arXiv:2305.02195v1 [cs.CV] 2 May 2023

CALM：面向可控虚拟角色的条件对抗潜变量模型

CHEN TESSLER，NVIDIA，以色列；YONI KASTEN，NVIDIA，以色列；YUNRONG GUO，NVIDIA，加拿大；SHIE MANNOR，NVIDIA，以色列及以色列理工学院；GAL CHECHIK，NVIDIA，以色列及以色列巴伊兰大学；XUE BIN PENG，NVIDIA，加拿大及加拿大西蒙弗雷泽大学

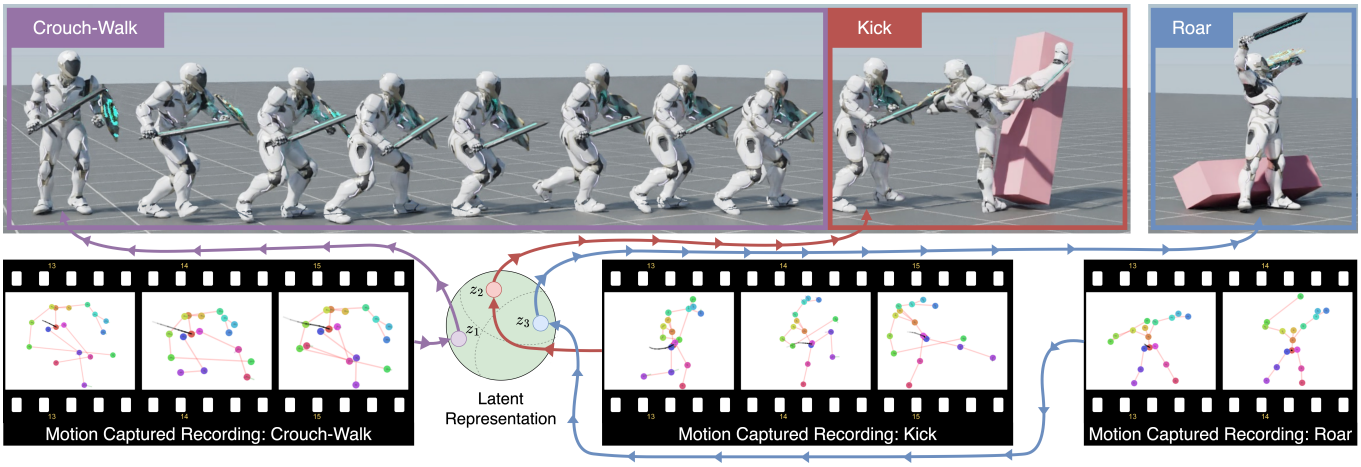


图 1. 我们的框架让用户能够通过示范来控制物理仿真角色的行为；这些示范以动作捕捉数据的低维潜嵌入形式编码。在此示例中，角色收到的指令是蹲伏行走至目标处、进入攻击范围后踢击，最后举起双臂庆祝。

在这项工作中，我们提出了条件对抗潜在模型（CALM），这是一种为用户控制的交互式虚拟角色生成多样且可控行为的方法。CALM 利用模仿学习，学习能够捕捉人体运动复杂性与多样性的运动表征，并实现对角色动作的直接控制。该方法联合学习控制策略和运动编码器，后者能够重建给定动作的关键特征，而非单纯照搬动作。结果表明，CALM 学会了语义化的动作表征，从而能够控制生成的动作，并通过风格条件训练更高层次的任务。训练完成后，用户便可通过直观的界面控制角色，类似于电子游戏中的操作界面。

1 引言

虚拟环境和交互式角色日益普及，也愈发易于使用，但由于人体运动的复杂性，为这些虚拟智能体创造逼真且多样化的行为仍是一项挑战。为了打造交互式和沉浸式体验，虚拟智能体必须能够以逼真的方式适应不同的环境和用户输入，这要求它们能够按需执行丰富多样的行为。为此，我们需要开发能够生成复杂且逼真的控制模型。

作者地址：Chen Tessler，NVIDIA，以色列，ctessler@nvidia.com；Yoni Kasten，NVIDIA，以色列；Yunrong Guo，NVIDIA，加拿大；Shie Mannor，NVIDIA，以色列及以色列理工学院；Gal Chechik，NVIDIA，以色列及以色列巴伊兰大学；Xue Bin Peng，NVIDIA，加拿大及加拿大西门菲莎大学。

行为，同时考虑环境的属性。例如，在虚拟现实游戏中，与虚拟角色和物体互动的玩家期望它们的行为符合现实。这包括响应用户指令，以及在虚拟环境中导航。当虚拟智能体无法自然地响应用户输入时，沉浸式体验可能会受到破坏。

机器学习的最新进展以及高质量人体动作捕捉数据的获取，推动了能够复现人类行为的控制策略的发展。该领域的早期研究，例如 [Peng 等人 2018]，，专注于模仿单个动作片段。然而，由于每个动作都是通过独立的控制器学习的，这种方法难以有效扩展。后续研究 [Peng 等，2022，ASE] 旨在通过学习潜变量条件控制器并最大化互信息目标，提升生成动作的多样性。在包含多种动作的数据集上训练后，虽然涌现出了不同的行为，但代价是失去了控制生成动作的能力。

在这些先前工作的基础上，我们提出条件对抗潜变量模型（CALM），这是一种学习动作表征的方法，既能捕捉人体运动的复杂性和多样性，又能提供一个可直接控制角色动作的接口。给定原始动作捕捉记录，

如图1所示，CALM 将这些记录编码为潜变量表征。CALM 还会将给定的潜在向量解码为物理仿真角色的一项技能，使其能够在以目标动作为条件的情况下执行高层任务。如图1所示，无需进一步训练，仅使用预先定义的原始动作数据，智能体便能运用一组目标技能解决具有挑战性的任务。我们的框架的一项关键优势在于，策略无需精确复现原始参考动作。相反，只要生成的动作符合特定动作片段的分布特征，策略就可以灵活地产生多样化的动作。这使策略能够偏离动作数据，生成新的多样化行为；尽管这些行为并未在原始数据集中明确呈现，却依然显得自然。

与先前工作相比，我们专注于无监督技术。数据未经过标注，我们也不假设事先了解动作之间的语义关联。我们提出了一种端到端方法，能够联合学习技能的有意义语义表示，以及能够生成所选技能的控制策略。虽然先前工作使用语言来推导语义关联 [Juravsky 等人 2022], 但我们证明，语义含义可以直接从动作中推断出来，而相似性则是在策略复现动作的情境下确定的。这种引导角色动作的能力，使得无需进一步重新训练即可利用类似人类的控制来解决任务，展现了 CALM 生成交互式且可控虚拟角色的能力。综上所述，我们的贡献如下：

(1) 我们提出了一种方法，利用无标签动作捕捉数据，对生成式动作控制器和运动编码器进行联合训练。由此得到的策略可以通过其编码 $z = E(M)$，引导生成动作 M。(2) 我们引入了精确训练，这是一种重用预训练策略并利用已学习潜在空间中的相似性的方法，从而在解决运动等高层任务时，实现对所生成动作的控制。(3) 最后，我们证明，结合步骤 1 和步骤 2，无需进一步训练，也无需精心设计奖励函数或终止条件，即可通过设计简单的有限状态机来完成任务。

2 相关工作

我们的目标是从多样化的无标注数据中学习丰富且可复用的技能表示，用于角色控制。数据驱动的动作生成领域大致可分为运动学模型和基于物理的模型。运动学模型直接生成姿态轨迹，而不显式考虑物理约束；基于物理的模型则通常使用物理仿真环境，在动作生成过程中确保真实动力学。

我们区分参数化模型的两种控制方法：直接预测和基于潜在表示/语言的控制。在直接预测中，模型学习为预定义任务生成动作；而基于潜变量和语言的控制则涉及学习动作生成模型。由于我们的重点是创建可交互、可控的角色，因此我们立足于物理约束生成领域的相关研究，并聚焦于基于潜空间的控制。

2.1 受物理约束的动作生成

在物理约束生成中，模型预测电机驱动信号，并将其输入模拟器。随后，模拟器在遵循各种支配定律（如重力和摩擦力）的同时，模拟角色的动作并生成下一个状态。深度学习领域的最新进展，特别是深度强化学习，显著提升了生成质量。DeepMimic [Peng 等人 2018], 便是一个例子，他们定义了一种运动跟踪奖励，用于训练策略模仿特定动作。然而，这些方法侧重于模仿单个预定义的动作片段。

直接预测。 在直接预测中，目标是直接学习解决下游任务。例如，可能要求智能体触及目标位置。在此，AMP [Peng 等人 2021] 将对抗模仿学习 [Ho 和 Ermon 2016、GAIL] 与经典强化学习相结合。智能体试图在最大化任务奖励的同时，成功欺骗判别器。然而，当示范数据分布与任务不匹配时，所得行为并不令人满意，生成的模型也无法泛化到新任务——这就需要针对每项任务重新优化所提供的数据并重新训练。

基于潜空间的控制。 这里的任务是学习一个可供采样的行为流形。Park 等人 [2019] 学习预测未来状态，再训练控制器来跟踪该行为。Won 等人 [2022]学习一个以下一状态为条件的条件变分自编码器（VAE）。最后，与我们的工作最为接近的是，ASE [Peng 等人 2022] 提出了一种无监督判别式学习流程。通过最大化潜空间与生成的下一状态之间的互信息，并同时优化模仿学习目标，智能体学会生成多样化的动作。

