arXiv:2301.02555v1 [cs.RO] 6 Jan 2023

“不，向右”——通过共享自主实现机器人操作的在线语言纠正

Yuchen Cui*
yuchenc
@cs.stanford.edu 斯坦福大学 美国加利福尼亚州斯坦福

Siddharth Karamcheti*
skara
mcheti@cs.stanford.edu 斯坦福大学 美国加利福尼亚州斯坦福

Raj Palleti 斯坦福大学 美国加利福尼亚州斯坦福

Nidhya Shivakumar 哈克学校 美国加利福尼亚州圣何塞

Percy Liang
斯坦福大学 美国加利福尼亚州斯坦福

Dorsa Sadigh
斯坦福大学 美国加利福尼亚州斯坦福

摘要

语言引导的人机交互系统若要得到广泛应用，必须满足两个关键要求：自适应性 和学习效率。遗憾的是，现有的指令跟随智能体无法适应，缺乏整合在线自然语言监督的能力；即便具备这种能力，也需要数百次演示才能学会简单的策略。在这项工作中，我们提出语言信息增强的带纠正潜在动作（LILAC）框架，以整合自然语言纠正并适应这些纠正——例如“向右”或“不，朝书的方向”——在执行过程中实时完成。我们在共享自主范式下探索丰富的操作领域。LILAC 并未采用人类与机器人轮流操作的离散方式，而是在人类与机器人之间分配能动性：语言作为输入传入一个经训练的模型，该模型生成有意义的低维控制空间，供人类用来引导机器人。每次实时纠正都会细化人类的控制空间，使其能够实现精准、持续的行为——此外，只需少量演示即可完成学习。我们通过一项用户研究评估了该方法：用户操控 Franka Emika Panda 机械臂完成复杂操纵任务。与涵盖开环指令跟随和单轮共享自主的现有学习基线相比，我们证明了这种能够处理纠正的方法取得了更高的任务完成率；用户也更青睐该方法，认为它可靠、精准且易用。[1]

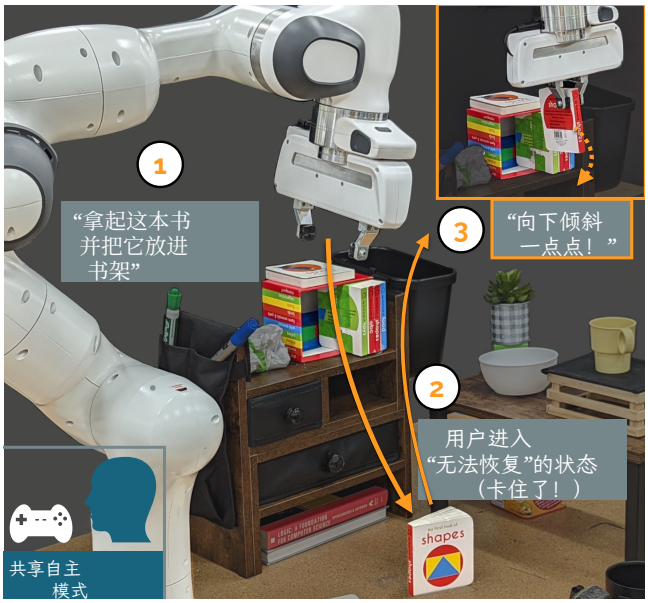


图1：LILAC：以往工作只能针对整个任务发出一条单一语言话语（“拿起书并把它放进书架”——实线），而我们的方法允许用户在执行过程中的任何时刻提供语言修正，使机器人能够在线调整（“稍微向下倾斜一点！”——右侧窗口）。

CCS概念

• **计算方法** → **合作与协调**；自然语言处理；从示范中学习。

关键词

在线纠正、语言与共享自主、机器人学习

ACM 引用格式：

Yuchen Cui、Siddharth Karamcheti、Raj Palleti、Nidhya Shivakumar、Percy Liang 和 Dorsa Sadigh。2023年。“不，向右”——通过共享自主实现机器人操作的在线语言纠正。载于*2023年ACM/IEEE人机交互国际会议论文集（HRI '23），2023年3月13日至16日，瑞典斯德哥尔摩。*ACM，纽约州纽约市，9 页。https://doi.org/10.1145/3568162.3578623

*两位作者对本研究的贡献相同。[1]项目网站（含视频和研究界面）：https://sites.google.com/view/hri-lilac。数据收集、模型定义、训练和评估的代码：https://github.com/Stanford-ILIAD/lilac。

允许免费为个人或课堂用途制作本作品部分或全部内容的数字副本或纸质副本，但不得为营利或商业利益制作或分发副本，且副本必须附有此声明及首页上的完整引文。本作品中第三方组件的版权必须得到尊重。其他用途请联系所有者/作者。*HRI '23，2023年3月13日至16日，瑞典斯德哥尔摩*©2023年版权归所有者/作者所有。ACM ISBN 978-1-4503-9964-7/23/03。
https://doi.org/10.1145/3568162.3578623

1 引言

机器人学中的自然语言研究主要关注人类与机器人之间的双边轮流式交互，通常采用指令跟随模式 [2, 3, 46, 49]。在这一范式中，人类



图2：用户与我们的系统交互。 [左侧]用户说出“拿起书并将其放入书架”，从而触发一个低维控制器（以操纵杆和阴影输入表示）。 [中间] 该控制空间由状态和语言共同决定：向下按可使末端执行器靠近书；抓取书后，按住向上/向左可使末端执行器朝架子移动。然而，这个静态 控制器还不够；用户卡住了！ [右侧] 我们的方法允许用户提供实时纠正（“稍微向下倾斜”），以优化控制空间，使用户能够完成任务。

发出指令，然后机器人自主执行——同时实现人类的目标，并规划在环境中执行的一系列动作，无需用户提供任何额外输入。人类与机器人之间这种明确的能动性划分给学习带来了巨大负担；现有系统要么需要大量语言对齐的示范数据来学习策略 [10, 34, 44, 45], , 要么对已知的环境动力学作出其他限制性假设；此外，还要求能够实现完美的物体定位和可供性预测，以便接入任务与运动规划器 [27, 37]。

与这些方法样本效率低下相伴的，是它们缺乏自适应性。考虑图1中的机器人，它正试图完成“拿起书并将其放入书架”这一任务。这是一项长时程任务，包含多个需要精确操作的关键状态——从抓住书脊，到将书抬离桌面且不碰到边桌，再到将书精确对准书架以便放入（两侧的间隙都不到几毫米）。在这种情况下，即便是现有的最佳方法，也无法稳定地完成任务。然而，这些“任务失败”往往是可预测且能够补救的。用户看到机器人朝着玻璃碗猛冲，就知道灾难即将发生，也知道如何避免；同样，诸如抓取落空或末端执行器未对准等细微错误也都可以修正——前提是用户获得恰当的机制来调整机器人的行为。

实现这种调整的一种方式是通过自然语言纠正——从简单的“左边！”或“稍微向下倾斜一点”（如图1所示），到更复杂的“不是，朝着蓝色标记去”。尽管近期研究尝试通过从对话中学习 [47, 48], 事后纠正 [7, 8, 12, 43], 或隐式反馈 [22], 来实现这一理念，但这些方法都无法实时发挥作用。相反，我们认为，可扩展的语言驱动人机交互系统必须能够以兼具自适应性和样本效率的方式处理在线纠正。

