 ±2.1	45.2 ±2.9	20.3 ±1.1	21.1 ±0.9
UMI-on-Air	500	24.9 ±1.9	7.6 ±1.6	9.4 ±2.1	6.3 ±1.5	24.1 ±2.1	44.3 ±2.8	20.8 ±1.1	21.4 ±0.9
UMI（所有视角）	500	65.4 ±2.1	48.7 ±3.0	39.8 ±3.5	47.5 ±3.1	60.6 ±2.4	68.9 ±2.7	60.1 ±1.4	57.9 ±1.1
GPT-6 Astra	200	6.9 ±4.7	0.0 ±0.0	–	–	17.2 ±7.0	–	9.6 ±3.4	9.6 ±3.4
GPT-6 Astra	500	44.8 ±9.2	26.7 ±11.4	–	–	44.8 ±9.2	–	41.1 ±5.8	41.1 ±5.8
GPT-6 Astra	1000	48.3 ±9.3	46。7 ±12.9	–	–	65.5 ±8.8	–	54.8 ±5.8	54.8 ±5.8
OMNIACT	200	58.7 ±2.1	22.0 ±2.5	20.9 ±2.9	22.7 ±2.6	39.4 ±2.4	56.7 ±2.8	43.9 ±1.4	40。9 ±1.1
OMNIACT	500	**70。4** ±2.0	**59.6** ±2.9	**48.7** ±3.6	**54.5** ±3.1	**60.9** ±2.3	**71.8** ±2.6	**64.7** ±1.4	**63.0** ±1.1

将结果与 OMNIACT 在图3中进行对比，图中显示 OMNIACT 的波动处于误差范围内，而 $\pi_{0.5}$和 MolmoAct2 的表现则有所下降。

3.2. 在仿真中对未见过的机器人进行评估

接下来，我们要测试基线方法推广到新型实体的效果。我们使用留出的 OMNIACT-BENCH基准测试（第2.3节）。

OMNIACT 可泛化至模拟环境中未见过的具身形态。 我们在基准测试中随机抽取的 2000 个回合子集上进行的 rollout成功率 如表 3 所示。结果表明，在 rollout 步数相同的情况下，OMNIACT 的表现优于基线方法，在六个任务类别中名列前茅；其总体成功率为 63.0%，比次优方法——使用所有相机视角的 UMI——高出 5.1 个百分点。除了策略之外，通用 GPT-6 Astra 模型在随机具身形态上同样表现不佳：在双方均进行评估的任务类别中，步数相同的情况下落后 OMNIACT 23.6 个百分点（41.1% 对 64.7%）。我们的实证观察表明，GPT-6 Astra 凭借其语言能力和多模态推理能力，会持续尝试不同的解决方案；然而，即使步数增加一倍，其表现仍落后 OMNIACT 9.9 个百分点。

3.3. 在真实机器人上进行评估

我们将 OMNIACT 部署在两台实体机械臂上，未进行微调：一台是 Franka Panda，即所有基线训练时使用的机器人；另一台是 OpenYAM 机械臂，只有 MolmoAct2-BimanualYAM 曾在其上训练过，而且它配备了一个所有策略都未见过的细尖法兰夹爪。

Franka。 我们在 DROID 配置下部署于 Franka Panda 上，

表4. **真实世界评估机械臂 OpenYAM 和 Franka 在训练语料库中的位置**。此外，我们的 OpenYAM 机械臂配备了训练期间未见过的细尖法兰夹爪。 P_k 表示百分位数。

	OpenYAM		Franka		
	数值	P_k	数值	P_k	中位数
臂展（米）	0.76	1.2	1.21	43.3	1.26
手臂质量（kg）	3.6	3.7	18.3	94.9	10.2
关节刚度	30	3.3	3500	90.0	1216
阻尼增益	1.75	2.0	350	85.4	102
电枢	0.017	15.6	0.100	44.8	0.130
力限制	19	1.2	100 ,000	82.2	2293
刚度与惯性比	3400	11.2	35,000	95.2	11,300
到最近具身的距离	0.545	98.9	0.065	15.8	0.116

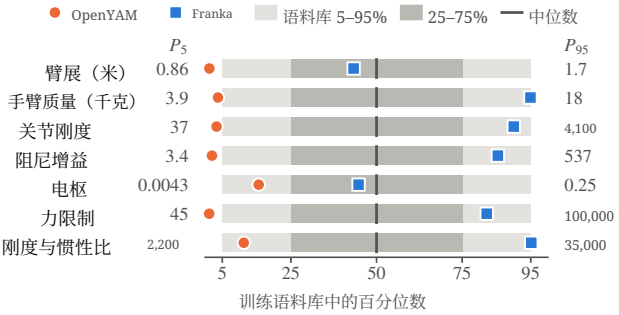


图4. **训练语料库中每种物理量对应的各真实世界机械臂百分位数** 见表4。OpenYAM 位于分布边缘，而 Franka 则介于分布范围之内与分布边缘之间。

表5. **在实体机械臂 Franka 和 OpenYAM 上进行真实世界部署**。在 MolmoAct2 这一行中，使用 MolmoAct2 针对每种具身形态的专用模型，以测量其最佳性能。**不适用**表示由于缺乏对某种具身形态的支撑，模型在该具身形态的所有尝试中均失败或表现得完全不安全。

