arXiv:2301.13868v1 [cs.LG] 31 Jan 2023

PADL：语言引导的基于物理的角色控制

Jordan Juravsky 英伟达 滑铁卢大学 加拿大
jjuravsky@nvidia.com

Yunrong Guo 英伟达 加拿大
kellyg@nvidia.com

Sanja Fidler 英伟达 多伦多大学 加拿大
sfidler@nvidia.com

Xue Bin Peng 英伟达 西蒙弗雷泽大学 加拿大
japeng@nvidia.com



（a）技能指令："跳跃并向下挥剑"。 (b) 技能指令："向前盾牌冲锋"。

图 1：我们的框架允许用户通过自然语言指令来引导物理模拟角色的行为。左：类人角色正在施展跳跃攻击。右：角色通过盾牌冲锋撞倒目标物体。

摘要

如何为模拟角色合成自然、逼真的动作，长期以来一直是计算机动画领域关注的重点。但要让这类系统适用于下游应用，它们不仅需要生成高质量动作，还必须提供易于使用且灵活多样的界面，让用户能够引导角色的行为。自然语言是一种简单易用且富有表现力的媒介，可用于表达用户意图。自然语言处理（NLP）领域近期的突破，已证明基于语言的界面可有效用于图像生成、程序合成等应用。在这项工作中，我们提出 PADL，借助自然语言处理领域的最新进展，探索开发用于基于物理的角色动画的语言驱动控制器。PADL 允许用户通过自然语言指令，指定角色应执行的高层任务和低层技能。我们提出一种对抗模仿学习方法，用于训练策略，将高级语言指令映射为低层控制，使角色能够执行用户指令所指定的任务和技能。此外，我们提出一种多任务聚合方法，利用基于语言的多项选择问答方法，根据语言指令确定高层任务目标。我们展示了，该框架能够有效引导模拟类人角色完成多种复杂的运动技能。

允许出于个人或课堂用途，免费制作本作品全部或部分内容的数字版或纸质副本，前提是副本不用于或分发以牟利或商业利益为目的，且副本须保留本声明及首页的完整引文。对于本作品中由 ACM 以外方所有的组件，必须尊重其版权。注明出处后可以摘录。若要以其他方式复制、再版、发布到服务器或分发至列表，须事先获得具体许可和/或支付费用。请通过 permissions@acm.org 申请许可。*SA '22* 会议论文集，*2022*年*12*月*6*日至*9*日，大邱，大韩民国© 2022年 美国计算机协会。
ACM ISBN 978-1-4503-9470-3/22/12...$15.00
https://doi.org/10.1145/3550469.3555391

CCS 概念

• **计算方法** → **程序化动画**；对抗学习。

关键词

角色动画，语言指令，强化学习，对抗模仿学习

ACM 引用格式：
Jordan Juravsky、Yunrong Guo、Sanja Fidler 和 Xue Bin Peng。2022年。PADL：语言引导的基于物理的角色控制。载于*SIGGRAPH Asia 2022* 会议论文集（*SA'22* 会议论文集），*2022*年*12*月*6*日至*9*日，大韩民国大邱。ACM，纽约，美国，12页。https://doi.org/10. 1145/3550469.3555391

1 引言

开发能够产生复杂且逼真行为的物理仿真角色，一直是计算机动画领域的核心挑战之一。该领域的研究催生了能够为多种技能生成高质量动作的系统 [Clegg 等人，2018；de Lasa 等人，2010；Hodgins 等人，1995；Lee 等人，2010a；Liu 和 Hodgins，2018；Liu 等人，2016；Mordatch 等人，2012；Peng 等人，2018a；Tan 等人，2014；Wang 等人， 2009]。然而，要使这些系统能够用于下游应用，控制模型不仅需要生成高质量动作，还应提供易于使用且灵活多样的界面，以便用户引导角色的行为。通常，这类界面通过简洁的控制抽象实现，例如操纵杆控制或目标航点。这些控制抽象允许用户通过高级指令轻松引导角色行为，但也可能大幅限制用户能够主动控制的行为种类与细致程度。另一种方式是采用动作跟踪模型，它能够通过直接指定目标运动轨迹，为角色的运动提供灵活且精细的控制界面。然而，制作运动轨迹是一个劳动密集型过程，需要丰富的领域专业知识或专用设备（例如动作捕捉）。

理想的动画系统应提供易于使用的界面，让用户能够轻松指定角色期望表现的行为，同时还应具备足够的灵活性，以便控制丰富多样的技能。自然语言是一种前景广阔的媒介，兼具易用性与灵活性。近年来，大型且表达能力强的语言模型为整合自然语言界面提供了强大工具，可应用于各种下游应用 [Brown 等人，2020年；Devlin等，2018；Radford 等人 2021], ，例如根据自然语言描述生成可运行的代码和逼真的图像 [Chen 等人，2021年；Ramesh 等人，2022年；Tan 等人，2018]。在这项工作中，我们旨在利用自然语言处理领域的这些技术，推动基于物理的角色动画语言驱动系统的开发。

这项工作的核心贡献是一个用于语言引导的基于物理的角色动画系统，使用户能够使用自然语言指令来控制物理仿真角色的行为。给定一个动作片段及其说明文字组成的数据集（这些说明文字描述了每个片段中呈现的行为），我们的系统会训练控制策略，将高级语言指令映射为低级运动指令，从而使角色能够重现相应的技能。我们提出了一种对抗模仿学习方法，使策略既能重现多种多样的技能，又能学习如何将每项技能与语言指令对应起来。我们的策略还可以接受训练，以完成其他辅助任务。我们提出了一种基于语言的多任务聚合模型，该模型根据给定的指令，在一组特定任务的策略中进行选择，从而使用户能够通过自然语言轻松指挥角色执行各种高层任务。我们提出了首批能够有效利用语言指令，指导全身物理仿真角色执行多种复杂运动技能的系统之一。本工作的代码可在 https://github.com/nv-tlabs/PADL 获取。

2 相关工作

为模拟角色合成自然且智能的行为，一直是计算机动画领域的核心研究课题。大量研究致力于构建运动学和基于物理的控制模型，以生成逼真的动作 [Clegg et al. 2018; da Silva et al. 2008; Hodgins et al. 1995; Holden et al. 2016; Lee et al. 2010a; Liu and Hodgins 2018; Tan et al. 2014; Wang et al. 2009, 2012]。尽管研究者高度重视动作质量，却很少关注这些模型在运行时的可操控性 。通常通过控制抽象将可操控性融入这些模型，让用户能够通过高级指令操控角色行为。这些抽象往往需要在易用性和灵活性之间取舍。简单的控制抽象，例如操纵杆指令或目标航点， [Agrawal and van de Panne 2016; Coros et al. 2009; Holden et al. 2017; Lee et al. 2021b,a, 2010b; Ling et al. 2020; Peng et al. 2018a, 2022, 2021; Starke et al. 2019; Treuille et al. 2007; Zhang et al. 2020], 提供了易于用户采用的直观界面。但这些抽象也可能限制用户能够主动操控的行为类型。另一种选择是通用运动跟踪模型，它们能提供灵活的界面，让用户对角色动作进行精细控制，通过目标运动轨迹操控角色的动作 [Bergamin et al. 2019; Park et al. 2019; Pollard et al. 2002; Wang et al. 2020; Won et al. 2020; Yamane et al. 2010]。这些目标轨迹规定了角色在每个时间步应达到的姿势，原则上能够引导角色完成任何可行的动作。然而，这种灵活性往往以牺牲易用性为代价，因为编制目标运动轨迹可能和手动制作关键帧动画一样繁琐、耗费人力。动作捕捉可以更快捷地为运动跟踪模型生成目标轨迹 [Peng et al. 2018b; Wang et al. 2020; Yu et al. 2021; Yuan et al. 2021], ，但通常需要专用设备，而且可能会将可复现的行为限制为用户能够亲身完成的动作。在本研究中，我们旨在利用自然语言，为基于物理的角色动画开发一种兼具易用性和灵活性的控制界面。

自然语言处理：研究表明，在日益庞大的数据集上训练的语言模型能够为文本数据形成强大的表征 [Devlin等，2018；Liu等，2019；Raffel等， 2019], ，并可用于各种下游应用。其中一个例子是文本引导合成：用户可以用自然语言表达提示，引导模型生成不同类型的内容。大型自回归模型能够根据用户提供的起始提示生成连贯的文本补全 [Brown等， 2020]。这些模型推动了“提示工程”的普及，其目标是构造最优提示模板，以引发语言模型所需的行为。此类基于提示的系统通常结合筛选或其他后处理技术，已成功用于解决小学数学题和竞赛编程挑战 [Cobbe等，2021；Li等，2022]。文本引导合成也可应用于不同模态。在这种情况下，语言模型不会直接生成所需内容，而是为用户的语言提示提供语义上有意义的编码，随后由单独训练的解码器生成另一种模态的内容。Nichol等， [2021] 和 Ramesh等， [2022] 成功运用这种方法，从自然语言生成逼真的图像，并借助了CLIP的文本编码器 [Radford等，2021]。在本研究中，我们旨在利用强大的语言模型，为基于物理的角色动画开发语言驱动控制器。

语言驱动动画：从语言合成动作是语音驱动面部动画的核心挑战之一，其目标是根据给定的话语生成面部运动。这些模型通常利用语音单位（音素）与面部发音动作（视素）之间的时间对应关系，为特定话语合成逼真的面部动画 [Brand 1999; Deena and Galata 2009; Hong et al. 2002; Karras et al. 2017; Pelachaud et al. 1996]。类似的时间对应关系也可用于根据语音生成全身手势 [Ahuja and Morency 2019; Alexanderson et al. 2020; Levine et al. 2009]。虽然这些技术能有效地根据语音生成逼真的动作，但并不适用于语言与动作之间不存在明确时间对应关系的更一般情形。例如，“撞倒红色方块”这样的高层指令

