SkillDiffuser：通过基于扩散的任务执行中的技能抽象实现可解释的分层规划

梁志轩[1] (,) Yao Mu[1] (,) 马恒波[2] 富塚雅义[2]
丁明宇[2]† 罗平[1] (,) (†)

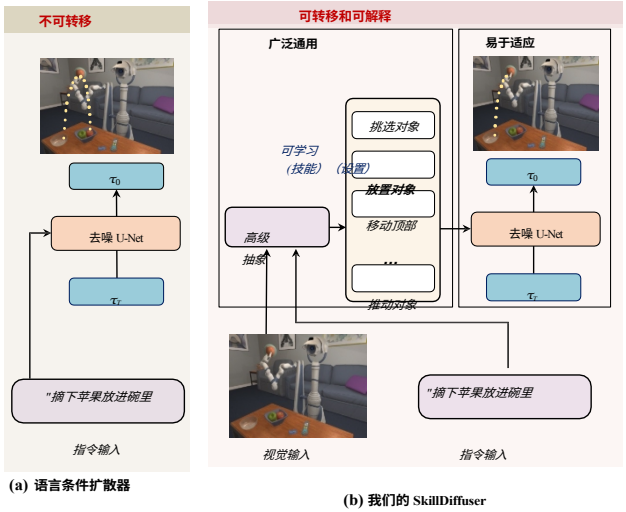
[1]香港大学 [2]加州大学伯克利分校 [3]上海人工智能实验室

{zxliang, ymu, pluo}@cs.hku.hk {hengbo ma, tomizuka, myding}@berkeley.edu
https://skilldiffuser.github.io/

摘要

扩散模型在机器人轨迹规划方面展现出强大的潜力。然而，从高级指令生成连贯的轨迹仍然是个难题，特别是对于需要多种连续技能的长距离组合任务。我们提出的 SkillDiffuser 是一个端到端的分层规划框架，它将可解释的技能学习与条件扩散规划整合在一起，以解决这一问题。在较高层次，技能抽象模块从视觉观察和语言指令中学习离散的、人类可理解的技能表征。然后，这些学习到的技能嵌入会被用来作为扩散模型的条件，以生成与技能相匹配的定制化拉登轨迹。这样就能生成符合可学习技能的各种状态轨迹。

技能。通过将技能学习与条件轨迹生成相结合，SkillDiffuser 可以在不同任务中按照抽象指令产生连贯的行为。在 Meta-World 和 LOReL 等多任务机器人操纵基准上进行的实验表明，SkillDiffuser 具有最先进的性能和人类可理解的技能表示。更多可视化结果和信息，请访问我们的网站。



(a) 语言条件扩散器 (b) 我们的 SkillDiffuser

图 1.**SkillDiffuser 与以前的语言条件扩散器的比较。**SkillDiffuser 利用高层次的抽象概念将视觉观察和语言指令转化为人类可理解的、以语言为基础的技能。然后，它将低级扩散模型条件化为这些技能，不仅提高了多步骤合成任务的执行性能，还大大增强了框架的通用性和适应性。

1. 引言

最近的研究[6, 7, 18, 19]表明，与以前的模型相比，扩散模型具有更强的生成能力，有助于在不同维度上增强强化学习，包括生成行动轨迹[2, 19]、策略表示[5, 50]和数据合成[14, 23]。然而，它们为复杂任务生成连贯轨迹的能力在性能和可推广性方面仍面临挑战，因为这些任务通常需要重新设计和优化行动轨迹。多步骤合成任务要求完成由众多协调密集型顺序步骤组成的抽象指令。

之前的方法 [2, 14]，如 Decision Diffuser，旨在通过将复杂任务分解为更简单的子技能，并在预定义的技能库中进行组织，来应对这一挑战。这些方法依赖于用单次技能表征来调节扩散模型，从而为每个子技能生成任务记录。然而，这些技术在尝试从大量数据集进行端到端自主学习时遇到了困难，从而影响了它们的可扩展性和实现端到端学习的能力[1]。此外，如果不明确学习可重复使用的技能，模型就无法捕捉错综复杂的步骤间依赖关系和制约因素。

†通讯作者。

cies and constraints，从而产生支离破碎、不合逻辑的运行轨迹。将模棱两可的指令分解为可学习的子目标和技能，可使模型更好地遵循逻辑步骤顺序，尊重任务结构，并在不同任务之间传递通用程序（*如*可重复使用的技能抽象化和基于技能的适应性扩散）。这种以技能为中心的范式为不同融合模型铺平了道路，使其能够解释和执行复杂、冗长的指令，而这些指令需要许多顺序步骤。

受上述观察结果的启发，我们提出了 SkillD- iffuser--一种将高级技能学习与基于条件扩散的低级执行相统一的分层规划框架。如图 1 所示，与以前的语言条件扩散策略相比，SkillDiffuser 能够端到端地解释和执行复杂指令，并具有更高的可移植性。SkillDiffuser 通过学习针对任务指令量身定制的可重复使用的技能来诱导可相互假装的子潜在目标。该框架以这些学习到的技能为条件，建立一个差异模型，从而生成与总体目标相一致的定制化连贯轨迹。通过将分层技能分解与常规轨迹生成相结合，SkillDiffuser 无需依赖预先确定的技能库即可实现一致的、以技能为导向的行为。此外，SkillDiffuser 的设计完全基于视觉观察，无需机器人本体感觉（*即*完全观察状态）。这种以可学习技能为特色的端到端方法使 SkillDiffuser 能够高效执行各种任务的抽象指令。