	成功率				中位数
模型	Franka	OpenYAM	总体（%）	参数量	步数
见过部分或全部目标具身实体					
MolmoBot	8/20	*N/S*	8/40（20%）	4B	142
LAP-3B	8/20	0/20	8/40（20%）	3B	241
$\pi_{0.5}$	13/20	*N/S*	13/40（33%）	3B	165
DreamZero	**15/20**	*N/S*	15/40 (38%)	14B	194
MolmoAct2	14/20	7/20	21/40 (53%)	5B	356
未见目标具身实体					
UMI	1/20	0/20	1/40 (3%)	0.6B	152
UMI-on-Air	1/20	0/20	1/40 (3%)	0.6B	298
UMI（所有视角）	11/20	0/20	11/40 (28%)	0.6B	166
OMNIACT	14/20	**13/20**	**27/40 (68%)**	0.4B	177

该配置中的机器人也是所有基线的训练对象。对于 MolmoAct2 基线，其 DROID 专用变体以确保最大程度的兼容性。

OpenYAM。 接下来，我们在一台实体 OpenYAM 机械臂上进行部署，该机械臂配备了细尖法兰夹爪。对于 MolmoAct2 基线，我们部署 BimanualYAM 变体，并启用两台 YAM 机械臂及其标准相机配置，以符合模型的预期。

OpenYAM 在我们的训练数据中所处的区域相对稀疏，且训练期间未见过其细尖法兰夹爪。表 4 和图 4 显示，其许多关键特征都超出了第 5/95 百分位数的范围。

在未见过的真实机器人上，OMNIACT 的表现优于其他未针对具身形态进行专门训练的模型，并且与针对具身形态专门训练的模型相当。我们在物体抓取与放置任务上测量成功率，改变物体和目标的位置，使其位于具身形态的臂展范围内及边界处。我们针对每个机器人测量20种初始设置；在每种设置中，每个策略都从相同的物体位置开始，并采用1000步时域（表5）。OMNIACT以零样本方式部署；在真实或模拟的任一机器人上都未进行微调。

在Franka上，OMNIACT在20种设置中的14种上成功，这与MolmoAct2-DROID相同，也与使用该机器人真实DROID数据训练的策略相当。在OpenYAM上，它在20种设置中的13种上成功；而专门在YAM上训练的MolmoAct2仅在7种设置上成功。根据经验，当物体相比容器距离机器人更远，或部分隐藏在容器后方时，MolmoAct2-BimanualYAM经常失败。尽管LAP-3B在I2RT YAM上的成功率为30%，但它在OpenYAM上无法成功，并且在抓取阶段表现不佳。

值得注意的是，尽管 UMI（所有视角）在 11/20 Franka 上取得了，但在所有 OpenYAM 设置下均失败了，因为这些设置处于训练分布的边缘。除场景约束的运动学图（第 2.1节）之外，UMI（所有视角）接收的训练数据与我们的模型相同。

3.4. 运动策略分析

在评估过程中，我们注意到，当物体拾取位置靠近容器时，OMNIACT 有时会利用 OpenYAM 通向夹爪的长连杆，仅通过绕 Z轴旋转夹爪来够到容器，而不是先抬起再放下。

这很可能是我们数据生成方法带来的副作用：我们允许不同阶段的位姿采用不同的方向；因此，当容器就在附近时，旋转是一种可行的策略，甚至可能更加直接。由于语料库中的每条轨迹都能由其对应的具身形态执行，因此数据并未将抓取方式固定为一种再重定向到不同具身形态的方式：根据身体形态和物体的不同，成功的抓取可能从上方、侧面或介于两者之间的方向接近。我们将考察 OMNIACT 是否能够灵活地根据具身形态和任务调整策略。

设置。我们在 OpenYAM（一种标准尺寸的机械臂）和 Lite6（一种臂展相对有限的机械臂）上进行这项实验。画面中可以看到一把刀、一个立方体和一个碗。刀放在表面上，很难从上方拾取。

在真实任务中，OMNIACT 会根据任务调整其运动策略。图 5（a）展示了同一部署中同一 OpenYAM 机械臂的两种抓取方式。对于立方体（上图），OMNI-ACT 从上方接近，并垂直闭合夹爪。对于刀（下图），它则调整腕部的俯仰角，直到夹爪与桌面齐平，然后从侧面夹住刀刃；此时刀身能够以最大接触面积嵌入指尖之间。该策略将腕部视为一个自由度，并选择物体所需的抓取方向，而不是始终保持固定的自上而下姿态。

