arXiv:2301.02667v2 [cs.CV] 8 Sep 2023

运动-动作-操作：在复杂三维环境中合成人-场景交互

李智艺 首尔国立大学kay2353@snu.ac.kr

Hanbyul Joo 首尔国立大学hbjoo@snu.ac.kr

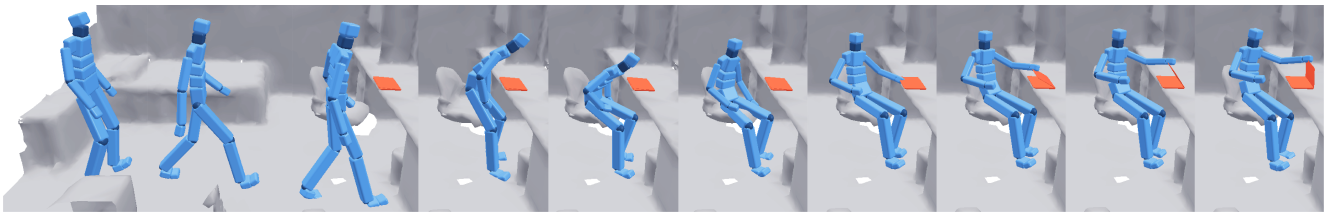

图 1：我们的方法 LAMA 可生成高质量、逼真的三维人体动作，其中包含在给定三维场景和指定交互提示下的移动、场景交互和操作。

摘要

由于三维环境高度复杂，且其中可能发生的人类行为种类繁多，合成涉及交互的人体动作一直颇具挑战。我们提出 *LAMA*，***L***运动-动作-操作，旨在复杂室内环境中合成自然且合理的长期人体移动。*LAMA* 的核心动机是构建一个统一框架，涵盖一系列日常动作，包括移动、场景交互和物体操作。与现有需要将动作数据与扫描的三维场景“配对”以进行监督的方法不同，我们将这一问题表述为测试时优化，仅使用人体动作捕捉数据进行合成。*LAMA* 采用结合动作匹配算法的强化学习框架进行优化，并进一步通过流形学习利用动作编辑框架，以覆盖交互和操作中的各种可能变化。通过大量实验，我们证明 *LAMA* 在各种具有挑战性的场景中合成逼真动作方面优于以往方法。项目页面：*https://jiyewise.github.io/projects/LAMA/*。

1. 引言

在现实生活的三维环境中合成交互一直是一项具有挑战性的研究课题，因为它复杂且多样。现实生活中的三维环境里，众多物体杂乱地摆放，由此产生的空间约束使运动合成受到诸多限制且十分复杂。此外，三维环境中可能的空间布局以及人类交互行为几乎无穷无尽的多样性，也使合成泛化变得困难。

由于人-场景交互涉及广泛的技术挑战，先前的方法主要关注一些子问题，例如：(1) 对静态姿势建模 [27, 75, 77, 19, 70, 53, 78] ，或 (2) 建模人与单个目标物体或交互类型之间的交互 [57, 76, 73, 58, 59, 51, 72, 12]。较新的方法 [64, 63, 17]则进一步尝试在真实世界三维场景中合成动态交互动作，并使用“场景配对”的动作数据集 [18] ，其中动作与周围的三维环境同步采集。由于此类配对数据集十分稀少且难以扩展，这些方法在充分覆盖真实世界三维场景中人类交互的复杂性与多样性方面，性能从根本上受到限制。

本文提出 LAMA（**移**动-**动**作-**操**作），用于在复杂室内环境中合成自然且合理的长时人体动作。LAMA 的核心动机是构建一个统一框架，涵盖真实世界三维场景中的一系列日常动作：穿行于杂乱区域、与场景交互，以及操作物体。不同于以往使用“场景配对”动作数据集进行监督的方法 [64, 17]

我们将其表述为一种测试时优化，仅利用人体动作捕捉数据。我们将强化学习（RL）作为优化工具，提出一种基于强化学习的框架，并结合动作匹配算法 [11, 7] ，以无缝合成移动和场景交互，同时通过碰撞规避来适应复杂三维场景。该框架中的物体操作则通过一种动作编辑方法实现，并在此基础上学习基于自编码器的动作流形空间 [22]。作为一种测试时优化框架，LAMA 适用于任何三维场景，例如公开数据集或新扫描的场景。通过与现有方法进行广泛的定量和定性评估，我们证明该方法在各种具有挑战性的场景中表现更优 [64, 17]。

我们的贡献总结如下：(1) 首个无需“配对”数据集，即可在复杂三维场景中生成结合移动、场景交互和操作的逼真长时动作的方法；(2) 一种新颖的测试时优化框架，仅需人体动作捕捉数据，通过引入与动作匹配相结合的强化学习框架，并配备针对碰撞规避和场景交互而精心设计的状态与奖励；(3) 达到最先进水平的动作合成质量，持续时间更长（接近 10 秒）；(4) 新近采集并精修的动作捕捉数据集，其中包含适用于动作匹配的移动和动作（例如坐下）。

2. 相关工作

生成人-场景交互。 生成自然的人体动作一直是计算机视觉领域广受研究的课题。早期方法主要利用神经网络 [15, 13, 39, 39, 42, 60, 62, 50]来合成或预测人体动作。然而，这些方法主要关注人体动作本身的合成，并未考虑周围的三维环境。近期方法开始着手建模和合成三维场景中的人-场景交互，或人与物体的交互。其中许多方法侧重于在给定的三维环境中静态摆放人物 [27, 75, 77, 18],, 通过各种类型的输入生成人人场景交互姿态，包括物体语义 [19], 、图像 [23, 74, 71, 70, 26, 25], 和文本描述 [53, 78]。

近年来，已有一些方法用于合成动态人-物交互（例如，坐在椅子上、搬运箱子）。Starke等人 [57] 提出了一种自回归学习框架，利用基于物体几何的环境编码来合成人-物交互。尽管这种编码包含场景中多个物体的信息，但正如 [17], 所示， [57]并没有用于在杂乱三维场景中导航的显式模块。后续工作 [17, 76]对此进行了扩展，通过合成

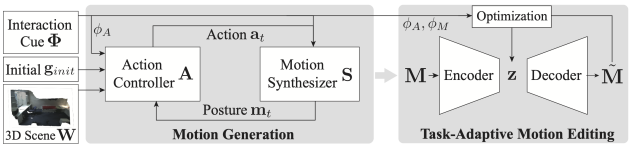

图2：LAMA 概览。

以物体变化和接触点为条件的动作。其他方法 [73, 58, 68, 59, 51, 72, 79]专注于生成人体操作时的自然手部动作，并进一步扩展为包含全身动作 [58, 68]。此外，也有研究探索了利用基于物理的角色控制来合成人-物交互[43, 12, 10, 51, 72, 35]。尽管这些方法涵盖了多种人-物交互，但大多数方法都聚焦于特定的交互类型或人与目标物体之间的关系，而未涉及杂乱三维场景中的长期导航。

较新的方法包括在杂乱三维场景中生成人与场景的自然交互 [64, 63, 8,65],, 这与我们的方法密切相关。这些方法使用与三维场景配对的人体动作数据集进行训练，因此需要同时获取真实动作和对应的三维场景，以提供监督。由于此类数据难以获取，一些方法利用合成数据集 [8, 65],, 即从深度视频 [64], 或短时长（1–3 秒）动作快照 [66]拟合得到的数据。在以往的方法中 [17,63],, 杂乱环境中的导航通常由独立模块通过路径规划来完成（例如， A^* 算法），并将人体体积近似为圆柱体。这些基于路径规划的方法对场景和身体的空间信息进行近似，因此在高度复杂的条件下存在局限。

运动合成与编辑。 利用动作捕捉数据合成自然人体运动，也是计算机图形学领域长期以来的研究课题。一些方法 [29, 41]构建动作图，将合理的过渡作为边插入其中，并通过遍历图来合成运动。类似的方法 [34, 55] 将动作补丁连接起来，以合成虚拟环境中的交互或多人交互。由于图方法具有通用性且简单易用，人们对其进行了多种改进，例如动作语法 [24] ，它会对动作图中的遍历规则加以约束。动作匹配 [7, 11]也可以看作动作图遍历的一种特殊形式：合理的过渡并非预先计算，而是在运行时搜索得到。深度学习的最新进展使得人们能够利用动作捕捉数据进行动作流形学习 [22, 56, 21]。基于变分自编码器（VAE）[40, 50] 和循环神经网络 [32, 16, 45] 的自回归方法也被用于根据过去的帧预测未来动作。这些框架被推广用于

合成多种动作，包括在地形上移动 [21]迷宫 [40], 动作指定的运动 [50],以及交互式体育运动 [32, 45]。据报道，基于神经网络的方法在多种动作编辑任务中也取得了成功，例如骨骼重定向 [4], 风格迁移 [22, 5], 中间帧生成 [16]。

强化学习（RL）与数据驱动方法和基于物理的方法相结合，在合成人体动作方面也取得了成功。与数据驱动方法结合时，RL 充当控制模块，通过遍历动作图 [31],潜空间 [38, 61, 40],以及预计算过渡表 [33], 根据给定的用户输入生成相应动作。深度强化学习（DRL）也被广泛用于合成具有多种运动技能的物理合理的动作 [47, 45, 36, 6, 67, 49, 48, 35]。这些方法的核心思想源于模仿学习：在 DRL 中，控制策略经过优化后，会依据角色的物理状态驱动角色，使其在物理模拟环境中实现对给定参考动作的跟踪目标。

3. 方法

3.1. 概述

我们的系统称为 $LAMA$，它以三维场景$\mathbf{W}$、期望的交互提示 $\mathbf{\Phi}$ 和初始状态 $\mathbf{g}_{init}$ =($\mathbf{p}^0_{root}$,$\mathbf{r}^0_{root}$) 为输入，输出一系列人体姿态 $\mathbf{M}$ = $\{\mathbf{m}_t\}_{t=1}^T$ ：

$$\mathbf{M} = \mathcal{LAMA}(\mathbf{W}, \mathbf{\Phi}, \mathbf{g}_{init}), \tag{1}$$