我们提出一种新颖的方法——**LILAC：语言信息引导的带纠正潜在动作**——它提供了一种可泛化的适应在线自然语言纠正的框架，并将其融入共享自主 [1, 14, 20, 30] 人机协作范式中。借助 LILAC，用户可以输入一连串语言话语，首先给出一个高层目标（“拿起这本书并把它放进书架”），每条话语都会影响用户能够如何控制机器人。在执行过程中的任何时刻，用户都可以输入新的话语——例如“稍微向下倾斜！”这样的纠正——从而在线更新控制空间，实时反映用户的意图。像这样的共享自主设置赋予了我们前文提到的自适应性，同时也使我们能够开发出具有极高样本效率的纠正感知系统。借助 LILAC，我们能够学会执行复杂操作任务，例如图1中所示的任务，只需 10 至 20 次示范，而完全自主的模仿或强化学习方法则需要数千至数万次示范[10, 19, 33, 34]。这些提升源于在人类与机器人之间分配能动性这一理念；在执行过程中，双方都会影响机器人的最终动作，共同承担对动作进行推理的责任。

我们通过一项被试内用户研究（n = 12）评估 LILAC。在研究中，用户使用 LILAC 和两种基线方法，在 Franka Emika Panda 机械臂上完成一组复杂的操作任务——一种是最先进的语言信息引导的潜在动作（LILA）模型 [23], , 另一种是完全自主的语言条件模仿学习方法。我们发现，LILAC 的任务成功率高于两种基线方法，这是因为它能够根据在线语言指令进行适应；此外，从定性评价来看，用户认为 LILAC 更可靠、更精准，也更易于使用。

2 动机示例

学习得到的潜在动作范式 [21, 32] 最初是在辅助遥操作的背景下提出的；对于行动能力有限的用户而言，如何通过低维控制器（例如，安装在轮椅上的 2 自由度操纵杆）直观地控制一台 7 自由度 + 辅助机器人臂，极具挑战性。朴素的

方法试图将高维机器人控制问题映射到低自由度接口，例如通过操纵杆分别控制末端执行器的（x/y、z/roll、pitch/yaw）轴，但这会让用户感到极度不适，且需要频繁进行模式切换，控制不精确，并带来很高的认知负荷 [1, 16]。学习得到的潜在动作——尤其是近期关于语言引导的潜在动作（LILA）的研究 [23] ——提供了一种折中方案：利用少量特定任务示范，学习从操纵杆轴到末端执行器控制轴的非线性映射，使操纵杆的每个轴都能表示在任务空间中具有语义意义的运动。

作为一个具体示例，请看图2，任务是“拿起书并将其放入书架”。LILA 学习一个静态的映射，并在整个回合中使用它，结果遭遇了一种关键失效模式：用户将书从桌上移向架子时，误差不断累积，导致末端执行器与架子未对齐，无法顺利放入！这正是我们需要在线语言修正的原因——这种机制让用户能够快速诊断问题并改进机器人的行为。使用 LILAC 时，用户在执行过程中给出“稍微向下倾斜”的修正，并切换到一个新的控制空间。此时按下操纵杆左侧按钮即可明确、精确地控制机器人的姿态，使用户能够调整末端执行器的水平位置并完成任务。

3 相关工作

LILAC 建立在丰富的研究成果之上，涵盖了整合语言修正、学习语言条件策略以及整合其他形式的纠正反馈等方法。

融入用于操作任务的语言纠正。 与我们的方法最相关的是近期将各种自然语言纠正融入机器人操作任务的研究。这些方法可以根据其所作的假设，以及用户在执行期间何时提供纠正来分类。例如，Broad 等人 [5]通过使用分布式对应图来落实语言，并借助将语言映射到预定义纠正落地表示空间的语义解析器，实现了数据高效的在线纠正（类似于 LILAC）；这些落地表示较为脆弱，且由人工设计，同时还需要使用运动规划器（以及完全已知的环境动力学），才能确定一组可令机器人执行的有效动作。相比之下，后续融入纠正的研究不再需要脆弱的人工设计纠正原语，而是利用事后纠正来定义可组合代价函数，并将其输入轨迹优化器 [43]；这类纠正的事后特性有所局限，尤其是在任务包含不可逆或“难以重置”的环节时，而且轨迹优化器需要对环境有相当深入的了解。后续研究放宽了对这些先验知识的要求，但仍只能在事后融入纠正信息：在 2D [7] 和 3D [8] 环境中，利用大量配对的纠正与示范数据集，在语言纠正之后直接“修改”轨迹航点。与这些方法不同，LILAC 是一种在线、实时运行的共享自主方法，无需海量数据、先验环境动力学或完整的状态知识。

学习语言条件策略。 相比融入纠正，在完全自主和共享自主模式下学习语言条件策略的研究领域更为广泛。该领域的早期研究使用语义解析器，将自然语言指令映射到预定义的运动规划基元，并使用规模适中的数据集 [2, 3, 27, 46]。虽然这些方法能够以很高的可靠性完成有限范围的任务，但由于依赖预定义基元和运动规划器，它们难以扩展到更复杂的操作领域；在这些领域中，环境知识难以获取，而人工定义的基元则脆弱且有局限。因此，该领域近期的研究转而直接通过大型语言与示范配对数据集进行模仿学习，以学习语言条件策略 [19, 34, 38, 44, 45]。尽管表达能力强，这些方法仍极其依赖数据，即使学习最简单的任务也需要数百或数千个示例。为解决样本效率问题，LILA [23]等其他研究转向共享自主模式，以少一个数量级以上的示范数据来学习人机协作策略。LILA 是我们所提出方法的起点。

整合其他形式的纠正反馈。 其他方法致力于从其他形式的纠正反馈中学习，例如物理纠正 [28, 31], 定向干预，即人类通过远程遥操作完全接管机器人的控制权 [17, 25, 35], 批评反馈 [9, 13], 以及轨迹偏好 [4, 11]。近期，Schmittle 等人 [42] 提出了一种元算法，可利用多种类型的纠正反馈（不包括语言）进行在线学习。尽管这项工作前景可观，我们仍将研究重点放在语言修正上，因为语言是人类用户自然的交流方式。学习整合语言修正，还能实现跨任务转移，因为“向左”等修正用语具有通用性，通常不依赖于机器人的当前状态；而其他修正方式则可能更依赖具体情境。

4 LILAC：纠正的框定

LILAC 基于 Karamcheti 等人提出的 LILA [23]，并在执行过程中融入自然语言纠正。LILAC 架构如图 3所示；实线表示推理，虚线表示训练。LILAC 以数据驱动的方式融入自然语言纠正；本文聚焦于方向性纠正（例如“向左”）和指称性纠正（例如“朝蓝色标记方向”），并采用一种通用且可扩展的流程，能够推广至其他更复杂类型的修正。本节将概述我们方法的所有要素。