这些方法有一个共同点——由此得到的潜空间难以控制。在这项工作中，我们联合学习人体运动的稠密表示和一种可直接控制的潜变量条件策略。

2.2 表示学习

表示学习旨在捕捉数据的底层结构，是机器学习领域的重要研究方向，尤其是在图像和视频的表示方面。定义数据点之间相似性的方法有多种，而与我们方法最接近的一种是直接利用下游任务。例如，生成对抗网络 [Goodfellow 等人 2020, GAN] ，以及变分自编码器 [Kingma 和 Welling 2013] 学习低维潜在表示，以便从参考分布中重建数据。

在学习动作表征的背景下，可以区分变分自编码器和基于对抗的训练方法。动作变分自编码器 [Ling 等，2020；Won 等， 2022] 能够联合学习动作的表征与生成。虽然基于变分自编码器的方法通常更容易训练，但其缺陷在于受重建损失驱动，因此无法偏离数据分布。

对抗方法的优势在于，它们能够生成符合参考数据分布的可能性较高的新动作。这在可控角色的应用场景中尤为重要。The

角色必须能够在每对动作之间自然过渡，即使没有提供明确的示范。

如 ASE [Peng 等人 2022], 所示，该策略能够学习生成多样化的行为，其泛化能力超越了动作重建。然而，其生成的潜在空间缺乏全局语义结构。因此，ASE 容易出现模式崩塌，且无法提供一种简便的方式将动作映射到潜在空间中，而这对于可引导性和控制而言是一项重要要求。其他研究工作，例如 PADL [Juravsky 等人 2022]，则利用了来自自然语言的监督。这些方法能够训练与语言对齐的动作表征，进而根据自然语言指令生成行为。然而，它们需要获取标注数据。

在这项工作中，数据没有标签，表征通过模仿学习策略端到端地学习。因此，动作之间的语义联系取决于策略复现这些动作的情境。

2.3 分层强化学习

潜在生成模型可以视为选项/技能框架的一部分 [Sutton 等人 1999]。选项框架区分高层控制器和低层控制器。低层控制器会生成微动作，这些高频动作能够产生多样的行为。另一方面，高层控制器通常以较低频率进行规划。在每个时间步，它都会选择要执行的技能。技能是具有时间延展性的动作，或者在我们的语境中，是长期动作。

以往在分层强化学习（HRL）方面的研究表明，利用有意义的技能不仅可以加快训练速度[Tessler 等人 2017],，还可以确保解决方案位于数据的支持范围内 [Peng 等人 2022]。

3 强化学习背景

在本研究中，预训练和下游任务均被建模为强化学习问题：智能体依据策略 π与环境交互。在每个步骤t，智能体观察一个状态 s_t，并从策略 $a_t \sim \pi(a_t|s_t)$中采样一个动作 a_t。随后，环境依据转移概率 $p(s_{t+1}|s_t, a_t)$转移至下一个状态 s_{t+1}。目标是最大化折扣累积奖励，其定义为

$$J = \mathbb{E}_{p(\tau|\pi)} \left[\sum_{t=0}^{T} \gamma^t r_t \middle| s_0 = s \right], \tag{1}$$

其中，$p(\tau|\pi) = p(s_0)\Pi_{t=0}^{T-1} p(s_{t+1}|s_t, a_t)\pi(a_t|s_t)$ 表示轨迹 $\tau = [s_0, a_0, r_0, \ldots, s_{T-1}, a_{T-1}, r_{T-1}, s_T]$ 的似然，而 $\gamma \in [0, 1)$ 是折扣因子，它决定智能体是目光短浅，还是会考虑更长远的结果。

4 概述

在本文中，我们介绍条件对抗潜变量模型（CALM），这是一种可扩展的数据驱动方法，用于为物理仿真角色创建可操控的控制器。这些角色的控制方式与玩家控制游戏中的虚拟角色类似，都是通过提供一系列移动和动作指令来实现的。图 2 展示了我们的框架。它由以下部分组成。

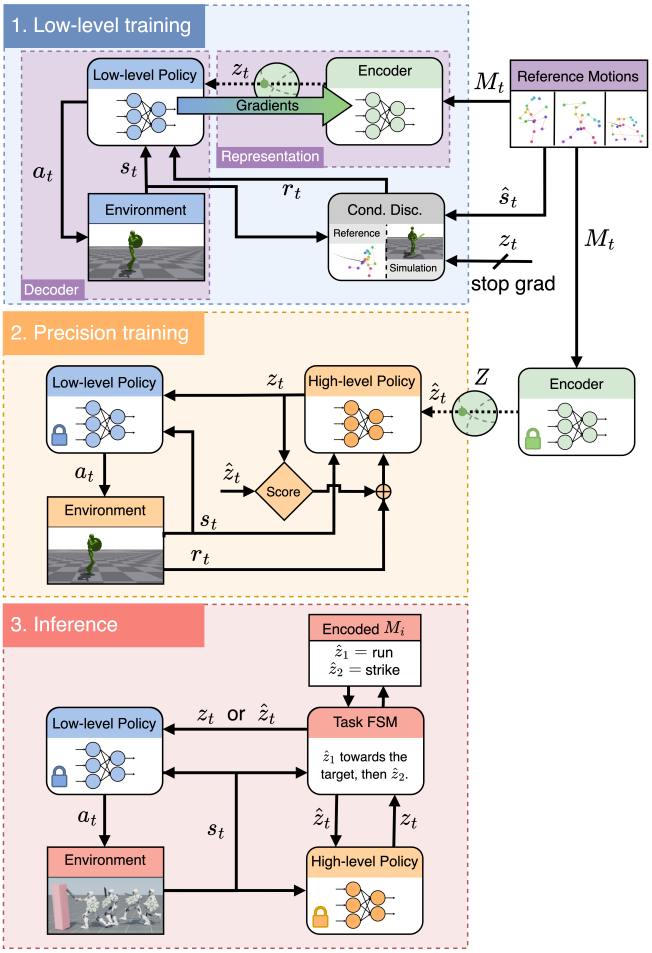


图 2. **CALM 框架**由三个阶段组成。**1**）在初始训练阶段，编码器和低层策略都通过利用条件判别器提供的反馈来构建。编码器学习生成一种精简的表征，以概括动作的核心；而与环境交互的低层策略则充当解码器。**2**）预训练阶段结束后，编码器和低层策略将被冻结。下一步是**精确训练**。在此阶段，将训练高层策略来控制低层策略。**3**）在最后的**推理**阶段，高层策略也会被冻结。随后，使用有限状态机（FSM）来描述复杂任务的解决方案。有限状态机是一种无需训练的基于规则的系统，它取代了奖励。根据任务状态，有限状态机要么向高层策略提供指令，要么直接向低层策略提供潜变量。

分为三个步骤。(1) 运动编码器和动作生成器的低层训练。(2) 使用高层策略进行精确训练。(3) 推理，在此过程中无需进一步训练即可完成任务。

在低层训练期间，CALM 学习一个编码器 E。它从动作参考数据集 M 中获取一个动作 M，即一段关节位置的时间序列，并将其映射为低维潜在表示 $z \in$ Z。此外，CALM 还会联合学习一个解码器。解码器是一个低层策略 $\pi(a|s, z)$，它与模拟器交互，并生成与参考数据集相似的动作

。该策略可按需生成多种行为，但不以动作的方向为条件。例如，可以指示它行走，但无法直观地控制行走方向。

接下来，为了控制运动方向，我们训练一个任务驱动的高层策略，以选择潜变量 z_t。这些潜变量会提供给低层策略，由其生成所请求的动作。在这里，运动编码器用于约束潜变量 z_t，使其接近预先指定的动作 $\hat{z}_t$，从而引导高层策略采用期望的行为风格。例如，在朝给定方向移动的同时执行蹲走动作。

最后，将先前训练好的模型组合起来，无需额外训练即可构成复杂动作。为此，用户创建一个包含标准规则和指令的有限状态机（FSM）。这些规则和指令决定执行哪种动作，其方式类似于用户控制电子游戏角色。例如，它们决定角色应执行一种简单动作（由低层策略直接执行），还是执行一种需要高层控制的定向动作。举例来说，可以构建如下有限状态机：(a)“朝目标蹲伏行走，直到距离 < 1米”，然后 (b)“踢击”，最后 (c)“庆祝”（图1）。

5 条件对抗潜变量模型

学习丰富且可复用的技能表示，能够让角色按需生成各种各样的动作，从而为游戏和视觉特效等应用开辟了广阔空间。在 CALM 中，这些技能通过运动条件策略 $\pi(a|s,\ z)$ 建模，其中动作 M被编码为潜变量 $z \in \mathrm{Z}$。这种映射由运动编码器 $z = E(M)$建模，并通过对从参考数据集 M 中均匀采样的动作M 求解条件模仿学习目标来学习：

$$\max_{\pi} -\mathbb{E}_{M\in\mathcal{M}}\left[D_{\mathrm{JS}}\left(d^{\pi}\left(s,s'|z\right)\big|_{z=E(M)}\middle\|d^{M}\left(\hat{s},\hat{s}'\right)\right)\right], \quad (2)$$

其中 DJS 表示 Jensen-Shannon散度，而 $d^{\pi}(s,\ s_{\prime}|z)\big|_{z=E(M)}$和 $d^{M}(\hat{s},\ \hat{s}_{\prime})$ 分别表示策略和参考动作的状态转移分布。