在真实机器人上，OMNIACT 会根据具身形态调整其运动策略。图 5（b）固定任务并改变机器人本体。当物体放置在靠近机器人处时，OpenYAM 和 Lite6 通常都会从上方接近。然而，当物体放置在 Lite6 相对较短臂展的边缘处（下图）时，它会从侧面接近，使夹爪大约悬在距桌面 20° 的位置；而在 OpenYAM（上图）上，它则从上方接近，因为 OpenYAM 可以轻松达到相同距离。此外，当尝试从侧面拾取时，OpenYAM 和 Lite6 的接近角度略有不同（图 5（a）（b）下图），使两台机器人都能尽可能保持平伏，同时不会对桌面施加过大的力；过大的力会在仿真中触发夹爪碰撞，并在真实机器人的驱动程序中触发碰撞停止。

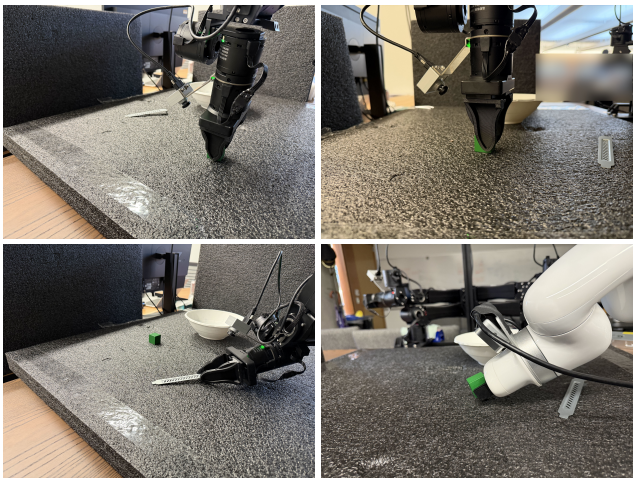
总体而言，我们可以看到 OMNIACT 会根据具身形态和当前任务调整其运动策略，而不是将一种规范抓取姿势重定向到不同身体上，导致其中一种身体被迫摆出不可行的姿势。

3.5. 消融研究

增加训练具身形态的数量有所帮助。我们将生成的训练具身形态下采样至语料库的一半和八分之一（分别为 400K 和 100K），同时固定架构和训练计划（表 6 和图 6）。成功率随之单调下降：−6.5%（一半时），以及 −26.8%（八分之一时）；每次 rollout 的步数上限为 500。

场景 grounded 运动学图和相机视角输入对成功率至关重要。接着，我们在数据固定的情况下移除输入（表 6）。移除逐节点运动学图标记会导致成功率下降 5.1 个百分点：虽然可以从像素和本体感知中部分推断出身体信息，但无法

充分展示了通过相机锚定节点而非机器人坐标系本体感知来提供场景锚定图的优势。进一步移除外部视角，仅保留腕部观测，还会导致性能下降 36.8%。



（a）相同具身形态，不同任务。 (b) 相同任务，不同具身形态。

图 5。**OMNIACT 为任务和具身形态选择其抓取方向**。（a）相同具身形态，不同任务：立方体采用自上而下的抓取，刀采用侧向抓取。（b）相同任务，不同具身形态：OpenYAM 采用自上而下的抓取，Lite6 采用侧向抓取。

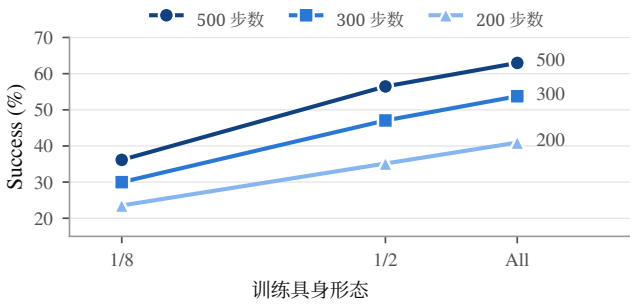


图 6。**成功率与训练具身形态数量的关系**。与表6中相同的留出回合上的成功率，在每个 rollout步数上限下统计。标准误差（≤ 1.1 点）小于标记的大小。

表 6。**消融研究**。在相同留出回合和超参数下的成功率。

	rollout步数为.....时的成功率 %		
模型	**200**	**300**	**500**
OMNIACT	40.9 ±1.1	53.8 ±1.1	63.0 ±1.1
配备 1/2 种具身形态	−5.8 ±1.1	−6.7 ±1.2	−6.5 ±1.2
具身形态的 1/8	−17.4 ±1.2	−23.8 ±1.2	−26.8 ±1.3
无场景接地	−4.6 ±1.1	−6.0 ±1.2	−5.1 ±1.2
不含场景接地 & 不含外部视角	−29.0 ±1.1	−37.9 ±1.2	−41.9 ±1.3