4.1 问题陈述

我们的设定是一个由元素 (S，A，T，U，C*，Z) 定义的序贯决策问题，其中 $s \in S \subseteq \mathbb{R}^n$ 表示机器人和环境的状态，$a \in A \subseteq \mathbb{R}^k$ 表示机器人的k维动作（在我们的例子中，是末端执行器位姿的 6 自由度增量——位置的笛卡尔坐标和方向的欧拉角），$T : S \times A \to S$ 是一个（随机的）、不可观测的转移函数。此外，$u \in U$ 表示用户提供的高级自然语言指令，$\mathbf{c} \in C^*$ 表示按顺序排列的

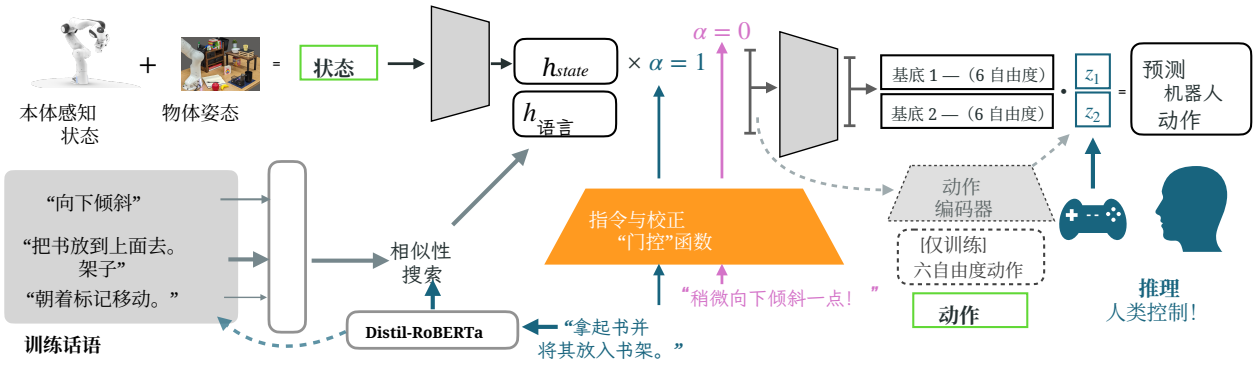


图 3：LILAC 概览——实线表示推理流程，虚线表示仅训练步骤。LILAC 能够高效融入语言修正，部分得益于其"门控"模块（橙色），该模块控制给定输入所需的状态上下文量。例如，要将"稍微向下倾斜"这样的修正落地，无需状态上下文（$\alpha = 0$）；而"拿起书并把它放进书架"这样的高层指令则需要上下文（$\alpha = 1$）。我们使用 GPT-3（一种预训练语言模型）来提供 α（参见 §4.3 中的讨论）。

（可能为空的）自然语言修正堆栈，即用户提供的修正；$z \in Z \subseteq \mathbb{R}^d$ where $d \ll k$ 表示用户通过其低维控制设备（例如，2 自由度操纵杆）提供的输入。用户可以在整个回合期间提供任意数量的在线纠正，以调整机器人的行为。

LILAC 的目标是学习一个函数 $F_\theta(s_t, z_t, u_t, \mathbf{c}_t)$：$S \times Z \times U \times C^* \rightarrow A$，将当前机器人和环境状态 s_t、低维控制输入 z_t、用户提供的初始高层指令 u（在给定回合中始终保持不变）以及语言修正的（可能为空的）堆栈 $\mathbf{c}_t$映射为要在环境中执行的高维机器人动作 a_t。相应的低自由度控制流形$\bigcup_{z_t \in Z} F_\theta(s_t, z_t, u_t, \mathbf{c}_t)$为用户提供了一个直观的界面，使其能够操控机器人完成相关任务。在每个新的时间步 $t+1$，用户可以提供新的语言修正 $c_\prime$，将其"推入"堆栈；也可以按下按钮，将最近的修正从堆栈中"弹出" $\mathbf{c}_t$，表示该修正已得到处理；或者提供一个控制输入 z_t，该输入会映射为相应的机器人动作 a_t。

4.2 建模：推理与学习

给定当前状态 s、语言 u 和 $\mathbf{c}$，F_θ 将低维用户控制输入 z映射为高维机器人动作 a。具体而言，我们将 $\hat{a} = F_\theta(s, z, u, \mathbf{c})$ 定义为：

$$
\begin{aligned}
h_{\text{state}} \in \mathbb{R}^m &= \texttt{EncodeState}_\theta(s) \\
h_{\text{language}} \in \mathbb{R}^m &= \texttt{EncodeLanguage}_\theta(u, \mathbf{c}) \\
\alpha \in [0,1] &= \texttt{GPTGating}(u, \mathbf{c}) \\
h_{\text{gated}} \in \mathbb{R}^m &= \alpha \cdot h_{\text{state}} + (1-\alpha) \cdot \text{bias}_\theta \\
h_{\text{fused}} \in \mathbb{R}^m &= \texttt{FiLM}_\theta(h_{\text{gated}}, h_{\text{language}}) \\
B_{\text{bases}} \in \mathbb{R}^{k \times d} &= \text{Gram-Schmidt}(\texttt{Projection}_\theta(h_{\text{fused}})) \\
\hat{a} \in \mathbb{R}^k &= B_{\text{bases}} \cdot z
\end{aligned}
$$

其中 m 是表示模型隐藏维度的超参数（本研究中我们设置为 $m = 128$）。

我们首先学习一个状态编码器 $\texttt{EncodeState}_\theta$。本研究使用的状态空间由机器人的本体感知状态（关节角度和末端执行器位姿）与一个向量拼接而成；该向量包含场景中每个物体的 (x, y, z) 位置。$\texttt{EncodeState}_\theta$ 是一个两层多层感知机，它接收该输入并输出 h状态。

为了对语言（u 和 $\mathbf{c}$）进行编码，我们采用后进先出的策略，有选择地对话语进行编码：仅对最近的话语进行编码——在交互开始时为 u，之后则为最近的纠正 $c_\prime$。我们使用 Sentence-BERT 项目发布的 Distil-RoBERTa 语言模型冻结变体 [41] 对该话语进行嵌入 [40]。这一过程由一个"非自然语言处理"最近邻索引 [36]提供支持：推理时的话语会映射到最接近的现有训练样本，并将其检索出来，输入模型的其余部分。与 LILA [23] 中采用的方法类似，这一过程可防止 LILAC 因语言的细微变化而退化，否则可能引发实际的用户安全问题。$\texttt{EncodeLanguage}_\theta$ 是另一个两层多层感知机，它接收检索到的嵌入，并输出 h语言。

接下来，我们将探讨如何融合状态嵌入与语言嵌入。LILAC 的关键组成部分及其保持数据高效的能力，源于 GPT-3 [6] "门控"模块（在图 3中以橙色显示），在上述公式中以标量值 $\alpha \in [0,1]$ 表示。门控值 α体现了一个简单的见解——不同的话语对对象／状态依赖的需求程度各不相同。例如，像"向左走"这样的话语，无需对环境中的任何物体进行推理（$\alpha = 0$）。关于门控以及我们如何将 GPT-3 实际化的详细讨论，见§4.3。我们使用这一门控值 α来调节状态信息在 h中的占比：通过将 α与 h状态及一个向量偏置$_\theta$进行凸组合来实现门控，其中 $\alpha = 0$ 消除了x状态。