隐含地编码了角色应执行的一系列技能。已有研究提出序列到序列模型，将高层语言描述映射为动作轨迹 [Lin 等人 2018; Plappert 等人 2017]。Ahuja 和 Morency [2019] 以及 Tevet 等人 [2022] 提出了自编码器框架，用于学习语言与动作的联合嵌入，并可据此根据语言描述生成全身动作。虽然这些技术已展现出令人期待的结果，但其重点主要在于开发运动学运动模型。在本研究中，我们旨在开发一种用于基于物理的角色动画的语言驱动模型，将高层语言指令映射为低层控制，使角色能够执行所需的行为。

3 背景

我们的角色通过目标条件强化学习框架进行训练。在该框架中，智能体依据控制策略 π 与环境交互，以完成从目标分布 $\mathbf{g} \sim p(\mathbf{g})$ 中抽取的给定目标 $\mathbf{g} \in \mathrm{G}$。在每个时间步$t$，智能体观察环境状态 $\mathbf{s}_t \in \mathrm{S}$，并通过执行从策略 $\mathbf{a}_t \sim \pi(\mathbf{a}_t|\mathbf{s}_t, \mathbf{g})$ 中采样的动作 $\mathbf{a}_t \in \mathrm{A}$ 来作出响应。执行动作 $\mathbf{a}_t$ 后，环境转移至新状态$\mathbf{s}_{t+1}$，智能体则获得标量奖励 $r_t = r(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1}, \mathbf{g})$，该奖励反映了对于给定目标 $\mathbf{g}$ 而言，此次状态转移的理想程度。智能体的目标是学习策略 π，使其期望折扣回报 $J(\pi)$ 最大化，

$$J(\pi) = \mathbb{E}_{p(\mathbf{g})}\mathbb{E}_{p(\tau|\pi,\mathbf{g})}\left[\sum_{t=0}^{T-1}\gamma^t r_t\right], \tag{1}$$

其中，$p(\tau|\pi, \mathbf{g}) = p(\mathbf{s}0)\prod_{t=0}^{T-1} p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)\pi(\mathbf{a}_t|\mathbf{s}_t, \mathbf{g})$ 表示在给定目标 $\mathbf{g}$ 的情况下，策略 π 下轨迹 $\tau = (\mathbf{s}0, \mathbf{a}0, \mathbf{s}1, ..., \mathbf{s}_T)$ 的可能性；$p(\mathbf{s}0)$ 是初始状态分布，$p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{s}a)$ 表示环境的转移动力学。T 是轨迹的时间范围，$\gamma \in [0, 1]$ 是折扣因子。

4 概述

本文介绍了基于物理的语言驱动动画系统 PADL（Physics-based Animation Directed with Language，读作“paddle”），用于开发由语言驱动的基于物理的角色动画控制模型。我们的框架允许用户通过指定要完成的任务以及完成该任务时要使用的特定技能来控制角色的运动。任务代表智能体必须实现的高级目标，例如前往目标位置或与特定物体互动。除了指定智能体必须完成什么任务外，用户还需要能够控制如何完成任务。例如，给定前往目标位置这一任务，智能体可以步行、奔跑或跳跃到目标位置。在我们的系统中，角色要执行的任务和技能通过自然语言分别以任务指令和技能指令的形式指定。

我们的框架包含三个阶段，系统示意图见图2。首先，在技能嵌入阶段，使用参考动作数据集 $\mathrm{M} = \{(\mathbf{m}^i, c^i)\}$，其中包含标注了自然语言描述的动作片段$\mathbf{m}^i$ c^i，以学习动作和文本的共享嵌入空间 Z。每个动作片段 $m^i = \{\hat{\mathbf{q}}_t^i\}$ 由一系列姿态 $\hat{\mathbf{q}}_t^i$ 表示。一个动作

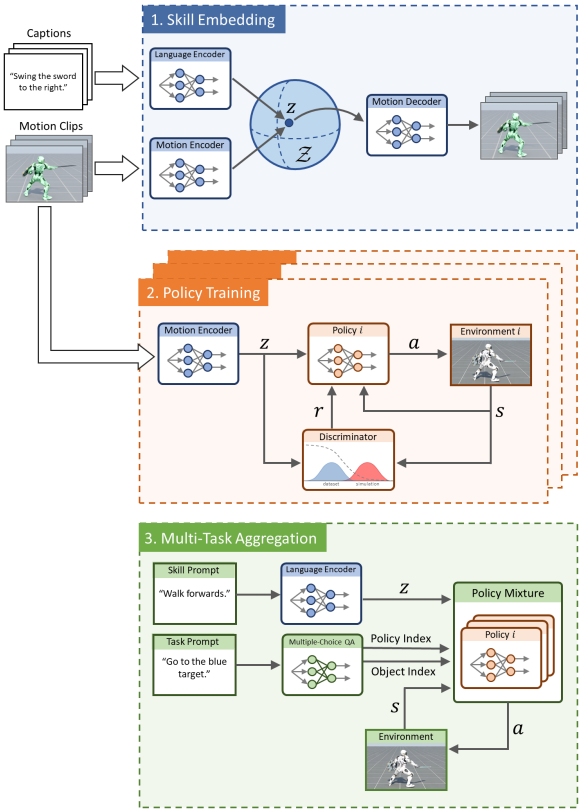


图2：PADL 框架由三个阶段组成。1）在技能嵌入阶段，使用由动作片段及其对应的文本描述组成的数据集，学习动作与描述文本的联合嵌入。2）在策略训练阶段，利用学到的技能嵌入训练一组策略，使其执行各种任务，同时模仿数据集中的行为。3）最后，在多任务聚合阶段，将针对不同任务训练的策略组合成一个多任务控制器，通过语言指令引导其执行不同的任务和技能。

编码器 $z_m^i = \mathrm{Enc}_m(\mathbf{m}^i)$ 和语言编码器 $z_l^i = \mathrm{Enc}_l(c^i)$ 经过训练，可将每对动作与描述映射到相似的嵌入 $z_m^i \approx z_l^i$。接下来，在策略训练 阶段，利用该嵌入训练一组强化学习策略，其中每个策略 $\pi^i(\mathbf{a}_t|\mathbf{s}_t, \mathbf{g}, \mathbf{z})$ 都经过训练，可利用嵌入空间中的多种技能$\mathbf{z} \in \mathrm{Z}$ 执行特定任务。训练完成后，可通过以给定语言指令的嵌入 $\pi\,\mathrm{Enc}_l(c)$ 为条件，引导策略执行特定技能。最后，在多任务聚合 阶段，将不同策略整合到一个多任务控制器中，并可通过语言指令引导它利用所需技能执行特定任务。

5 技能嵌入

在技能嵌入阶段，我们的目标是构建一个嵌入空间，使动作与相应的自然语言描述对齐。为此，我们采用与 MotionCLIP [Tevet 等人 2022], 类似的方法，训练一个变换器自编码器，使其

将动作序列编码为潜在表示，使其与预训练 CLIP 文本编码器 [Radford 等人 2021]生成的语言嵌入"对齐"。给定一个动作片段 $\hat{\mathbf{m}} = (\hat{\mathbf{q}}_1, ..., \hat{\mathbf{q}}_n)$ 及其描述文本 c，动作编码器 $\mathbf{z} = \text{Enc}_m(\hat{\mathbf{m}})$ 将动作映射为嵌入 $\mathbf{z}$。该嵌入经过归一化，使其位于单位球面上 $||\mathbf{z}|| = 1$。遵循 Tevet 等人 [2022], 的做法，$\text{Enc}_m(\mathbf{m})$ 由一个双向Transformer [Devlin 等人 2018]建模。动作解码器与编码器联合训练，以生成重建序列 $\mathbf{m} = (\mathbf{q}_1, ..., \mathbf{q}_n)$，从 $\mathbf{z}$ 中还原 $\hat{\mathbf{m}}$。解码器也被建模为双向Transformer $\mathbf{m} = \text{Dec}(\mathbf{z}, \mathbf{U})$，它使用学习得到的常量查询序列 $\mathbf{U} = (\mathbf{u}_1, ..., \mathbf{u}_n)$ 并行解码所有帧，其结构类似于 Carion 等人 [2020]的最后一层。自编码器使用以下损失函数进行训练：

$$\mathcal{L}_{\text{auto}} = \mathcal{L}_{\text{recon}} + 0.1\mathcal{L}_{\text{align}}. \tag{2}$$

重建损失函数 Lrecon 衡量重建序列与原始动作之间的误差：

$$\mathcal{L}_{\text{recon}} = \frac{1}{n}\sum_{t=1}^{n} ||\hat{\mathbf{q}}_t - \text{Dec}\left(\text{Enc}_m\left(\hat{\mathbf{m}}\right), \mathbf{U}\right)||_2^2. \tag{3}$$

对齐损失 Lalign 衡量动作嵌入与语言嵌入之间的余弦距离：

$$\mathcal{L}_{\text{align}} = 1 - d_{\cos}\left(\text{Enc}_m\left(\hat{\mathbf{m}}\right), \text{Enc}_l(c)\right). \tag{4}$$

语言编码器 $\text{Enc}_l(\mathbf{m})$ 使用预训练的 CLIP 文本编码器建模，并在其后添加一个由两层全连接层组成的输出头；根据公式 4，仅对该输出头进行微调。为帮助避免过拟合，对于从数据集中采样的每个动作序列小批量，我们还会从每个动作中提取一个随机子序列，并将这些切片添加到模型的训练批次中。这些子序列仅参与重建损失函数的计算。