SkillDiffuser 的工作原理如下。**1)** 高级技能学习使用向量量化技能预测器[45]，将任务提炼为离散和可解释的技能。我们采用的不是预测技能持续时间，而是固定的预测范围--以固定的时间间隔预测技能。这种基于水平线的离散化过程可将视觉和语言输入无缝整合为一个具有凝聚力的技能集，从而为低级扩散模型提供指导。**2)** 对于基于技能的轨迹生成，我们利用无分类器扩散模型作为策略，并直接嵌入技能作为指导。这种设置允许生成与技能规范一致的多模式状态轨迹，同时避免过度拟合到封闭数据集。**3)** 为了从预测的状态中进行动作推理，我们训练了一个反动力学网络，以解码生成的两个连续帧之间的运动。通过将状态预测与运动解码分离，SkillDiffuser 生成了一个完全自适应的框架，可通过可转移的状态空间计划来指导不同的体现。

我们在 LOReL [29] Sawyer 和 Meta-World [51] 数据集上评估了该模型在技能学习和多任务规划方面的性能，并考虑到真实世界场景中的机器人必须在有限的状态信息（主要是视觉观察）下运行，因此进行了重要的实验设置。此外，我们还介绍了未此外，我们还介绍了未见过的组合任务的成功率、可重用性和所学技能的可视化，以说明该模型确实有能力抽象出不仅有效而且人类可以理解的高级技能，从而使我们更接近于以直接方式获取技能的智能代理。

我们的贡献有三个方面：**1）**我们提出了一个端到端分层规划框架，通过技能学习实现子目标抽象；**2）**我们采用了一个以所学技能为条件的无分类器扩散模型，以生成以技能为导向的可转移状态轨迹；**3）**我们在复杂的基准测试中展示了最先进的性能，并提供了人类可理解的可视化技能表征。

2. 相关作品

2.1. 模仿学习和多任务学习

模仿学习（IL）已经从基础的行为克隆发展到复杂的多任务学习框架。传统方法依赖于专家示范的监督学习[33, 37, 38]，最近的发展则转向从专家数据中学习奖励[16]或 Q 函数[12]，从而提高了模仿复杂行为的能力。新的挑战在于多任务 IL [41]，即在不同的任务中训练模仿者，目的是将模仿者推广到新的场景中，任务规格从向量状态 [28] 到视觉和语言描述 [8, 11, 13, 48]。

多任务学习方法通常利用共享表征来同时学习一系列任务，从而提高学习过程的灵活性和效率。元世界基准[51]对多任务和元强化学习进行了评估，强调了对能够快速适应的算法的需求。在此基础上，"提示决策转换器"[49] 利用特定任务的提示展示了少数几个策略的通用化。扩散策略也在多任务设置中进行了探索[14]，显示了在不同任务中生成不同行为的能力。不过，与利用状态输入 [14, 49] 或访问机器人本体感觉 [30] 的方法不同，SkillDiffuser 仅使用原始视觉输入。

2.2. 技能发现与分层学习

技能学习是机器人获取新能力或完善现有能力的过程，由于其在使自主系统适应新任务、随时间推移提高性能以及与人类和复杂环境自然交互方面发挥着关键作用，因此越来越受到关注。传统上，这一领域受到手工制作的特征和专家示范的影响[32]。

随着深度学习的发展，Eysenbach *等人*[9] 和 A. Sharma *等人*[40] 研究了学习方法中的技能识别，实现了以学习到的潜在变量为条件的策略，并保持了一致性。

在学习到的潜变量基础上实现策略，并保持一致的技能代码。在通过语言指令学习技能领域，LISA[13]通过对每个轨迹的多种技能进行采样，以独特的方式整合了语言调节，脱颖而出。

我们的技能扩散器（SkillDiffuser）就是按照这种方法，在高层提取每条指令的子技能。但不同的是，SkillD-iffuser 在较低层次采用了可适应的扩散策略，以不同的子技能为条件生成不同的行动，这就形成了一个创造性的分层规划框架，同时也推进了强化学习分层技术的研究[22, 27, 52]。

2.3. 利用扩散模型进行规划

近年来，扩散模型[18]在图像合成[18]领域取得了重大突破，并在各种生成应用中显示出良好的效果。直接使用扩散模型进行规划的开创性工作是 Diffuser [19]，它为在行为合成中使用扩散模型奠定了基础。然后，这类规划的一个分支

方法在各种决策任务中取得了最先进的性能[3, 5, 7, 23, 31]。其中，Chi *等人*在[5]中所做的工作引入了学习行动分布得分函数梯度并对其进行相对优化的概念，证明了其在视觉运动策略学习中的巨大潜力。这些工作进一步扩展了这一方向，并加强了基于扩散的规划器的通用性和普适性。

我们的方法受到无分类器扩散引导[17]的启发，与分类器引导模型[6]相比，无分类器扩散引导具有显著优势。通过采用无分类器方法，我们可以规避与训练奖励模型和 Q *函数*相关的挑战，在许多复杂度非常高的实际规划场景中，训练奖励模型和 Q *函数*尤其麻烦。最近的研究也扩展了这一方向，使用条件扩散模型生成定制轨迹。Decision Diffuser [2] 就是一个例子，它可以为预定义的技能库生成轨迹。然而，与我们的方法不同的是，它无法以端到端的方式自主学习技能抽象，因此难以扩展到更多任务。这凸显了具有动态、可学习技能抽象的扩散模式的必要性，从而促进了复杂指令的执行。