我们通过 FiLM [39], 将 h语言映射为仿射变换参数 $\gamma \in \mathbb{R}^m$，并 $\beta \in \mathbb{R}^m$ 使用分别训练的两层多层感知机。随后，我们将得到的变换逐元素应用于 h门控后的特征，得到 h融合后的 $= \gamma \cdot h$门控特征 $+ \beta$。

最后，我们预测基向量 $B\text{bases} \in \mathbb{R}^{k \times d}$（请回想，$k$表示高自由度机器人动作空间的维度，而 d 表示用户的低自由度控制接口的维度）。Bbases 唯一确定给定时间步下用户的控制流形；我们训练一个两层多层感知机 Projection_θ，其输入为 h融合后的特征，输出具有所需维度的矩阵。为了得到经过适当条件化的控制流形，我们使用修正格拉姆-施密特法对基向量进行正交归一化。最终的高自由度机器人动作 $\hat{a} \in \mathbb{R}^k$ 是 B基向量与用户输入 $z \in \mathbb{R}^d$的矩阵-向量乘积。

从语言和示范中学习。为了学习 F_θ，我们假设有一个由（u = 语言，τ = 轨迹）对组成的数据集，其中每条轨迹由一系列（s，a）对构成；$\tau = \{(s_1, a_1) \ldots (s_T, a_T)\}$，并设定所需的潜在动作维度 d（例如，本研究中操纵杆的轴数为 $d = 2$）。我们使用的动作空间是末端执行器空间中的增量（笛卡尔位置、欧拉角方向）。请注意，这些输入与 F_θ 的签名并不完全一致——尤其是，对于给定的每个机器人动作 a，我们都无法获得对应的“真实”潜在动作 z。为了解决这个问题，我们借鉴了先前关于学习得到的潜在动作 [23, 24, 32]的研究思路：将训练过程设定为学习一个状态与语言条件自编码器，并利用压缩来诱导出有意义的潜在动作控制流形。

具体来说，我们通过添加一个层，将高自由度机器人动作压缩到一个 d维潜在空间中来实现：

$$z_{\text{compressed}} \in \mathbb{R}^d = \texttt{Compress}_\theta(a)$$

$$a_{\text{reconstruct}} \in \mathbb{R}^k = B_{\text{bases}} \cdot z_{\text{compressed}}$$

其中，B基向量按照上述方法根据状态和语言计算得出。

有了这个重建目标，我们可以写出一个紧凑的训练损失函数：$\text{L}(\theta) = ||a - a\text{重建}||_2^2$ ——换句话说，最小化动作高自由度机器人动作 a与a重建之间的均方误差。压缩器$_\theta$由一个两层多层感知机实现，并在训练后弃用。

4.3 指令与纠正的门控

LILAC 得以扩展的关键，在于认识到各种形式的纠正语言能够泛化到不同状态 ——换句话说，不同的语言话语对物体/环境状态依赖性的要求各不相同。要正式定义语言话语的“状态依赖性”很困难；一种启发式方法是根据话语中包含的指称对象数量对不同话语进行分类。例如，图图1中的“拿起边桌上的东西并把它放到桌子上”包含3个指称对象，表明其状态依赖性很强；机器人必须将话语与环境中的物体关联起来，才能确定正确的行为。然而，“不，往左一点！”这样的话语没有明确的指称对象；只需依靠用户的静态参考坐标系以及在末端执行器空间中产生的增量，就能理解这句话。

为了用 LILAC 将这一想法付诸实践，并进一步提升我们方法的样本效率，[2] 我们使用一个门控函数（橙色，见图 3），该函数根据语言预测一个离散值 $\alpha \in \{0, 1\}$。值 0 表示状态无关话语——例如，纠正“向下倾斜一点”。相应地，在我们的架构中，该值会将任何状态相关信息归零（见 α 项，位于图 3中），并仅根据所提供的语言预测动作。关键在于，提供给网络其余部分（网络定义于$\mathcal{F}_\theta$）的语言与状态的融合表示由 α 进行调制——更多细节见§4.4。

使用 GPT-3 识别纠正。在这项工作中，我们构建了一个基于 GPT-3 [6] 的提示框架，用于输出 α。这样做是因为刻画话语的状态依赖性很困难；虽然前面提到的指称计数启发式在某些情况下可能有效，但诸如“不，是蓝色的那个！”这样的话语包含隐式指称对象（在此例中，可能是标记笔或杯子），而这些对象需要根据环境状态进行定位。还有许多其他现象也让定义用于计算 α 的启发式变得困难——照应、空指称对象（“把机器人向左移动”）等。我们没有精心设计语法或启发式，而是选择借助大语言模型的上下文学习能力：这类模型能够根据提示和少量示例进行外推。我们具体采用 GPT-3 text-davinci-002（175B 参数） [6]。为设计提示，我们用 10 分钟迭代地构建输入／输出示例和任务描述，并使用一组留出的 5 条语言话语作为参考信号。[3]

4.4 可复现性

为促进可复现性和未来工作，我们发布了一个开源代码库（https://github.com/Stanford-ILIAD/lilac），其中包含涵盖数据收集、模型定义、训练和真实机器人部署的完整流程。

模型架构。 §4.2 中详细介绍的所有多层感知机 §4.2 使用 $m = 128$ 以及 GELU 激活函数 [15]。为提高稳定性，我们在将拼接后的状态表示输入状态编码器之前，添加了一个批归一化层 [18] 。按照当前实现，LILAC 极其轻量，仅有 188K 个参数。

训练细节。 训练 LILAC 效率很高，可在消费级笔记本电脑 CPU 上运行，无需昂贵的 GPU。我们训练 50 轮，批次大小为 512，使用 AdamW 优化器 [26] ，默认学习率为 0.001，权重衰减为 0.01。我们不采用其他形式的正则化（例如随机失活）。我们根据少量 ($n = 5$) 留出（语言、轨迹）对的验证损失来选择模型。

如 §4.2所述，我们假设有一个由话语及其对应轨迹构成的数据集。这些话语既包括高层任务话语（例如，“拿起书并将其放入书架”），也包括纠正话语（例如，“稍微向下倾斜一点”）。我们运行 GPT-3 α 标注流程作为预处理步骤，为每个示例标注所需的上下文依赖性程度，然后使用所有数据联合训练 $\mathcal{F}_\theta$。

5 用户研究预备工作

为评估 LILAC 相较于先前语言引导策略学习方法的表现，我们开展了一项被试内用户研究，共有 $n = 12$ 名参与者，每位参与者都对 LILAC 进行了评估

[2]一个合理的问题是，为什么不把所有话语都视为需要统一程度（或相同程度）的状态依赖性；答案根植于我们所处的小数据规模。我们需要在不同状态下收集同一条纠正语句“向左”的多个实例才能实现泛化，而采用 LILAC 的“门控”方法时，我们只需要一个实例！