6 策略训练

一旦我们获得动作与描述文本的联合嵌入，接下来将使用该嵌入来训练控制策略，使物理模拟的角色能够执行各种高层任务，同时运用语言指令指定的技能。在每个时间步 t，策略 $\pi(\mathbf{a}_t|\mathbf{s}_t, \mathbf{g}, \mathbf{z})$接收角色的状态 $\mathbf{s}_t$、任务特定目标 $\mathbf{g}$以及技能潜变量 $\mathbf{z}$作为输入。目标 $\mathbf{g}$规定了角色应实现的高层任务目标，例如移动到目标位置或朝向期望方向。技能潜变量 $\mathbf{z}$ 规定了角色应使用何种技能来实现期望目标，例如行走或跑步前往目标位置。这些潜变量通过对从数据集 M 中采样的动作片段 $\mathbf{z} = \text{Enc}_m(\mathbf{m})$进行编码生成。为了训练策略使用指定技能执行给定任务，我们采用由以下两个部分组成的奖励函数：

$$r_t = r_t^{\text{skill}} + \lambda^{\text{task}} r_t^{\text{task}}, \tag{5}$$

其中 r技能$_t$ 是技能奖励，而 r任务$_t$ 是任务奖励，其系数为 λ任务。

6.1 技能目标

为了训练策略执行特定$\mathbf{z}_i$所指定的技能，我们要求策略的状态转移分布（$\mathbf{s}$，$\mathbf{s}'$）与相应动作片段$\mathbf{m}^i$的分布相匹配。为此，我们在状态转移与技能编码的联合分布上训练对抗判别器 $D(\mathbf{s}, \mathbf{s}', \mathbf{z})$ [Ho 和 Er-mon，2016年；Merel 等，2017；Peng 等，2021]。判别器经过训练后，能够判断给定的状态转移（$\mathbf{s}$，$\mathbf{s}'$）是来自与$\mathbf{z}$对应的动作片段，还是来自模拟角色或数据集中的其他动作片段。判别器通过最小化以下损失函数进行训练：

$$\mathcal{L}_D = \mathbb{E}_{p_{\mathcal{M}}(\mathbf{m})}\Bigg[- \mathbb{E}_{p_{\mathbf{m}}(\mathbf{s},\mathbf{s}')}\left[\log(D(\mathbf{s}, \mathbf{s}', \mathbf{z}))\right] \tag{6}$$

$$- w_D\, \mathbb{E}_{p_\pi(\mathbf{s},\mathbf{s}'|\mathbf{z})}\left[\log(1 - D(\mathbf{s}, \mathbf{s}', \mathbf{z}))\right] \tag{7}$$

$$- (1 - w_D)\, \mathbb{E}_{p_{\mathcal{M}\setminus\mathbf{m}}(\mathbf{s},\mathbf{s}')}\left[\log(1 - D(\mathbf{s}, \mathbf{s}', \mathbf{z}))\right] \tag{8}$$

$$+ w_{\text{gp}}\, \mathbb{E}_{p_{\mathbf{m}}(\mathbf{s},\mathbf{s}')}\left[\left\|\nabla_\phi D(\phi, \mathbf{z})\Big|_{\phi=(\mathbf{s},\mathbf{s}')}\right\|^2\right]\Bigg]. \tag{9}$$

$p_{\mathcal{M}}(\mathbf{m})$ 表示从数据集 M 中采样到动作片段 $\mathbf{m}$ 的可能性，而 $\mathbf{z} = \text{Enc}_m(\mathbf{m})$ 是该动作片段的编码。$p_{\mathbf{m}}(\mathbf{s}, \mathbf{s}')$ 表示观测到给定动作片段中的状态转移的可能性，而 $p_\pi(\mathbf{s},\mathbf{s}'|\mathbf{z})$ 表示在以 $\mathbf{z}$ 为条件时，从策略 π中观测到状态转移的可能性。$p_{\mathcal{M}\setminus\mathbf{m}}(\mathbf{s}, \mathbf{s}')$ 表示从数据集中除 $\mathbf{m}$ 以外的其他动作片段中随机采样状态转移并观测到该状态转移的可能性，w_D 是手动指定的系数。损失函数中的最后一项是梯度惩罚，其系数为 w_{gp} [Peng 等人 2021], ，该项可提高对抗训练过程的稳定性。技能奖励定义如下：

$$r_t^{\text{skill}} = -\log\left(1 - D(\mathbf{s}_t, \mathbf{s}_{t+1}, \mathbf{z})\right). \tag{10}$$

训练完成后，为了通过技能指令 $c$skill 引导策略，模型会接收编码 $\mathbf{z} = \text{Enc}_l(c\text{skill})$。通过让判别器同时以状态转移和潜变量为条件，我们的方法明确鼓励策略模仿数据集中的每个动作片段，从而大幅减少模式坍塌。我们将在附录 D 中详细阐述这一优势，并将我们的方法与相关的对抗式强化学习框架进行比较。

7 多任务聚合

策略训练阶段的每个策略都能够执行多种技能，但每个策略只能执行涉及单个目标物体的一项高层任务。我们展示了，这些独立策略可以聚合为一个更灵活的复合策略，使用户能够指挥角色在包含多个物体的环境中执行各种不同任务。然而，在实验中，我们发现，尝试使用第 6 节中的流程训练一个能够执行所有任务的单一多任务策略，会导致性能不佳。在强化学习中，有效训练多任务策略仍然是一个具有挑战性且尚未解决的问题；以往的系统通常采用分而治之方法来处理多任务强化学习 [Ghosh 等，2018；Ruder，2017；Rusu 等，2015]。

为了构建更灵活的多任务多物体控制器，我们将一组单任务策略整合在一起。在每个时间步，系统都会使用用户当前的任务指令生成提示，并将其输入多项选择问答（QA）模型。QA 模型会识别用户所指的是哪个任务和环境物体。随后，将已识别任务对应的单任务控制器

设为控制角色的活动策略，并将已识别物体的状态传递给所选策略。附录中的算法 1 通过伪代码概述了这一流程。请注意，由于角色在每个时间步都由第 6 节中的单一策略驱动，整合后的控制器一次只能执行涉及单个物体的一个高层任务。不过，借助此控制器，用户可以通过自然语言动态控制当前关注的任务和物体。

7.1 多项选择问答

基于语言的选择模型概述如图3所示。多项选择问答模型采用预训练的 BERT 模型构建，并在 SWAG数据集上进行微调 [Zellers 等人 2018]。每道多项选择题都由一个初始提示句（句子 A）以及 n 个候选后续句（句子 B）构成 [Devlin 等人 2018]。随后，模型会为 n 个不同的序列输出分数，其中序列 i 是提示句与第 i个候选句的拼接。与得分最高的候选句对应的物体将被选为策略的目标物体。识别用户指令中的任务也采用类似的流程。

对于用户提供的每条任务指令，模型会分别接收两道独立的多项选择题，以识别相应的任务和物体。第一道题用于识别任务，其中每个多项选择选项都对应一个经过训练的策略。问答模型的输入采用类似故事的格式，以模仿其微调所用的SWAG数据集中的元素。例如，如果任务指令是“撞倒蓝色塔楼”，则击打策略的候选序列为：

- “Bob想要推倒蓝色高塔。由于他具备推倒指定物体的能力，这对他来说应该很容易。”同样，位置策略的候选句子序列如下：

- “Bob想要推倒蓝色高塔。由于他具备前往指定目的地的能力，这对他来说应该很容易。”

随后，多项选择问答模型将预测哪组句子最有可能成立。同样，在用于提取目标物体的多项选择题中，每个物体都有一个描述其外观的选项。绿色方块对应的候选句子序列如下：

- “Bob想要推倒蓝色高塔。他首先将注意力转向附近的绿色物体 。”

8 实验设置

我们通过为一个3D模拟人形角色训练语言引导的控制策略，评估框架的有效性。该角色配备了剑和盾牌，与Peng等人使用的角色相似， [2022], 具有37个自由度，并采用相似的状态与动作表示。数据集共包含131段独立片段，动作数据总时长约为9分钟。每段片段均由人工标注1至4条说明文字，用于描述角色在该片段中的行为，

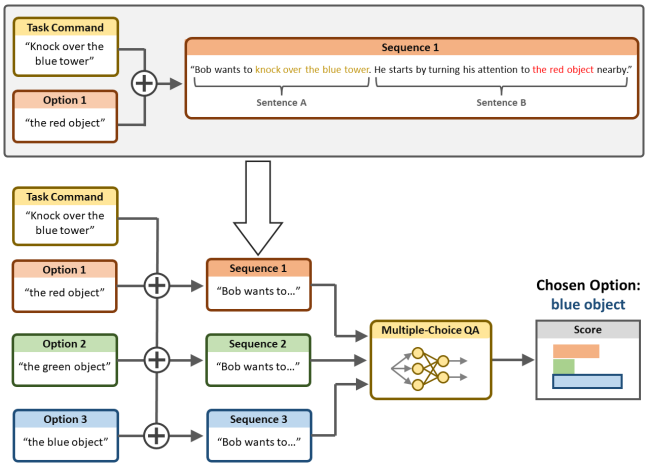


图3：基于语言的选择模型概述，该模型根据用户的任务指令选择目标物体。任务指令用于生成一组候选句子，每个句子对应环境中的一个特定物体。随后，模型使用多项选择问答模型，根据任务指令预测最可能的候选句子。模型的预测结果用于识别用户所指的目标物体。