3. 初步

3.1. 通过状态扩散进行规划

正如之前的文章[2, 19]所介绍的，扩散模型是解决强化学习中的规划问题的一个很有前途的工具，它被视为马尔可夫决策过程（Markov Deci- sion Process，MDP）[34]。在 MDP 框架 $\mathcal{M}=(S, A, T, R, \gamma)$中，规划策略的目标是确定识别最优行动序列 $\boldsymbol{a}^*_{0:T}$

S 是状态空间，A 是行动空间。

将状态轨迹视为序列数据 $\boldsymbol{\tau}$，其中

在序列建模中，扩散概率模型将规划视为一个迭代去噪过程。该模型通过逆转被模拟为高斯过程的正向扩散过程来逐步完善轨迹，在此过程中，噪声被逐步添加到数据中，记为 $p_\vartheta(\boldsymbol{\tau}^{i-1}|\boldsymbol{\tau}^{(i)})$。训练包括最小化数据负对数似然的 ELBO，类似于贝叶斯推理的变异推理，目标是最优化：

$$\vartheta^* = \arg\min_{\vartheta} -E_{(\boldsymbol{\tau})^0} \log p_\vartheta \boldsymbol{\tau}^0 \quad 、 \tag{1}$$

其中 $p\,\boldsymbol{\tau}^N$为标准正态分布，$\boldsymbol{\tau}^0$为无噪声序列数据。

在实际应用中，[18] 提出了一种简化的替代损失函数，重点预测反向扩散步骤的高斯平均值：

$$L_{\text{denoise}}(\vartheta) = \mathsf{E}_{i,\boldsymbol{\tau}^0\sim(q,\epsilon)\ (\sim N)}[\,||\,\epsilon - \epsilon_{(\vartheta)}(\boldsymbol{\tau}, i)\,||^2\,]. \tag{2}$$

3.2. 无分类器扩散引导

在基于无条件扩散的方法基础上，离线强化学习领域的一个关键目标是生成回报率最高的轨迹。随着条件扩散模型的蓬勃发展[6]，分类器引导的方法通过在扩散过程中注入特定的轨迹信息（编码为 $\boldsymbol{y}(\boldsymbol{\tau})$），如返回值 $J\ (\boldsymbol{\tau}^0)$ 或指定的约束条件，来实现这一目标：

$$q(\boldsymbol{\tau}^{(i)}|\boldsymbol{\tau}^{(i)-1}), \quad p_{(\vartheta)}(\boldsymbol{\tau}^{(i)-1}|\boldsymbol{\tau}^{(i)}, \boldsymbol{y}(\boldsymbol{\tau})). \tag{3}$$

根据 [10] 中的假设，我们可以得出

$$\boldsymbol{\tau}^{(i)-1} \sim N\ (\mu_\vartheta + \alpha\Sigma\nabla_\tau \log p\,\boldsymbol{y}_{(}\boldsymbol{\tau}^{(}), \Sigma), \tag{4}$$

其中，α 是调整引导强度的超参数，Σ 是指定的噪声协方差，μ_ϑ是无条件扩散中噪声的学习均值。

然而，分类器引导的扩散模型需要根据轨迹分类器 $\boldsymbol{y}(\boldsymbol{\tau})$ 精确估计引导梯度，这可能并不可行，需要在训练过程中引入单独的动态编程过程来估计 Q *函数*。

因此，无分类器引导提供了另一种选择，即在轨迹生成过程中添加引导信号，放大数据中隐含的高回报或最优特征，*即* $\boldsymbol{y}(\boldsymbol{\tau})$。从数学上讲，在反向去噪过程中需要添加的噪声是：

$$\hat{\epsilon} = \epsilon_{(}\theta_{)}(\boldsymbol{\tau}^{i}, \varnothing, i) + \omega^{(}\epsilon_{(}\theta_{)}(\boldsymbol{\tau}^{i}, \boldsymbol{y}, i) - \epsilon_{(\vartheta)}(\boldsymbol{\tau}^{(i)}, \varnothing, i)), \tag{5}$$

其中 ω 是引导尺度，$\varnothing$ 代表引导的不确定性。将 $\omega=$ 设置为 0，可消除无分类器

ω值越大，轨迹生成过程中条件信息的影响就越大。

另外，要最小化的损失函数可以重写为

$$L_{diff}(\vartheta) = \mathbb{E}_{i,\tau,\epsilon}\left[\epsilon - \epsilon_\vartheta(\tau^i, (1-\beta)\boldsymbol{y}(\tau^i) + \beta\varnothing, i)\right]^2, \quad (6)$$

其中，β 是一个超参数，用于控制放弃特定条件 $\boldsymbol{y}(\tau)$ 的概率，从而在保持上下文相关性的同时提高样本的多样性。

用上述 L_ϑ训练噪声预测模型ϵ_ϑ

$L(\vartheta)$进行训练后，从高斯噪声开始对轨迹进行采样，并逐步去噪。

sian 噪声采样，然后用修正的ϵ逐步去噪。

通过公式 5，采用重新参数化技术。

总之，通过无分类器引导，我们可以调节轨迹采样过程，使生成的轨迹更符合 $\boldsymbol{y}(\tau)$ 所代表的预期特征。这一过程反复应用条件噪声模型，以完善包含满足约束条件的未来状态的目标轨迹。