4. 讨论

我们发现，扩大多样化的机器人实体规模，并提供场景 grounded 运动学图，能够提升策略在现实世界任务中对未见过的机器人的泛化能力，从而实现

针对当前任务制定适合特定身体的运动策略。扩大训练具身形态的数量，能够提升策略在留出的模拟具身形态上的表现。场景 grounded 运动学图也能带来小幅提升；但当机器人与训练分布的差异进一步增大时，图的贡献似乎会显著增大（例如 OpenYAM）。这些结果表明，扩大具身形态规模有助于覆盖与训练对象相似的身体，而运动学图则能将策略迁移到处于该分布边缘的身体上。

一个尚待解决的问题是进一步扩展具身形态的类型，包括双臂、人形机器人，以及工具、设备和车辆等日常物品。

致谢

本研究部分由美国国家科学基金会通过项目 IIS-2211133 资助。作者声明不存在其他利益冲突。

参考文献

[1] Bo Ai、Liu Dai、Nico Bohlinger、Dichen Li、Tongzhou Mu、Zhanxin Wu、K. Fay、Henrik I. Christensen、Jan Peters 和 Hao Su。迈向机器人运动中的具身规模定律，2025年。2[2] Shuai Bai 、 Keqin Chen 、 Xuejing Liu 、 Jialin Wang 、 Wenbin Ge、Sibo Song、Kai Dang、Peng Wang、Shijie Wang、Jun Tang、Humen Zhong、Yuanzhi Zhu、Mingkun Yang、Zhaohai Li、Jianqiang Wan、Pengfei Wang、Wei Ding、Zheren Fu、Yiheng Xu、Jiabo Ye、Xi Zhang、Tianbao Xie、Zesen Cheng、Hang Zhang、Zhibo Yang、Haiyang Xu 和 Junyang Lin。Qwen2.5- vl 技术报告，2025年。3， 5[3] Homanga Bharadhwaj 、 Roozbeh Mottaghi 、 Abhinav Gupta 和 Shubham Tulsiani。Track2act：从互联网视频预测点轨迹，实现可泛化的机器人操作，2024年。 2[4] Kevin Black、Noah Brown、Danny Driess、Adnan Esmail、Michael Equi、Chelsea Finn、Niccolo Fusai、Lachy Groom、Karol Hausman、Brian Ichter、Szymon Jakubczak、Tim Jones、Liyiming Ke、Sergey Levine、Adrian Li-Bell、Mohith Mothukuri、Suraj Nair、Karl Pertsch、Lucy Xiaoyang Shi、James Tanner、Quan Vuong、Anna Walling、Haohuan Wang 和 Ury Zhilinsky。 π_0：用于通用机器人控制的视觉-语言-动作流模型，2026年。 2[5] Qingwen Bu、Yanting Yang、Jisong Cai、Shenyuan Gao、Guanghui Ren、Maoqing Yao、Ping Luo 和 Hongyang Li。Univla：借助以任务为中心的潜在动作学习在任何地方执行动作，2025年。[6] Lawrence Yunliang Chen、Kush Hari、Karthik Dharmarajan、Chenfeng Xu、Quan Vuong 和 Ken Goldberg。Mirage：通过跨绘制实现跨具身零样本策略迁移，2024年。[7] Tao Chen、Adithyavairavan Murali 和 Abhinav Gupta。面向多机器人迁移学习的硬- 件条件策略，2019年。 2[8] Cheng Chi、Zhenjia Xu、Chuer Pan、Eric Cousineau、Ben- jamin Burchfiel、Siyuan Feng、Russ Tedrake 和 Shuran Song。通用操作接口：在真实环境中的机器人教学，无需真实环境中的机器人，2024年。 5