这一目标与模仿学习文献中的既有研究相关，具体而言，是从观测中学习（LfO）[Torabi 等人 2018]。在这种设定中，智能体获得一个示范数据集，其中仅包含状态转移/观测，不包含产生这些转移的底层动作。生成对抗观测模仿 [Torabi 等人，2018 年，GAIfO] 联合训练策略和判别器。策略生成状态转移（$s,\ s_{\prime}$），而判别器则试图区分这些转移与从数据中采样的示范（$\hat{s},\ \hat{s}_{\prime}$）。然而，面对多种多样的动作时，GAIfO 容易发生模式崩塌，即策略无法对大型数据集中的所有不同动作行为进行建模，最终得到的模型也不会显式学习可用于引导策略完成下游任务的技能嵌入。

在先前的研究中，ASE [Peng 等人 2022] 尝试通过引入潜变量并最大化互信息来解决这些问题。他们证明，模型在输入随机采样的变量时能够生成多样化的行为。然而，仅依赖互信息损失还不够。为避免模式崩塌，ASE 还采用了多样性损失，该损失规定相似的潜变量应产生相似的动作分布。我们的研究表明，这对于可控性尤为重要。编码器学习到的动作与潜变量之间的映射并不精确，导致控制器在以相同潜变量为条件时无法生成相似的动作，从而难以引导策略执行特定动作。

CALM 通过使用条件判别器来缓解先前方法中的问题，该判别器会迫使策略重现数据集中的每个动作。在每次迭代中，都会从参考数据集中随机采样一个动作 M。编码器将该动作映射为潜在编码 $z = E(M)$。随后，策略和条件判别器都以该潜变量 z 为条件。让判别器以潜变量为条件有助于缓解模式坍塌，因为这会迫使策略生成与给定潜变量对应动作相似的动作。由此得到以下判别器损失

$$\mathcal{L}_{\mathcal{D}} = -\mathbb{E}_{M\in\mathcal{M}}\Bigg[\mathbb{E}_{d^{M}(\hat{s},\hat{s}')}\left(\log\mathcal{D}(\hat{s},\hat{s}'|z)\right) + \mathbb{E}_{d^{\pi(z)}(s,s')}\left(\log(1-\mathcal{D}(s,s'|z))\right)\Bigg|z = E(M)\Bigg], \quad (3)$$

以及策略目标

$$J = \mathbb{E}_{M\in\mathcal{M}}\left[\mathbb{E}_{p(\tau|\pi,z)}\left(\sum_t \gamma^t r(s_t,s_{t+1},z)\Big|s_0 = s\right)\Bigg|z = E(M)\right], \quad (4)$$

奖励为 $r(s_t,\ s_{t+1},\ z) = -\log(1-\mathrm{D}(s_t,\ s_{t+1}|z))$。

5.1 实际考量

本节介绍实现 CALM 的实用方案所需的设计决策，以及详细的实现细节。

5.1.1 **编码器。**理想的表征应针对控制策略进行优化，而不是像对比学习方法那样依靠辅助目标来驱动潜在表示的结构 [Oord 等人 2018]。我们将展示，在优化预训练目标公式（4）时，可以利用来自策略的梯度，以端到端的方式训练出有效的编码器。此外，这种方法能够形成具有清晰语义结构的潜在空间，使相似动作彼此聚集，从而实现具有语义意义的插值。

编码器的输出被投影到 l_2 单位超球面上。这一约束受到该领域先前研究的启发 [Bojanowski 和 Joulin 2017；Chen 等人 2020；Parkhi 等人 2015；Peng 等人 2022；Wang 等人 2017；Wang 和 Isola 2020；Xu 和 Durrett 2018]，，并具有多项优势。例如，已知固定范数向量有助于提升机器学习中的训练稳定性，而机器学习中通常会用到点积 [Wang 等人 2017；Xu 和 Durrett 2018]。在动作生成领域，单位 l_2 范数约束所施加的潜在空间结构，降低了推理期间因采样分布外潜变量而产生不自然行为的可能性，Peng 等人 [2022]对此已有证明。

由于动作片段的长度可以任意延长，我们将数据划分为长度为 2 秒且相互重叠的子动作。这样得到的动作足够长，能够呈现鲜明且连贯的特征。此外，在时间上彼此接近的子动作很可能具有相似的特征。我们利用这一认识来

通过对预测嵌入 [施加对齐损失和均匀性损失来改进潜在空间结构 2020]Wang 和 Isola。

$$\mathcal{L}_{\text{align}} = \mathbb{E}_{(M,M')\overset{\text{overlapping}}{\sim}\mathcal{M}}\left[\left\|E(M) - E(M')\right\|_2^2\right], \tag{5}$$

$$\mathcal{L}_{\text{uniform}} = \log \mathbb{E}_{(M,M')\overset{\text{i.i.d}}{\sim}\mathcal{M}}\left[\exp\left(-2\left\|E(M) - E(M')\right\|_2^2\right)\right], \tag{6}$$

其中，重叠表示来自同一原始动作序列、具有非零重叠的两个秒级子动作（M，$M\prime$），而 独立同分布表示从数据中随机采样的子动作。

5.1.2 技能转换。执行新任务时，智能体通常需要依次在不同行为之间转换。即使是简单的任务，例如到达某个位置，角色也需要结合行走、转身和站立等技能。为了实现动作之间平滑且稳健的转换，我们通过在随机时间步改变条件动作 M，明确训练模型在不同动作之间进行转换。这样可以教会策略成功地在彼此迥异的动作之间转换。

5.1.3 判别损失。在对抗模仿学习中，判别奖励为智能体提供学习信号。为改善训练动态， [Juravsky 等，2022；Peng 等，2022]2021] 提出使用梯度惩罚正则项和负采样。这种正则化有助于缓解判别器过拟合，并为策略生成更平滑的优化景观。由此提高了训练稳定性和生成动作的整体质量。此外，由于我们的目标是学习一种针对控制任务进行了优化的动作编码，因此我们阻止梯度从判别器的目标函数流向编码器。由此得到以下目标函数：

$$\begin{aligned}\mathcal{L}_{\mathcal{D}} = -\mathbb{E}_{M\in\mathcal{M}}\Big(&\mathbb{E}_{d^{\pi}(s,s'|z)}\left[\log\left(1-\mathcal{D}(s,s'|z)\right)\right] \\ &+\mathbb{E}_{d^{M}(\hat{s},\hat{s}')}\left[\log\mathcal{D}(\hat{s},\hat{s}'|z)+\log\left(1-\mathcal{D}(s,s'|z'\sim\mathcal{Z})\right)\right] \\ &+w_{\text{gp}}\mathbb{E}_{d^{M}(\hat{s},\hat{s}')}\left[||\nabla_{\theta}\mathcal{D}(\theta)|_{\theta=(\hat{s},\hat{s}'|z)}||^2\right]\Big|z=\text{stop grad}(E(M))\Big).\end{aligned} \tag{7}$$

这里，w_{gp} 是梯度惩罚系数。

6 高层控制

低层控制器训练完成后，会将其用作动作生成器，使角色的动作以数据中观察到的动作为基础。本节介绍如何训练高层策略来控制动作执行的方向；随后便可利用该策略，以不同方式解决复杂任务，而无需专门针对这些任务进行训练。

6.1 精确训练

给定一个动作编码 $\hat{z} = E(M)$，低层策略会生成与之匹配的动作。然而，向低层策略提供“跑步”的编码，并不能控制跑步方向。为了实现更好的控制，我们训练一个高层策略，使其能够按照指定的动作形式和方向生成动作。

具体而言，高层策略会生成潜变量 z_t。随后，这些潜变量会提供给控制角色的低层策略。任务要求角色朝指定方向移动 $\mathbf{d}_t^*$，同时进行蹲伏行走 M_1，或者改为冲刺 M_2。

我们通过结合任务奖励和潜变量相似性损失来实现这一点。具体而言，给定动作编码 $\hat{z} = E(M)$，我们使用以下奖励来训练高层策略

$$r_t^{\text{locomotion}} = \exp\left(-0.25\left\|\mathbf{d}_t^* - \frac{\dot{\mathbf{x}}_t^{\text{root}}}{||\dot{\mathbf{x}}_t^{\text{root}}||}\right\|^2\right) + \exp\left(-4||z_t - \hat{z}||^2\right), \tag{8}$$

其中，$\dot{\mathbf{x}}\text{root}_t$ 是角色的速度。

6.2 示范引导

给定预训练的编码器和低层策略（第 5 节），以及预训练的高层策略（第 6.1 节），任务设计者可以用自然直观的方式描述如何完成任务，从而克服奖励设计的脆弱性。在这里，任务设计者为智能体应执行的各种动作提供示范，并将其置于有限状态机（FSM）中，由该状态机决定何时在不同行为之间转换。例如，击打物体这一任务可以分为三个阶段：“跑向物体”；进入 0.5 米范围后，“执行一次攻击”；然后“保持待机”，直到收到下一条指令。

要实现如此高水平的控制，需要兼具低层策略所提供的灵活性，以及预训练高层策略所提供的精确性。在需要精确控制的阶段（例如朝指定方向移动），有限状态机（FSM）会向高层策略提供所需的动作嵌入 $\hat{z}$ 以及执行该动作的方向。当转换到独立动作（例如特定的挥剑动作）时，FSM 会将动作编码直接提供给低层策略。

这使得无需重新训练即可重复使用，并且类似于用户使用游戏控制器与角色交互的方式。通过单一组合的 (a) 低层策略、(b) 编码器和 (c) 高层策略，即可解决形式各异的未见任务。