整个数据集中共有 265 条描述文本。图 4 展示了数据集中的动作片段示例及其对应的描述文本。

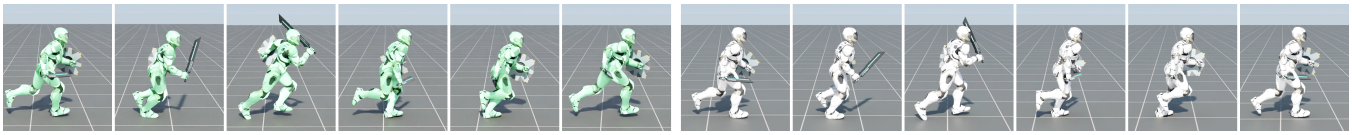
8.1 任务

除了训练策略模仿数据集中的技能外，每个策略还会接受训练，以执行一项额外的高层任务。这里我们概述了各种任务，详细说明请参见附录 B。

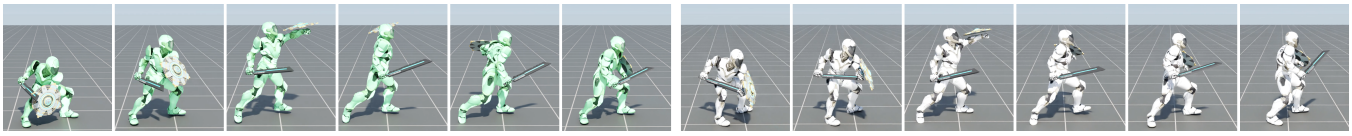
(1) **朝向：** 首先，我们设置了一个简单的朝向任务，目标是让角色转身面向目标方向 $\mathbf{d}^*$，该方向以水平面上的二维向量表示。策略的目标输入 $\mathbf{g}_t = \mathbf{d}_t^*$ 记录了角色局部坐标系中的目标方向。(2) **位置：** 接下来是目标位置任务，目标是让角色移动到目标位置 $\mathbf{x}^*$。目标 $\mathbf{g}_t = \tilde{\mathbf{x}}_t^*$ 记录了角色局部坐标系中的目标位置 $\tilde{\mathbf{x}}_t^*$。(3) **击打：** 最后是击打任务，目标是让角色击倒一个目标物体。目标 $\mathbf{g}_t = (\tilde{\mathbf{x}}_t^*, \dot{\tilde{\mathbf{x}}}_t^*, \tilde{q}_t^*, \dot{\tilde{q}}_t^*)$ 记录了目标物体的位置 $\tilde{\mathbf{x}}_t^*$、旋转 $\tilde{q}_t^*$、线速度 $\dot{\tilde{\mathbf{x}}}_t^*$和角速度 $\dot{\tilde{q}}_t^*$。所有特征均以角色的局部坐标系表示。

8.2 训练

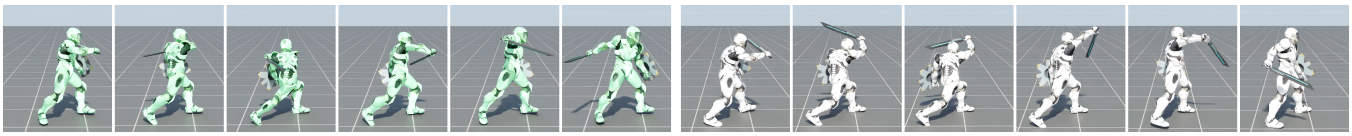
所有物理模拟均使用 Isaac Gym 进行，这是一款大规模并行的基于 GPU 的物理模拟器 [Makoviychuk 等人 2021]。模拟频率为 120Hz，策略运行频率为 30Hz。单个 A100 GPU 上并行模拟 4096 个环境。技能嵌入使用 128D 潜在空间。策略、价值函数和判别器分别使用独立的多层感知机进行建模



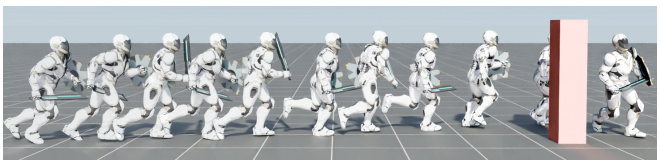
(a) “向前冲刺，同时挥动手臂”。



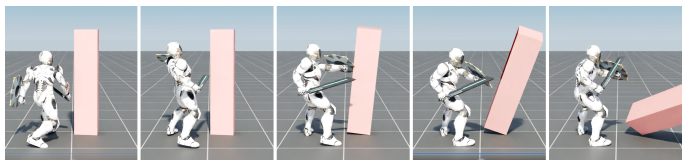
(b) “向左盾击”、“向左盾击”、“静止站立时向左进行盾牌猛击”



(c) “向右斩击”、“向右挥击”、“向右挥剑”、“静止站立并向右斩击”。



(d) 任务：位置。技能：“挥动手臂的同时向前冲刺”。



(e) 任务：击打。技能：“向右盾牌猛击”。

图 4：(a)–(c)：参考动作片段（左侧）及其对应的描述文本，以及模拟角色根据语言指令执行参考技能时产生的动作（右侧）。附录 C 的图 7 展示了更多参考动作和策略轨迹。(d)–(e)：训练得到的策略使用不同技能完成任务。

其使用 ReLU 单元，隐藏层包含 [1024, 1024, 512] 个单元。每个策略均采用近端策略优化进行训练，使用约 70 亿个样本 [Schulman 等人 2017],，相当于约 7 年的模拟时间，需要约 2.5 天的现实时间。为任务奖励选择一个能有效平衡任务奖励与技能奖励的权重 λ^{task} 可能颇具挑战，且可能需要针对特定任务进行调优。因此，我们采用一种自适应方法，根据目标任务奖励值动态调整 λ^{task} [Mentzer 等人 2021]。更多详情请参见附录 B.4。

9 结果

我们首先在不使用辅助任务的情况下训练策略，以评估模型从动作数据集中复现技能的能力。图 4 展示了策略在接收各种技能指令时的行为示例。该策略能够遵循多种语言指令，范围从行走和跑步等移动技能，到挥剑和盾牌猛击等更具运动性的行为。由于语言编码器基于大型 CLIP 模型 [Radford 等人 2021], 构建，因此它对数据集中未出现的新指令具有一定的鲁棒性。例如，即使数据集中的描述文本都不包含“悠闲”一词，也没有将行走表述为“散步”，在收到“悠闲地散个步”这一提示时，模型仍能正确做出轻松行走的动作。不过，由于用于训练编码器的描述文本数量相对较少，面对某些新指令时，模型仍可能产生不正确的行为。该角色能够成功执行右挥砍在收到提示“右挥砍”时。然而，“用剑向右挥砍”却会使角色执行左挥砍。

除了从动作数据集中学习技能外，我们的策略还可以经过训练来执行其他高层任务，如第 8.1 节所述。图 4 展示了这些任务的示例。我们针对每项任务分别训练策略，随后可将它们整合为一个多任务控制器，由该控制器根据任务指令激活相应的策略。我们在一个包含多个可供角色交互的物体的环境中，展示了多任务控制器的有效性。用户可以发出任务指令，指定目标物体以及角色应执行的任务。我们的多项选择问答框架能够持续从用户指令中正确识别任务和目标物体。例如，给出指令：“撞倒蓝色方块”。选择模型会正确识别出击打任务的策略，并将蓝色方块选为目标。选择模型还可以解析较为特别的指令，例如“悠闲地走到那家栗色酒馆”，并正确识别出位置任务和红色方块。尽管大语言模型具备泛化能力，某些指令仍可能导致错误行为。附录 C 提供了更多任务指令示例，以及模型据此产生的行为。

9.1 数据集覆盖范围

为确定学习能对齐动作与文本的技能嵌入所产生的影响，我们评估模型复现

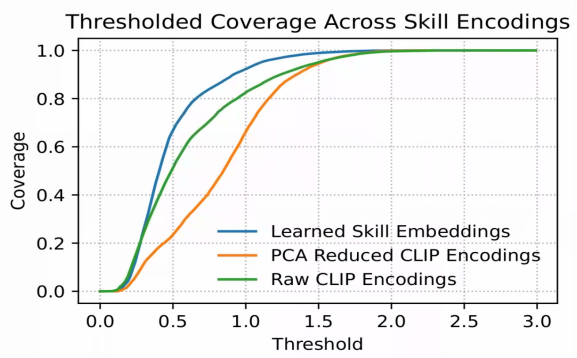


图 5：比较在策略训练阶段使用不同技能编码时的数据集覆盖范围。“学习得到的技能嵌入”使用第 5 节所述的学习得到的动作编码器生成的 128D 嵌入。我们将其与以下基线方法进行比较：策略直接使用数据集说明文字的 512D CLIP 文本编码进行训练，以及使用主成分分析将这些编码降至 128D 的方法。

数据集中各种动作的能力，具体做法是向模型提供相应的指令。我们采用带阈值的覆盖率指标进行评估。给定由动作片段 $\hat{\mathbf{m}} = (\hat{\mathbf{s}}_0, \hat{\mathbf{s}}_2, \ldots, \hat{\mathbf{s}}_n)$ 指定的状态序列、技能编码 $\mathbf{z} = \text{Enc}_I(c)$ 对应的策略轨迹 $\tau = (\mathbf{s}_0, \mathbf{s}_2, \ldots, \mathbf{s}_k)$（其中 c 是 $\hat{\mathbf{m}}$ 的描述文本），以及阈值参数 $\epsilon > 0$，我们将覆盖率定义为：

$$\text{coverage}(\tau, \hat{\mathbf{m}}, c, \epsilon) = \frac{1}{n}\sum_{i=0}^{n} \mathcal{I}\left(\left(\min_{j\in\{0,\ldots,k\}} ||\hat{\mathbf{s}}_i - \mathbf{s}_j||_2\right) \leq \epsilon\right) \quad (11)$$