其中，初始时刻（即 $t = 0$）角色根节点的全局位置和方向分别表示为 $\mathbf{p}^0_{root} \in \mathbb{R}^3$ 和 $\mathbf{r}^0_{root} \in so(3)$。时刻 t 的输出姿势 $\mathbf{m}_t$ =($\mathbf{p}^t_{root}$, $\mathbf{r}^t_{root}$, $\mathbf{r}^t_1$, ..., $\mathbf{r}^t_J$) $\in \mathbb{R}^{3J+6}$, 由根节点的全局位置和方向，以及 J 个关节的局部关节方向构成的拼接向量表示，其中第 j 个关节采用轴角表示 $\mathbf{r}^t_j \in so(3)$。在整个系统中，骨骼树结构和关节偏移均固定，如图 5(a) 所示。我们将三维场景 W = $\{\mathbf{w}_i\}$ 表示为一组三维对象和场景网格，其中包括背景场景网格以及其他用于操作的目标物体网格。

交互提示 $\mathbf{\Phi} = [\phi_A, \phi_M]$ 表示输出需要实现的预期目标，由动作任务（例如坐下）的动作提示 ϕ_A 和操作任务的操作提示 ϕ_M 组成。动作提示 ϕ_A = {$\mathbf{q}_{root}$、$\mathbf{r}_{root}$、$\mathbf{q}_{rFoot}$、$\mathbf{q}_{lFoot}$、} 分别表示根节点的期望位置和方向，以及左脚和右脚的位置（即 $\mathbf{q}_j \in \mathbb{R}^3$）。足部位置是可选的；若未提供，则可自动确定。$\phi_A$ 可以手动指定以引导角色，也可以通过现成的估计器（例如 [17]中的 GoalNet）自动生成。操作提示 ϕ_M = $\{\mathbf{q}^t_j\}_{j\in J_M}$ 表示用于控制的选定关节 J_M 的期望位置，主要用于动作编辑流程（第 3.6节）。例如，ϕ_M 可指定手部关节轨迹，以完成图 1 和图 6中的打开笔记本电脑动作。三维场景$\mathbf{W}$、动作提示 ϕ_A 和操作提示 ϕ_M 的示例见图 6 （左）。

LAMA 采用由三级系统构成的设计，包括动作控制器 A、动作合成器 S，以及基于流形的动作编辑器 E。移动和动作部分通过动作控制器 A 与动作合成器 S 无缝完成。我们的设计核心思想是将强化学习框架与动作匹配相结合 [11, 7]，在完成所需场景交互任务的同时合成逼真的人体动作。动作控制器 A 以三维场景 W、动作提示 ϕ_A 和初始状态 $\mathbf{g}_{init}$ 为输入，利用强化学习进行测试时优化，以合成相应动作。控制策略 π 经过优化 [1]，用于在时刻 t 采样动作 $\pi(\mathbf{a}_t|\mathbf{s}_t,$ W $\phi_A)$，其中 $\mathbf{a}_t$ 表示可能的下一步动作，包含预测的动作类型和短期未来预测。$\mathbf{s}_t$ 是用于表示人物角色当前状态的状态提示，包括其身体姿势、周围场景占用情况以及目标动作提示。直观而言，动作控制器 A 通过考虑当前角色—场景状态 $\mathbf{s}_t$，经过优化以生成合理的下一步动作 $\mathbf{a}_t$ 。动作控制器 A 生成的动作信号 $\mathbf{a}_t$ 会作为输入提供给动作合成器 S，随后由后者确定下一时间步的姿势 $\mathbf{m}_{t+1}$，即 S($\mathbf{m}_t$,$\mathbf{a}_t$) = $\mathbf{m}_{t+1}$。人物角色的下一个状态 $\mathbf{s}_{t+1}$ 可由 $\mathbf{m}_{t+1}$ 再次计算得到，随后动作控制器 A 会将其作为下一帧的输入。

由 $\mathbf{A}$ 和 $\mathbf{S}$ 合成的初始输出动作$\mathbf{E}(\mathbf{M})$ = $\{\tilde{\ }_t\}_{t=1}^T$ $\tilde{\mathbf{M}}$，其中 $\mathbf{M}$ $\tilde{\mathbf{M}}$ 是编辑后的动作。编辑模块 $\mathbf{E}$ 的目标是：(1) 对 $\mathbf{M}$ 进行后处理，使其适用于动作任务 ϕ_A 所针对的各种物体（例如，坐在不同高度的椅子上）；(2) 按照 ϕ_M 的指示执行人-物体交互（例如移动物体、打开门）。图2 展示了 LAMA 的概览。

3.2. 场景感知动作控制器

不同于以往的方法，[65, 17] 这些方法使用路径规划进行导航，并使用通过场景配对动作数据集训练的基于学习的模块进行交互；我们的动作控制器$\mathbf{A}$ 通过完成动作提示 ϕ_A 并避开三维场景$\mathbf{W}$中的碰撞，无缝执行移动和期望动作。重要的是，给定场景$\mathbf{W}$、动作 ϕ_A 和初始状态$\mathbf{g}_{init}$ 作为输入，我们的动作控制器会直接进行优化，以便在每个状态下从动作数据库中选择最合理的动作片段，合成——

[1]对于该策略，我们使用“优化的”而非“学习的”这一说法，因为我们执行的是测试时优化。

自然的人体动作，同时考虑三维场景，而无需任何场景配对动作数据集或训练流程。直观而言，给定当前状态$\mathbf{s}_t$，动作控制器的目标是输出最佳的下一步动作$\mathbf{a}_t$，并利用该动作在动作合成器$\mathbf{S}$中搜索下一个动作片段。

状态。时刻 t 的状态 $\mathbf{s}_t = \psi(\mathbf{m}_{t-1}, \mathbf{m}_t, \mathbf{W}\ \phi_A)$ 是一个表示当前人类角色状态的特征向量，其中 ψ 是根据其他输入计算状态的函数。$\mathbf{s}_t = (\mathbf{s}_t^{body}, \mathbf{s}_t^{scene}, \mathbf{s}_t^{inter})$ 由身体构型 $\mathbf{s}^{body}$、场景占用 $\mathbf{s}^{scene}$ 和期望的当前目标交互 $\mathbf{s}^{inter}$ 组成。身体构型 $\mathbf{s}^{body} = \{\mathbf{r}\ \dot{\mathbf{r}}\ \theta_{up}, h, \mathbf{p}_e\}$，其中 $\mathbf{r}\ \dot{\mathbf{r}} \in \mathbb{R}^{J\times 6}$ 分别表示除根节点外 J 个关节的关节旋转和速度，采用六维表示 [80]；$\theta_{up} \in \mathbb{R}$ 是根节点的向上向量（以相对于 Y 轴的角度表示），$h \in \mathbb{R}$ 是根节点距地面的高度，$\mathbf{p}_e \in \mathbb{R}^{e\times 3}$ 是人物中心坐标系中的末端执行器位置（其中 e 是末端执行器的数量）。$\mathbf{s}^{scene} = \{\mathbf{g}_{occ}, \mathbf{g}_{root}\}$ 包含地面平面上的场景占用信息，如图 4 所示。$\mathbf{g}_{occ} \in \mathbb{R}^{n^2}$ 表示智能体周围邻近 n 个单元在地面平面上的占据栅格，$\mathbf{g}_{root} \in \mathbb{R}^2$ 表示角色在离散网格平面上的当前全局根节点位置。请注意，虽然出于效率考虑，我们使用二维地面网格而非三维网格，但第 3.4 节中的碰撞奖励项仍会考虑三维场景。$\mathbf{s}^{inter}$ 表示角色所瞄准的动作提示，即 $\mathbf{s}^{inter} = \phi_A$。

动作。给定角色的当前状态 $\mathbf{s}_t$，控制策略 π 输出可行动作 $\mathbf{a}_t = (\mathbf{a}_t^{type}, \mathbf{a}_t^{future}, \mathbf{a}_t^{offset})$。$\mathbf{a}_t^{type}$ 提供了可能动作（如行走、坐下或停止）中下一动作类型的概率，从而决定动作之间的转变时机（例如，从移动转变为坐下）。$\mathbf{a}_t^{future}$ 预测未来的动作提示，例如未来 10、20 和 30 帧时合理的根节点位置。姿态偏移 $\mathbf{a}_t^{offset}$ 旨在修改动作合成器模块 $\mathbf{S}$的动作数据库中检索到的原始动作数据。直观而言，我们优化后的控制策略会生成姿态偏移 $\mathbf{a}_t^{offset}$，以调整数据库中选取的最接近的合理原始姿态。这样一来，仅凭人体动作数据，角色便能做出更合理的场景感知人体姿态。更多详情见第 3.3 节。

3.3. 动作合成器

将当前姿态$\mathbf{m}_t$ 和动作控制器$\mathbf{A}$ 输出的动作信号$\mathbf{a}_t$作为输入，动作合成器$\mathbf{S}$ 会生成下一个合理的姿势：$\mathbf{S}(\mathbf{m}_t, \mathbf{a}_t) = \mathbf{m}^{t+1}$。作为第一步，动作合成器会从动作数据库中搜索与最接近的动作特征最匹配的动作，然后修改搜索到的原始运动，使其更适合场景。为此，动作合成器的输出$\mathbf{m}^{t+1}$ 会接着递归输入到动作控制器中——

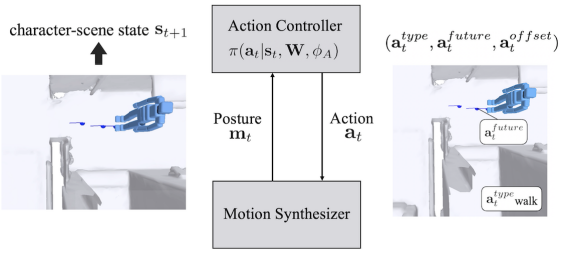

图 3：动作控制器与动作合成器之间关系的可视化。

。我们采用改进版的动作匹配算法 [7, 11, 20] 来完成运动合成的第一步。在动作匹配中，运动合成会定期执行：从动作数据库中搜索最合理的下一个镜头动作片段，并将其拼接成一段连贯的长序列。