[3]提示：https://github.com/Stanford-ILIAD/lilac/tree/main/scripts/alphas.py。

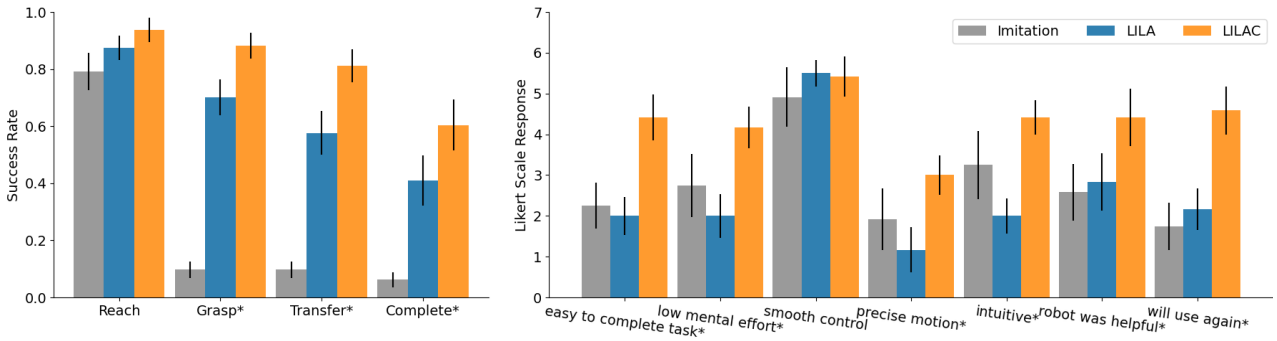


图 4：我们用户研究的结果（$n = 12$），涵盖三种条件：1）语言条件模仿学习；2）语言引导的潜在动作（LILA）——语言引导的共享自主的一种实现方式，不包含在线纠正；以及 3）LILAC——我们提出的方法，用户可以在机器人执行过程中的任何时刻提供在线纠正。

并将其与用于完全自主和共享自主的语言条件方法进行比较——即在与 LILAC 相同的示范上训练的语言引导模仿学习基线方法（“模仿”），以及同样在相同示范上训练的非纠正式共享自主基线方法（“LILA”）。以下章节将详细介绍环境、任务、数据收集流程以及用户研究流程。最后，我们将列出自变量、因变量测量指标和具体假设。

环境与任务。我们考虑一个多任务“桌面”环境（图 5），其中包含以下按复杂程度排列的任务：

(1) **清理垃圾**：将一团揉皱的纸（可变形）扔进黑色垃圾桶。(2) **转移笔**：将蓝色标记（图 5左上方）从架子转移到金属笔筒（左下方）。(3) **打开抽屉**：抓住小旋钮并水平拉出架子最底层的抽屉（需要精细控制末端执行器方向）。

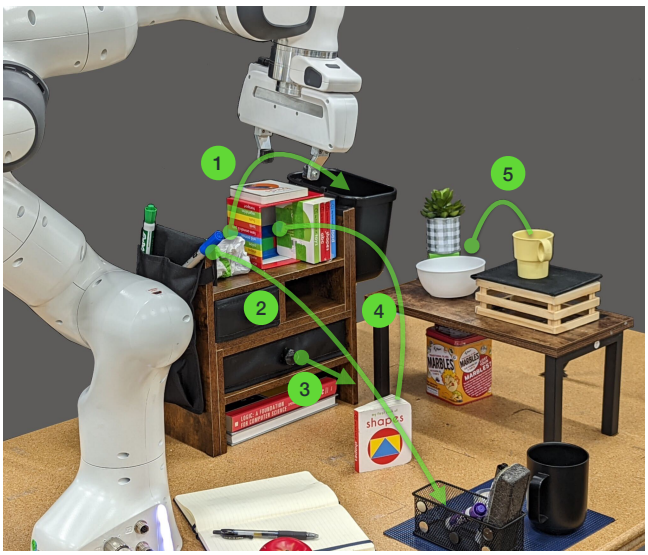


图 5：桌面操作环境的设置，以及高层任务示意图（更多详情见 §5）。

(4) **放入书本**：抓住书脊，将桌上的书拿起并插入书架（两侧的间隙只有几毫米）。(5) **给植物浇水**：用黄色杯子里的水给多肉植物浇水（图 5右上方的白色碗；为方便清理，我们用弹珠代替实际的水）。

我们定义的 5 项任务从操作角度来看都颇具难度，尤其是在我们所处的小数据规模下。为了细致比较所研究的三种方法，我们定义了一组子任务来衡量部分任务成功（将稀疏的完整任务成功率转化为更密集的进展度量）：a）抵达要操作的目标物体（例如，在**放入书本**任务中抵达书本）；b）成功抓取可操作物体；c）将目标物体转移到目标位置（例如，在**给植物浇水**任务中，将装水的杯子移至植物上方）；最后，d）完成整个任务。

示范与纠正数据收集。对于每项任务，我们都收集了密集的人类引导动觉示范——总计 50 次完整任务示范（与完全自主指令跟随方法通常所需的示范数量相比，少了几个数量级 [34, 38]）。机器人的本体感知状态由其关节状态（7 自由度；单位为弧度）以及通过正向运动学计算得出的末端执行器位姿（表示为 3 自由度笛卡尔位置和 3 自由度欧拉角方向）拼接而成；我们将示范中相邻时间步之间末端执行器空间的增量计算为高层机器人动作（得到 6 自由度高层动作）。我们以 10 Hz 频率记录示范并运行控制器；机器人控制平台基于 Polymetis [29]构建。

对于 LILAC，我们还收集了一小组纠正示范（通过与机器人互动不到 2 小时收集），并为其配上相应的纠正语言话语，涵盖两大类宽泛的纠正：1）方向性纠正，例如“向下倾斜”“向右”“逆时针旋转”；2）上下文指称纠正，例如“朝蓝色标记移动”或“不，朝抽屉上的旋钮向下移动”。我们通过回放完整任务示范来收集这些示范，并在回放过程中随机抽取中间状态以启动纠正。

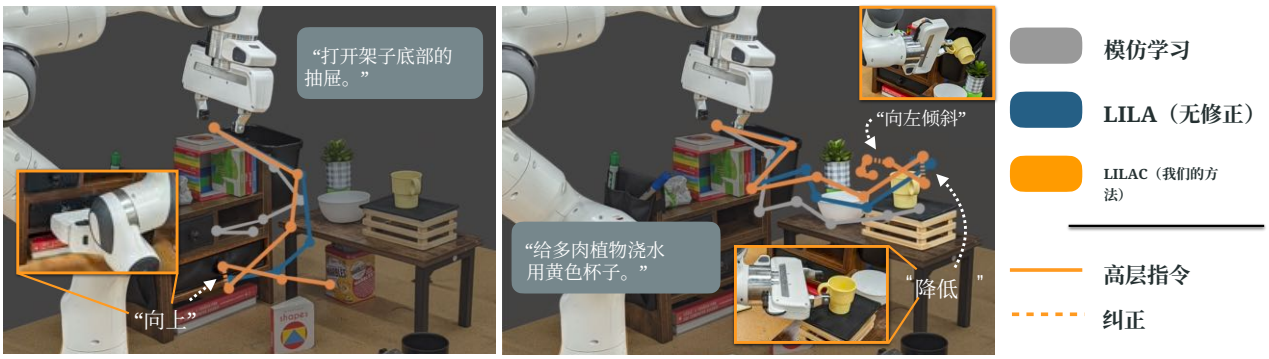


图 6：打开抽屉和给植物浇水任务中，不同控制策略下的定性轨迹。完全自主的模仿学习方法未能完成任务的第一阶段，而 LILA 能够到达抽屉和杯子，但无法精确瞄准并抓取物体。LILAC 也卡在同一位置，但随着用户发出低层修正来精确操控末端执行器，能够恢复并完整完成任务。