该指标用于确定动作片段中与策略轨迹中的某个状态足够接近的状态所占比例。在实验中，我们为每条轨迹采集 300 个时间步（10 秒）。我们不选取固定阈值 ϵ，而是对 ϵ 在 $[0, 3]$ 之间取不同值应用公式 11，从而生成覆盖率曲线。

图 5 比较了 PADL 模型与基线模型的性能，后者直接将描述文本的 CLIP 编码作为策略的输入。覆盖率统计先对数据集中每个动作片段的所有描述文本取平均，再对所有动作片段取平均。原始 CLIP 编码为 512D，而我们学习得到的技能嵌入为 128D。我们还加入了一个额外的基线模型，该模型使用主成分分析将 CLIP 编码的维度降至 128D。我们学习得到的嵌入能够更好地复现数据集中的行为。直接将 CLIP 编码用作策略输入往往会导致动作质量较低，而且在语言指令引导下更容易执行错误行为。

9.2 技能插值

除了实现语言控制外，学习到的技能嵌入还能够在不同技能之间产生语义上有意义的插值。给定两个技能指令 c_1 和 c_2，我们使用语言编码器将每个描述文本编码为相应的潜变量 $\mathbf{z}_1$ 和 $\mathbf{z}_2$。随后，我们对这两个潜变量进行球面插值，并以插值后的潜变量作为策略的条件，生成一条轨迹。例如，给定两个指令：“向前走”和“向前冲刺，同时挥动手臂”，在这两个潜变量之间插值，会产生这样的移动行为：

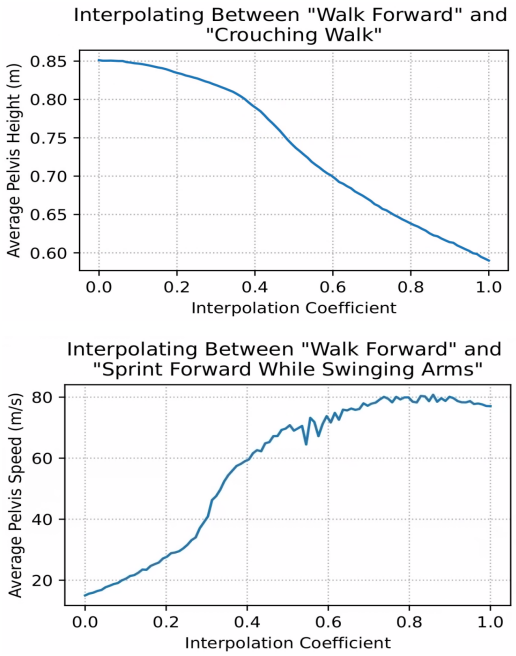


图 6：在潜在空间中对技能进行插值，可生成具有语义意义的中间行为，例如以不同的行走高度和速度移动。

以不同速度移动。图 6记录了策略以不同插值潜变量为条件时角色的平均速度。同样，在“向前走”和“向前蹲着走”之间插值，会产生行走高度各异的步态。不过，并非所有指令组合都能产生直观的中间行为。

10 结论

在本研究中，我们提出了 PADL，这是一种用于学习基于语言的控制器的框架，面向基于物理的角色动画。语言既用于指定角色应执行的高层任务，也用于指定角色完成这些任务时应使用的低层技能。虽然我们的模型能够模仿动作数据中的多种技能，但其可执行的高层任务种类仍然有限。我们希望探索更具可扩展性的方法来建模角色与环境的交互，以更通用的策略取代有限的先验任务集合，使用户能够通过自然语言指定任意的环境交互。我们还希望将 PADL 扩展到规模大得多的带标注的动作捕捉数据集 [Punnakkal 等人 2021]，，这或许能使智能体和语言编码器建模更多样的技能，同时更能适应改述，并能够泛化到新的指令。具体而言，我们预计，随着文本数据增多，技能嵌入阶段的语言编码器将得到显著提升。我们期待语言引导的基于物理的角色动画领域取得进一步进展，并希望我们的工作能助力开发功能强大、质量出色的动画工具，使其界面易于广泛使用、灵活多样且简单易用。

致谢

我们感谢 Reallusion[1] 为本项目提供动作捕捉参考数据。此外，我们感谢匿名审稿人提供反馈，也感谢 Steve Masseroni 和 Margaret Albrecht 协助制作补充视频。