4. 方法论

4.1. 方法概述

基于上文讨论的动机，我们提出了 SkillDiffuser，这是一种先进的方法框架，用于在各种机器人之间进行稳健的多任务学习。这种动态方法利用了高层次技能学习和低层次条件扩散模型之间的合作。整体框架如图 2 所示。值得注意的是，我们利用以语言为基础的表述来抽象技能，从而通过我们的扩散策略使任务的执行对人类来说既可解释又可理解。

4.2. 高级可解释技能抽象

在我们的 SkillDiffuser 框架中，高级可解释技能抽象模块在理解和执行复杂任务方面发挥着至关重要的作用。然而，在我们考虑的多任务环境中，每个只有单一语言指令的任务可能会被分解成一系列子任务或技能，而这些子任务或技能在指令本身中并没有明确划分。此外，假设线性训练数据集表示为 D，它由针对各种任务的次优策略得出的轨迹组成。A

轨迹 $\tau = (l, \{\boldsymbol{i}_t, \boldsymbol{a}_{(t)}\}^T_{t=1})$包括语言描述 $R_{(L)}$，其中 L *是语言空间*，而 $R_{(t)}$是在时间步长 T 上的图像观察和行动序列（$\boldsymbol{i}_{(t)}, \boldsymbol{a}_t$），没有附加奖励标签。但是，轨迹并不能指明子任务之间的界限，这就要求所提出的方法能够分割和解释这些子任务。

这就要求所提出的方法能够以无监督的方式将任务分割成子目标并加以解释。

为了应对这一挑战，我们在基于地平线的技能预测器基础上构建了一个技能抽象组件，该组件采用了基于地平线的技能预测器和 R(L)。

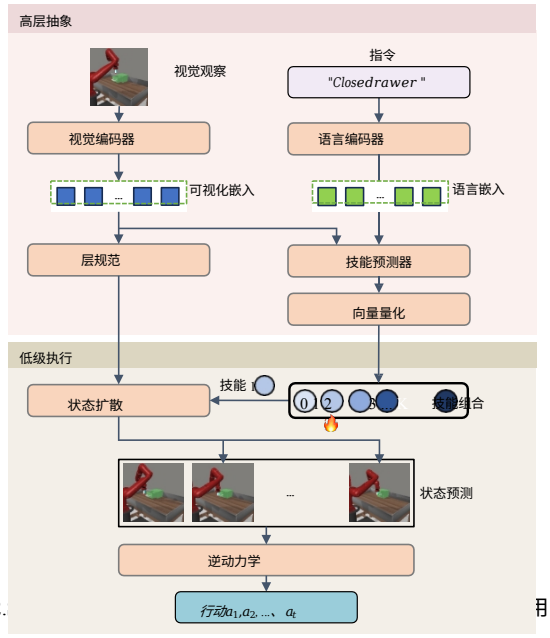


图 2. 用高层可解释技能抽象和低层技能条件扩散模型的合作，在多任务学习环境中执行任务。高层技能抽象通过技能预测器和矢量量化操作实现，生成子目标（技能集），扩散模型利用这些子目标确定适当的未来状态。未来状态通过反动力学模型转换为行动。这种独特的融合方式使不同任务的基础规划器保持一致，而差异仅存在于逆动力学模型中。

该计划器来自 GPT-2 [35]。它的设计目的是通过融合视觉输入和自然语言指令来解析和分解任务，并通过矢量量化（VQ）子模块将学习到的技能离散化为一个技能集。该组件的具体情况如下。

首先，由于我们仅使用图像作为机器人状态信息，因此我们使用固定的图像编码器（*如* R3M [30]）将图像转换为潜在空间特征。为方便起见，我们将图像编码器表示为 Φ_{im}: $I \rightarrow R^I$，其中 I 表示输入图像空间，$R^{(I)}$表示结果特征空间，其作用是将视觉信息浓缩为有利于高级语义的形式。同时，我们使用语言编码器对自然语言指令进行预处理，其形式化为 Φ_{lang}: $L \rightarrow R_{(L)}$，其中 L 是语言输入的空间。

编码器和 R^L语言特征空间。然后，两个编码器的输出都被输入到技能预测器中，技能预测器将这两种模式进行整合。

技能预测过程如下：时间步长为 t 的图像 $\boldsymbol{i}_t \in I$ 被编码为视觉嵌入 $\boldsymbol{s}_t = \Phi_{im}(\boldsymbol{i}_t)$ 。然后，通过语言编码器的输出 $\boldsymbol{l}_t = \Phi_{lang}(\boldsymbol{l})$，将该嵌入 $\boldsymbol{s}_t$与语言指令 $\boldsymbol{l} \in L$ 一起输入技能预测器。图 3