本文作者担任了完整任务示范和纠正示范的专家示范者。

参与者与流程。所有用户研究均依据经大学批准的 IRB审查方案开展，参与者从 12 名大学生中招募（8 男 4 女，年龄范围 20 至 30 岁，平均年龄 24.8 岁）。12 名参与者中，只有 4 人此前有遥操作机器人经验。所有研究均使用 Franka Emika Panda 机器人（如图 5所示），这是一种配有平行夹爪的七自由度固定臂机械臂。在所有设置中，Panda 的最大关节速度范数限制为 1 弧度/秒，扭矩限制则保守地设为 40.0 牛米。

我们开展了一项被试内用户研究，每位参与者都使用三种候选方法（分别称为模仿、*LILA*和*LILAC*（我们的方法），如图 4所示），完成 5 项高层任务中的 3 项。我们对不同用户使用候选方法的顺序进行了随机调整，以确保公平比较。开始时，每位用户都会阅读各控制方法的详细文字说明（具体文本可在项目网站上查阅），观看一段展示待执行高层任务的视频，并获准使用相应控制方法进行一次“练习”。每位用户针对给定任务进行两次试验；我们记录部分成功率，并请用户填写定性问卷，然后再切换到下一种策略。

假设。在这项研究中，我们改变控制策略（模仿、LILA 和 LILAC），并采用子任务成功率和完整目标成功率这两项客观指标来评估效果。此外，我们还通过 7 点李克特量表向用户发放问卷，跟踪“易用性”“平滑控制”以及“再次使用该控制策略的可能性”等定性指标（所有定性指标详见图 4）。我们针对 LILAC 相对于基线策略的表现检验以下两个假设：

H1 –LILAC 在完成复杂操作任务时，能够使用户获得更高的子任务成功率和完整目标成功率；相比之下，用户需为语言条件模仿学习智能体指定任务，或者使用以相同数据量训练的 LILA。

H2 – 根据调查问卷问题中的主观回答所衡量的整体可用性，用户在定性上更偏好 LILAC，而非两种基线策略。

基线实现。 LILA 和模仿（语言条件行为克隆）基线方法采用与 LILAC 完全相同的数据进行训练，训练流程也几乎完全相同。[4] LILA 按照与 LILAC 完全相同的架构进行训练，如 图 3所示，但不包含GPT-3 门控组件。模仿学习模型采用具备历史感知能力的策略，能够关注先前状态（在本研究中，我们将历史截断为 1 秒）。我们使用与 LILAC 相同的状态编码器和语言编码器，但采用 2 层 Transformer [50] 对完整历史序列进行编码。我们再次使用 FiLM 层融合语言和状态嵌入，然后通过最后一个 2 层多层感知机直接预测要在环境中执行的动作（用于开环控制）。

6 用户研究结果

我们报告了各子任务的成功率，该数值为所有用户的平均值，见图 4（左）。从客观结果来看，LILAC 在所有子任务中均取得了最高成功率；对于后三项子任务，其表现显著优于（$p < 0.05$）模仿学习和 LILA 基线方法——抓取、转移以及完整任务完成——这些结果充分支持**H1**。此外，图 4（右）显示，用户主观上也更偏好 LILAC。根据问卷结果，在 7 项指标中的 6 项上，用户对 LILAC 的偏好显著（$p < 0.05$），其中包括“易用性”“直观性”和“再次使用意愿”等。主观结果支持**H2**；用户青睐 LILAC 的适应性，因为它能让用户执行有针对性的精确运动。

可视化。 为了进一步理解 LILAC 以及融入在线纠正的价值，我们在 图 6 中可视化了两个高层任务采用三种控制策略时的示例轨迹。左侧展示的是简单的 打开抽屉任务：我们看到，完全自主的模仿学习模型甚至无法抵达抽屉，而 LILA 和 LILAC 虽然能够成功到达抽屉，却在尝试精准瞄准并抓取小旋钮时陷入困境。LILA 无法恢复，而 LILAC 能够响应用户修正，生成更精细的控制空间，使用户得以完成抓取并结束任务。在难度更高的 给植物浇水

[4]实现代码：https://github.com/Stanford-ILIAD/lilac/tree/main/models。

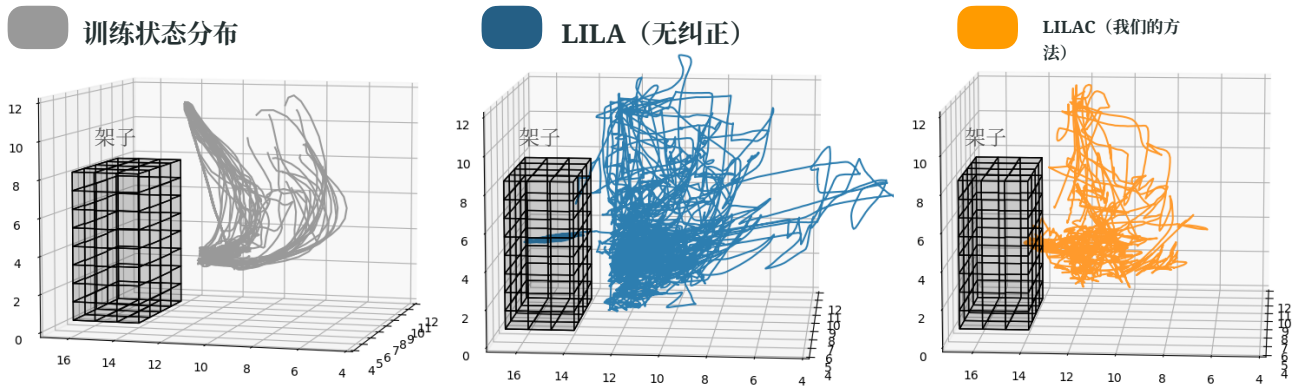


图7：LILA 和 LILAC 在打开抽屉任务中的观测轨迹（左侧显示训练轨迹）。虽然 LILA 偏离了观测到的状态分布，但使用 LILAC 遍历的状态接近训练期间观察到的状态。

任务中，我们看到类似的情况：模仿学习因撞翻杯子而灾难性失败，对环境造成不可逆的损害；LILA 能够到达杯子，但无法为用户提供足够的精度来成功抓取。借助 LILAC，用户可以通过两次紧密衔接的修正实现精准、有针对性的控制：首先抓取杯子，随后调整末端执行器的姿态，以成功完成倾倒动作。这些轨迹可视化有助于我们了解修正在何处以及何时最有用，具体展现了关键状态下对自适应性的需求。

图7 还绘制了所有用户在打开抽屉 任务中的三维末端执行器轨迹（位置；为清晰起见，省略了姿态），其中同时展示了 LILA 和 LILAC 的轨迹，以及训练数据中表示的轨迹。我们发现，与 LILA 相比，LILAC 使用户能够保持更接近训练状态分布的状态，这进一步解释了 LILAC 的出色表现。