[9] Embodiment Collaboration，Abby O'Neill，Abdul Rehman，Abhinav Gupta，Abhiram Maddukuri，Abhishek Gupta，Abhishek Padalkar，Abraham Lee，Acorn Pooley，Agrim Gupta，Ajay Mandlekar，Ajinkya Jain，Albert Tung，Alex Bewley，Alex Herzog，Alex Irpan，Alexander Khazatsky，Anant Rai，Anchit Gupta，Andrew Wang，Andrey Kolobov，Anikait Singh，Animesh Garg，Aniruddha Kembhavi，Annie Xie，Anthony Brohan，Antonin Raffin，Archit Sharma，Arefeh Yavary，Arhan Jain，Ashwin Balakrishna，Ayzaan Wahid，Ben Burgess-Limerick，Beomjoon Kim，Bernhard Sch¨olkopf，Blake Wulfe，Brian Ichter，Cewu Lu，Charles Xu，Charlotte Le，Chelsea Finn，Chen Wang，Chenfeng Xu，Cheng Chi，Chenguang Huang，Christine Chan，Christopher Agia，Chuer Pan，Chuyuan Fu，Coline Devin，Danfei Xu，Daniel Mor- ton，Danny Driess，Daphne Chen，Deepak Pathak，Dhruv Shah，Dieter B¨uchler，Dinesh Jayaraman，Dmitry Kalash- nikov，Dorsa Sadigh，Edward Johns，Ethan Foster，Fangchen Liu，Federico Ceola，Fei Xia，Feiyu Zhao，Felipe Vieira Fru- jeri，Freek Stulp，Gaoyue Zhou，Gaurav S. Sukhatme，Gautam Salhotra，Ge Yan，Gilbert Feng，Giulio Schiavi，Glen Berseth，Gregory Kahn，Guangwen Yang，Guanzhi Wang，Hao Su，Hao-Shu Fang，Haochen Shi，Henghui Bao，Heni Ben Amor，Henrik I Christensen，Hiroki Furuta，Homanga Bharadhwaj，Homer Walke，Hongjie Fang，Huy Ha，Igor Mordatch，Ilija Radosavovic，Isabel Leal，Jacky Liang，Jad Abou-Chakra，J ae- hyungKim，JaimynDrake，JanPeters，JanSchneider，Jasmine Hsu，Jay Vakil，Jeannette Bohg，Jeffrey Bingham，Jeffrey Wu，Jensen Gao，Jiaheng Hu，Jiajun Wu，Jialin Wu，Jiankai Sun，Jianlan Luo，Jiayuan Gu，Jie Tan，Jihoon Oh，Jimmy Wu，Jingpei Lu，Jingyun Yang，Jitendra Malik，Jo˜ao Silv´erio，Joey Hejna，Jonathan Booher，Jonathan Tompson，Jonathan Yang，Jordi Salvador，Joseph J. Lim，Junhyek Han，Kaiyuan Wang，Kanishka Rao，Karl Pertsch，Karol Hausman，Keegan Go，Keerthana Gopalakrishnan，Ken Goldberg，Kendra Byrne，Kenneth Oslund，Kento Kawaharazuka，Kevin Black，Kevin Lin，Kevin Zhang，Kiana Ehsani，Kiran Lekkala，Kirsty Ellis，Krishan Rana，Krishnan Srinivasan，Kuan Fang，Kunal Pratap Singh，Kuo-Hao Zeng，Kyle Hatch，Kyle Hsu，Laurent Itti，Lawrence Yunliang Chen，Lerrel Pinto，李飞飞，Liam Tan，Linxi "Jim" Fan，Lionel Ott，Lisa Lee，Luca Weihs，Magnum Chen，Marion Lepert，Marius Memmel，Masayoshi Tomizuka，Masha Itkina，Mateo Guaman Castro，Max Spero，Maxim- ilian Du，Michael Ahn，Michael C. Yip，Mingtong Zhang，Mingyu Ding，Minho Heo，Mohan Kumar Srirama，Mohit Sharma，Moo Jin Kim，Muhammad Zubair Irshad，Naoaki Kanazawa，Nicklas Hansen，Nicolas Heess，Nikhil J Joshi，Niko Suenderhauf，Ning Liu，Norman Di Palo，Nur Muham- mad Mahi Shafiullah，Oier Mees，Oliver Kroemer，Osbert Bastani，Pannag R Sanketi，Patrick "Tree" Miller，Patrick Yin，Paul Wohlhart，Peng Xu，Peter David Fagan，Peter Mitrano，Pierre Ser manet，Pieter Abbeel，Priya Sundaresan，Qiuyu Chen，Quan Vuong，Rafael Rafailov，Ran Tian，Ria Doshi，Roberto Mart´ın-Mart´ın，Rohan Baijal，Rosario Scalise，Rose Hendrix，Roy Lin，Runjia Qian，Ruohan Zhang，Russell Men- donca，Rutav Shah，Ryan Hoque，Ryan Julian，Samuel Bus- tamante，Sean Kirmani，Sergey Levine，Shan Lin，Sherry

Moore, Shikhar Bahl, Shivin Dass, Shubham Sonawani, Shubham Tulsiani, Shuran Song, Sichun Xu, Siddhant Haldar, Siddharth Karamcheti, Simeon Adebola, Simon Guist, Soroush Nasiriany, Stefan Schaal, Stefan Welker, Stephen Tian, Subramanian Ramamoorthy, Sudeep Dasari, Suneel Belkhale, Sungjae Park, Suraj Nair, Suvir Mirchandani, Takayuki Osa, Tanmay Gupta, Tatsuya Harada, Tatsuya Matsushima, Ted Xiao, Thomas Kollar, Tianhe Yu, Tianli Ding, Todor Davchev, Tony Z. Zhao, Travis Armstrong, Trevor Darrell, Trinity Chung, Vidhi Jain, Vikash Kumar, Vincent Vanhoucke, Vitor Guizilini, Wei Zhan, Wenxuan Zhou, Wolfram Burgard, Xi Chen, Xiangyu Chen, Xiaolong Wang, Xinghao Zhu, Xinyang Geng, Xiyuan Liu, Xu Liangwei, Xuanlin Li, Yansong Pang, Yao Lu, Yecheng Jason Ma, Yejin Kim, Yevgen Chebotar, Yifan Zhou, Yifeng Zhu, Yilin Wu, Ying Xu, Yixuan Wang, Yonatan Bisk, Yongqiang Dou, Yoonyoung Cho, Youngwoon Lee, Yuchen Cui, Yue Cao, Yueh-Hua Wu, Yujin Tang, Yuke Zhu, Yunchu Zhang, Yunfan Jiang, Yunshuang Li, Yunzhu Li, Yusuke Iwasawa, Yutaka Matsuo, Zehan Ma, Zhuo Xu, Zichen Jeff Cui, Zichen Zhang, Zipeng Fu, and Zipeng Lin. Open x-embodiment: 机器人学习数据集和 rt-x 模型，2025年。 2