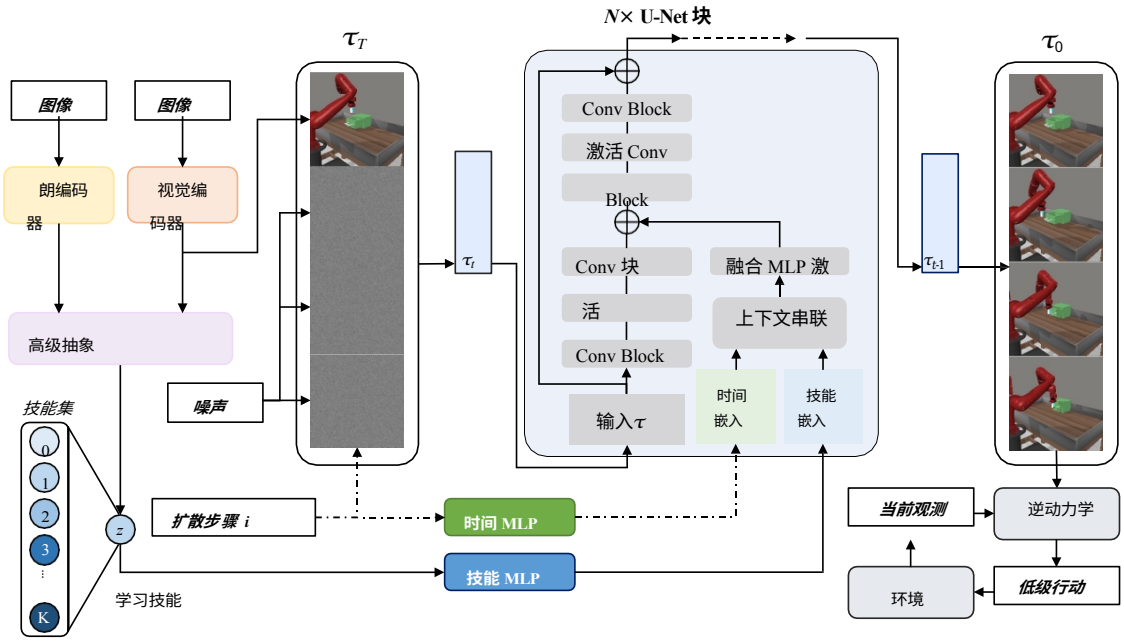


图 3.**SkillDiffuser 的低级技能条件扩散规划模型。**值得注意的是，这里的示意图使用图像来表示视觉特征，以达到说明的目的，而在实际应用中，扩散模型的输入和采样输出都是状态嵌入。当前观测也是当前视觉观测的特征嵌入。

技能预测器表示为 $f: R^I \times R^L \to C$，通过 $z = f(s_t, l_t)$ 生成技能代码 z，该代码封装了从视觉和语言输入中解读出的任务再要求。之后，矢量量化 [45] 操作 $\mathbf{q}(\cdot)$ 离散技能集包含 K 个技能嵌入 $z^1, z^2, \ldots, z^{(K)}$，分别代表不同的潜在技能。$z^K$ 代表不同的潜在技能。VQ 操作是通过将潜在的 $\tilde{z}$ 映射到其最接近的技能集条目上实现的，技能向量更新为最接近它们的嵌入 z 的移动平均值，这与 [45] 相同。这一过程如下：

$$\tilde{z} = f(\Phi_{im}(i_t), \Phi_l(lang_l(l))) \quad (7)$$

$$z = \mathbf{q}(\tilde{z}) = \arg\min_{z_k \in C} \|\tilde{z} - z^k\|_2. \quad (8)$$

VQ 强制每个学习到的技能 z 位于 C 中，这相当于利用 *k-means* [24] 算法学习 K 个语言嵌入原型。这就像一个瓶颈，限制了语言信息的流动，有助于离散技能代码的学习。通过无差别量化的反向传播是由直通梯度估计器实现的，它只需复制梯度，就能实现端到端的模型训练。

在我们的方法中，我们将一致的技能代码 z 的技能代码。这种在多个水平线上的一致应用，巧妙地解决了在不改变水平线本身的情况下改变子任务持续时间的难题。因此，这种策略不仅保持了执行不同任务所需的灵活性，还通过避免水平线引起的结构变化保持了模型的架构稳定性。

重要的是，所学技能代码的离散性增强了系统行为的可解释性和可控性，因为每个技能代码都与一些人类可理解的语言短语相关联。图 6 是一个例子。通过这种方法，SkillDiffuser 不仅能根据语言指令学习执行任务，还能以人类可理解的方式完成任务，从而能更深入地理解和控制具体化形代理的决策过程。

4.3. 低水平技能条件扩散规划

正如第 1 节所强调的，虽然现有的方法如

决策扩散器 [2] 引入了受技能约束的条件扩散模型，但其能力有限

目前，很多模型都只能生成仅满足预定技能要求的轨迹。因此，这些模型无法实现能够对任意学习技能进行调节的扩散模型。为了克服这些限制，并使扩散模型能够对学习到的连续技能谱进行规划，我们提出了一种方法，利用嵌入技能的无分类器引导扩散模型。

SkillDiffuser 首先采用一个扩散模型，对在高级技能抽象过程中学到的连续技能空间 Z 进行操作。我们采用一个 U-Net 作为噪声预测模型 $\epsilon_\theta(\cdot)$，并通过上下文条件对其进行引导。更具体地说，我们首先利用技能 MLP（类似于点向前馈网络[46]）（记为 Λ）将技能特征与去噪模型对齐。然后，我们将技能嵌入 $\Lambda(z)$ 融合到整个 U-Net 剩余块的状态特征中。详细信息

如图 3 所示。通过这种方式，我们可以使扩散模型在每一步都能根据上下文调节技能嵌入的影响。根据公式 5，我们可以根据这些给定的技能合成未来的状态。这是从以前的静态调节框架向更加动态和自适应的轨迹生成过程的重大转变。