动作特征。运动特征 $\mathbf{y}_t$ 表示短运动片段中每一帧的特征，其计算方式为 $f(\mathbf{m}) = \mathbf{y}_t = \{\{\mathbf{p}_j\}, \{\dot{\mathbf{p}}_j\}, \theta_{up}, c, \mathbf{o}_{\text{future}}\}$。从姿势 $\mathbf{m}$ 中提取选定关节 $j \in$ {头部、手部、脚部}的位置和速度 $\mathbf{p}_j, \dot{\mathbf{p}}_j \in \mathbb{R}^3$，这些数据均定义在 $\mathbf{m}$ 的人物中心坐标系中。$\theta_{up} \in \mathbb{R}^3$ 是根关节的上方向向量，$c \in \{0, 0.5, 1\}$ 表示自动计算的左右脚接触提示（不接触时为 0，接触时为 1，未接触但距地面小于某一阈值时为 0.5）。$\mathbf{o}_{\text{future}} = \{\{\mathbf{p}_{root}^{\Delta t}\}, \{\mathbf{r}_{root}^{\Delta t}\}\}$ 包含短期未来姿态的提示，其中 $\mathbf{p}_{root}^{\Delta t}$ 和 $\mathbf{r}_{root}^{\Delta t}$ 分别表示当前目标帧之后 Δt 帧处根关节的位置和方向。$\mathbf{o}_{\text{future}}$ 是在当前目标动作 $\mathbf{m}$ 的人物中心坐标系中的二维 XZ 平面上计算的，因此 $\mathbf{p}_{root}^{\Delta t}, \mathbf{r}_{root}^{\Delta t} \in \mathbb{R}^2$。选取的未来帧因动作类型而异；对于移动，我们提取之后第 10、20 和 30 帧（频率为 30Hz），以 [11]为基准。直观而言，运动特征通过考量相邻帧，提取目标帧的姿势和时间提示。为提高效率，我们预先计算数据库中动作片段每一帧的运动特征 $\mathbf{y}_t$。

角色当前状态的动作特征$\mathbf{x}_t$，或称查询特征，也以相同的方式根据动作控制器生成的姿势$\mathbf{m}_{t-1}$、$\mathbf{m}_t$ 和 $\mathbf{a}_t^{future}$ 进行计算，即$\mathbf{x}_t = f(\mathbf{m}_{t-1}, \mathbf{m}_t, \mathbf{a}_t^{type}, \mathbf{a}_t^{future})$。组件$\mathbf{a}_t^{future}$ 在查询特征中充当$\mathbf{o}_{\text{future}}$，可以将其理解为动作控制器为预测的未来姿态提供线索。

运动搜索与更新。给定查询动作特征$\mathbf{x}_t$ 和动作数据库中的动作特征$\mathbf{y}_k$（其中 k 是片段的索引），运动搜索通过计算查询特征与动作数据库特征之间的加权欧氏距离，在动作数据库中找到最佳匹配项 k^*，

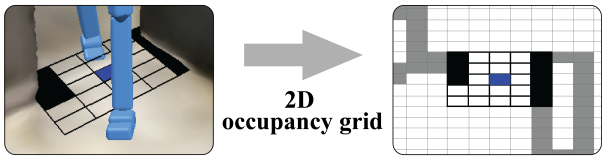

图4：占据栅格的可视化表示。右侧的栅格表示俯视图。灰色和黑色表示被占据的单元格，蓝色表示根节点。

特征和动作数据库特征：

$$k^* = \arg\min_k ||\mathbf{w}_f^T(\mathbf{x}_t - \mathbf{y}_k)||^2, \tag{2}$$

其中，$\mathbf{w}^f$ 是用于控制特征元素重要性的固定权重向量。在从动作数据库中找到最佳匹配项 $\hat{\mathbf{m}}_{k^*}$后，动作合成器使用来自$\mathbf{a}_t$的预测动作偏移量$\mathbf{a}_t^{offset}$ 对其进行更新，即$\tau(\hat{\mathbf{m}}^{k^*+1}, \mathbf{a}^{offset}) = \mathbf{m}^{t+1}$，其中 $\hat{\mathbf{m}}^{k^*+1}$ 是下一个合理的角色姿态，τ是用于更新 $\hat{\mathbf{m}}^{k^*+1}$中选定关节的更新函数。实际应用中，会定期执行运动搜索（例如，每隔 N 帧一次），以使合成动作在时间上更加连贯。

3.4. 优化场景感知动作控制器

我们的强化学习框架旨在通过最大化折扣累积奖励来优化策略。在我们的方法中，我们设计奖励以引导角色在常见约束条件下（例如，平滑过渡和碰撞规避）执行移动和期望的动作（例如，坐下）。我们的奖励函数由以下几项组成：

$$R_{\text{total}} = w_{\text{tr}}R_{\text{tr}} + w_{\text{act}}R_{\text{act}} + w_{\text{reg}}R_{\text{reg}}, \tag{3}$$

其中，w_{tr}、w_{act}和 w_{reg} 是用于平衡各奖励项的权重。当角色在满足来自三维场景的空间约束的同时朝向动作 ϕ_A 移动时，即可获得轨迹奖励 R_{tr}，具体如下：

$$R_{\text{tr}} = r_{\text{coli}} \cdot r_{\text{pos}} \cdot r_{\text{root}}, \text{ where} \tag{4}$$

$$r_{\text{coli}} = \exp\left(-\frac{1}{\sigma_{\text{coli}}^2}\sum_{b\in\mathbf{B}} w_b\rho(b,\mathbf{W})\right), \tag{5}$$

$$r_{\text{pos}} = \exp\left(-\frac{1}{\sigma_{\text{root}}^2}\sum_{j\in\mathbf{J}} \|\mathbf{p}_0 - \mathbf{q}_j\|^2\right), \tag{6}$$

$$r_{\text{vel}} = \begin{cases} 1 & \text{when } \dot{\mathbf{p}}_{root} \geq \sigma_{th} \\ \sigma_{vel}\|\dot{\mathbf{p}}_0\|^2 & \text{else.} \end{cases} \tag{7}$$

碰撞避免奖励 r_{coli} 会对与三维场景发生的碰撞进行惩罚。如图5 (a) 所示，骨骼结构中的身体肢体表示为一组具有固定宽度的箱形节点$\mathbf{B}$，其中每个元素 $b \in \mathbf{B}$ 是

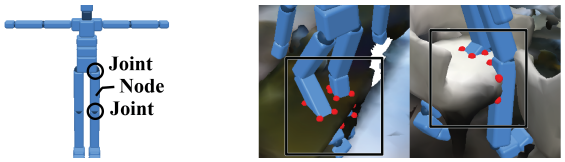

图 5：（a）带有关节和盒状节点的骨架。（b）自动检测到的碰撞点（以红色标示）。

腿部和手臂的三维箱体表示（不包括躯干和头部）。函数 $\rho(b,\ \mathbf{W})$ 用于检测箱形节点 b 与三维场景$\mathbf{W}$ 边缘之间的碰撞，并返回交点数量。（图5 (b)）。 w_b是用于控制各肢体 b重要性的权重。当不存在穿透时，碰撞避免奖励达到最大值，从而促使策略 π 生成适当的动作$\mathbf{a}_t$ 以避免物理上不合理的穿透。 r_{pos} 在智能体朝目标动作提示移动时获得 ϕ_A。 r_{vel} 通过在根节点速度 $\dot{p}_{root}$ 低于阈值时进行惩罚，鼓励角色移动 σ_{th}。σ_{coli}，σ_{root}，以及 σ_{vel} 是用于控制平衡的权重。

动作奖励 R_{act} 鼓励完成给定的动作提示 $\phi_A = \{\mathbf{q}_{root}, \mathbf{r}_{root}, \mathbf{q}_{rFoot}, \mathbf{q}_{lFoot}\}$:

$$R_{\text{act}} = r_{\text{inter}} \cdot r_{\Delta\text{t}} \cdot r_{\Delta\text{v}}, \quad \text{where}$$

$$r_{\text{inter}} = \exp\left(-\frac{1}{\sigma_{\text{inter}}^2}\sum_{j\in J_A}\|\mathbf{p}_j - \mathbf{q}_j\|^2\right), \tag{8}$$

$$r_{\Delta\text{t}} = \exp\left(-\sigma_{\Delta\text{t}}^2 C_{\text{tr}}\right), r_{\Delta\text{v}} = \exp\left(-\sigma_{\Delta\text{v}}^2 C_{\text{vel}}\right)$$

其中，交互奖励项 r_{inter} 在角色从导航切换到相应动作时给出ϕ_A，并且当执行的动作满足 ϕ_A的位置约束时达到最大值。平滑性奖励项 $r_{\Delta\text{t}}$和 $r_{\Delta\text{v}}$ 用于最小化过渡代价，该代价基于式2中定义的特征距离子集，其中 C_{tr}是 p_j、 θ_{up}和 c的加权特征距离，而 C_{vel}来自 $\dot{p}$。这些奖励项旨在抑制角色产生突变。

正则化奖励 R_{reg} 会惩罚$\mathbf{a}_t^{offset}$ 对在$\mathbf{S}$的动作数据库中搜索到的原始姿势进行过度修改（记作 $\hat{\mathbf{m}}_t$），并维持帧间的时间一致性。

$$R_{\text{reg}} = \exp\Big(-\frac{1}{\sigma_{\text{reg}}^2}\Big(\|\hat{\mathbf{m}}_t - \mathbf{m}_t\|^2 + \|\mathbf{m}_t - \mathbf{m}_{t-1}\|^2\Big)\Big).$$

如 [37, 45], 所述，将奖励与一致的目标相乘，可以确保所有奖励项同时得到满足。我们还使用提前终止 [47] 和有限的动作转换来加快学习。详情见补充材料。

3.5. 泛化动作控制器

虽然我们主要将强化学习用于给定单个目标任务的测试时优化，但优化后的

策略凭借强化学习的特性，在一定程度上可以处理任务的变化。如第 4.4 节的实验所示，我们证明了优化后的控制器可以直接用于各种动作提示 ϕ_A 和初始状态$\mathbf{g}_{init}$，无需针对同一场景进行进一步优化$\mathbf{W}$。