参考文献

Shailen Agrawal 和 Michiel van de Panne。2016年。基于任务的运动。《*ACM*图形学汇刊》（*SIGGRAPH 2016*论文集）35，4（2016年）。C. Ahuja 和 L. Morency。2019年。Language2Pose：基于自然语言的姿势预测。载于 *2019*年国际三维视觉会议（*3DV*）。IEEE计算机学会，洛斯阿拉米托斯，加利福尼亚州，美国，719–728。https://doi.org/10.1109/3DV.2019.00084 Simon Alexanderson、Gustav Eje Henter、Taras Kucherenko 和 Jonas Beskow。2020年。风格可控的语音驱动手势合成：使用归一化流。《计算机图形学论坛》（2020年）。https://doi.org/10.1111/cgf.13946 Kevin Bergamin、Simon Clavet、Daniel Holden 和 James Richard Forbes。2019年。DReCon：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，文章206（2019年11月），11页。https://doi.org/10.1145/ 3355089.3356536 Matthew Brand。1999年。语音木偶表演。载于 第二十六届计算机图形学与交互技术年会论文集（*SIGGRAPH '99*）。ACM出版社/Addison-Wesley出版公司，美国，21–28。https://doi.org/10.1145/311535.311537 Tom B. Brown、Benjamin Mann、Nick Ryder、Melanie Subbiah、Jared Kaplan、Prafulla Dhariwal、Arvind Neelakantan、Pranav Shyam、Girish Sastry、Amanda Askell、Sand-hini Agarwal、Ariel Herbert-Voss、Gretchen Krueger、Tom Henighan、Rewon Child、Aditya Ramesh、Daniel M. Ziegler、Jeffrey Wu、Clemens Winter、Christopher Hesse、Mark Chen、Eric Sigler、Mateusz Litwin、Scott Gray、Benjamin Chess、Jack Clark、Christopher Berner、Sam McCandlish、Alec Radford、Ilya Sutskever 和 Dario Amodei。2020年。语言模型是少样本学习者。*CoRR* abs/2005.14165（2020年）。arXiv:2005.14165 https://arxiv.org/abs/2005.14165 Nicolas Carion、Francisco Massa、Gabriel Synnaeve、Nicolas Usunier、Alexander Kirillov 和 Sergey Zagoruyko。2020年。基于变换器的端到端目标检测。https://doi.org/10.48550/ARXIV.2005.12872 Mark Chen、Jerry Tworek、Heewoo Jun、Qiming Yuan、Henrique Ponde de Oliveira Pinto、Jared Kaplan、Harrison Edwards、Yuri Burda、Nicholas Joseph、Greg Brockman、Alex Ray、Raul Puri、Gretchen Krueger、Michael Petrov、Heidy Khlaaf、Girish Sastry、Pamela Mishkin、Brooke Chan、Scott Gray、Nick Ryder、Mikhail Pavlov、Alethea Power、Lukasz Kaiser、Mohammad Bavarian、Clemens Winter、Philippe Tillet、Felipe Petroski Such、Dave Cummings、Matthias Plappert、Fotios Chantzis、Elizabeth Barnes、Ariel Herbert-Voss、William Hebgen Guss、Alex Nichol、Alex Paino、Nikolas Tezak、Jie Tang、Igor Babuschkin、Suchir Balaji、Shantanu Jain、William Saunders、Christopher Hesse、Andrew N. Carr、Jan Leike、Joshua Achiam、Vedant Misra、Evan Morikawa、Alec Radford、Matthew Knight、Miles Brundage、Mira Murati、Katie Mayer、Peter Welinder、Bob McGrew、Dario Amodei、Sam McCandlish、Ilya Sutskever 和 Wojciech Zaremba。2021年。评估基于代码训练的大语言模型。*CoRR* abs/2107.03374（2021年）。arXiv:2107.03374 https://arxiv.org/abs/2107.03374 Alexander Clegg、Wenhao Yu、Jie Tan、C. Karen Liu 和 Greg Turk。2018年。学习穿衣：通过深度强化学习合成人体穿衣动作。《*ACM*图形学汇刊》37，6，文章179（2018年12月），10页。https://doi.org/10.1145/3272127.3275048 Karl Cobbe、Vineet Kosaraju、Mohammad Bavarian、Mark Chen、Heewoo Jun、Lukasz Kaiser、Matthias Plappert、Jerry Tworek、Jacob Hilton、Reiichiro Nakano、Christo-pher Hesse 和 John Schulman。2021年。训练验证器解决数学应用题。https://doi.org/10.48550/ARXIV.2110.14168 Stelian Coros、Philippe Beaudoin 和 Michiel van de Panne。2009年。面向基于物理的角色的稳健的基于任务的控制策略。《*ACM*图形学汇刊》（*SIGGRAPH Asia*论文集）28，5（2009年），文章170。Marco da Silva、Yeuhi Abe 和 Jovan Popović。2008年。使用短时域模型预测控制仿真人体运动数据。《计算机图形学论坛》27（2008年）。Martin de Lasa、Igor Mordatch 和 Aaron Hertzmann。2010年。基于特征的运动控制器。《*ACM*图形学汇刊》29，3（2010年）。Salil Deena 和 Aphrodite Galata。2009年。使用共享高斯过程潜变量模型的语音驱动面部动画。载于 第五届视觉计算进展国际研讨会论文集：第一部分（拉斯维加斯，内华达州）（*ISVC '09*）。施普林格出版社，柏林，海德堡，89–100。https://doi.org/10.1007/978-3-642-10331-5 9_ Jacob Devlin、Ming-Wei Chang、Kenton Lee 和 Kristina Toutanova。2018年。BERT：用于语言理解的深度双向变换器预训练。https://doi.org/10.48550/ARXIV.1810.04805 Dibya Ghosh、Avi Singh、Aravind Rajeswaran、Vikash Kumar 和 Sergey Levine。2018年。分治强化学习。载于国际学习表征会议。https://openreview.net/forum?id=rJwelMbR- F. Sebastin Grassia。1998年。使用指数映射对旋转进行实用参数化。*J. Graph. Tools* 3，3（1998年3月），29–48。https://doi.org/10.1080/10867651. 1998.10487493 Jonathan Ho 和 Stefano Ermon。2016年。生成对抗模仿学习。载于神经信息处理系统进展，D. Lee、M. Sugiyama、U. Luxburg、I. Guyon 和 R. Garnett（编），第29卷。Curran Associates, Inc. https://proceedings.neurips.cc/paper/2016/file/cc7e2b878868cbae992d1fb743995d8f-Paper.pdf Jessica K. Hodgins、Wayne L. Wooten、David C. Brogan 和 James F. O'Brien。1995年。人体运动动画。载于第*22*届计算机图形学与交互技术年度会议论文集（*SIGGRAPH '95*）。美国计算机协会，纽约，美国，71–78。https://doi.org/10.1145/218380. 218414 Daniel Holden、Taku Komura 和 Jun Saito。2017年。用于角色控制的相位函数神经网络。《*ACM*图形学汇刊》36，4，文章42（2017年7月），13页。https://doi.org/10.1145/3072959.3073663 Daniel Holden、Jun Saito 和 Taku Komura。2016年。用于角色运动合成与编辑的深度学习框架。《*ACM*图形学汇刊》35，4，文章138（2016年7月），11页。https://doi.org/10.1145/2897824.2925975 Pengyu Hong、Zhen Wen 和 T.S. Huang。2002年。利用神经网络实时合成由语音驱动且带有表情的面部动画。*IEEE*神经网络汇刊13，4（2002年），916–927。https://doi.org/10.1109/TNN.2002.1021892 Tero Karras、Timo Aila、Samuli Laine、Antti Herva 和 Jaakko Lehtinen。2017年。通过联合端到端学习姿态与情绪实现音频驱动的面部动画。《*ACM*图形学汇刊》36，4，文章94（2017年7月），12页。https://doi.org/10.1145/3072959. 3073658 Kyungho Lee、Sehee Min、Sunmin Lee 和 Jehee Lee。2021b年。为交互式角色控制学习时间关键响应。《*ACM*图形学汇刊》40，4，文章147（2021年7月），11页。https://doi.org/10.1145/3450626.3459826 Seyoung Lee、Sunmin Lee、Yongwoo Lee 和 Jehee Lee。2021a年。从单个动作片段中学习一组运动技能。《*ACM*图形学汇刊》40，4，文章93（2021年）。Yoonsang Lee、Sungeun Kim 和 Jehee Lee。2010a年。数据驱动的双足控制。《*ACM*图形学汇刊》29，4，文章129（2010年7月），8页。https://doi.org/10.1145/1778765. 1781155 Yongjoon Lee、Kevin Wampler、Gilbert Bernstein、Jovan Popović 和 Zoran Popović。2010 29，6，138 2010 8 10 1145 1882261 1866160 b. 用于交互式角色移动的运动场。《*ACM*图形学汇刊》文章（dec），页。https://doi.org/ . / . S. Levine、C. Theobalt 和 V. Koltun。2009年。实时韵律驱动的肢体语言合成。《*ACM*图形学汇刊》28（2009年12月），1–10。https: // doi.org/10.1145/1618452.1618518 Yujia Li、David Choi、Junyoung Chung、Nate Kushman、Julian Schrittwieser、Rémi Leblond、Tom Eccles、James Keeling、Felix Gimeno、Agustin Dal Lago、Thomas Hubert、Peter Choy、Cyprien de Masson d'Autume、Igor Babuschkin、Xinyun Chen、Po-Sen Huang、Johannes Welbl、Sven Gowal、Alexey Cherepanov、James Molloy、Daniel J. Mankowitz、Esme Sutherland Robson、Pushmeet Kohli、Nando de Freitas、Koray Kavukcuoglu 和 Oriol Vinyals。2022年。AlphaCode：竞赛级代码生成。https://doi.org/10.48550/ARXIV.2203.07814 Angela S. Lin、Lemeng Wu、Rodolfo Corona、Kevin Tai、Qixing Huang 和 Raymond J. Mooney。2018年。根据自然语言描述生成人类活动动画视频。载于*NeurIPS 2018*视觉基础交互与语言研讨会论文集。http://www.cs.utexas.edu/users/ai-labpub- view.php?PubID=127730 Hung Yu Ling、Fabio Zinno、George Cheng 和 Michiel van de Panne。2020年。使用运动变分自编码器的角色控制器。《*ACM*图形学汇刊》39，4（2020年）。Libin Liu 和 Jessica Hodgins。2018年8月。使用轨迹优化和深度强化学习学习篮球运球技能。《*ACM*图形学汇刊》37，4（2018年8月）。Libin Liu、Michiel van de Panne 和 KangKang Yin。2016年。为基于物理的角色引导学习控制图。《*ACM*图形学汇刊》35，3（2016年）。Yinhan Liu、Myle Ott、Naman Goyal、Jingfei Du、Mandar Joshi、Danqi Chen、Omer Levy、Mike Lewis、Luke Zettlemoyer 和 Veselin Stoyanov。2019年。RoBERTa：稳健优化的BERT预训练方法。https://doi.org/10.48550/ARXIV.1907.11692 Viktor Makoviychuk、Lukasz Wawrzyniak、Yunrong Guo、Michelle Lu、Kier Storey、Miles Macklin、David Hoeller、Nikita Rudin、Arthur Allshire、Ankur Handa 和 Gavriel State。2021年。Isaac Gym：面向机器人学习的高性能GPU物理仿真。*CoRR* abs/2108.10470（2021年）。arXiv:2108.10470 https://arxiv. org/abs/2108.10470 Fabian Mentzer、Eirikur Agustsson、Johannes Ballé、David Minnen、Nick Johnston 和 George Toderici。2021年。使用生成对抗网络进行细节合成与传播的神经视频压缩。https://doi.org/10.48550/ARXIV.2107.12038 Josh Merel、Yuval Tassa、Dhruva TB、Sriram Srinivasan、Jay Lemmon、Ziyu Wang、Greg Wayne 和 Nicolas Heess。2017年。通过对抗模仿从动作捕捉中学习人类行为。*CoRR* abs/1707.02201（2017年）。arXiv:1707.02201 http://arxiv.org/abs/1707.02201 Igor Mordatch、Emanuel Todorov 和 Zoran Popović。2012年。通过接触不变优化发现复杂行为。《*ACM*图形学汇刊》31，4，文章