此外，根据之前的研究[2, 7, 23]，我们采用了纯状态扩散模型，避免了在扩散模型中直接生成行动。取而代之的是，我们利用另一个 MLP（记为$\Psi(\cdot)$）在状态生成后进行反演，以推断出能实现两个连续状态之间转换的可行行动。通常，我们会整合当前帧的观测数据，为运动 预测提供更详细的信息，实现闭环控制。在数学上，它是

$$\boldsymbol{a}_t = \Psi([\tilde{\boldsymbol{s}}_t, \tilde{\boldsymbol{s}}_{t+(1)}], \boldsymbol{i}_{(t)}) \quad \text{for t} = 0, \ldots, T-1, \qquad (9)$$

其中，$\tilde{s}_t$和$\tilde{s}_{(t)\ (+1)}$是扩散模型生成的$\tilde{\tau}$中的连续观测嵌入，$\boldsymbol{i}_t$是当前观测，$\boldsymbol{a}_t$是推断的行动。

由此产生的行动序列$\{\boldsymbol{a}_0, \boldsymbol{a}_1, ., \boldsymbol{a}_{T-1}\}$ de-从生成的状态中提取的$\Psi(\cdot)$，包含了执行任务的技能，因此在多个任务中都具有出色的适应性。面对新任务时，我们只需根据新任务的运动学特性改变反动力学模型$\Psi(\cdot)$，而扩散模型的结构和参数则保持不变。这种模块化确保了 SkillDif- fuser 的生成能力不是针对特定任务的，而是可以在动态变化的各种任务中加以利用。底层模块示意图如图 3 所示。

4.4. 训练 SkillDiffuser

我们为 SkillDiffuser 设计了三重损失函数。首先，对于任务特定的反向动力学模型，我们采用行为克隆损失[44]来训练我们的反向 MLP，通过观察状态转换来模仿专家的行动。这种损失称为L_{inv}，其计算公式为

$$L_{inv} = \mathbb{E}_{\tau \sim D} \left[\boldsymbol{a} - \Psi(\tilde{\boldsymbol{s}}\tilde{\boldsymbol{s}}', \boldsymbol{i}) \right]^2, \qquad (10)$$

其中的符号与公式 9 类似。

相应地，包括技能抽取和低级执行在内的其他部分与任务无关。对于高级技能抽象模块，我们采用向量量化（VQ）损失来完善技能预测器。这种向量量化损失（L_{VQ}）可确保技能预测器生成的嵌入与技能集向量密切匹配，从而提高技能表示的可解释性和一致性。受 VQ-VAE [45] 的启发，我们将其表述为

$$L_{\text{VQ}} = \mathbb{E}_{\tau} \left[\| \mathbf{q}(\boldsymbol{z}) - \tilde{\boldsymbol{z}} \|_2^2 \right] \qquad (11)$$

根据公式 6，确保我们模型的状态预测与专家演示中的技能指导和时间动态一致。这里，我们将$\boldsymbol{y}(\tau) = \Lambda(\boldsymbol{z})$，$\boldsymbol{z}$ 由公式 7 和公式 8 得出。

值得注意的是，我们在训练 SkillDiffuser 时使用了两个优化器，一个是反动力学模型L_{inv}，另一个是高级技能抽象和低级规划的总体参数$L_{\text{VQ}} + \lambda L_{diff}$，其中$\lambda$是损失权重。这种精心构建的损耗架构使 SkillDiffuser 能够在多任务环境中发挥出色作用，只需根据每个新任务的具体要求替换反动力学模型$\Psi(\cdot)$，即可实现任务间的通用化。

此外，我们使用预先训练好的 distilBERT [39] 作为语言编码器，采用与 LOReL [29] 和 LISA [13] 一致的配置，同时冻结其参数，以保证语言理解的稳定性。我们采用不同的设置作为视觉编码器，以确保在不同数据集上进行公平比较。我们将在第 5 节的相应部分详细阐述这些细节。更多训练细节见附录 G，我们还在附录 B 中提供了一些算法的伪代码。

5. 实验

我们首先对 LOReL Sawyer 数据集进行了全面评估，然后在 Meta-World 基准上进行了消融研究，以说明我们方法的效率。最后，我们对我们的方法在 LOReL 和 Meta-World MT10 上学习到的技能进行了可视化展示。

5.1. 数据集

LOReL Sawyer 数据集[29] 是语言条件离线奖励学习（Language-conditioned Offline Reward learning）的缩写，由从任意强化学习策略中收集的*伪专家*轨迹或*游戏数据*组成，并标注了事后的众包语言指令。LOReL Sawyer 数据集包含 50k 条轨迹，每条轨迹都有自己的语言指令。
在模拟的 Sawyer 机器人环境中，以 20 个步骤进行计算。我们我们使用与原版相同的六项任务来评估我们的方法。
本文[29]在五种不同情况下（*即*看到、未看到动词、未看到无、未看到动词+ 名词和人类提供）对指令进行了转述。这样，所有 6 项任务加起来共有 77 条指令。有关该数据集的更多详情，请参见附录 D.1。

元世界数据集[51]。该数据集是为评估多任务和元强化学习算法而设计的综合基准。它引入了一套完整的 50 个不同的机器人操作任务，所有任务都位于
在一个统一的桌面环境中，由一个模拟的
模拟索耶臂控制的统一桌面环境中。元数据中心内的多任务 10（MT10）