7 讨论

尽管用户研究的结果颇具说服力，但我们认为有必要如实说明当前方法的不足，并探讨未来工作的可能方向。

局限性。 未来工作应研究不同类型的、对上下文更加敏感的语言修正；具体而言，由于我们对传入修正的编码方式（如 §4 所述），我们会独立地解读每条话语，而不考虑之前所说的内容。这限制了我们对照应或隐式共指等现象的解读——例如“不是，另一个方向”或“撤销那个”之类的修正，无法使用当前的框架实例得到正确解读。此外，我们发现，尽管修正相较于基础共享自主控制空间提供了更大的灵活性，但它们很容易被过度使用——我们注意到，研究中的一些参与者很快就偏离了高层指令所诱导的控制空间，转而在修正模式下完成大部分任务，实际上将 LILAC 变成了一个美化版的末端执行器控制界面，让用户一次只移动一个轴。要使底层高层控制空间更加自然直观——即 自然化控制界面——对于将 LILAC 扩展到更复杂、时间跨度更长的任务至关重要，在这些任务中，低层修正单独使用可能无法为用户提供足够的表达能力来完成任务。可能出现这种情况的场景包括：需要同时调节 3+ 个自由度的任务，或者连续的低层修正只能让用户以令人沮丧的缓慢速度推进当前任务（例如制作一杯茶这样的长时程任务）。最后，我们发现，“旋转”或“倾斜”等修正可能会产生歧义：有些用户希望根据自己的参考坐标系而不是机器人的参考坐标系来解读修正，另一些用户则恰恰相反。

结论。 在这项工作中，我们论证了，可扩展的语言驱动人机交互系统必须 同时具备自适应性 和样本效率。我们指出，处理在线自然语言修正 的能力能够为现有系统增添这种自适应性，并提出 LILAC——语言信息增强的带纠正潜在动作——作为一种潜在的解决方案。LILAC 建立在共享自主范式之上：自然语言话语被映射到有意义的低维控制空间，人类可以利用这些空间引导机器人；用户提供的每项修正都会优化 底层控制空间，从而实现精准、针对性的控制。我们通过用户研究将 LILAC 与语言条件模仿学习和语言引导的共享自主进行比较，结果表明，具备适应在线修正的能力十分重要：与两种基线方法相比，用户主观上更偏好 LILAC，且其客观表现也更出色。LILAC 为共享自主中的自适应语言驱动方法迈出了坚实的一步。我们希望，其可靠性、精确性和易用性这几项核心原则能在未来工作中得到延续。

致谢

丰田研究院（“TRI”）为本研究提供了经费支持。本项目还获得了美国海军研究办公室以及美国国家科学基金会项目 2006388 和 2132847 的支持。Siddharth Karamcheti 感谢开放慈善项目人工智能奖学金对他的资助。最后，我们谨向匿名评审致谢。