7 实验

数据：为了获得多样化的动作控制能力，低层策略使用 160 个动作片段进行训练，这些片段总时长超过 30 分钟 [Reallusion 2022]。每个动作片段都会被拆分为连续重叠的 2 秒子序列，不考虑过渡边界。因此，如果一个动作片段包含多种技能，这些 2 秒片段可能会包含来自多种技能的动作。。其中包括各种形式的行走等基本动作，以及剑击组合等更复杂的动作。补充材料对此进行了更详细的说明。

训练流程：在整个预训练过程中，智能体与环境交互，进行长度为 K 步的轨迹采样。在随机时间步，从参考数据集中采样一个随机动作，然后将其编码并提供给低层策略。随后，轨迹由低层策略在潜变量 z的条件下与环境交互所产生的观测状态构成。接着，使用 PPO [Schulman 等人 2017]，基于收集到的轨迹训练低层策略。

我们在单张 A100 GPU 上，将训练并行化至 4096 个 Isaac Gym [Makoviychuk et al.2021] 环境中，总计执行 50 亿步。低层（高层）策略以 30（6）Hz 的频率做出决策。潜在空间 Z 定义为一个 64 维超球面。

	编码器质量 ↓	多样性 ↑	可控性 ↑
CALM	**0.23**	**19.8±0.1**	**78%**
ASE	0.68	18.6±0.4	35%

表 1. **预训练：** 对学习得到的编码器和低层策略进行定量评估。我们测量 Fisher 集中系数（编码器质量）、初始分数（多样性）和生成准确率（可控性）。

模型架构编码器是一个标准的 MLP，将$E(M) \mapsto z$ 映射为 $\mapsto z$，策略和条件判别器各自包含一个额外的输入头 $H(z)$，用于潜变量解析，随后接入一个 MLP $\pi(s,\ H(z)) \mapsto a$，其中 $a \in \mathbb{R}^{31}$。

8 结果

我们通过使用 CALM 学习技能嵌入来测试该方法的有效性，这些技能嵌入使仿真人形机器人能够执行各种运动控制任务。我们展示了 CALM 能够学习多样化人体运动在语义上有意义的潜在表示，以及可控策略。随后，我们训练了一个高层策略，用于控制这些动作执行的方向。最后，我们展示了如何重复利用这些低层和高层策略，在无需进一步训练的情况下解决未见过的任务。请在提供的补充视频 [1] 中观看 CALM 生成的动作。

8.1 可控动作生成

我们首先从三个方面对 CALM 进行分析：（1）编码器质量，（2）低层控制器的多样性，以及（3）组合系统的可控性。结果见表 1。我们重点将 CALM 与 ASE 进行比较。ASE [Peng 等人 2022], 是一种潜在生成模型，能够学习将任意潜变量映射为动作。

实验 *1*：编码器质量。我们使用 Fisher 类别可分性度量 [Bishop 等人 1995]，评估编码器学习到的表征，据此衡量潜在空间中不同动作类别之间的可分性。这里，动作类别定义为单个动作文件中的子动作。如表 1 所示，CALM 学会将动作编码为区分度更高的表征。

实验 *2*：多样性。我们使用第 5 节中的参考数据集训练了一个分类器，将动作序列映射到其来源动作的索引。对于从随机采样的潜变量 $z \sim \mathrm{Z}$ 生成的动作，我们报告其初始分数 [Salimans 等人 2016]。如表 1 所示，CALM 显著提升了生成动作的多样性。

实验 *3*：可控性。最后，我们通过用户研究量化了 CALM 生成所请求动作的效果。我们向评审者提供参考动作和文本描述（摘自 Juravsky 等人 [2022]），并请他们判断生成的动作是否相似。对于每个模型，我们向评审者展示了 40 个参考动作，以及每个参考动作对应的 3 个生成结果。我们报告准确率，即控制器生成准确结果的百分比。结果表明，与 ASE 相比，CALM 能更好地控制生成的动作，将感知生成准确率从 35% 提高到 78%。

[1]有关可视化过程中渲染伪影的说明，请参见第 9 节。

动作	朝向		位置		打击	
	风格	得分	结尾	得分	结尾	得分
Run	1	0.92		0.98		0.96
				0.99		1
				0.96		0.99
Walk	0.81	0.92		0.99		0.98
				1		1
				0.97		0.96
蹲下 Walk	0.94	0.91		0.98		0.99
				0.99		1
				0.97		0.99

表2. **方向动作控制**（朝向）和**零样本任务求解**的定量评估。我们考虑三种运动形式：跑步、行走和蹲伏行走，每种形式都具有不同的速度和风格。对于每项任务，我们考虑多种结束动作。对于定位任务，我们考虑♂（站立待机）、（庆祝，举起双臂）和 l（蹲伏待机）。对于击打任务，我们考虑：Œ（踢击）、（持盾冲锋）和 s（挥剑）。

这是通过使用条件判别目标，对表征（编码器）和生成式动作模型（低层策略）进行端到端学习实现的。因此，CALM 学会将动作编码为具有语义意义的表征，并训练出能够生成与演示具有相似特征的动作的控制器。

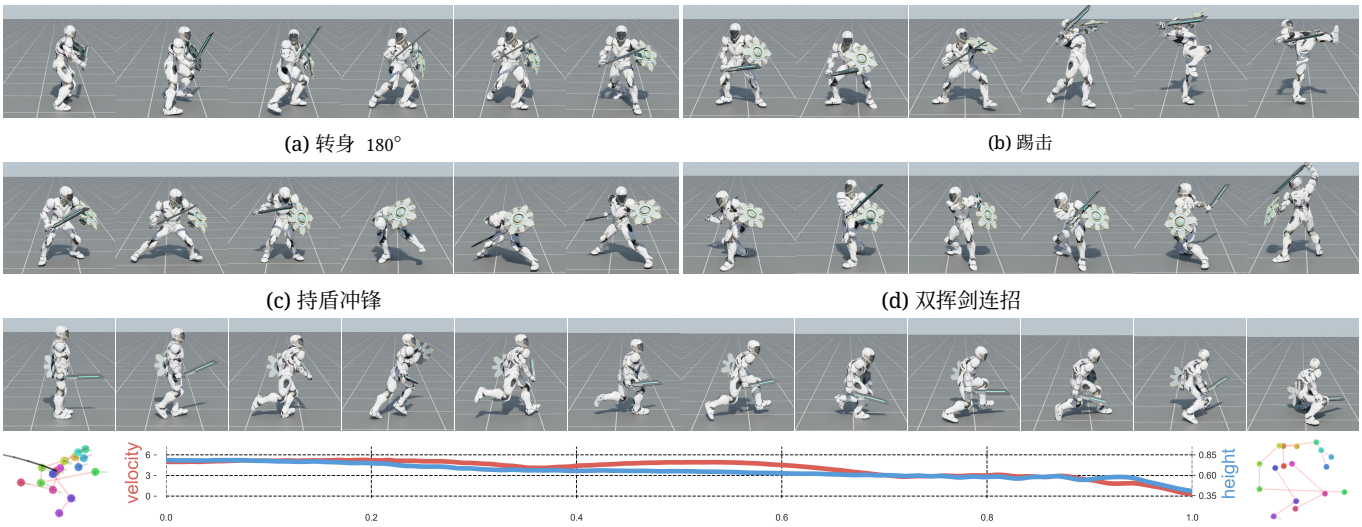
8.1.1 定性分析。在图3中，我们展示了 CALM 生成的动作。在单个片段中，条件动作不断变化，从而在人类般自然地过渡到所请求的动作。此外，为了展示潜在空间的语义结构，我们对两个语义相关的动作“冲刺”和“蹲伏待机”进行编码，并在一段时间内对其编码进行插值。如图3e所示，CALM 在这两个动作之间平滑过渡，同时降低速度和身体高度，并持续以某种形式行走。

8.2 解决下游任务

利用第 8.1 节中的编码器和低层策略，我们展示了如何使用类似电子游戏控制的指令组合动作，以解决未见过的任务。

8.2.1 方向性动作控制。首先，我们展示了给定参考动作 M 和一个方向 d^*时，高层策略可以学会控制低层策略。我们将此任务称为朝向。角色应朝请求的方向产生具有相似特征的动作。我们在图4中展示了学习到的动作，并在表2的“朝向”列中对其进行量化。在此，高层策略通过联合学习“跑步”、“举盾行走”和“蹲下行走”进行条件化。

评估期间，高层策略以固定风格为条件，并在随机时间步改变方向。我们报告了生成请求风格的成功率（由人工评审员评估），以及运动方向的成功率（通过请求方向与实际运动方向之间的余弦距离衡量）。