作为我们框架的扩展，我们可以通过针对随机输入变化优化策略，使控制器具有更强的泛化能力，$\mathbf{g}_{init}$ 并且 ϕ_A 在策略优化期间的每个回合中进行优化。此过程更类似于常见的强化学习框架，在该框架中，策略会预先针对目标场景进行“学习” $\mathbf{W}$，并在推理期间应用于提供的输入。我们还证明，通过这一增强过程，控制器能够处理范围更广的输入变化。对于在固定三维场景 $\mathbf{W}$下指定不同任务的情况，这一框架扩展能够提高效率。如第 4.4 节所示，通过泛化过程，我们可以直接将策略用于多种输入，而无需进一步优化。或者，如有必要，也可以高效地微调策略。需要注意的是，这一扩展仍不同于其他基于学习的方法 [64, 17]，因为我们不需要任何场景配对动作数据集或其他监督。

3.6. 任务自适应动作编辑

为涵盖交互的多样性，我们在动作合成框架中加入了任务自适应动作编辑模块。具体而言，在物体操作场景中，会提供操作提示 ϕ_M，以便约束末端执行器（例如手）沿着所需轨迹运动，从而在目标物体上完成操控任务，如图6所示。操作提示 ϕ_M 可以通过任何可行方式提供；在我们的实验中，我们以半自动方式生成该提示。我们通过模拟目标关节物体的运动 [69] 来计算所需轨迹，计算时考虑物体网格表面上的一个接触点。

˜编辑后的动作$\mathbf{M} = E(\mathbf{M})$不仅应满足稀疏给定的位置约束，还应保留关节之间的时间一致性和空间相关性，以维持动作的自然度。我们采用基于卷积自编码器 [22] 的动作流形学习方法，将动作压缩为动作流形空间中的潜向量。动作编辑通过在流形中搜索最优潜向量来完成。训练自编码器时，将由$\mathbf{M}$转换而来的动作序列$\mathbf{X}$ 表示为姿势的时间序列，具体做法是拼接以6D表示的关节旋转 [80], 、根节点高度、投影到XZ平面的相对于前一帧的根节点变换，以及脚部接触标签。编码器和解码器模块基于重建损失进行训练，$||\mathbf{X} - \Psi^{-1}(\Psi(\mathbf{X}))||^2$，其中 Ψ 是编码器，Ψ^{-1} 是解码器。

来自编码器的潜向量$\mathbf{z} = \Psi(\mathbf{X})$能够保留动作流形空间中自然人体动作所具有的时空关-

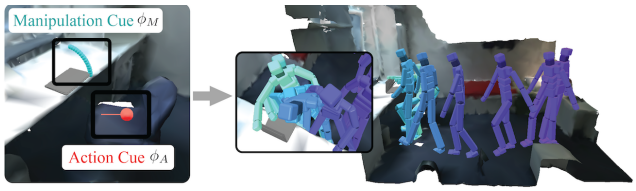

图 6：系统输入 Φ、$\mathbf{W}$ 和˜输出$\mathbf{M}$的可视化表示。左侧分别以红色和青色显示动作提示 ϕ_A 和操作提示ϕ_M。右侧为˜合成动作$\mathbf{M}$。

节与帧之间的关系。如 [22], 所示，在流形空间中编辑动作可确保编辑后的动作自然且连贯。通过最小化损失函数 $\mathcal{L}$，并约束输出动作遵循操控约束 ϕ_M，即可找到最优潜向量$\mathbf{z}^*$。我们还在 $\mathcal{L}$ 中加入了额外的正则项，使输出动作能够保留原始动作的脚部位置和根轨迹。关于 $\mathcal{L}$的更多详情，请参阅补充材料。最后，编辑后的动作 ˜$\mathbf{M}$ 可通过 $\Psi^{-1}(\mathbf{z}^*)$计算得到。

4. 实验

我们评估 LAMA 在真实世界三维场景中合成长时动作的能力，场景中涉及各种人与场景及物体交互。我们采用大量定量指标和感知研究进行评估。

数据集。 为构建动作合成器的数据库，我们采集了一个新的动作捕捉数据集，其中包含移动和动作。动作由基于惯性测量单元的系统 XSens MVN Link [3]捕获。采集的数据包含高质量的人体动作，涵盖各种场景中的移动和交互，例如从随机起点以不同角度行走，以及坐在椅子上。捕获的动作数据经过后处理，以适用于动作匹配。本系统使用的所有数据均为动作捕捉数据（采用BVH格式），不包含与场景或物体相关的先验信息。我们使用 PROX [18] 和 Matterport3D [9] 数据集提供三维场景，并使用 SAPIEN [69]物体网格进行操作。详情请参阅补充材料。

4.1. 实验设置

评估指标。 由于我们的系统在运动合成中不依赖监督，因此缺少真实标注数据或官方评估指标，使得量化合成质量颇具挑战。我们尝试从物理合理性和自然度两个方面进行评估。

·物理合理性：我们使用接触和穿透指标来评估合成动作的物理合理性。若足部处于接触状态，接触度量会对脚部运动施加惩罚。由于足部接触是动力学中的关键要素，基于接触的度量与



图 7：包含移动、动作和操作的动作示例。上：打开、关闭垃圾桶盖并坐在椅子上。下：打开门并坐在椅子上。

方法	接触	穿透	自然度
Wang 等人 [64]	6.32	2.75	16.27
Wang 等人 [64]*	22.98	14.73	-
SAMP [17]	11.75	7.18	42.04
LAMA（我们的方法）	**4.34**	**1.30**	**100**

表 1：**基线比较**所有帧上的平均足部接触距离（厘米，↓）和穿透率（百分比，↓）得分。自然度得分（百分比，↑）表示相对于 LAMA 的选择比例（对于 LAMA，设为 100）。带星号的 Wang 等人表示未进行后处理。

动作的物理合理性密切相关。穿透损失（表1中的“穿透”）用于衡量身体穿透场景中物体等不合理情况。我们通过统计交点（第3.2节）超过特定阈值的帧数来计算穿透度量。[2]

- **自然度：**我们通过在亚马逊土耳其机器人上开展感知研究（A/B 测试），评估合成动作的自然度。用于测试的动作采用完全相同的视角和 3D 角色进行渲染，因此从外观上无法区分。我们要求人类观察者根据以下两个标准选择更自然的动作：（1）角色的运动具有人类特征；（2）该运动在给定场景中合理可信。研究设置的详细信息见补充材料。

基线。我们将 LAMA 与当前最先进的方法以及我们方法的变体进行比较。

- **Wang 等人** [64] 是针对给定三维场景中人-场景交互的最先进长期动作合成方法。我们使用作者提供的代码进行评估。由于 Wang 等人会对合成动作进行后处理，以改善足部接触并减少碰撞，而这些因素与我们的指标直接相关，因此我们同时比较 Wang 等人经过后处理和未经后处理的结果。

- **SAMP** [17] 生成的交互不仅能够泛化到不同的物体变化，也能泛化到给定三维场景内的随机起点。SAMP 明确利用路径规划，在杂乱的三维场景中导航。

- **消融基线**我们对动作控制器和动作编辑模块进行了消融研究。我们对场景奖励进行了消融研究 r_{coli}和动作偏移$\mathbf{a}_t^{offset}$，以展示这两项对生成场景感知运动的贡献。我们还比较了去除转移奖励 $r_{\Delta t}$ 和 $r_{\Delta v}$ 项（第3.2节）后的动作控制器。最后，我们将动作编辑模块与逆运动学（IK）进行比较，展示其自然编辑动作的优势（第3.6节）。

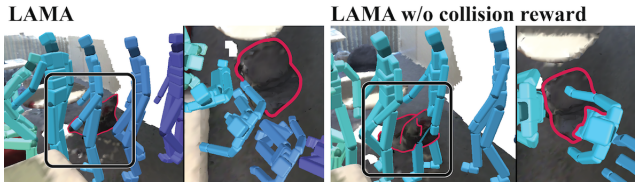

图 8：与 LAMA（左）以及未使用碰撞奖励的 LAMA（右）进行比较。没有碰撞奖励时，角色无法避开障碍物（红色所示），发生了碰撞。

[2] 10 腿部为 7，手臂为 7

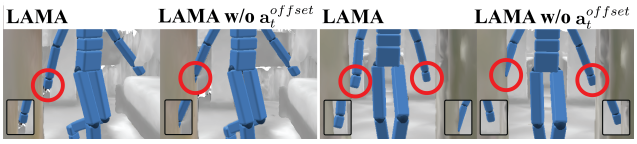

图 9：与 LAMA（左）以及不使用动作偏移的 LAMA（右）进行比较。原始 LAMA 中的角色向前移动，同时倾斜手臂以避开墙壁碰撞；而不使用动作偏移的 LAMA 中的角色则不会这样做。

4.2. 与先前工作的比较

评估设置与细节。为与基线方法进行比较，我们使用随机输入$\mathbf{g}_{init}$ 和 ϕ_A，在 Wang 等人用于测试的 4 个 PROX 三维场景中总共生成 50 个动作序列 [64]。由于我们的方法基于测试时优化，不需要显式划分训练集和测试集，因此动作控制器会针对每个输入单独优化，并且在策略优化之前不会获得输入的先验信息。优化一个策略需要 4 到 20 分钟，而动作编辑模块中的优化需要 3 到 4 分钟（500 个周期）。我们只考虑移动和动作（行走到坐下）动作序列，不包括操作，因为基线方法不处理操作。接触度量通过发生接触时足部的位置差异来衡量，其中接触状态根据足部速度自动标注。为公平地计算穿透度量，我们将 Wang 等人和 SAMP 输出的 SMPL-X 转换为与我们的方法相同的盒状骨架，并统计交点。表1 展示了结果。

物理合理性。如图所示，LAMA 在物理合理性方面优于 Wang 等人和 SAMP。Wang 等人对合成动作进行后处理，以确保接触并减少穿透，但 LAMA 的表现仍然更优。此外，与 SAMP 中基于路径规划的导航相比，我们基于强化学习并结合动作匹配的方法在杂乱三维场景中的碰撞规避方面展现出优势。

自然度。为了开展感知研究，我们分别构建了两组数据，用于与 Wang 等人和 SAMP 进行比较；每组测试均由互不重叠的参与者完成。每组包含 50 个动作序列，每个序列收集 5 份独立回答进行比较。与 Wang 等人比较时，LAMA 获得 215 票，Wang 等人获得 35 票（相对比例为 16.27%）。与 SAMP 比较时，LAMA 获得 176 票，SAMP 获得 74 票（相对比例为 42.04%）。结果表明，我们的方法在自然度方面同样大幅优于基线方法。