[10] AgiBot World Contributors、Qingwen Bu、Jisong Cai、Li Chen、Xiuqi Cui、Yan Ding、Siyuan Feng、Shenyuan Gao、Xindong He、Xuan Hu、Xu Huang、Shu Jiang、Yuxin Jiang、Cheng Jing、Hongyang Li、Jialu Li、Chiming Liu、Yi Liu、Yuxiang Lu、Jianlan Luo、Ping Luo、Yao Mu、Yuehan Niu、Yixuan Pan、Jiangmiao Pang、Yu Qiao、Guanghui Ren、Cheng Ruan、Jiaqi Shan、Yongjian Shen、Chengshi Shi、Mingkang Shi、Modi Shi、Chonghao Sima、Jianheng Song、Huijie Wang、Wenhao Wang、Dafeng Wei、Chengen Xie、Guo Xu、Junchi Yan、Cunbiao Yang、Lei Yang、Shukai Yang、Maoqing Yao、Jia Zeng、Chi Zhang、Qinglin Zhang、Bin Zhao、Chengyue Zhao、Jiaqi Zhao 和 Jianchao Zhu。Agibot World Colosseo：面向可扩展智能具身系统的大规模操作平台，2026年。

[11] Abhay Deshpande、Maya Guru、Rose Hendrix、Snehal Jauhri、Ainaz Eftekhar、Rohun Tripathi、Max Argus、Jordi Salvador、Haoquan Fang、Matthew Wallingford、Wilbert Pumacay、Yejin Kim、Quinn Pfeifer、Ying-Chun Lee、Piper Wolters、Omar Rayyan、Mingtong Zhang、Jiafei Duan、Karen Farley、Winson Han、Eli Vanderbilt、Dieter Fox、Ali Farhadi、Georgia Chalvatzaki、Dhruv Shah 和 Ranjay Krishna。Moltbot：大规模仿真助力零样本操作，2026年。5

[12] Ria Doshi、Homer Walke、Oier Mees、Sudeep Dasari 和 Sergey Levine。扩展跨本体学习：一个策略用于操作、导航、运动和航空，2024年。[13] Haoquan Fang、Jiafei Duan、Donovan Clay、Sam Wang、Shuo Liu、Weikai Huang、Xiang Fan、Wei-Chuan Tsai、Shirui Chen、Yi Ru Wang、Shanli Xing、Jaemin Cho、Jae Sung Park、Ainaz Eftekhar、Peter Sushko、Karen Farley、Angad Wadhwa、Cole Harrison、Winson Han、Ying-Chun Lee、Eli VanderBilt、Rose Hendrix、Suveen Ellawela、Lucas Ngoo、Joyce Chai、Zhongzheng Ren、Ali Farhadi、Dieter Fox 和 Ranjay Kr- ishna。Molmoact2：面向真实世界部署的动作推理模型，2026年。5[14] Raktim Gautam Goswami、Prashanth Krishnamurthy、Yann LeCun、Farshad Khorrami。Robopepp：通过嵌入预测式预训练进行基于视觉的机器人姿态与关节角度估计，2025年。 [15] Agrim Gupta、Linxi Fan、Surya Ganguli 和 李飞飞。Metamorph：使用 Transformer 学习通用控制器，2022年。 2 [16] Harsh Gupta、Xiaofeng Guo、Huy Ha、Chuer Pan、Muqing Cao、Dongjae Lee、Sebastian Scherer、Shuran S ong 和 Guanya Shi。Umi-on-air：为本体无关的视觉运动策略提供本体感知引导，2026年。 5[17] Huy Ha、C. Karen Liu 和 Shuran Song。Transformer 变换器：用于运动条件机器人协同设计的统一模型，2026年。[18] Wenlong Huang、Chen Wang、Yunzhu Li、Ruohan Zhang 和 李飞飞。Rekep：用于机器人操作的关系关键点约束时空推理，2024年。 2[19] Physical Intelligence、Kevin Black、Noah Brown、James Darpinian、Karan Dhabalia、Danny Driess、Adnan Esmail、Michael Equi、Chelsea Finn、Niccolo Fusai、Manuel Y. Galliker、Dibya Ghosh、Lachy Groom、Karol Hausman、Brian Ichter、Szymon Jakubczak、Tim Jones、Liyiming Ke、Devin LeBlanc、Sergey Levine、Adrian Li-Bell、Mohith Mothukuri、Suraj Nair、Karl Pertsch、Allen Z. Ren、Lucy Xiaoyang Shi、Laura Smith、Jost Tobias Springenberg、Kyle Stachowicz、James Tanner、Quan Vuong、Homer Walke、Anna Walling、Haohuan Wang、Lili Yu 和 Ury Zhilinsky。 $\pi_{0.5}$：一种具有开放世界泛化能力的视觉-语言-动作模型，2025年。5[20] Byungjun Kim、Taeksoo Kim、Hyunsoo Cha 和 Hanbyul Joo。通过机器人渲染构建机器人分解式世界模型，2026年。[21] Moo Jin Kim、Karl Pertsch、Siddharth Karamcheti、Ted Xiao、Ashwin Balakrishna、Suraj Nair、Rafael Rafailov、Ethan Foster、Grace Lam、Pannag Sanketi、Quan Vuong、Thomas Kollar、Benjamin Burchfiel、Russ Tedrake、Dorsa Sadigh、Sergey Levine、Percy Liang 和 Chelsea Finn。Openvla：一种开源视觉-语言-动作模型，2024年。2[22] Yejin Kim 、 Wilbert Pumacay 、 Omar Rayyan 、 Max Argus、Winson Han、Eli VanderBilt、Jordi Salvador、Abhay Deshpande、Rose Hendrix、Snehal Jauhri、Shuo Liu、Nur Muhammad Mahi Shafiullah、Maya Guru、Ainaz Eftekhar、Karen Farley、Donovan Clay、Jiafei Duan、Arjun Guru、Piper Wolters、Alvaro Herrasti、Ying-Chun Lee、Georgia Chalvatzaki、Yuchen Cui、Ali Farhadi、Dieter Fox 和 Ranjay Krishna。MolmoSpaces：面向机器人导航与操作的大规模开放生态系统，