参考文献

[1] Brenna D Argall。2018。康复机器人技术中的自主性：一个交叉领域。《控制、机器人与自主系统年度评论》1（2018），441–463。[2] Yoav Artzi 和 Luke Zettlemoyer。2013。用于将指令映射到动作的语义解析器弱监督学习。《计算语言学协会汇刊》（TACL）1（2013），49–62。[3] Dilip Arumugam、Siddharth Karamcheti、Nakul Gopalan、Lawson L. S. Wong 和 Stefanie Tellex。2017。准确且高效地理解不同粒度的人机指令。载于《机器人：科学与系统》（RSS）。[4] Erdem Biyik 和 Dorsa Sadigh。2018。奖励函数的批量主动偏好学习。载于机器人学习会议（CoRL）。[5] Alexander Broad、Jacob Arkin、Nathan D. Ratliff、Thomas M. Howard 和 Brenna Argall。2017。面向辅助机器人操作器的实时自然语言修正。《国际机器人研究杂志》（IJRR）36（2017），684–698。[6] Tom B. Brown、Benjamin Mann、Nick Ryder、Melanie Subbiah、Jared Kaplan、Prafulla Dhariwal、Arvind Neelakantan、Pranav Shyam、Girish Sastry、Amanda Askell、Sandhini 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。语言模型是少样本学习者。arXiv 预印本 arXiv:2005.14165（2020）。[7] Arthur Fender C. Bucker、Luis F. C. Figueredo、Sami Haddadin、Ashish Kapoor、Shuang Ma 和 Rogerio Bonatti。2022。使用自然语言指令重塑机器人轨迹：基于 Transformer 的多模态数据对齐研究。载于智能机器人与系统国际会议（IROS）。978–984。[8] Arthur Fender C. Bucker、Luis F. C. Figueredo、Sami Haddadin、Ashish Kapoor、Shuang Ma、Sai Vemprala 和 Rogerio Bonatti。2022。LaTTe：语言轨迹变换器。arXiv 预印本 arXiv:2208.02918（2022）。[9] S. Chernova 和 Andrea Lockerd Thomaz。2014。从人类教师学习机器人。载于《从人类教师学习机器人》。[10] Maxime Chevalier-Boisvert、Dzmitry Bahdanau、Salem Lahlou、Lucas Willems、Chitwan Saharia、Thien Huu Nguyen 和 Yoshua Bengio。2019。BabyAI：研究具身语言学习样本效率的平台。载于国际学习表征会议（ICLR）。[11] Paul Christiano、Jan Leike、Tom B. Brown、Miljan Martic、Shane Legg 和 Dario Amodei。2017。从人类偏好中进行深度强化学习。载于神经信息处理系统进展（NeurIPS）。[12] John D. Co-Reyes、Abhishek Gupta、Suvansh Sanjeev、Nick Altieri、John DeNero、Pieter Abbeel 和 Sergey Levine。2019。通过元学习利用语言引导策略。载于国际学习表征会议（ICLR）。[13] Yuchen Cui 和 Scott Niekum。2018。基于批评反馈的主动奖励学习。载于国际机器人与自动化会议（ICRA）。6907–6914。[14] Anca D Dragan 和 Siddhartha S Srinivasa。2013。一种用于共享控制的策略融合形式体系。《国际机器人研究杂志》（IJRR）32（2013），790–805。[15] Dan Hendrycks 和 Kevin Gimpel。2016。高斯误差线性单元（GELU）。arXiv 预印本 arXiv:1606.08415（2016）。[16] Laura V Herlant、Rachel M Holladay 和 Siddhartha S Srinivasa。2016。通过自动时间最优模式切换实现机械臂辅助遥操作。载于 ACM/IEEE人机交互国际会议（HRI）。35–42。[17] Ryan Hoque、Ashwin Balakrishna、Ellen R. Novoseller、Albert Wilcox、Daniel S. Brown 和 Ken Goldberg。2021。ThriftyDAgger：面向交互式模仿学习的预算感知新颖性与风险门控。载于机器人学习会议（CoRL）。[18] Sergey Ioffe 和 Christian Szegedy。2015。批归一化：通过减少内部协变量偏移加速深度网络训练。载于国际机器学习会议（ICML）。448–456。[19] Eric Jang、Alex Irpan、Mohi Khansari、Daniel Kappler、Frederik Ebert、Corey Lynch、Sergey Levine 和 Chelsea Finn。2021。BC-Z：通过机器人模仿学习实现零样本任务泛化。载于机器人学习会议（CoRL）。[20] Shervin Javdani、Henny Admoni、Stefania Pellegrinelli、Siddhartha S Srinivasa 和 J Andrew Bagnell。2018。通过事后优化实现遥操作与协同作业中的共享自主。《国际机器人研究杂志》（IJRR）37（2018），717–742。[21] Hong Jun Jeon、Dylan P. Losey 和 Dorsa Sadigh。2020。利用学习得到的潜在动作实现共享自主。载于《机器人：科学与系统》（RSS）。[22] Siddharth Karamcheti、Dorsa Sadigh 和 Percy Liang。2020。通过分解学习自适应语言界面。载于 EMNLP交互式与可执行语义解析研讨会（IntEx-SemPar）。[23] Siddharth Karamcheti、Megha Srivastava、Percy Liang 和 Dorsa Sadigh。2021。LILA：语言引导的潜在动作。载于机器人学习会议（CoRL）。[24] Siddharth Karamcheti、A. Zhai、Dylan P. Losey 和 Dorsa Sadigh。2021。学习视觉引导的潜在动作以实现辅助遥操作。载于动力学与控制学习会议（L4DC）。[25] Michael Kelly、Chelsea Sidrane、K. Driggs-Campbell 和 Mykel J. Kochender- fer。2019年。HG-DAgger：与人类专家进行交互式模仿学习。载于国际机器人与自动化会议（ICRA）。8077–8083。[26] Diederik Kingma 和 Jimmy Ba。2015年。Adam：一种随机优化方法。载于国际学习表征会议（ICLR）。[27] T. Kollar、J. Krishnamurthy 和 Grant P. Strimel。2013年。迈向交互式落地语言习得。载于机器人：科学与系统（RSS）。[28] Mengxi Li、Alper Canberk、Dylan P. Losey 和 Dorsa Sadigh。2021年。从一系列物理纠正中学习人类目标。载于国际机器人与自动化会议（ICRA）。2877–2883。[29] Yixin Lin、Austin S. Wang、Giovanni Sutanto、Akshara Rai 和 Franziska Meier。2021年。Polymetis。https://facebookresearch.github.io/fairo/polymetis/。[30] Dylan P. Losey、Hong Jun Jeon、Mengxi Li、Krishna Parasuram Srinivasan、Ajay Mandlekar、Animesh Garg、Jeannette Bohg 和 Dorsa Sadigh。2021年。学习潜在动作以控制辅助机器人。《自主机器人》（AURO）（2021），1–33。[31] Dylan P Losey、Craig G McDonald、Edoardo Battaglia 和 Marcia K O'Malley。2018年。物理人机交互中共享控制的意图检测、仲裁与沟通方面综述。《应用力学评论》70（2018）。[32] Dylan P. Losey、Krishnan Srinivasan、Ajay Mandlekar、Animesh Garg 和 Dorsa Sadigh。2020年。使用学习得到的潜在动作控制辅助机器人。载于国际机器人与自动化会议（ICRA）。378–384。[33] Jelena Luketina、Nantas Nardelli、Gregory Farquhar、Jakob Foerster、Jacob An-dreas、Edward Grefenstette、Shimon Whiteson 和 Tim Rocktäschel。2019年。自然语言辅助强化学习综述。载于国际人工智能联合会议（IJCAI）。[34] Corey Lynch 和 Pierre Sermanet。2020年。在游戏中落地语言。arXiv预印本 arXiv:2005.07648（2020）。[35] Ajay Mandlekar、Danfei Xu、Roberto Martín-Martín、Yuke Zhu、Li Fei-Fei 和 Silvio Savarese。2020年。使用远程遥操作进行人在回路中的模仿学习。arXiv预印本 arXiv:2012.06733（2020）。[36] Alana Marzoev、S. Madden、M. Kaashoek、Michael J. Cafarella 和 Jacob Andreas。2020年。非自然语言处理：弥合合成语言数据与自然语言数据之间的差距。arXiv预印本 arXiv:2004.13645（2020）。[37] Cynthia Matuszek、Nicholas FitzGerald、Luke Zettlemoyer、Liefeng Bo 和 Dieter Fox。2012年。用于落地属性学习的语言与感知联合模型。载于国际机器学习会议（ICML）。1671–1678。[38] Oier Mees、Lukás Hermann 和 Wolfram Burgard。2022年。非结构化数据中的语言条件机器人模仿学习，哪些因素至关重要。《IEEE机器人与自动化快报》（RA-L）7（2022），11205–11212。[39] Ethan Perez、Florian Strub、Harm D. Vries、Vincent Dumoulin 和 Aaron C. Courville。2018年。FiLM：使用通用条件层进行视觉推理。载于人工智能促进协会（AAAI）。[40] Nils Reimers 和 Iryna Gurevych。2019年。Sentence-BERT：使用孪生BERT网络进行句子嵌入。载于自然语言处理实证方法会议（EMNLP）。[41] Victor Sanh、Lysandre Debut、Julien Chaumond 和 Thomas Wolf。2019年。DistilBERT：BERT的蒸馏版本，更小、更快、更便宜、更轻量。arXiv预印本 arXiv:1910.01108（2019）。[42] Matthew Schmittle、Sanjiban Choudhury 和 Siddhartha S Srinivasa。2020年。从纠正反馈中在线学习：一种用于机器人学的元算法。载于机器人学习会议（CoRL）。[43] Pratyusha Sharma、Balakumar Sundaralingam、Valts Blukis、Chris Paxton、Tucker Hermans、Antonio Torralba、Jacob Andreas 和 Dieter Fox。2022年。利用自然语言反馈纠正机器人计划。载于机器人：科学与系统（RSS）。[44] Mohit Shridhar、Lucas Manuelli 和 Dieter Fox。2021年。CLIPort：用于机器人操作的"What"和"Where"路径。载于机器人学习会议（CoRL）。[45] Simon Stepputtis、J. Campbell、Mariano Phielipp、Stefan Lee、Chitta Baral 和 H. B. Amor。2020年。用于机器人操作任务的语言条件模仿学习。载于神经信息处理系统进展（NeurIPS）。[46] Stefanie Tellex、Thomas Kollar、Steven Dickerson、Matthew R Walter、Ashis Gopal Banerjee、Seth J Teller 和 Nicholas Roy。2011年。理解用于机器人导航和移动操作的自然语言指令。载于人工智能促进协会（AAAI）。[47] Jesse Thomason、Michael Murray、Maya Cakmak 和 Luke Zettlemoyer。2019年。视觉与对话导航。载于机器人学习会议（CoRL）。[48] Jesse Thomason、Aishwarya Padmakumar、Jivko Sinapov、Nick Walker、Yuqian Jiang、Harel Yedidsion、Justin W. Hart、Peter Stone 和 Raymond J. Mooney。2019年。通过人机对话提升落地自然语言理解。载于国际机器人与自动化会议（ICRA）。[49] Jesse Thomason、Shiqi Zhang、Raymond J. Mooney 和 Peter Stone。2015年。通过人机对话学习理解自然语言指令。载于国际人工智能联合会议（IJCAI）。[50] Ashish Vaswani、Noam Shazeer、Niki Parmar、Jakob Uszkoreit、Llion Jones、Aidan N Gomez、Lukasz Kaiser 和 Illia Polosukhin。2017年。注意力就是你所需要的一切。arXiv预印本 arXiv:1706.03762（2017）。