4.3. 消融研究

动作控制器的消融研究。在定量消融实验中，我们比较了原始 LAMA 与不含碰撞奖励 r_{coli} 的 LAMA。消融研究在 5 个 PROX 场景中进行。在原始 LAMA 中，穿透

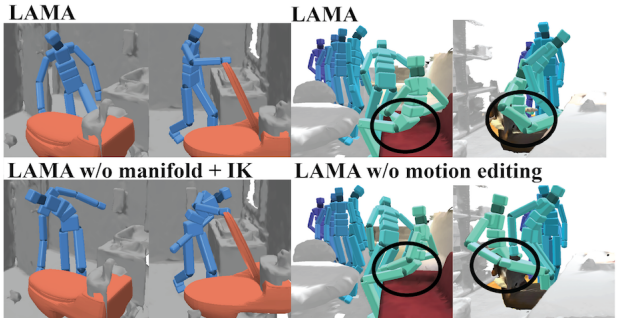

图 10：（a）角色打开马桶盖的对比：LAMA（上）与移除流形并以逆运动学替代的 LAMA（下）。（b）坐下时的对比：LAMA（上）与不含动作编辑的 LAMA（下）。

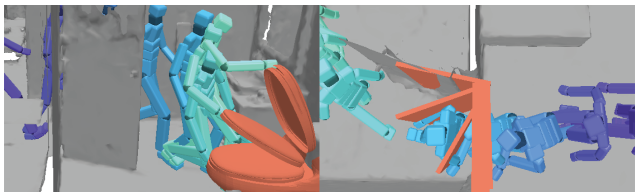

图 11：合成的操作动作示例。目标物体以橙色显示，角色在起始时为紫色，结束时为青色。左：行走并打开马桶盖。右：行走并开门。

仅发生在整个动作序列 **1.1%** 的帧中，而在不含 **15.7%** 的 LAMA 中，该比例为 r_{coli}。结果表明，r_{coli} 会促使动作控制器根据给定的三维场景合成动作。示例结果见图 8。我们还定性比较了动作控制器中其他组件的作用。如图 9所示，若没有动作偏移 $\mathbf{a}_t^{offset}$ 角色不会倾斜四肢以避免与物体或墙壁发生穿透，因为从动作数据库获取的原始运动不包含任何场景信息。这表明 $\mathbf{a}_t^{offset}$ 也有助于生成精细的场景感知姿势。此外，缺少平滑度奖励 $r_{\Delta t}$ 和 $r_{\Delta v}$ 时，生成的动作不够流畅，呈现出不自然且突兀的运动。

关于任务自适应动作编辑的消融研究。我们通过采用基于逆运动学的替代方法，替换动作编辑模块，对其进行消融实验。与 ϕ_M相同，我们仅将接触关节（例如右手）的轨迹提供给逆运动学求解器。如图10 （左）所示，采用动作编辑模块的 LAMA 展现出自然的动作，例如弯曲膝盖和倾斜髋部以实现接触。然而，逆运动学的结果姿势显得很不自然，因为逆运动学求解器无法体现自然人体运动中的时空关联。此外，如图10 （右）所示，动作编辑模块能让角色恰当地坐在不同形状的椅子上。

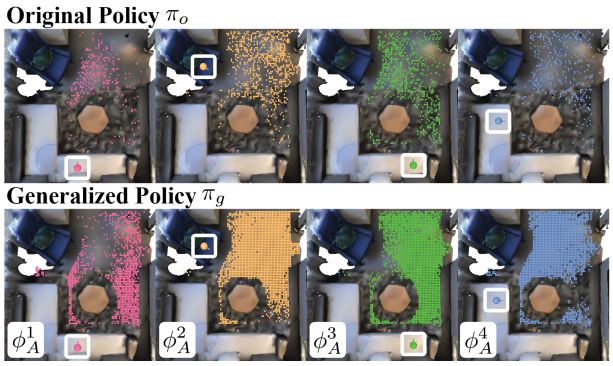

图 12：可视化展示策略能够覆盖的范围，以及泛化前后的差异。彩色点表示初始起点$\mathbf{g}_{init}$，策略可从这些起点合成符合动作提示（白色）的动作。

4.4. 动作控制器鲁棒性测试

如第3.5节所述，在测试时利用强化学习进行优化，可使优化策略处理输入变化。在本实验中，我们旨在衡量针对单一任务 ϕ_A 和初始状态的$_{init}$ 策略对不同输入的泛化能力。为测试其对不同初始状态和任务的鲁棒性，我们将优化策略应用于场景中的所有可能输入变化，并统计策略成功合成动作的输入数量。在采样得到的所有可能初始状态中，图12中的彩色点标示了策略能够合成满足给定动作提示 ϕ_A的动作的初始位置。如表2所示，最初针对一组输入（图12中的红色）进行优化的策略，无需额外优化，即使面对截然不同的输入，也能成功合成动作。此外，为测试泛化策略（见第3.5节）的鲁棒性，我们对优化过程中采用增强策略训练的策略进行了相同测试。正如预期，它在面对输入变化时展现出更强的鲁棒性。

我们进一步展示了采用增强策略优化的策略在未见场景中的泛化能力。在场景$\mathbf{W}_0$（图12中的场景）优化的泛化策略（第3.5节）在来自 PROX [18] 的两个未见场景$\mathbf{W}_1$ 和$\mathbf{W}_2$ 中进行了测试，如图13所示。如表3所示，在场景$\mathbf{W}_0$ 中优化的泛化策略（第3.5节）在一定程度上可泛化到场景$\mathbf{W}_1$,，因为场景$\mathbf{W}_0$ 和$\mathbf{W}_1$ 具有相似的结构（沙发和椅子围绕着桌子）。然而，正如预期的那样，在完全不同的场景$\mathbf{W}_2$中进行测试时，泛化能力会下降。

请注意，由于无需进一步进行策略优化，这里的推理时间约为每个输入 0.2–3 秒。测试设置的详细信息见补充材料。

方法	ϕ_A^1	ϕ_A^2	ϕ_A^3	ϕ_A^4
原始策略	10.5%	25.9%	25.5%	13.9%
泛化策略	40.2%	81.7%	77.0%	71.7%

表2：**鲁棒性测试**。在场景中，优化策略成功合成满足ϕ_A^n的动作的$\mathbf{g}_{init}$ 占有效$\mathbf{g}_{init}$ 总数的比例（百分比）。

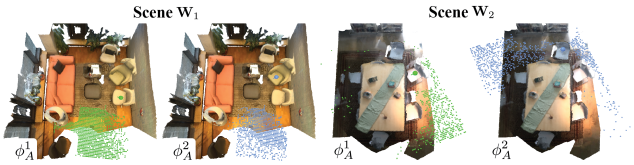

图13：在未见场景$\mathbf{W}_1$ 和$\mathbf{W}_2$上的鲁棒性测试。彩色点表示策略能够合成满足动作提示的动作的初始状态。

方法	$\mathbf{W}_1, \phi^1{}_A$	$\mathbf{W}_1, \phi^2{}_A$	$\mathbf{W}_2, \phi^1{}_A$	$\mathbf{W}_2, \phi^2{}_A$
-	70.5%	53.4%	11.9%	21.2%

表 3：**未见场景中的鲁棒性测试**。在场景 $\mathbf{W}_n$ 中，优化策略成功合成满足 ϕ_A^n 要求的动作的比例（占场景中所有有效$\mathbf{g}_{init}$ 的百分比）$\mathbf{g}_{init}$。

5. 讨论

我们提出了一种统一框架，仅使用动作数据集，即可在复杂的真实世界三维场景中合成人体动作。我们将其构建为一种测试时优化方法，利用结合动作匹配的强化学习来实现逼真的运动合成，并利用动作流形进一步覆盖多样化的操作行为。我们的方法已在多种场景中经过全面评估，表现优于以往方法 [64, 17]。

尽管测试时优化采用了强化学习，但单一策略能够涵盖输入的变化，也能够泛化到广泛的变化。将此框架与监督学习相结合以进一步提高效率，可能是一个有趣的未来研究方向。此外，虽然我们假设整个系统使用固定骨架，但交互运动可能会因角色的身体形状和尺寸而异。我们将针对不同身体形状合成动作留作未来工作。

致谢。本研究得到首尔大学- Naver 超大规模 AI 中心、首尔大学创意先锋研究人员计划、韩国政府（科学技术信息通信部）资助的韩国国家研究基金会项目（编号：2022 R1A2C2092724），以及韩国政府（科学技术信息通信部）资助的信息通信技术规划评估院项目（编号：2022-0-

00156 和编号：2021-0-01343）的支持。H. Joo 为通讯作者。

A. 补充视频

补充视频展示了我们的方法 LAMA 在各种场景中的结果。视频中，我们展示了在 PROX [18],Matterport3D [9], 以及使用 Polycam App 扫描的自有三维场景 [1]上的人体运动合成结果。我们使用 iPad Pro 和 SAPIEN [69]物体网格，以半自动方式生成操作提示，视频中也展示了这一过程。如视频所示，我们的方法能够在许多具有挑战性的场景中成功生成合理、自然的人体动作。

虽然我们原本的流程是为测试时优化而设计的，但视频还通过展示单一优化策略如何在无需进一步优化的情况下处理不同输入，定性地证明了我们框架在泛化场景中的优势。视频还展示了通过增强策略（第 3.5 节）优化的策略能够处理更多样的输入变化。

我们的补充视频通过展示多项消融研究，说明了方法中各部分的重要性：碰撞奖励 r_{coli} （公式 (4)）、转移奖励（$r_{\Delta \text{t}}$、 $r_{\Delta \text{v}}$）（公式 (8)）、动作控制器（第 3.2 节）中的姿态偏移 $\mathbf{a}_t^{offset}$，以及我们的动作编辑模块（第 3.5 节）；并将其与传统的逆运动学（IK）进行比较。我们还展示了与先前最先进方法的比较 [17, 63, 64]，并证明我们的结果能生成质量更高的动作，在复杂三维场景中具有更好的碰撞规避性能。