2026 4, 5 。Marion Lepert、[23] Jiaying Fang 和 Jeannette Bohg。Masquerade：利用数据编辑从自然场景人类视频中学习，2025年。 [24] Yaron Lipman、Ricky T. Q. Chen、Heli Ben-Hamu、Maximilian Nickel 和 Matt Le。用于生成建模的流匹配，2023年。 4[25] Fangchen Liu、Kuan Fang、Pieter Abbeel 和 Sergey Levine。Moka：通过基于标记的视觉提示实现开放世界机器人操作，2024年。 2[26] Ruoshi Liu、Alper Canberk、Shuran Song 和 Carl Vondrick。可微机器人渲染，2024年。[27] Songming Liu、Lingxuan Wu、Bangguo Li、Hengkai Tan、Huayu Chen、Zhengyi Wang、Ke Xu、Hang Su 和 Jun Zhu。

Rdt-1b：用于双臂操作的扩散基础模型，2025年。 2[28] Kangchen Lv、Mingrui Yu、Yongyi Jia、Chenyu Zhang 和 Xiang Li。用于全臂机器人操作的运动学感知扩散策略，具有一致的三维观测和动作空间， 2025 2 。Xiao Ma、[29] Sumit Patidar、Iain Haughton 和 Stephen James。用于运动学感知多任务机器人操作的分层扩散策略，2024年。 [30] Pengjun Niu、Yujia Xie、Rui Peng、Hang Zhao 和 Ke Liu。Skelwam：用于零样本跨具身操作的骨架引导世界-动作模型，2026年。[31] 英伟达、Johan Bjorck、Fernando Casta˜neda、Nikita Cherniadev、Xingye Da、Runyu Ding、Linxi “Jim” Fan、Yu Fang、Dieter Fox、Fengyuan Hu、Spencer Huang、Joel Ja ng、Zhenyu Jiang、Jan Kautz、Kaushil Kundalia、La wrence Lao、Zhiqi Li、Zongyu Lin、Kevin Lin、Guilin Liu、Edith Llontop、Loic Magne、Ajay Mandlekar、Avnish Narayan、Soroush Nasiriany、Scott Reed、You Liang Tan、Guanzhi Wang、Zu Wang、Jing Wang、Qi Wang、Jiannan Xiang、Yuqi Xie、Yinzhen Xu、Zhenjia Xu、Seonghyeon Ye、Zhiding Yu、Ao Zhang、Hao Zhang、Yizhou Zhao、Ruijie Zheng 和 Yuke Zhu。Gr00t n1：面向通用型人形机器人的开放式基础模型，2025年。 [32] Austin Patel 和 Shuran Song。Get-zero：用于零样本具身泛化的图具身变换器，2024年。2[33] Prabin Kumar Rath 和 Nakul Gopalan。Xmop：用于零样本跨具身神经运动规划的全身控制策略，2025年。[34] Mingyo Seo、H. Andy Park、Shenli Yuan、Yuke Zhu 和 Luis Sentis。Legato：使用抓取工具进行跨具身模仿，2025年。[35] Kei Suzuki、Jing Liu、Ye Wang、Chiori Hori、Matthew Brand、Diego Romeres 和 Toshiaki Koike-Akino。将形态嵌入 Transformer，以实现跨机器人策略学习，2026年。2[36] Octo Model Team、Dibya Ghosh、Homer Walke、Karl Pertsch、Kevin Black、Oier Mees、Sudeep Dasari、Joey Hejna、Tobias Kreiman、Charles Xu、Jianlan Luo、You Liang Tan、Lawrence Yunliang Chen、Pannag Sanketi、Quan Vuong、Ted Xiao、Dorsa Sadigh、Chelsea Finn 和 Sergey Levine。Octo：一种开源通用机器人策略，2024年。 