(a) 转身 180°　(b) 踢击

(c) 持盾冲锋　(d) 双挥剑连招

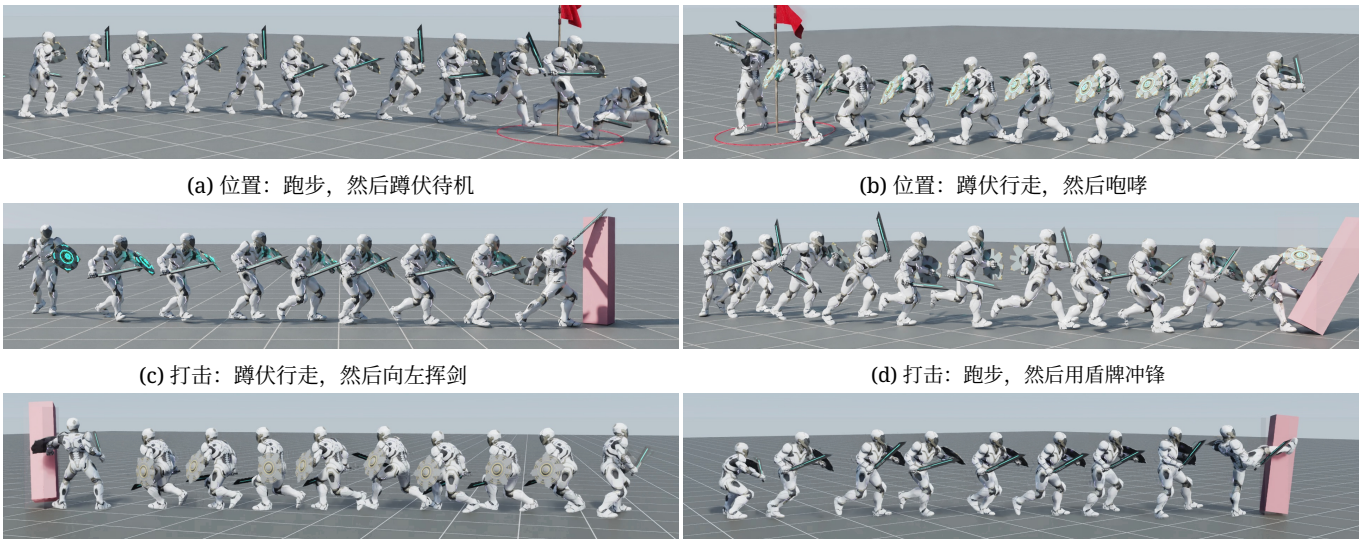
(e) 时间插值：从冲刺到蹲伏待机

图 3. **低层训练：**低层控制器根据演示动作的编码生成的技能。



(a) 朝向：跑步　(b) 朝向：行走，举盾　(c) 朝向：蹲伏行走

图 4. **精确训练：** 训练单个高层控制器，通过奖励引导生成风格约束运动。



(a) 位置：跑步，然后蹲伏待机　(b) 位置：蹲伏行走，然后咆哮

(c) 打击：蹲伏行走，然后向左挥剑　(d) 打击：跑步，然后用盾牌冲锋

(e) 打击：蹲伏行走，然后用盾牌横扫　(f) 打击：行走，然后踢击

图 5. **推理：** 无需进行任何额外训练，CALM 即可解决多种不同的任务，并以多种不同形式完成同一任务。

结果表明，通过将潜变量约束在参考动作编码附近，高层策略能够生成指定风格的动作，同时确保其沿着所要求的方向移动。

8.2.2 无需进一步训练即可解决任务。 在最后一项实验中，我们将可控低层控制器与高层运动策略结合，为未见过的任务提供零样本解决方案。我们考虑两项任务：定位和打击。对于定位任务，智能体应抵达并停留在目标位置——目标位置以旗杆周围的圆圈表示。打击是一项更复杂的任务，要求智能体抵达目标并将其击倒。在这两种情况下，角色都通过对一系列参考动作进行条件化来控制。为此，系统会提供指向目标位置的方向向量，该向量以角色的局部坐标系表示。一旦进入范围，系统便直接向低层策略提供与所请求动作对应的潜变量，例如踢击、持盾冲锋或挥剑。

如图 5 所示的定性结果和表2所示的定量结果，CALM 可以用于解决任务，其方式类似于人类使用游戏控制器完成任务的方式。得益于 CALM 的可控性，在无需进一步训练或针对特定任务设计奖励的情况下，有限状态机（FSM）会依次安排角色在不同动作之间进行转换。最终形成的类人动作组合能够完成相应任务。

9 局限性

表1和表2中的结果展示了 CALM 如何在不牺牲可控性的情况下学会丰富多样的动作库。随后，可以利用这一点学习风格条件运动，并最终组合动作，以完成多步骤的未见任务。本节将重点介绍这项工作引发的开放挑战与问题。

预训练——模式坍塌：我们已经展示了，与现有方法 ASE相比，我们的CALM算法提高了生成的动作的可控性，使性能显著提升，从35%提高到78%。然而，模式坍塌仍然是一个尚未解决的挑战。例如，我们观察到，以待机动作作为低层控制器的条件，可能会导致不真实的微小动作，使角色缓慢移动。尽管我们的方法在很大程度上解决了模式坍塌问题，但仍有进一步改进的空间。

预训练——未见过的动作：我们的工作着重于学习一个用于分布内动作的潜变量生成式动作控制器。然而，当使用来自未见过动作的编码来对角色进行条件控制时，我们无法保证生成的动作质量。虽然我们观察到，一些未见过的动作会映射到数据中语义相似的动作，例如，踮脚会映射为弹跳步行，但我们预计，随着动作越来越偏离分布，模型可能会失效。

精细化训练——超越运动：表2显示，我们的方法能够利用学习到的潜在空间，实现风格条件化的运动。然而，要控制复杂动作（例如攻击时剑或盾牌的运动轨迹），可能需要在预训练阶段进行额外创新，例如学习与数据具有更大分布差异的动作。

有限状态机——鲁棒性：我们的方法已证明能够利用游戏和动画行业中的经典工具（如有限状态机和行为树）来解决任务，并提供类似游戏控制器的界面，如图 5 所示。然而，我们预计，策略的鲁棒性范围可能会限制其解决与训练期间所见动力学特性截然不同的任务的能力，例如爬楼梯或在不平坦的地形上行走。因此，要解决此类任务，可能还需要进一步的创新和训练。

渲染——伪影：为了展示我们的工作如何集成到游戏行业中，我们使用在 Omniverse (OV) 中渲染的高分辨率角色来呈现动作。尽管在 IsaacGym (IG) 中记录的动作在物理上准确，并且满足物理约束且没有视觉伪影，但 OV 中用于可视化的角色在渲染时不受这些约束。因此，由于角色几何形状不同，可视化角色可能会出现穿透及其他在 IG 中不会出现的视觉伪影。值得注意的是，渲染伪影并非我们提出的 CALM 算法本身固有的问题，而是由角色几何形状的差异造成的。减少这些伪影的一种方法是确保模拟角色的几何形状更接近可视化角色。另一种解决方案是增强训练过程，使其能够适应不同的角色形态，从而直接控制可视化角色，同时施加物理约束。

10 讨论与未来工作

在这项工作中，我们提出了 CALM，这是一种用于学习可复用且可控的运动技能的框架，适用于基于物理的角色动画。我们的模型能够利用动作片段来引导角色的行为。给定一个无标签的动作数据集，CALM 可以同时学习编码器和低层控制器。编码器将动作映射到具有语义意义的低维表征空间中，而低层控制器则充当解码器的角色，生成具有与学习到的表征中编码的动作相似特征的动作。这些参考动作既可用于控制低层技能，也可用于引导更高层级的控制器，并指定角色在解决复杂任务时应采用哪些动作。对角色生成动作的控制能力，使其能够以零样本方式解决复杂的多步任务，这是实现交互式虚拟角色真正集成应用的重要一步。

我们通过运动约束训练，使解决方案能够朝着利用预先指定动作的方向发展。然而，要成功学会沿指定方向生成所要求的动作，需要仔细调整奖励参数。我们希望探索将方向表征与动作内容表征解耦的方法。这种解耦将使高层策略能够通过简化的奖励进行学习，同时确保生成的动作具有所需特征。最后，有些动作需要与环境进行协调交互。例如，后手翻和越障等特技动作，只有与跳马台或高台箱互动时才能完成。我们计划研究自动化方法，以理解动作与物体之间的配对关系，并在训练过程中引入此类物体，从而学习相应的动作。