B. 实验的更多细节

本节进一步介绍我们在动作控制器鲁棒性测试（第4.4节）和感知研究（第4.2节）中的实验细节。

B.1. 动作控制器鲁棒性测试（第4.4节）

在主论文的第4.4节、图 10 和表 2 中，我们展示了针对特定输入优化的单个策略能够处理不同的目标动作 ϕ_A 和初始$\mathbf{g}_{init}$。我们在第 4.4 节中详细介绍了该实验。对于实验设置，我们考虑输入的所有可能变化，以测试针对特定输入训练的策略的泛化能力。具体而言，我们通过对地面平面进行网格采样，自动选取一组“所有”有效初始状态 $\{\mathbf{g}_{init}\}$，以确定位置$\mathbf{p}^0_{root}$，并排除被物体占据的点；身体朝向$\mathbf{r}^0_{root}$则随机设定。对于动作目标 $\{\phi_A\}$，我们手动选取多个合理的位置（例如椅子）作为动作目标。在图 10 使用的测试场景$\mathbf{W}_0$ 中，有 2635 个合理的初始位置；我们考虑了图 10 白色方框中显示的 4 个目标动作提示。

Name	足部接触	穿透
单一优化策略	4.16	1.35
泛化策略	4.47	1.41

表 4：鲁棒性测试合成动作的物理合理性测量（第 4.4 节）

原始策略 π_o（图 10 上方）针对特定输入$\mathbf{g}_{init}$ 和动作提示 ϕ_A^1进行了优化，它们在图 10 左上方以红色标出。图 10 中的彩色点显示了策略 π_o 无需进一步优化即可成功实现目标的位置。对于每组输入对$\mathbf{g}_{init}$ 和 ϕ_A^n，我们使用策略 π_o 5 次进行运动合成。在每次试验中，初始身体朝向均随机选择，以增加变化。若未触发任何提前终止条件（碰撞、停滞、移出场景），且在 5 次试验中至少有 2 次满足 ϕ_A^n，我们便判定策略在当前初始位置取得成功。如图 10 和表 2 所示，针对特定目标优化的动作控制器可以适用于许多输入变化。

我们在图 10 和表 2 下方，对泛化策略（第 3.5 节所述）进行了相同的测试。如图所示，该策略能够在同一场景中覆盖范围广得多的输入变化。

计算时间比较。作为一种无需场景配对动作数据集的测试时优化方法，对于给定的输入对，我们的原始框架需要从头开始训练策略。然而，将针对特定输入优化后的同一策略用于其他输入，可以大幅减少计算时间，因为无需进一步优化策略。为了比较仅执行推理与从头优化策略所需的时间，我们使用 5 组输入对进行测试，这些输入对由初始$\mathbf{g}_{init}$ 和 ϕ_A组成。这里，“仅推理”指使用预先优化的策略处理不同输入，而不进行任何进一步优化。结果表明，在仅推理的情况下，每组输入对的运动合成平均耗时**0.15** 秒；而对每组输入对从头优化策略，平均耗时**6.32** 分钟（379 秒）。由此可见，强化学习框架的能力有望大幅提升我们方法的效率。

动作质量测量。我们还在第4.4节的鲁棒性测试中评估合成动作的物理合理性。一个优化策略针对不同的输入对合成了15个动作序列（不包括该策略最初针对其进行优化的输入对）。我们还测量了采用增强策略优化的泛化策略所合成的动作。结果如表4所示。这表明，尽管策略能够处理输入变化，合成运动质量并未

Name	价值
策略网络学习率	2e-4
价值网络学习率	0.001
折扣因子 (γ)	0.95
广义优势估计和时序差分 (λ)	0.95
裁剪参数 (ϵ)	0.2
每次策略更新的元组数	30000
策略/价值更新的批量大小	512

表5：动作控制器**A**控制策略学习的超参数详情。

合成运动质量下降。

B.2. 感知研究设置

感知研究所用的视频见补充视频。我们在补充视频中为每组提供了3个视频。

C. 实现细节补充说明

C.1. 动作控制器

实现细节。 动作控制器模块的策略网络和值网络分别由 4 层和 2 层全连接层组成，每层包含 256 个节点。控制策略通过近端策略优化（PPO）算法进行优化 [54]。采用 Adam优化器 [28] ，并使用 Nvidia RTX 3090 GPU。对于动作控制器 **A**和动作合成器模块 **S**，我们使用动画库 DART [30]。我们还使用了一个公开可用的 PPO 实现 [45, 36], ，并按照原始 PPO 算法移除了其中的可变时间步进函数 [36] 。关于动作控制器策略网络和值网络的优化细节列于表 5。

加速技术。如正文所述，我们使用提前终止条件来加速策略优化。当出现以下情况时，回合将终止：(1) 角色移出场景边界框；(2) 碰撞奖励 r_{coli} 低于特定阈值；(3) 根节点速度在特定时长（50 帧）内低于特定阈值，以防止角色长时间静止。此外，动作控制器在从移动切换到其他动作时，会预先检查动作信号是否有效。当动作合成器（第 3.3 节）中的公式 2 所定义的最近特征距离超过特定阈值时，动作控制器会丢弃此次切换，并继续导航。

C.2. 动作合成器

动作数据库信息。动作由基于惯性测量单元的系统 XSens MVN Link 捕获 [3] ，并通过 XSens MotionCloud 软件进行后处理 [2]。随后，使用 Autodesk MotionBuilder 将捕获的动作重定向到单一统一骨架，并进行后处理，使其适用于动作匹配。对于动作片段，我们会镜像动作片段以进行数据增强。动作片段的长度（以帧计，表中的“片段长度”）、动作片段数量（表中的“片段数”）以及总帧数（表中的“总帧数”）汇总于表 6。

动作特定特征定义。 如论文正文第 3.3 节所定义，动作特征同时表示动作的当前状态和短期未来运动： $f(\mathbf{m}) = \{\{p_{\mathrm{j}}\}, \{\dot{p}_{\mathrm{j}}\}, \theta_{up}, c, \mathbf{o}_{\mathrm{future}}\}$。具体而言，动作特定特征 $\mathbf{o}_{\mathrm{future}} = \{\{p_0^{\Delta t}\}, \{r_0^{\Delta t}\}\}$包含未来动作，使动作搜索过程能够考虑未来动作一致性，其中$p_0^{\Delta t}$, $r_0^{\Delta t} \in \mathbb{R}^2$ 分别表示当前目标帧之后 Δt 帧处根关节的位置和方向。对于移动，我们提取未来 $\Delta t = 10$、20 和 30 帧（30Hz），遵循 [11],的设定，详见论文正文。对于坐下，我们特别选择 Δt 作为角色完成坐下动作的帧。该设计的主要动机是促使动作合成器搜索包含所需目标动作的动作片段。

C.3. 通过动作流形进行动作编辑

模型与训练的实现细节。 任务自适应动作编辑模块的编码器和解码器由三个卷积层组成。对于任务自适应动作编辑的卷积自编码器，我们使用 PyTorch [46], FairMotion [14], 和 PyTorch3d [52]。自编码器使用 Adam优化器进行训练 [28]，学习率为 0.0001。我们使用 Nvidia RTX 3090 GPU。我们采用 3 层一维时间卷积，卷积核宽度为 25，步幅为 2，每个输出特征的通道维度为 256。训练任务自适应动作编辑中的自编码器模块时，我们使用 Mixamo [44],Lafan1 [16], COUCH [76], 和我们自己的数据。训练数据集汇总于表 7。请注意，用于训练自编码器的数据也不包含与场景相关的信息（BVH格式），并且我们在动作编辑模块与动作合成器之间采用了不同的预处理步骤。

重建损失。 编码器 Ψ 和解码器 Ψ^{-1}基于重建损失 $\mathcal{L}_{\mathrm{recon}} = ||\mathbf{X}$ –进行训练

$\Psi^{-1}(\Psi(\mathbf{X}))||^2$，其中：

$$\mathcal{L}_{\text{recon}} = w_c\mathcal{L}_{\text{contact}} + w_r\mathcal{L}_{\text{root}} + w_q\mathcal{L}_{\text{quat}} + w_p\mathcal{L}_{\text{pos}}. \quad (9)$$

$\mathcal{L}_{\text{contact}}$、 $\mathcal{L}_{\text{root}}$和 $\mathcal{L}_{\text{quat}}$ 分别是足部接触标签、根节点状态（高度以及相对于前一帧并投影到XZ平面的变换）和六维表示中的关节旋转的均方误差损失 [80]。为了惩罚沿运动学链累积的误差，我们执行正向运动学（FK），并测量原始动作与重建动作中关节的全局位置距离。由于关节的全局位置高度依赖根节点位置，在训练早期，我们基于以根节点为中心的坐标测量距离，以忽略根节点的全局位置；我们通过经验发现，这样更加稳定。此外，在训练过程中，我们使用了一种数据增强技术，即向归一化后的输入添加噪声。噪声从正态分布 $\mathcal{N}(0,\ 1)$ 中采样，并乘以0.01的尺度。我们通过经验发现，该技术能够提高重建动作的质量。

动作编辑损失。 对于动作编辑，位置损失和正则化损失定义如下。

$$\begin{aligned}
\mathcal{L} &= w_p\mathcal{L}_{\text{pos}} + w_f\mathcal{L}_{\text{foot}} + w_r\mathcal{L}_{\text{root}}, \quad \text{where} \\
\mathcal{L}_{\text{pos}} &= \sum_{\mathbf{q}_j^t \in \phi_M} \|\mathbf{p}_j^t - \mathbf{q}_j^t\|^2, \\
\mathcal{L}_{\text{foot}} &= \sum_{foot} \|\mathbf{p}_{foot}^e - \mathbf{p}_{foot}^i\|^2, \\
\mathcal{L}_{\text{root}} &= w_r\|\mathbf{r}_{\text{xz}}^e - \mathbf{r}_{\text{xz}}^i\|^2 + w_{\Delta r}\|\dot{\mathbf{r}}_{\text{xz}}^e - \dot{\mathbf{r}}_{\text{xz}}^i\|^2.
\end{aligned} \quad (10)$$