2[37] 小米具身智能团队、澳门大学、Shaoqing Xu、Fang Li、Guozhi Zhan、Zhixiang Duan、Yuhan Wang、Yuechen Luo、Shengyin Jiang、Hanbing Li、Zhiying Du、Longlong Wang、Longmei Jiang、Weixiang Liang、Ying Gong、Yong Pan、Ziping Zhao、Zhiyuan Chen、Yangwei You、Kun Ma、Qinyuan Liu、Hangjun Ye 和 Zhi xin Yang。一项策略，多种具身形态：面向异构具身操作的统一以相机为中心的动作几何预训练，2026年。[38] Michael Tschannen、Alexey Gritsenko、Xiao Wang、Muhammad Ferjad Naeem、Ibrahim Alabdulmohsin、Nikhil Parthasarathy、Talfan Evans、Lucas Beyer、Ye Xia、Basil Mustafa、Olivier H´enaff、Jeremiah Harmsen、Andreas Steiner 和 Xiaohua Zhai。Siglip 2：具备更强语义理解、定位和稠密特征能力的多语言视觉语言编码器，2025年。 3，5 [39] Vitalis Vosylius、Younggyo Seo、Jafar Uruc和 Stephen James。《渲染与扩散：对齐基于扩散的行为克隆中的图像空间与动作空间》，2024年。 2[40] Lirui Wang、Xinlei Chen、Jialiang Zhao 和 Kaiming He。《利用异构预训练Transformer扩展本体感知视觉学习》，2024年。[41] Junjie Wen、Yichen Zhu、Jinming Li、Zhibin Tang、Chaomin Shen 和 Feifei Feng。《Dexvla：采用插件式扩散专家的视觉语言模型，实现通用机器人控制》，2025年。[42] Seonghyeon Ye、Yunhao Ge、Kaiyuan Zheng、Shenyuan Gao、Sihyun Yu、George Kurian、Suneel Indupuru、You Liang Tan、Chuning Zhu、Jiannan Xiang、Ayaan Malik、Kyungmin Lee、William Liang、Nadun Ranawaka、Jiasheng Gu、Yinzhen Xu、Guanzhi Wang、Fengyuan Hu、Avnish Narayan、Johan Bjorck、Jing Wang、Gwanghyun Kim、Dantong Niu、Ruijie Zheng、Yuqi Xie、Jimmy Wu、Qi Wang、Ryan Julian、Danfei Xu、Yilun Du、Yevgen Chebotar、Scott Reed、Jan Kautz、Yuke Zhu、Linxi ” Jim” Fan 和 Joel Jang。《世界动作模型是零样本策略》，2026年。5[43] Chengxuan Ying 、 Tianle Cai 、 Shengjie Luo 、 Shuxin Zheng、Guolin Ke、Di He、Yanming Shen 和 Tie-Yan Liu。《Transformer在图表示方面的表现真的很差吗？》，2021年。 3 [44] Chengbo Yuan、Chuan Wen、Tong Zhang 和 Yang Gao。《通用流作为可扩展机器人学习的基础可供性》，2024年。 2[45] Lihan Zha、Asher J. Hancock、Mingtong Zhang、Tenny Yin、Yixuan Huang、Dhruv Shah、Allen Z. Ren 和 Anirudha Majumdar。《Lap：语言-动作预训练支持零样本跨具身迁移》，2026年。 5[46] Ke Zhang、Lixin Xu、Chengyi Song、Junzhe Xu、Xiaoyi Lin、Zeyu Jiang 和 Renjing Xu。《Dexformer：通过历史条件Transformer实现跨具身灵巧操作》，2026年。[47] Guoyang Zhao、Quanhao Qian、Gongjie Zhang、Wenhao Li、Jiuniu Wang、Xiaowei Lu、Deli Zhao 和 Ran Xu。《Geoprop：在视觉中锚定机器人状态，实现通用型操作》，2026年。 2[48] Yi Zhou、Connelly Barnes、Jingwan Lu、Jimei Yang 和 Hao Li。《论神经网络中旋转表示的连续性》，2020年。 3