[1]https://actorcore.reallusion.com/

43（2012年7月），8页。https://doi.org/10.1145/2185520.2185539 Alex Nichol、Prafulla Dhariwal、Aditya Ramesh、Pranav Shyam、Pamela Mishkin、Bob McGrew、Ilya Sutskever 和 Mark Chen。2021年。GLIDE：通过文本引导扩散模型实现逼真图像生成与编辑。https://doi. org/10.48550/ARXIV.2112.10741 Soohwan Park、Hoseok Ryu、Seyoung Lee、Sunmin Lee 和 Jehee Lee。2019年。从无组织的人体运动数据中学习预测与模拟策略。《*ACM*图形学汇刊》38, 6，文章205（2019年11月），11页。https://doi.org/10.1145/3355089. 3356501 Catherine Pelachaud、Norman Badler 和 Mark Steedman。1996年。为语音生成面部表情。*Cognitive Science* 20（1996年3月），1–46。https://doi.org/10. 1016/S0364-0213(99)80001-9 Xue Bin Peng、Pieter Abbeel、Sergey Levine 和 Michiel van de Panne。2018a年。Deep-Mimic：示例引导的深度强化学习，用于学习基于物理的角色技能。《*ACM*图形学汇刊》37, 4，文章143（2018年7月），14页。https: //doi.org/10.1145/3197517.3201311 Xue Bin Peng、Yunrong Guo、Lina Halper、Sergey Levine 和 Sanja Fidler。2022年。ASE：适用于物理仿真角色的大规模可复用对抗性技能嵌入。《*ACM*图形学汇刊》41, 4，文章94（2022年7月）。Xue Bin Peng、Angjoo Kanazawa、Jitendra Malik、Pieter Abbeel 和 Sergey Levine。2018b年。SFV：从视频中通过强化学习获取物理技能。《*ACM*图形学汇刊》37, 6，文章178（2018年11月），14页。Xue Bin Peng、Ze Ma、Pieter Abbeel、Sergey Levine 和 Angjoo Kanazawa。2021年。AMP：用于风格化的基于物理的角色控制的对抗性运动先验。《*ACM*图形学汇刊》40, 4，文章1（2021年7月），15页。https://doi.org/10.1145/3450626.3459670 Matthias Plappert、Christian Mandery 和 Tamim Asfour。2017年。使用深度循环神经网络学习人体全身运动与自然语言之间的双向映射。*CoRR* abs/1705.06400（2017年）。arXiv:1705.06400 http://arxiv.org/abs/1705.06400 Nancy Pollard、Jessica Hodgins、Marcia Riley 和 Christopher Atkeson。2002年。调整人体运动以控制人形机器人。2（2002年4月）。https://doi.org/ 10.1109/ROBOT.2002.1014737 Abhinanda R. Punnakkal、Arjun Chandrasekaran、Nikos Athanasiou、Alejandra Quiros-Ramirez 和 Michael J. Black。2021年。BABEL：身体、动作与行为（附英文标签）。载于*IEEE/CVF*计算机视觉与模式识别会议论文集（*CVPR*）。722–731。Alec Radford、Jong Wook Kim、Chris Hallacy、Aditya Ramesh、Gabriel Goh、Sandhini Agarwal、Girish Sastry、Amanda Askell、Pamela Mishkin、Jack Clark、Gretchen Krueger 和 Ilya Sutskever。2021年。通过自然语言监督学习可迁移视觉模型。*CoRR* abs/2103.00020（2021年）。arXiv:2103.00020 https://arxiv.org/abs/2103.00020 Colin Raffel、Noam Shazeer、Adam Roberts、Katherine Lee、Sharan Narang、Michael Matena、Yanqi Zhou、Wei Li 和 Peter J. Liu。2019年。探索统一文本到文本转换器的迁移学习极限。https://doi.org/10.48550/ARXIV. 1910.10683 Aditya Ramesh、Prafulla Dhariwal、Alex Nichol、Casey Chu 和 Mark Chen。2022年。使用CLIP潜变量进行分层文本条件图像生成。https://doi. org/10.48550/ARXIV.2204.06125 Sebastian Ruder。2017年。深度神经网络中的多任务学习概述。*CoRR* abs/1706.05098（2017年）。arXiv:1706.05098 http://arxiv.org/abs/1706.05098 Andrei A. Rusu、Sergio Gomez Colmenarejo、Caglar Gulcehre、Guillaume Desjardins、James Kirkpatrick、Razvan Pascanu、Volodymyr Mnih、Koray Kavukcuoglu 和 Raia Hadsell。2015年。策略蒸馏。https://doi.org/10.48550/ARXIV.1511.06295 John Schulman、Filip Wolski、Prafulla Dhariwal、Alec Radford 和 Oleg Klimov。2017年。近端策略优化算法。*CoRR* abs/1707.06347（2017年）。arXiv:1707.06347 http://arxiv.org/abs/1707.06347 Sebastian Starke、He Zhang、Taku Komura 和 Jun Saito。2019年。用于角色-场景交互的神经状态机。《*ACM*图形学汇刊》38, 6，文章209（2019年11月），14页。https://doi.org/10.1145/3355089.3356505 Fuwen Tan、Song Feng 和 Vicente Ordonez。2018年。Text2Scene：根据文本描述生成抽象场景。*CoRR* abs/1809.01110（2018年）。arXiv:1809.01110 http://arxiv.org/abs/1809.01110 Jie Tan、Yuting Gu、C. Karen Liu 和 Greg Turk。2014年。学习自行车特技。《*ACM*图形学汇刊》33, 4，文章50（2014年7月），12页。https://doi.org/10.1145/2601097. 2601121 Guy Tevet、Brian Gordon、Amir Hertz、Amit H. Bermano 和 Daniel Cohen-Or。2022年。MotionCLIP：将人体运动生成引入CLIP空间。https://doi.org/ 10.48550/ARXIV.2203.08063 Adrien Treuille、Yongjoon Lee 和 Zoran Popović。2007年。通过连续控制实现近乎最优的角色动画。载于*ACM SIGGRAPH 2007*论文集（加利福尼亚州圣迭戈）（*SIGGRAPH '07*）。美国计算机协会，纽约，美国，7–es。https://doi.org/10.1145/1275808.1276386 Jack M. Wang、David J. Fleet 和 Aaron Hertzmann。2009年。优化行走控制器。载于*ACM SIGGRAPH Asia 2009*论文集（日本横滨）（*SIGGRAPH Asia '09*168, 8 10 1145 1661412 1618514）。美国计算机协会，纽约，美国，文章页。https://doi.org/ . / . Jack M. Wang、Samuel R. Hamner、Scott L. Delp 和 Vladlen Koltun。2012年。利用基于生物学的执行器和目标优化运动控制器。*ACM* 《*ACM*图形学汇刊》31, 4, 第25篇文章（2012年7月），11页。https://doi.org/10.1145/2185520. 2185521 Tingwu Wang、Yunrong Guo、Maria Shugrina 和 Sanja Fidler。2020年。UniCon：基于物理的角色运动通用神经控制器。arXiv:2011.15119 [cs.GR] Jungdam Won、Deepak Gopinath 和 Jessica Hodgins。2020年。控制物理仿真角色多样行为的可扩展方法。*ACM*图学汇刊39, 4, 第33篇文章（2020年7月），12页。https://doi.org/10.1145/3386569.3392381 Katsu Yamane、Stuart O. Anderson 和 Jessica K. Hodgins。2010年。利用人体运动数据控制人形机器人：实验验证。载于*2010*年第*10*届*IEEE-RAS*国际人形机器人会议。504–510。https://doi.org/10.1109/ ICHR.2010.5686312 Ri Yu、Hwangpil Park 和 Jehee Lee。2021年。从带动态摄像机运动的单目视频中获取人体动态。*ACM*图学汇刊40, 6, 第208篇文章（2021年12月），14页。https://doi.org/10.1145/3478513.3480504 Y. Yuan、S. Wei、T. Simon、K. Kitani 和 J. Saragih。2021年。SimPoE：用于三维人体姿态估计的模拟角色控制。载于*2021*年*IEEE/CVF*计算机视觉与模式识别会议（*CVPR*）。IEEE计算机学会，洛斯阿拉米托斯，加利福尼亚州，美国，7155–7165。https://doi.org/10.1109/CVPR46437.2021.00708 Rowan Zellers、Yonatan Bisk、Roy Schwartz 和 Yejin Choi。2018年。SWAG：用于基于语境的常识推理的大规模对抗数据集。https://doi.org/10.48550/ARXIV.1808.05326 Yunbo Zhang、Wenhao Yu、C. Karen Liu、Charlie Kemp 和 Greg Turk。2020年。学习操纵非晶态材料。*ACM*图学汇刊39, 6, 第189篇文章（2020年11月），11页。https://doi.org/10.1145/3414685.3417868

状态与动作表示

我们通过为一个3D模拟人形角色训练基于语言的控制策略来评估框架的有效性。该角色装备有一把剑和一面盾牌，与 Peng 等人使用的角色类似。 [2022], 其总计具有37个自由度。角色的状态$\mathbf{s}_t$由一组描述角色身体构型的特征表示。这些特征包括：

- 根节点距地面的高度。
- 根节点在角色局部坐标系中的旋转。
- 每个关节的局部旋转。
- 每个关节的局部速度。
- 手、脚、剑和盾牌在角色局部坐标系中的位置。

根节点指定为骨盆。角色的局部坐标系定义为：原点位于角色的骨盆处，x轴沿根连杆的朝向，y轴则与全局向上向量对齐。每个关节的旋转都用两个三维向量编码，分别表示以关节的父坐标系表示的该连杆局部坐标系的切向量和法向量 [Peng et al. 2021]。每个动作$\mathbf{a}_t$都指定了位于各关节处的比例-微分控制器的目标旋转。根据 Peng et al. [2021], , 三维关节的目标旋转使用三维指数映射表示 Grassia [1998]。

B 任务详情

B.1 朝向任务

朝向任务的奖励定义如下：

$$r_t^{\text{task}} = \min\left(\mathbf{d}_t \cdot \mathbf{d}_t^*, 0.5\right) \tag{12}$$

其中，$\mathbf{d}_t$表示智能体的朝向。我们对奖励进行阈值处理，从而形成一个奖励达到饱和的最佳“锥形”区域，使智能体能够稍微偏离目标朝向，以便更好地模仿技能。

B.2 位置任务

位置任务奖励按以下方式计算：

$$r_t^{\text{task}} = \begin{cases} 0.2r_t^{\text{pos}} + 0.8r_t^{\text{vel}} & ||\mathbf{x}^* - \mathbf{x}||_2 > \delta_{\text{pos}} \\ 0.8 & ||\mathbf{x}^* - \mathbf{x}||_2 \leq \delta_{\text{pos}} \end{cases} \tag{13}$$

其中，$\mathbf{x}$ 表示角色根节点的位置，而 $r\text{pos}_t$ 用于鼓励角色靠近目标：

$$r_t^{\text{pos}} = \exp\left(-0.25||\mathbf{x}^* - \mathbf{x}||_2^2\right), \tag{14}$$

$r\text{vel}_t$ 用于鼓励角色朝目标移动。该速度奖励鼓励智能体以至少$\delta\text{vel} = 0.5$ m/s 的速度朝目标方向移动：

$$r_t^{\text{vel}} = \exp\left(-0.25\left(\max(\delta_{\text{vel}} - v_t^{\text{proj}}, 0) + 0.1v_t^{\text{perp}}\right)\right) \tag{15}$$

其中

$$v_t^{\text{proj}} = ||\text{proj}_{\mathbf{x}^*}(\mathbf{v}_t)||_2 \tag{16}$$

$$v_t^{\text{perp}} = ||\text{perp}_{\mathbf{x}^*}(\mathbf{v}_t)||_2 \tag{17}$$

分别定义智能体沿目标方向的速度和切向速度。当智能体进入目标 $\delta\text{pos} = 2$米范围内时，我们将任务奖励设为上限；当方块被撞倒时，则终止回合，以免智能体只是跑去撞方块。