参考文献

Christopher M Bishop 等人。1995年。用于模式识别的神经网络。牛津大学出版社。Piotr Bojanowski 和 Armand Joulin。2017年。通过预测噪声进行无监督学习。载于国际机器学习会议。PMLR，517–526。Ting Chen、Simon Kornblith、Mohammad Norouzi 和 Geoffrey Hinton。2020年。用于视觉表征对比学习的简单框架。载于国际机器学习会议。PMLR，1597–1607。Ian Goodfellow、Jean Pouget-Abadie、Mehdi Mirza、Bing Xu、David Warde-Farley、Sherjil Ozair、Aaron Courville 和 Yoshua Bengio。2020年。生成对抗网络。*ACM* 通讯 63，11（2020），139–144。F Sebastian Grassia。1998年。使用指数映射对旋转进行实用参数化。图形工具期刊 3，3（1998），29–48。Jonathan Ho 和 Stefano Ermon。2016年。生成对抗模仿学习。神经信息处理系统进展 29（2016）。Jordan Juravsky、Yunrong Guo、Sanja Fidler 和 Xue Bin Peng。2022年。PADL：基于语言的物理角色控制。载于*SIGGRAPH Asia 2022* 会议论文集 （韩国大邱）*(SA '22)*。计算机协会，纽约，纽约州，美国，第 19 篇文章，共 9 页。https://doi.org/10.1145/3550469.3555391 Diederik P Kingma 和 Max Welling。2013年。自编码变分贝叶斯。*arXiv* 预印本 *arXiv:1312.6114* （2013）。Hung Yu Ling、Fabio Zinno、George Cheng 和 Michiel Van De Panne。2020年。使用动作变分自编码器的角色控制器。*ACM* 图形学汇刊（*TOG*）39，4（2020），40–1。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。载于神经信息处理系统第三十五届会议数据集与基准赛道（第二轮）。https://openreview.net/forum? id=fgFBtYgJQX_ Aaron van den Oord、Yazhe Li 和 Oriol Vinyals。2018年。使用对比预测编码进行表示学习。*arXiv* 预印本 *arXiv:1807.03748* （2018）。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（*TOG*）38，6（2019），1–11。Omkar M Parkhi、Andrea Vedaldi 和 Andrew Zisserman。2015年。深度人脸识别。（2015）。Xue Bin Peng、Pieter Abbeel、Sergey Levine 和 Michiel Van de Panne。2018年。DeepMimic：以示例引导的基于物理的角色技能深度强化学习。*ACM* 图形学汇刊（*TOG*）37，4（2018），1–14。Xue Bin Peng、Yunrong Guo、Lina Halper、Sergey Levine 和 Sanja Fidler。2022年。ASE：面向物理仿真角色的大规模可复用对抗技能嵌入。*ACM* 图形学汇刊 41，4，第 94 篇文章（2022年7月）。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运动先验。*ACM* 图形学汇刊（*TOG*）40，4（2021），1–20。Reallusion。2022年。轻松制作三维动画和二维卡通。（2022）。http://www. reallusion.com Tim Salimans、Ian Goodfellow、Wojciech Zaremba、Vicki Cheung、Alec Radford 和 Xi Chen。2016年。改进生成对抗网络的训练技术。神经信息处理系统进展 29（2016）。John Schulman、Filip Wolski、Prafulla Dhariwal、Alec Radford 和 Oleg Klimov。2017年。近端策略优化算法。*arXiv* 预印本 *arXiv:1707.06347* （2017）。Richard S Sutton、Doina Precup 和 Satinder Singh。1999年。马尔可夫决策过程与半马尔可夫决策过程之间：强化学习中时间抽象的框架。人工智能 112，1–2（1999），181–211。Chen Tessler、Shahar Givony、Tom Zahavy、Daniel Mankowitz 和 Shie Mannor。2017年。在 Minecraft 中进行终身学习的深度分层方法。载于*AAAI* 人工智能会议论文集，第 31 卷。Faraz Torabi、Garrett Warnell 和 Peter Stone。2018年。生成对抗观测模仿。*CoRR* abs/1807.06158（2018）。http://arxiv.org/abs/1807.06158 Laurens Van der Maaten 和 Geoffrey Hinton。2008。使用 t-SNE 可视化数据。机器学习研究期刊 9，11（2008）。Feng Wang、Xiang Xiang、Jian Cheng 和 Alan Loddon Yuille。2017年。NormFace：用于人脸验证的 L2 超球面嵌入。载于第 *25* 届 *ACM* 多媒体国际会议论文集。1041–1049。Tongzhou Wang 和 Phillip Isola。2020年。通过超球面上的对齐与均匀性理解对比表示学习。载于国际机器学习会议。PMLR，9929–9939。Jungdam Won、Deepak Gopinath 和 Jessica Hodgins。2022年。使用条件变分自编码器的基于物理的角色控制器。*ACM* 图形学汇刊（*TOG*）41，4（2022），1–12。Jiacheng Xu 和 Greg Durrett。2018年。用于稳定变分自编码器的球形潜在空间。载于*2018* 年自然语言处理实证方法会议论文集。4503–4513。

状态与动作空间
在这项工作中，我们研究一个在物理模拟环境中运行的三维仿真人形角色，该角色手持剑和盾牌，拥有 37 个自由度。Peng 等人曾使用过类似的角色 [2022]。为了对状态进行编码，我们沿用 Peng 等人的相同表征技术 [2021]。智能体观察到：

- 根部（角色的骨盆）高度。
- 根部相对于角色局部坐标系的旋转。
- 各关节的局部旋转。- 各关节的局部速度。
- 手、脚、盾牌以及剑尖的位置，全部以角色的局部坐标系表示。

角色的局部坐标系定义为：原点位于根节点，x 轴沿根节点链接的朝向，y 轴与全局向上向量对齐。每个关节的 3D 旋转使用两个 3D 向量进行编码，分别对应于在链接父节点坐标系中表示的链接局部坐标系的切向量 **u** 和法向量 **v** [Peng 等人。2021]总计得到一个 120D 的状态空间。

A.1 低层策略
此外，低层策略还观测动作的 64D 潜在表示。

为了控制角色，智能体提供一个动作 a，该动作表示用于 PD 控制器的目标旋转，这些控制器分别位于角色的各个关节处。与 Juravsky 等人 [2022]；Peng 等人 [2022, 2021], 类似，3D 球形关节的目标旋转使用 3D 指数映射进行编码 [，从而得到一个 31D 的动作空间。

A.2 编码器
编码器学习将定长动作映射到低维表征。我们考虑时长为 2 秒的动作。由于低层控制器以 30Hz 运行，因此共有 60 帧。训练期间，我们随机采样动作片段。如果动作片段超过 2 秒，我们会随机采样一个连续的 2 秒窗口，并在动作的离散支撑点之间进行插值。另一方面，如果动作短于 2 秒，我们会进行零填充。

A.3 判别器
判别器的训练方式与 ASE [Peng 等人 2022]类似。给定一个 2 秒的动作片段 M，判别器以对应的潜变量编码 $z = E(M)$ 为条件，任务是区分从参考动作 M 中随机采样的序列 ($\hat{}s_1$，……，$\hat{}s_{10}$)，以及由策略生成、且以相同潜变量 z 为条件的转移序列 (s_1，……，s_{10})。

A.4 高层策略
高层策略会观察额外的任务信息，并生成提供给低层策略的潜变量 $z \in Z$。

A.4.1 格挡。在格挡任务中，一个投射物会朝智能体发射。智能体需要使用盾牌挡住投射物。在每个回合中，投射物分为两个阶段：预热阶段中保持静止，随后进入发射阶段，朝角色身体上的目标区域移动。

高层策略观察
- 此过程中的相对进度。
- 原点在角色局部坐标系中的位置。
- 投射物在角色局部坐标系中的局部位置。
- 与投射物之间的角度。- 投射物的速度和角速度。

这会产生一个额外的 18 维任务特定观测。

A.4.2 触及。 触及任务的目标是将剑尖置于指定位置并保持在该位置。在此，智能体观测目标位置在角色局部坐标系中的三维位置。

A.4.3 运动。 运动任务是一项以动作为条件的任务。给定一组动作 $\tilde{M}$，在时间 t，高层策略的任务是朝指定方向移动，同时采用与 $M_t \in \tilde{M}$具有相似特征的动作。为此，它会观测

- 目标方向在角色局部坐标系中的表示。
- 当前指定动作 M_t 的独热编码 $|\tilde{M}|$。

B 架构详情
在本研究中，所有网络均由全连接层构成。编码器和高层策略均采用标准的多层感知机（MLP）网络。在将潜在表示与观测拼接之前，低层策略和条件判别器还会对潜变量 z进行预处理，所用模块也是一个 MLP。

C CALM 与其他受物理约束的动作生成框架的比较

以往研究也探讨了物理角色动画这一挑战。本节将我们的 CALM 方法与以往研究进行比较，着重说明方法和结果上的差异。

C.1 动作匹配
早期生成复杂角色运动的研究主要聚焦于动作匹配的启发式方法。在 DeepMimic [中，Peng 等人 2018],针对角色当前姿势与数据中相应姿势之间的偏差设计了奖励。之后，借助动作变分自编码器，这一方法得到了扩展，并通过采用变分自编码器范式构建角色控制器，实现了一定程度的控制。值得关注的工作包括 Ling 等人 [2020] 和 Won 等人 [2022] 的研究，他们分别学习了无条件和条件动作变分自编码器，用于动作重建。

引用 Won 等人 [2022], ，我们指出动作匹配方案面临的主要挑战：

“我们的控制器也可能出现[Ling 等人 2020], 提到的下沉问题：如果输入轨迹中缺少不同行为之间的转换，控制器就无法在这些行为之间切换。使用由大量异质行为组成的数据集，而其中单条记录的时长相对较短，可能会导致这一问题。通过构建运动图来扩充数据集（我们在实验中采用了这种方法）或许能快速缓解这一问题；不过，准备包含丰富转换的数据集，才能生成看起来最自然的动作。”

由于低层控制器基于状态重建损失进行学习，因此它本身并不具备泛化到数据中未出现情形的能力。对抗技术的一项核心优势在于，它们会迫使行为符合参考数据分布，从而催生新的动作，例如在不同动作类别之间转换，如图 7 所示。

另一方面，直接模仿示范的方法有一项核心优势：逐轨迹损失能够充分反映智能体重建给定动作的能力。虽然在对抗方法中，这一点可能并不容易实现，但可以轻松利用它进行自适应采样 [Park 等人 2019], , 从而减少模式崩塌并提升整体质量。

C.2 对抗技术

与我们的研究方向更为接近的是对抗技术。在这类方法中，运动匹配奖励被判别信号取代。判别器会根据数据生成一个信号，用以表示生成的动作出现的可能性。

最初，AMP [Peng 等人 2021] 专注于使用无条件策略和无条件判别器来解决单项任务。缺乏条件约束使其无法引导行为。因此，优化对抗目标会导致生成整个参考数据分布。为了强制生成类人解，需要仔细调整参考数据分布，使其与预期的任务统计特征相匹配。