其中 $\tilde{\mathbf{z}}$ 遵循公式 7。

最后，对于低层次的仅以技能为条件的扩散执行，我们将扩散损失 L_{diff} 计算为

世界是一个子集，由十个精心挑选的任务组成，在多样性和复杂性方面达到了平衡。这十项任务的详情可参见附录 D.3。

任务指令	随机	LCBC [42]	LCRL [20]	Lang DT [13]	LISA [13]	技能扩散器
关闭抽屉	52%	50%	58%	10%	**100%**	**95**± **3.2**
打开抽屉	14%	0%	8%	**60%**	20%	**55**± **13.3%**
左旋水龙头	24%	12%	13%	0%	0%	**55**± **9.3**
向右转动水龙头	15%	**31%**	0%	0%	**30%**	**25**± **4.4**
向右移动黑色马克杯	12%	**73%**	0%	20%	60%	18± 3.9
将白色马克杯下移	5%	6%	0%	0%	**30%**	10± 1.7
任务平均数	20%	29%	13%	15%	40%	**43**± **1.1%**

表 1.**LOReL Sawyer 数据集的任务成功率。**我们展示了与随机策略、语言条件模仿学习（LCBC）、语言条件 Q 学习（LCRL）、平面非层次决策转换器（Lang- DT）和 LISA 相比的成功率。每个数据集的结果都是通过 3 个种子计算得出的。就所有任务的平均 性能而言，SkillDiffuser 优于所有其他方法。最佳方法和与 最佳方法相差 6% 以内的方法用**粗体**标出。

重述类型	朗 DT	LISA [13]	SkillDiffuser
所见	15	40	**43.65**± 4.7
未见名词	13.33	33.33	**36.01**± 6.3
未见动词	28.33	30	**36.70**± 9.5
未见名词+动词	6.7	20	**42.02**± 3.8
人类提供	26.98	27.35	**40.16**± 2.1
平均值	18.07	30.14	**39.71**

表 2.**LOReL Sawyer 的重述成功率（单位：%）。**这里显示的是 Lang DT、LISA 和我们的 SkillDiffuser 的结果。标准误差是根据 3 个随机种子计算得出的。

均值如下

5.2. LOReL Sawyer 数据集的评估结果

基线。我们采用随机策略、语言条件限制模仿学习（LCBC）[42]、语言条件限制 Q 学习（LCRL）[20]、Lang-DT（在[13]中也称为 Flat Baseline）和最先进的技能学习方法 LISA [13] 作为基线。我们对前三种算法的设置与 LOReL [29] 相同，对后两种算法的设置与 LISA [13] 相同。随机策略作为基线。LCBC 根据指令模仿离线数据集的行为，这与之前专注于模仿学习以实现语言条件行为的工作相一致。相比之下，LCRL 采用的是强化学习方法，将每集的最终状态标记为语言指令奖励。Lang-DT 采用因果转换器[25]作为非层次基准，而 LISA 则采用双转换器结构，一个用于技能预测，另一个用于行动规划。我们没有将其与 LOReL planner [29] 进行比较，因为 LOReL planner 使用的是基于人类注释的学习奖励函数 MPC，而 LISA 和我们的计划只使用轨迹数据进行学习。

结果为确保公平比较，我们的方法 SkillD- iffuser 在设计上保持了与基线模型相似的参数数量。它采用了与 LISA [13] 相同的视觉和语言编码器架构，并对 SkillDiffuser 各层的嵌入维度和头部数量进行了严格匹配。

表 1 和表 2 显示了 LOReL 的任务成功率和重述成功率，平

方法	朗 DT	LOReL [29]	LISA [13]
SkillDiffuser 成功率	13.33± 1.3	18.18± 1.8	20.89± 0.6
25.21± 2.7			

表 3.**LOReL 多步骤合成任务的性能。**

以 20 步的时间跨度运行 10 次。我们的方法 SkillDiffuser 在六个不同任务中取得了最高的平均性能，这表明它具有卓越的跨任务适应能力，尤其是与基于决策转换器 [4] 的类似方法（如 LISA [13]）相比。此外，在 LOReL 测试任务中，SkillDiffuser 在所有重述类型中都表现出色，平均比 LISA 高出 9.6%。这证明了该模型在处理各种技能表征时的鲁棒性，标志着技能条件扩散模型的显著进步。

5.3. 在 LOReL 组合任务中的表现

设置。我们按照表 3 所示的 LISA [13]未见合成任务设置进行实验，其中包含 12 条合成指令。3.详细指令见附录 D.2，例如 "打开抽屉并向右移动黑色马克杯"。与 LISA 一样，我们将最多插曲步骤数从传统的 20 步扩展到 40 步。

结果我们观察到，SkillDiffuser 的性能是非分层基线（即不含技能分层）的 2 倍，而且比 LISA 提高了约 25%，这凸显了它的有效性。基于 MPC 的 LOReL 计划程序在开放场景（如合成任务）中表现不佳。

5.4. 元世界数据集的消融研究

设置。我们在带有精细注释指令的 Meta-World MT10 基准上进行了实验。我们还只使用了视觉观察。详情见附录 D.3。

基线。我们用三个基准来评估我们的方法，它们都是从现有模型修改而来。第一种是 Flat R3M，改编自 R3M [30] 论文中的规划器。由于最初的计划器使用了机器人本体感觉的前四项，我们取消了它们，使计划器只关注视觉观察。第二条基线是 SkillD- iffuser 的变体，缺少高级技能抽象模块，但保留了基于条件扩散的低级规划器、