$\mathbf{p}^j$ 表示关节 j的位置，$\mathbf{r}$和 $\dot{\mathbf{r}}$ 分别表示根节点的位置和速度。上标 e 和 i 分别表示其来自编辑后的动作还是初始动作。下标xz表示该向量投影到XZ平面上。损失项 $\mathcal{L}$ 促使编辑后的动作保持初始动作的接触状态和根节点轨迹（在XZ平面上），同时生成其他关节的自然动作，以满足稀疏位置约束。为最小化损失，同样使用Adam优化器 [28]，学习率为0.005。

从 SAPIEN 生成操作提示 [69]。虽然操作提示 $\phi_M = [\mathbf{v}(R_t,\ T_t,\ \theta_t)]_t$ 可根据应用场景通过多种方式提供，但我们主要考虑与可动对象交互的场景。为此，我们从 SAPIEN数据集 [69]中提取可动对象各部分所需的目标顶点轨迹，以半自动方式生成操作提示。具体而言，我们将目标物体放置在三维场景中，并选择物体上的一个目标顶点$\mathbf{v}$，假设角色的手会接触该顶点以操作目标部件（例如，垃圾桶盖上的一个顶点）。随后，通过以固定间隔改变可动对象的关节运动 θ 参数，即可获得该顶点$\phi_M = [\mathbf{v}(R_t,\ T_t,\ \theta_t)]_t$ 的轨迹。其中， R_t、 T_t分别表示物体的全局朝向和平移，θ_t 表示物体在时间t的关节参数（例如笔记本电脑盖子的铰链角度）。$\mathbf{v}(\cdot)$表示给定参数时所选顶点$\mathbf{v}$的三维位置。由此得到的操作提示 ϕ_M 是手部关节在操作动作中应遵循的目标轨迹。请注意，我们的系统只需要操作提示 ϕ_M；三维物体网格仅用于可视化，我们会将其与同步的 θ_t 一同展示。

标签	片段长度	片段数量	总帧数
移动	10	23832	24267
Sit	50 – 85	6230	15130

表 6：动作数据库中每个动作类别对应的预处理运动数据集详情**S**。

Name	值
运动序列长度	120
序列数量（训练）	45713
序列数量（测试）	11040
序列数量（验证）	5268

表7：用于训练动作编辑模块的预处理运动数据集详情**M**。

操作的更多实现细节给定由动作控制器合成的初始运动输出M synthesized from the Ac- tion Controller, $\mathbf{M} = \{\mathbf{m}_t\}_{t=1}^T$ 以及操作提示$\phi_M = \{\mathbf{m}_{t'}\}_{t'=1}^{\tau}$，系统还会获得相应的时间段 $[t_i, t_f]$，在此时间段内，我们希望编辑动作，使其遵循操作提示（假设持续时间相同，即 $t_f - t_i = \tau$）。随后，针对目标运动片段执行动作编辑~，即$\mathbf{M}_{t_i:t_f} = \mathbf{E}(\mathbf{M}_{t_i:t_f})$，并用其替换$\mathbf{M}$ 中的对应部分，从而得到$\mathbf{M}$作为最终输出。

根据可能的应用场景（例如坐下并打开笔记本电脑），可能需要在合成动作$\mathbf{M}$的中间或末尾“添加”操作动作。在这种情况下，我们只需将目标帧重复 τ，以构建更长的动作$\mathbf{M} = \{\mathbf{m}_t\}_{t=1}^{T+\tau}$，并对通过重复生成的静止运动这一目标运动片段进行动作编辑。

参考文献

[1] Polycam——适用于 iPhone 和 Android 的激光雷达与 3D 扫描仪。https://poly.cam/. 10[2]Xsens MotionCloud。https://www. movella.com/products/motion-capture/xsens-motioncloud. 11[3]Xsens MVN Link。https://www.movella.com/products/motion-capture/xsens-mvn-link. 6, 11[4] Kfir Aberman、Peizhuo Li、Dani Lischinski、Olga Sorkine-Hornung、Daniel Cohen-Or 和 Baoquan Chen。用于深度运动重定向的骨架感知网络。ACM 图形学汇刊，39(4)，2020年。3[5] Kfir Aberman、Yijia Weng、Dani Lischinski、Daniel Cohen- Or 和 Baoquan Chen。将运动风格从视频迁移到动画的无配对方法。ACM 图形学汇刊，39(4)，2020年。3[6] Kevin Bergamin、Simon Clavet、Daniel Holden 和 James Richard Forbes。Drecon：基于数据驱动的基于物理的角色响应式控制。ACM 图形学汇刊，38(6)，2019年。3[7] Michael B¨uttner 和 Simon Clavet。动作匹配——迈向新一代动画之路。载于 Nucl.ai 会议论文集，2015年。2, 3, 4[8] Zhe Cao、Hang Gao、Karttikeya Mangalam、Qi-Zhi Cai、Minh Vo 和 Jitendra Malik。利用场景上下文进行长期人体动作预测。载于 ECCV，2020年。2[9] Angel Chang、Angela Dai、Thomas Funkhouser、Maciej Hal- ber、Matthias Niessner、Manolis Savva、Shuran Song、Andy Zeng 和 Yinda Z hang。Matterport3D：在室内环境中从 RGB-D数据中学习。载于 3D视觉会议，2017年。6, 10[10] Yu-Wei Chao、Jimei Yang、Weifeng Chen 和 Jia Deng。学习坐下：通过分层控制合成人-椅交互。载于人工智能促进协会年会，2021年。2[11] Simon Clavet。动作匹配与迈向新一代动画之路。载于游戏开发者大会论文集，2016年。2, 3, 4, 11[12] Haegwang Eom、Daseong Han、Joseph S Shin 和 Junyong Noh。基于物理的角色动画中结合视觉运动系统的模型预测控制。ACM 图形学汇刊，39(1)，2019年。1, 2[13] Katerina Fragkiadaki、Sergey Levine、Panna Felsen 和 Ji- tendra Malik。用于人体动力学的循环网络模型。载于 ICCV，2015年。2[14] Deepak Gopinath 和 Jungdam Won。FairMotion——用于加载、处理和可视化动作捕捉数据的工具。GitHub，2020年。11[15] Ikhsanul Habibie、Daniel Holden、Jonathan Schwarz、Joe Yearsley 和 Taku Komura。用于人体运动合成的循环变分自编码器。载于英国机器视觉会议，2017年。2[16] F´elix G Harvey、Mike Yurick、Derek Nowrouzezahrai 和 Christopher Pal。稳健的运动中间帧生成。ACM图形学汇刊，39(4)，2020年。2, 3, 11[17] Mohamed Hassan、Duygu Ceylan、Ruben Villegas、Jun Saito、Jimei Yang、Yi Zhou 和 Michael Black。随机的场景感知动作预测。载于 ICCV，2021年。1, 2, 3, 6, 7, 9, 10[18] Mohamed Hassan、Vasileios Choutas、Dimitrios Tzionas 和 Michael J. Black。解决三维人体姿态歧义具有三维场景约束的能力。在 ICCV，2019年。1, 2, 6, 9, 10[19] Mohamed Hassan、Partha Ghosh、Joachim Tesch、Dimitrios Tzionas 和 Michael J Black。通过学习人-场景交互来填充三维场景。在 CVPR，2021年。1, 2[20] Daniel Holden、Oussama Kanoun、Maksym Perepichka 和 Tiberiu Popa。学习式动作匹配。ACM 图形学汇刊，39(4)，2020年。4[21]Daniel Holden、Taku Komura 和 Jun Saito。用于角色控制的相位函数神经网络。ACM 图形学汇刊，36(4)，2017年。2, 3[22] Daniel Holden、Jun Saito 和 Taku Komura。用于角色动作合成与编辑的深度学习框架。ACM 图形学汇刊，35(4)，2016年。2, 3, 6[23] Chun-Hao P Huang、Hongwei Yi、Markus H¨oschle、Matvey Safroshkin、Tsvetelina Alexiadis、Senya Polikovsky、Daniel Scharstein 和 Michael J Black。捕捉并推断稠密全身人-场景接触。在 CVPR，2022年。2[24] Kyunglyul Hyun、Kyungho Lee 和 Jehee Lee。角色动画的动作语法。在《计算机图形论坛》，第35卷，2016年。2[25] Nan Jiang、Tengyu Liu、Zhexuan Cao、Jieming Cui、Yixin Chen、He Wang、Yixin Zhu 和 Siyuan Huang。椅子：迈向全身铰接人-物交互。arXiv 预印本 arXiv:2212.10621，2022年。2[26] Yuheng Jiang、Suyi Jiang、Guoxing Sun、Zhuo Su、Kaiwen Guo、Minye Wu、Jingyi Yu 和 Lan Xu。Neuralhofusion：人-物交互下的神经体积渲染。在 CVPR，2022年。2[27] Vladimir G Kim、Siddhartha Chaudhuri、Leonidas Guibas 和 Thomas Funkhouser。Shape2pose：以人为中心的形状分析。ACM 图形学汇刊，33(4)，2014年。1, 2[28] Diederik P Kingma 和 Jimmy Ba。Adam：一种随机优化方法。arXiv 预印本 arXiv:1412 6980，2014年 11, 12 . . [29] Jehee Lee 、 Jinxiang Chai 、 Paul SA Reitsma 、 Jessica K Hod-gins 和 Nancy S Pollard。使用人体动作数据驱动动画化身的交互式控制。在第29届计算机图形学与交互技术年会论文集，2002年。2[30] Jeongseok Lee、Michael X Grey、Sehoon Ha、Tobias Kunz、Sumit J ain、Yuting Ye、Siddhartha S Srinivasa、Mike Stilman 和 C Karen Liu。DART：动态动画与机器人学工具包。《开放源代码软件期刊》，3(22)，2018年。11[31] Jehee Lee 和 Kang Hoon Lee。根据人体动作数据预计算化身行为。在2004年 ACM SIGGRAPH/Eurographics 动画研讨会论文集，2004年。3[32] Kyungho Lee、Seyoung Lee 和 Jehee Lee。通过学习多目标控制实现交互式角色动画。ACM 图形学汇刊，37(6)，2018年。2, 3[33] Kyungho Lee、Sehee Min、Sunmin Lee 和 Jehee Lee。为交互式角色控制学习时间关键响应。ACM 图形学汇刊，40(4)，2021年。3[34] Kang Hoon Lee、Myung Geol Choi 和 Jehee Lee。动作补丁：构建以动作数据标注的虚拟环境的基本模块