ASE [Peng 等人 2022] 后来将这一工作扩展为两个阶段的学习——先进行低层训练，再进行高层训练。首先，在多样化数据集上使用无条件判别器和信息最大化项训练低层条件策略。这使模型能够学习多样化的行为，并通过固定条件变量来控制这些行为 z。然而，动作与潜变量之间缺乏对应关系，导致模式崩塌，因此需要采用多样性损失等技巧来提升质量。因此，控制 ASE 生成的动作并非易事。

最后，与我们的工作最接近的是 PADL，Juravsky 等人 2022] 提出了一种同时结合条件策略和判别器的方案。CALM 与 PADL 之间的主要区别在于：（1）PADL 需要标注数据，而标注数据不仅获取成本高，而且表达能力不如示范。我们不假设存在任何额外的监督，只要求能够获取动作序列。（2）PADL 将表征学习与控制学习分开进行，而我们联合学习控制和表征。（3）PADL 直接学习任务驱动控制器，而我们学习一种通用低层条件技能生成器，该生成器无需进一步重新训练即可重复使用。

因此，CALM 无需监督即可学习到：(1) 一种结构良好的潜在表示，能够捕捉动作的语义含义。

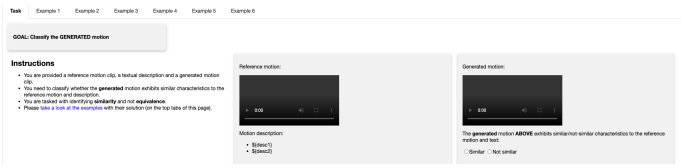


图 6. 动作分类系统

这一点在图 3e 的插值实验中有所体现：角色沿着一条具有语义含义的路径平滑过渡。(2) CALM 提供了一种更具可扩展性的方法，同一个低层控制器可以重复用于多个任务，如图 5 所示。

D 低层控制器的定性分析

在本节中，我们将介绍有关第 8.1 节定量结果的一些补充信息。

为了分析 CALM 学到了什么，我们进行了两项测试。第一项旨在评估编码器及其所学内容。我们报告了费 Fisher 距离，它用于衡量集中度。给定一个动作文件 M 以及从中截取的 2 秒滑动窗口片段 M_i

$$coeff(M) = \frac{\sum_{M_i,M_j \in M} \frac{||E(M_i)-E(M_j)||}{|M|^2}}{\sum_{M_i \in M, M_k \in \mathcal{M}} \frac{||E(M_i)-E(M_k)||}{|\mathcal{M}|\cdot|M|}} \tag{9}$$

得分定义为所有动作文件的集中度系数的平均值。

接下来，我们测试了低层控制器按需生成动作的能力。我们从数据集中随机选取了 40 个动作的子集。我们分别使用各自的编码器对 CALM 和 ASE 进行条件设定，使其生成具有相似特征的动作。初始状态从数据中随机采样。我们请人工评审员对每个生成的动作进行分类，每个参考片段生成 3 个动作。

我们报告所有生成的动作的平均准确率。此外，我们在图6中展示了问卷中的一张截图。

E 额外实验

作为对正文中实验的补充，我们展示了一些额外结果。补充视频最能展现这些动作。

E.1 消融分析

为了分析预训练阶段各项设计决策的重要性，我们进行了消融实验。我们比较了完整模型、从判别器训练中移除负样本的模型，以及移除潜在空间正则化项的模型。

如表3所示，加入每个设计元素都能进一步提升模型的生成质量。

E.2 技能转换

为了测试智能体能否以自然的方式按需在不同技能之间切换，我们让低层策略在一个长时段轨迹上运行。在该轨迹过程中，我们从中采样一个动作片段。

	集中度 ↓	生成 ↑	准确率 ↑
CALM	**0.23**	**19.8±0.11**	**78%**
不使用负样本	0.24	15.7±0.07	62%
不使用负样本，不使用正则化	0.35	12.8±0.05	61%

表3。**预训练消融：**我们对预训练阶段的各种设计选择进行了消融实验。

参考数据集，并以相应的潜变量作为智能体的条件。

如图 7 所示，智能体学会了在复杂动作之间进行转换。

E.3 下游任务

在论文中，我们着重介绍了我们的方法所具有的独特优势，即无需额外训练即可解决未见过的任务。尽管零样本应用备受关注，但有时仍需要自动化解决方案。

在图 8 中，我们展示了高层策略经过训练后，无需风格条件即可直接解决下游任务的场景。

E.3.1 格挡。 在格挡任务中，投射物会被投向智能体，智能体需要用盾牌将其挡住。为完成该任务，当投射物击中盾牌时，高层策略会获得正奖励

$$r_t^{\text{block}} = \mathbf{1}(\text{projectile blocked by shield}) . \tag{10}$$

E.3.2 触及。 由于“格挡”任务考验的是对持盾手臂的控制，我们还展示了“触及”任务，该任务要求精准的剑术动作。在此任务中，智能体需要将剑尖移动到指定 x^* 位置。为此，智能体会获得奖励

$$r_t^{\text{reach}} = \exp\left(-4||x_t^{\text{sword}} - x^*||_2^2\right) , \tag{11}$$

其中 $x_{\text{剑}t}$ 表示时间 t 时剑尖的位置。

E.3.3 位置。 在“格挡”和“触及”任务中，角色始终处于同一位置，但仍需要精准的低层动作。为了展示如何像 Peng 等人所示那样直接根据奖励组合动作， [2022], 我们还训练了“位置”和“打击”任务。

在定位任务中，角色需要到达目标位置。高层策略会收到奖励

$$r_t^{\text{location}} = \exp\left(-0.5||x^* - x_t^{\text{root}}||^2\right) , \tag{12}$$

其中 x^* 是目标位置， $x\text{root}_t$ 是角色根节点的位置。该奖励促使高层控制器生成能够尽快到达目标位置的动作。如图 8c 所示，控制器学会选择能够实现跑步的潜变量 z_t 。

E.3.4 打击。 “打击”任务考验高层控制器在复杂的多步骤任务中切换动作的能力。在此任务中，智能体需要击倒一个目标。为此，高层策略会获得奖励

$$r_t^{\text{strike}} = 1 - \mathbf{u}^{\text{up}} \cdot \mathbf{u}_t^* , \tag{13}$$

其中 $\mathbf{u}\text{up}$ 是全局向上向量，$\mathbf{u}_t^*$是目标物体在全局坐标系中表示的局部向上向量。此外，为了防止角色只是撞向目标，Peng 等人[2022] 提供了一个终止条件：如果角色用剑以外的任何部位触碰目标，环境就会终止。

奖励促使角色迅速触及目标，而终止条件则确保它只有使用剑才能做到这一点。如图 8d 所示，角色朝目标跑去，然后施展剑击命中目标。

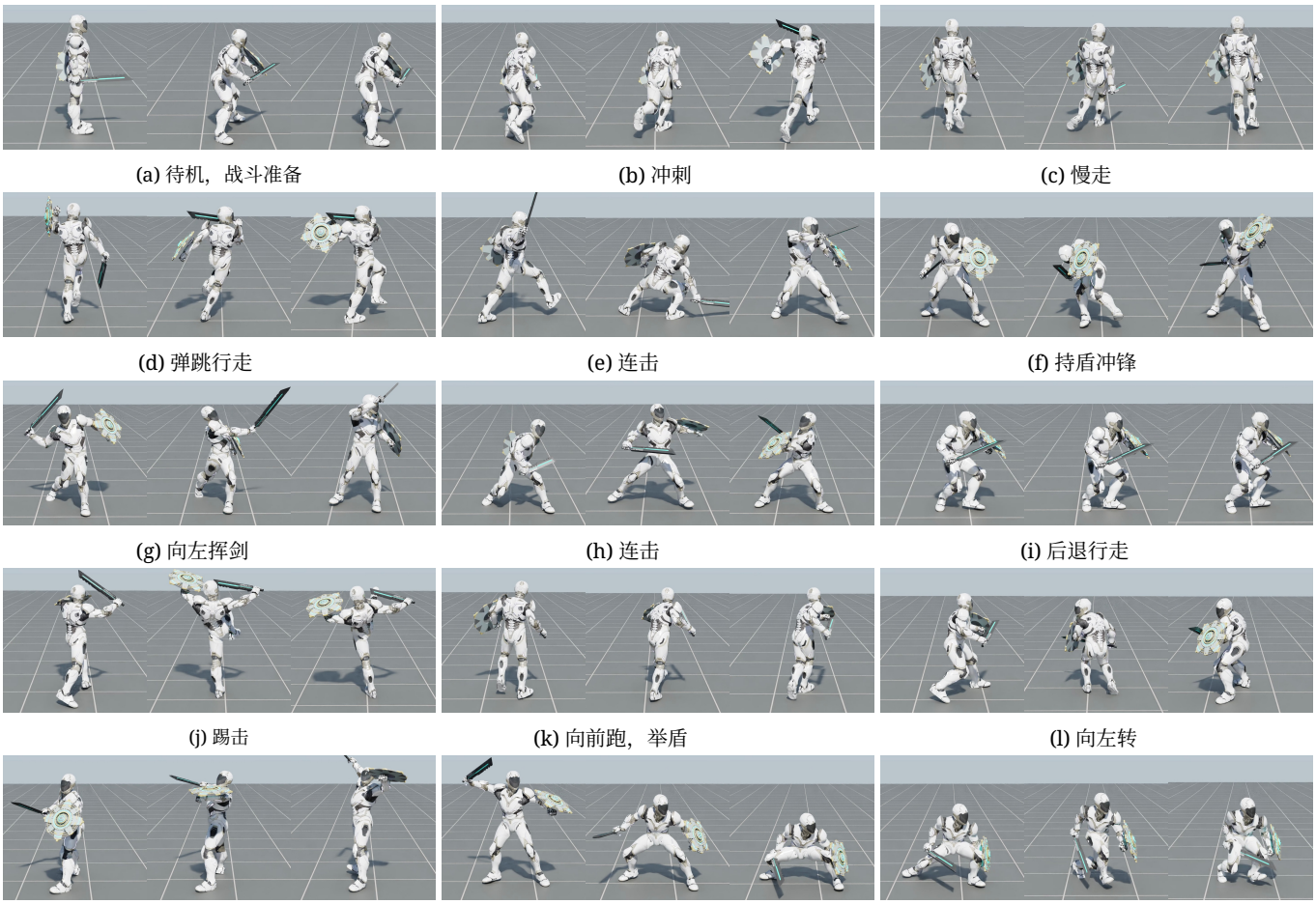
E.3.5 任务结论。尽管低层策略经过训练，能够生成长时程行为，但通过迭代潜在控制，高层策略仍能生成新动作来完成任务。例如，为了格挡投射物，它学会控制盾牌的瞄准方向；在触及任务中，它则学会控制剑的位置，并在目标位置附近保持相对静止。这些特定行为展现了对抗训练方案的优势：能够泛化到参考数据集中未曾观察到的内容。最后，在面对复杂的长时程任务时，角色学会运用类人动作，同时在不同动作之间自然过渡。