方法	语言	技能集	扩散器	性能
平面 R3M [30]	✗	✗	✗	13.3%
LISA [13]	✓	✓	✗	13.8%
朗扩散器	✓	✗	✓	16.7%
技能扩散器	✓	✓	✓	**23.3%**

表 4.**元世界语言技能条件消融研究。**这里显示的是 Flat R3M、语言条件扩散器、LISA 和我们的 SkillDiffuser 的结果。所有结果均为 3 次运行的平均值。

以评估技能抽象化的影响。该版本集成了一个双层 MLP，可从视觉和语言输入中预测选项，起到了语言条件扩散规划器的作用。最后一个基线是我们针对 Meta-World 数据集重新实施的 LISA [13]，以验证扩散模型的效率。为确保公平性，我们使用 R3M 作为 Meta-World 上所有这些方法的视觉编码器。

消融对语言技能调节的影响。表 4 显示，我们的 "元世界 MT10 "任务与之前使用状态作为观察结果的任务[49]或考虑机器人本体感觉的任务[30]有很大不同，而且更具挑战性。我们只使用单帧视觉输入和指令。扁平 R3M 方法缺乏语言条件和技能组合，只能通过行为克隆成功完成*抽屉关闭*和*伸手*等任务。语言条件扩散器和改进型 LISA 均优于 Flat R3M，这表明了每个相应模块的价值。我们的 SkillDiffuser 将技能离散为一个技能集，其性能比语言条件扩散器高出 6%，比 LISA 高出 9.5%，证明了这种组合架构的有效性。

5.5. 关于所学技能可重复使用性的消融研究

为了评估我们所学技能的可重用性，我们计算了 LOReL Sawyer 数据集中每条指令所使用的不同技能的平均数量以及使用每种技能的指令总数，如表 5 所示。(最大插曲步长为 20，我们实验的技能范围为
10.)我们观察到，每条指令平均使用 1.55 种子技能，每种技能被调用的次数多于指令数（5 个评估集为 75 次），这验证了所学技能的可迁移性。正如文献[13]表 10 所指出的，除了很小的范围会影响性能外，学习子技能以获得精细语义有助于在不同阶段执行不同的操作。此外，我们还在附录 E.1 中可视化了应用离散技能产生的图像，以进一步验证技能的可解释性。

5.6. 学习技能集的可视化结果

我们在图 4 中展示了 LOReL 组合任务中技能集的可视化，在图 5 中展示了原始 LOReL 数据集的结果，在图 6 中展示了 Meta-World 数据集的结果（见附录 C）。通过对 SkillDiffuser 在 LOReL 组合任务中学会的技能进行直观分析，我们发现在 20 个

# 学习技能数量	# 实例数	# 成功次数	使用 1 种技能	使用 2 种技能	平均
17	375	144	64	80	1.55
17 种技能的频率	30, 8, 14, 10, 5, 19, 4, 7, 20, 2, 9, 20, 25, 6, 10, 3, 18				

大小的技能集中，我们的方法学会了 11 个技能（*例如，拉手*）。

表 5.**单个 结构所使用的不同技能的平均数量以及每个技能所使用的指令总数。**

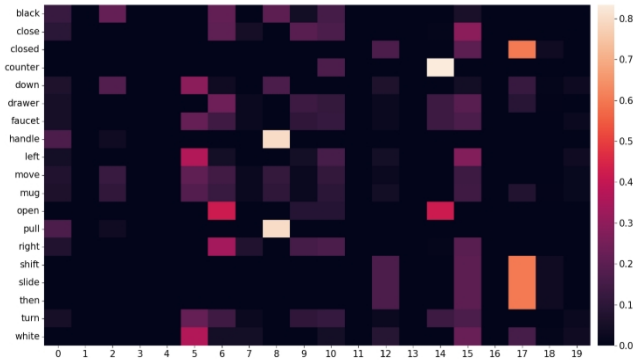


图 4.**LOReL Sawyer 构成任务的技能热图可视化。**我们显示了与 LOReL 中大小为 20 的技能集相关的词频，并按列进行了归一化处理。数据的稀疏性和明显的亮点表明，某些语言词与特定技能有着独特的联系。通过我们的方法可以学习到 11 种技能。(放大以获得最佳视图)

技能 0]、*打开计数器*[技能 14]*等*）的独特词汇亮点明显。这些亮点横跨热图中的 11 列（与最初只有一列对应默认 BC 相比有所变化），证明了该模型能够在没有明确定义的技能库的情况下，从视觉输入中识别并分离出独特的技能。这表明，与以往基于扩散的规划相比，该模型不仅在解释方面有了重大进步，而且还成功地抽象出了高级技能表征。

6. 结论

本文介绍的 SkillDiffuser 是一个集成框架，通过实现可解释的技能学习和条件扩散规划，使机器人能够根据自然语言指令执行任务。它采用向量量化技术，直接从视觉和语言演示中学习离散、可理解的技能代表。随后，这些技能作为扩散模型的条件，生成符合所学技能的状态轨迹。通过将分层技能分解与条件轨迹生成相结合，SkillDiffuser 可以编译和执行各种操作任务的抽象指令。大量的操作实验表明，SkillDiffuser 的性能达到了最先进的水平，突出显示了它对多步骤组合任务的有效性以及自动学习可解释技能的能力。

致谢

本文得到国家重点研发计划（2022ZD0161000）和香港普通基金（17200622）的部分资助。