B.3 击打任务

最后，我们有一个击打任务，其目标是让角色击倒目标物体。目标$\mathbf{g}_t = (\tilde{\mathbf{x}}_t^*, \tilde{\mathbf{x}}_t^*, \tilde{q}_t^*, \tilde{q}_t^*)$记录了目标物体的位置 $\tilde{\mathbf{x}}_t^*$、旋转 $\tilde{q}_t^*$、线速度 $\tilde{\mathbf{x}}_t^*$和角速度 $\tilde{q}_t^*$。所有特征均以角色的局部坐标系表示。任务奖励定义如下：

$$r_t^{\text{task}} = \begin{cases} 0.2r_t^{\text{pos}} + 0.8r_t^{\text{vel}} + 0.8r_t^{\text{knock}} & u_t^* \cdot u^{\text{up}} \geq 0.3 \\ 1.4 & u_t^* \cdot u^{\text{up}} < 0.3 \end{cases} \tag{18}$$

其中，击倒奖励会激励智能体推倒方块：

$$r_t^{\text{knock}} = 1 - u_t^* \cdot u^{\text{up}}. \tag{19}$$

此处， $u\text{up}$ 是全局向上向量， u_t^*是目标物体在全局坐标系中表示的局部向上向量。位置奖励 $r\text{pos}_t$和速度奖励 $r\text{vel}_t$与位置任务中使用的相同。当方块倾倒到足够程度时，任务奖励便会达到饱和。

B.4 自适应任务权重调度

为任务奖励选择权重 λtask，以有效平衡任务奖励与技能奖励，可能颇具挑战性，而且可能需要针对具体任务进行调优。将 λ task 设得过低，可能导致策略只学会模仿技能，而不顾及任务。同样，若 λtask 过高，策略可能会通过不自然的行为完成任务，完全忽略技能指令。因此，我们不使用固定的任务权重，也不手动为 λtask 构造退火调度，而是采用比例控制器，在整个训练过程中动态调整 λtask，方式与 Mentzer 等人 [2021]类似。该控制器由目标任务奖励 $\hat{r}\text{tar}$、控制器增益 k_p 以及一个用于确保数值稳定性的小正数 ϵ参数化。在训练周期 i，我们计算经验缓冲区中的平均任务奖励 $r\text{task}_i$。随后，根据 $r\text{task}_i$与 $\hat{r}$ tar 在对数空间中的误差，更新任务权重 λtask_i：

$$\lambda_{i+1}^{\text{task}} = \exp\left(\log\left(\lambda_i^{\text{task}}\right) + k_p\left(\log\left(\hat{r}^{\text{tar}} + \epsilon\right) - \log\left(\bar{r}_i^{\text{task}} + \epsilon\right)\right)\right) \tag{20}$$

任务权重初始化为 $\lambda\text{任务}_0 = 3$，并将 $\lambda\text{任务}_i$限制在范围 $[0.5, 3]$内。对于位置任务，我们将目标任务奖励权重设为 0.15；对于击打任务，我们将目标奖励设为 0.3。对于朝向任务，我们发现控制器并非必要，因此使用了常数 $\lambda\text{任务} = 1$。

C 多项选择模型示例输出

在表1中，我们提供了任务指令的示例，以及多项选择问答模型所对应的物体和策略



(a)“向前行走”、“一边摆动手臂一边向前行走”。

(b)“一边挥动手臂一边向前冲刺”。

(c)“踢”、“用右腿踢”、“右踢后向前迈步”、“右 腿踢”。

(d) “向左盾击”、“盾牌向左猛击”、“静止站立时向左盾牌猛击”。

(e) “向右斩击”、“向右挥击”、“向右挥剑”、“站定后向右斩击”。

图 7：参考运动 c 嘴唇（左侧）及其对应的说明文字，以及模拟器生成的动作 ed 当角色被指示通过语言指令执行参考技能时（右侧）。

已识别。我们观察到，即使用户使用一些别出心裁的任务指令，例如“消灭那个绿色家伙”或“溜达到那家栗色酒馆”，问答模型仍能正确识别用户的意图。我们还提供了几个问答模型错误识别任务和/或物体的示例。例如，模型预测任务指令“前往蓝色目标”指的是击打任务，而非位置任务；但它能正确将“前往蓝色方块”和“前往蓝色塔楼”识别为位置任务。问答模型有时也会受到改述方式的影响，例如，它能正确识别“导航到酸橙色长方体”中的任务，却无法识别“朝酸橙色长方体方向导航”中的任务。

D PADL 与其他对抗式强化学习框架的比较

训练 PADL 智能体（详见第 6 节）时，技能目标会明确奖励智能体模仿数据集中每个动作片段的能力；为此，判别器是在状态转移与技能嵌入的联合分布上进行训练的。我们发现，与其他在对抗强化学习中仅使用基于状态转移边缘分布训练的判别器的工作相比，联合判别器有助于缓解 PADL 训练期间的模式坍塌。这里我们具体将本方法与两个相关的对抗式强化学习框架进行比较：AMP [Peng 等人 2021] 和 ASE [Peng 等人 2022]。

D.1 与 AMP 的比较

与 PADL 类似，AMP 通过结合任务奖励和技能奖励来训练智能体。然而，由于 AMP 的技能奖励使用边缘判别器，可能会发生模式坍塌：智能体在完成高层任务时，只专注于模仿参考动作数据中特定子集的技能。PADL 在技能奖励中使用联合判别器，并明确训练策略使用不同的参考技能来完成高层任务，因此能够提升策略对数据集的覆盖度。此外，与 AMP 智能体不同，PADL 智能体以编码待使用技能的潜变量为条件。这使用户能够实时控制

表1：多项选择问答模型识别出的示例任务指令及相应的物体和任务。

任务指令	识别出的物体	识别出的任务
“撞倒蓝色方块”	“附近的蓝色物体。” √	“撞倒指定物体。” √
“撞倒绿色方块”	“附近的绿色物体。” √	“撞倒指定物体。” √
“前往红色方块”	“附近的红色物体。”√	“导航至指定目的地。”√
“前往橙色方块”	“附近的橙色物体。”√	“导航至指定目的地。”√
“面朝紫色方块”	“附近的紫色物体。” √	“调整自身朝向，使其面朝指定朝向。”√
“撞倒紫色目标”	“附近的紫色物体。” √	“撞倒指定物体。” √
“转向蓝色目标”	“附近的蓝色物体。” √	“调整自身朝向，使其面向指定朝向。”√
“转向橙色目标”	“附近的橙色物体。” √	“调整自身朝向，使其面向指定朝向。”√
"面朝橙色目标"	"附近的橙色物体。" √	"调整自身朝向，使其面朝指定朝向。"√
"面朝紫色目标"	"附近的紫色物体。" √	"调整自身朝向，使其面朝指定朝向。"√
“前往蓝色目标”	“附近的蓝色物体。” √	“撞倒指定物体。”
“推倒红色塔”	“附近的红色物体。”√	“撞倒指定物体。” √
“面向橙色方尖碑”	“附近的橙色物体。” √	“调整自身朝向，使其面向指定朝向。” √
“前往酸橙色长方体”	“附近的绿色物体。” √	“前往指定目的地。” √
"朝酸橙色长方体移动"	"附近的绿色物体。" √	"调整自身朝向，使其面向指定朝向。"
"看向停车标志"	"附近的红色物体。" √	"调整自身朝向，使其面向指定朝向。" √
"观赏日落"	"附近的红色物体。" √	"调整自身朝向，使其面向指定朝向。" √
"撞倒钴蓝色方块"	"附近的红色物体。"	"撞倒指定物体。" √
“靠近紫罗兰色标记”	“附近的紫色物体。”√	“让自己面朝指定朝向。”
“摧毁绿色家伙”	“附近的绿色物体。” √	“撞倒指定物体。” √
“慢悠悠地晃到那家栗色酒馆去”	“附近的红色物体。”√	“前往指定目的地。” √

受训智能体用来完成任务的技能，这对有效的语言控制至关重要。

D.2 与 ASE 的比较

ASE 低层控制器和 PADL 控制器都以技能潜变量为条件，因此可以动态控制智能体使用的技能。在 ASE 训练期间，潜变量从先验分布（例如单位球面）中随机抽取；在整个训练过程中，策略会在这个潜在空间中学习有意义的表示，这一过程通过边缘判别器与编码器结合实现，编码器会促使潜变量与其对应的策略轨迹之间具有较高的互信息。这种方法也可能导致模式坍塌，使参考数据集中的技能只有一部分能在潜在空间中得到表示。PADL 通过为参考数据集中的每个动作片段分配一个独特的动作潜变量来缓解此类模式坍塌（这些潜变量在技能嵌入阶段学习），从而保证每个动作片段都能在潜在空间中得到表示。

在开发语言控制动画系统的早期实验中，我们曾在 ASE 低级控制器之上添加一个语言头。我们通过从单位球面采样潜变量，使用这些潜变量从预训练控制器检查点中记录轨迹，并用自然语言为轨迹添加标注，从而创建了一个由（潜变量、描述文本）对组成的数据集。随后，我们训练了一个小型多层感知机来逆转标注过程，将轨迹描述文本的 BERT 嵌入映射到生成该轨迹的对应潜变量。这种方法使策略的技能能够由语言控制，但标注效率低下，因为每个（潜变量、描述文本）对数据集都只适用于特定检查点学习到的潜在空间。另一个 ASE 检查点（具有不同的

算法 1： 多任务聚合

```
智能体状态 ← 智能体状态;
当 尚未完成 时
    技能潜变量 = Enc_l(getSkillCommand());
    策略Idx，物体Idx = QAModel(getTaskCommand());
    策略 = 策略[策略Idx];
    目标物体状态 = 物体[物体索引];
    动作 = 策略(智能体状态, 技能潜变量, 目标物体状态);
    智能体状态 = 环境.step(动作);
end
```

学习到的潜在空间）则需要重新收集一套全新的标注数据集。此外，由于模式崩溃，数据集中更复杂的技能往往无法在策略的潜在空间中得到体现。