E.3.6 有限状态机与奖励设计。本文提出了一种无需进行特定任务训练即可解决位置和打击等任务的方法。该方法利用了有限状态机设计，以及低层策略能够按需生成指定动作这一特点。与奖励设计相比，其优势显而易见。要让角色学会蹲伏行走接近目标，然后踢击目标，最后再做出庆祝性的咆哮动作，需要精心设计奖励和终止机制。另一方面，采用有限状态机方法，可以指引角色朝指定方向执行特定动作，从而无需奖励设计或任何特定任务训练，就能以多样方式完成任务。

F 潜在空间分析

为了进一步了解所学习的表征，我们分析了潜在空间的结构。我们大致将动作分为 5 类：

- 行走动作
- 剑攻击
- 盾牌攻击
- 转向动作
- 和待机动作



(a) 待机，战斗准备　(b) 冲刺　(c) 慢走

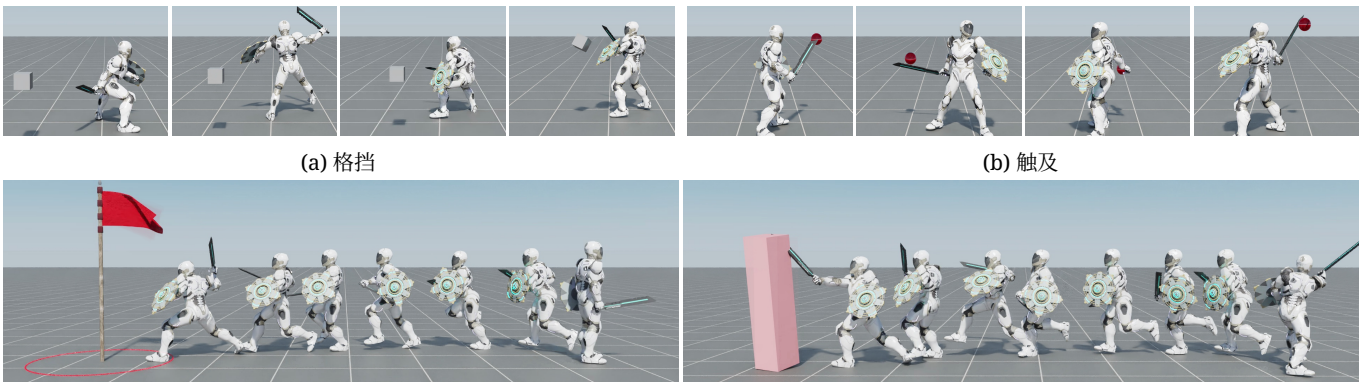
(d) 弹跳行走　(e) 连击　(f) 持盾冲锋

(g) 向左挥剑　(h) 连击　(i) 后退行走

(j) 踢击　(k) 向前跑，举盾　(l) 向左转

(m) 咆哮，举起双手　(n) 蹲下，待机　(o) 蹲下，行走

图 7. 由低层控制器根据示范动作的编码生成的技能。每隔 3 秒采样一个新动作，并将相应的潜在表示输入低层策略。这些图像取自一个持续时间较长的单次回合，在此期间，各个动作依次执行。



(a) 格挡　(b) 触及

(c) 位置　(d) 打击

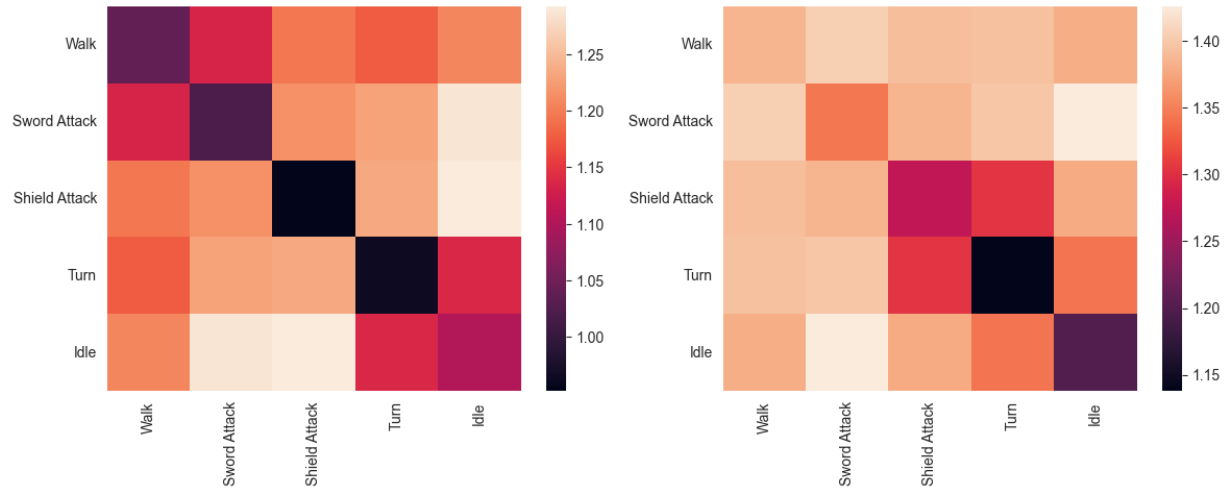
图 8. 标准的分层　任务求解。高层策略直接控制低层策略，并优化特定于任务的目标　e.

对于参考数据集中的每个动作，我们分别使用各方法对应的编码器对其进行编码。随后，我们计算不同组之间的成对距离平均值。如图 9 所示，CALM 能够在潜在空间中使各动作组聚集得更加紧密。这一点可以通过对角线方向上较低的距离观察到。这意味着 CALM 学习到了具有更强语义含义的表征，即相似的动作会在潜在空间中聚集得更加紧密。

此外，对 ASE 编码内部距离的分析表明，它并未在动作类别之间保持语义关系。具体而言，与行走对应的动作分散在潜在空间中。

此外，我们还展示了 CALM 学得的表征的 TSNE 图 [Van der Maaten 和 Hinton 2008] 。虽然绘制了所有数据，但由于动作数量庞大（约 180 个），为清晰起见，我们只标注了其中一部分。

虽然 TSNE 无法保留全局结构，如图 10 所示，但它确实表明，来自同一动作片段的子序列获得了相似的表征，也就是说，它们位于潜在空间中的相近位置。



(a) CALM (b) ASE

图 9. 潜在空间中各动作类别之间的距离。颜色越深，表示两个动作类别之间的成对潜在空间邻近度越高。

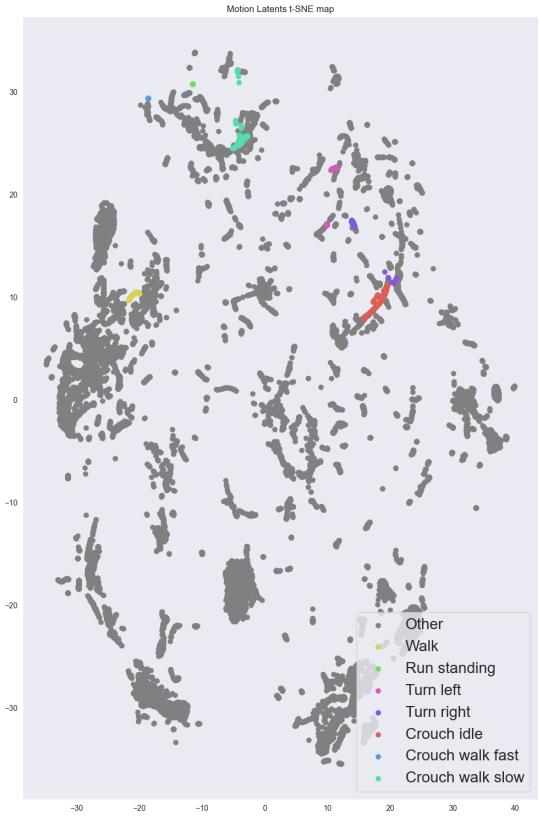


图 10. CALM TSNE分析