。在 ACM SIGGRAPH 2006 论文集。2006年。2[35] Seunghwan Lee、Phil Sik Chang 和 Jehee Lee。深度柔顺控制。在 ACM SIGGRAPH 2022 会议论文集，2022年。2, 3[36] Seyoung Lee、Sunmin Lee、Yongwoo Lee 和 Jehee Lee。从单个动作片段中学习一系列运动技能。ACM 图形学汇刊，40(4)，2021年。3, 11[37] Seunghwan Lee、Moonseok Park、Kyoungmin Lee 和 Jehee Lee。可扩展的肌肉驱动人体模拟与控制。ACM 图形学汇刊，38(4)，2019年。5[38] Sergey Levine、Jack M Wang、Alexis Haraux、Zoran Popovi´c 和 Vladlen Koltun。使用低维嵌入的连续角色控制。ACM 图形学汇刊，31(4)，2012年。3[39] Ruilong Li、Shan Yang、David A. Ross 和 Angjoo Kanazawa。AI 编舞师：基于音乐条件的三维舞蹈生成，使用 AIST++。在 ICCV，2021年。2[40] Hung Yu Ling、Fabio Zinno、George Cheng 和 Michiel Van De Panne。使用动作变分自编码器的角色控制器。ACM 图形学汇刊，39(4)，2020年。2, 3[41] Kovar Lucas、Gleicher Michael 和 Pighin Fr´ed´eric。动作图。在第29届计算机图形学与交互技术年会论文集，2002年。2[42] Julieta Martinez、Michael J Black 和 Javier Romero。关于使用循环神经网络进行人体动作预测。在 CVPR，2017年。2[43] Josh Merel、Saran Tunyasuvunakool、Arun Ahuja、Yuval Tassa、Leonard Hasenclever、Vu P ham、Tom Erez、Greg Wayne 和 Nicolas Heess。捕捉与搬运：用于视觉引导全身任务的可复用神经控制器。ACM 图形学汇刊，39(4)，2020年。2[44] Adobe 的 Mixamo。https://www.mixamo.com，2017年。11[45] Soohwan Park、Hoseok Ryu、Seyoung Lee、Sunmin Lee 和 Jehee Lee。从无序人体动作数据中学习预测与模拟策略。ACM 图形学汇刊，38(6)，2019年。2, 3, 5, 11[46] Adam Paszke、Sam Gross、Francisco Massa、Adam Lerer、James Bradbury、Gregory Chanan、Trevor Killeen、Zeming Lin、Natalia Gimelshein、Luca Antiga、Alban Desmaison、Andreas Kopf、Edward Yang、Zachary DeVito、Martin Rai-son、Alykhan Tejani、Sasank Chilamkurthy、Benoit Steiner、Lu Fang、Junjie Bai 和 Soumith Chintala。PyTorch：一种命令式风格的高性能深度学习库。在《神经信息处理系统进展》第32卷。2019年。11[47] Xue Bin Peng、Pieter Abbeel、Sergey Levine 和 Michiel Van de Panne。DeepMimic：通过示例引导、基于物理的角色技能深度强化学习。ACM 图形学汇刊，37(4)，2018年。3, 5[48] Xue Bin Peng、Yunrong Guo、Lina Halper、Sergey Levine 和 Sanja Fidler。ASE：面向物理模拟角色的大规模可复用对抗技能嵌入。ACM 图形学汇刊，41(4)，2022年。3[49] Xue Bin Peng、Ze Ma、Pieter Abbeel、Sergey Levine 和 Angjoo Kanazawa。AMP：用于风格化的对抗动作先验基于物理的角色控制。ACM图形学汇刊，40(4)，2021年。3[50] Mathis Petrovich、Michael J Black 和 G¨ul Varol。使用 Transformer 变分自编码器的动作条件3D人体运动合成。载于 ICCV，2021年。2，3[51] Yuzhe Qin、Yueh-Hua Wu、Shaowei Liu、Hanwen Jiang、Rui-han Yang、Yang Fu 和 Xiaolong Wang。DexMV：从人类视频中进行灵巧操作的模仿学习。载于 ECCV，2022年。1，2[52]Nikhila Ravi、Jeremy Reizenstein、David Novotny、Taylor Gordon、Wan-Yen Lo、Justin Johnson 和 Georgia Gkioxari。使用 PyTorch3D 加速3D深度学习。arXiv:2007.08501，2020年。11[53] Manolis Savva、Angel X Chang、Pat Hanrahan、Matthew Fisher 和 Matthias Nießner。PiGraphs：从观察中学习交互快照。ACM 图形学汇刊，35(4)，2016年。1，2[54] John Schulman、Filip Wolski、Prafulla Dhariwal、Alec Radford 和 Oleg Klimov。近端策略优化算法。arXiv 预印本 arXiv:1707.06347，2017年。11[55] Hubert PH Shum、Taku Komura、Masashi Shiraishi 和 Shuntaro Yamazaki。用于多角色动画的交互补丁。ACM 图形学汇刊，27(5)，2008年。2[56] Sebastian Starke、Ian Mason 和 Taku Komura。DeepPhase：用于学习运动相位流形的周期自编码器。ACM 图形学汇刊，41(4)，2022年。2[57] Sebastian Starke、He Zhang、Taku Komura 和 Jun Saito。用于角色-场景交互的神经状态机。ACM 图形学汇刊，38(6)，2019年。1，2[58] Omid Taheri、Vasileios Choutas、Michael J Black 和 Dimitrios Tzionas。GOAL：生成用于手-物体抓取的4D全身运动。载于 CVPR，2022年。1，2[59] Omid Taheri、Nima Ghorbani、Michael J Black 和 Dimitrios Tzionas。GRAB：全身人体抓取物体的数据集。载于 ECCV，2020年。1，2[60] Graham W Taylor 和 Geoffrey E Hinton。用于建模运动风格的因子化条件受限玻尔兹曼机。载于 ICML，2009年。2[61] Adrien Treuille、Yongjoon Lee 和 Zoran Popovi´c。通过连续控制实现近似最优的角色动画。载于 ACM SIGGRAPH 2007 论文集。2007年。3[62] Ruben Villegas、Jimei Yang、Yuliang Zou、Sungryull Sohn、Xunyu Lin 和 Honglak Lee。通过层次化预测学习生成长期未来。载于 ICML，2017年。2[63] Jingbo Wang、Yu Rong、Jingyuan Liu、Sijie Yan、Dahua Lin 和 Bo Dai。迈向多样且自然的场景感知3D人体运动合成。载于 CVPR，2022年。1，2，10[64] Jiashun Wang、Huazhe Xu、Jingwei Xu、Sifei Liu 和 Xiaolong Wang。在三维场景中合成长时段3D人体运动与交互。载于 CVPR，2021年。1，2，6，7，8，9，10[65] Jingbo Wang、Sijie Yan、Bo Dai 和 Dahua Lin。用于人体运动合成的场景感知生成网络。载于 CVPR，2021年。2，3[66] Zan Wang、Yixin Chen、Tengyu Liu、Yixin Zhu、Wei Liang 和 Siyuan Huang。Humanise：三维场景中的语言条件人体运动生成。NeurIPS，2022年。2

[67] Jungdam Won、Deepak Gopinath 和 Jessica Hodgins。控制物理模拟角色多样行为的可扩展方法。ACM 图形学汇刊，39(4)，2020年。3[68] Yan Wu、Jiahao Wang、Yan Zhang、Siwei Zhang、Otmar Hilliges、Fisher Yu 和 Siyu Tang。Saga：带接触的随机全身抓取。载于 ECCV，2022年。2[69] Fanbo Xiang、Yuzhe Qin、Kaichun Mo、Yikuan Xia、Hao Zhu、Fangchen Liu、Minghua Liu、Hanxiao Jiang、Yifu Yuan、He Wang 等。SAPIEN：仿真部件交互环境。载于 CVPR，2020 年。6，10，12[70] Xianghui Xie 、 Bharat Lal Bhatnagar 和 Gerard Pons-Moll。Chore：从单张 RGB图像中进行接触、人体与物体重建。载于 ECCV，2022年。1，2[71] Xiang Xu、Hanbyul Joo、Greg Mori 和 Manolis Savva。D3D-HOI：从视频中获取动态三维人-物交互。arXiv 预印本 arXiv:2108.08420，2021年。2[72] Zeshi Yang、Kangkang Yin 和 Libin Liu。学习以多样的抓握风格使用筷子。ACM 图形学汇刊，41(4)，2022年。1，2[73] He Zhang、Yuting Ye、Takaaki Shiratori 和 Taku Komura。ManipNet：基于手-物空间表征的神经操控合成。ACM 图形学汇刊，40(4)，2021年。1，2[74] Jason Y. Zhang、Sam Pepose、Hanbyul Joo、Deva Ramanan、Jitendra Malik 和 Angjoo Kanazawa。从野外场景中的单张图像感知三维人-物空间布局。载于 ECCV，2020年。2[75] Siwei Zhang、Yan Zhang、Qianli Ma、Michael J Black 和 Siyu Tang。PLACE：三维环境中的关节运动与接触邻近性学习。载于 3D视觉会议，2020年。1，2[76] Xiaohan Zhang、Bharat Lal Bhatnagar、Sebastian Starke、Vladimir Guzov 和 Gerard Pons-Moll。COUCH：迈向可控的人-椅交互。载于 ECCV，2022年。1，2，11[77] Yan Zhang、Mohamed Hassan、Heiko Neumann、Michael J Black 和 Siyu Tang。在没有人物的场景中生成人物。载于 CVPR，2020年。1，2[78] Kaifeng Zhao、Shaofei Wang、Yan Zhang、Thabo Beeler、，和 Siyu Tang。带语义控制的组合式人-场景交互合成。载于 ECCV，2022年。1，2[79] Keyang Zhou、Bharat Lal Bhatnagar、Jan Eric Lenssen 和 Gerard Pons-Moll。TOCH：用于运动细化的时空物体到手部的对应关系。载于 ECCV，2022年。2[80] Yi Zhou、Connelly Barnes、Lu Jingwan、Yang Jimei 和 Li Hao。关于神经网络中旋转表示的连续性。载于 CVPR，2019年。4，6，12