OpenVLA：开源视觉—语言—动作模型

Moo Jin Kim[*,1] Karl Pertsch[*,1,2] Siddharth Karamcheti[*,1,3]
Ted Xiao[4] Ashwin Balakrishna[3] Suraj Nair[3] Rafael Rafailov[1] Ethan Foster[1] Grace Lam
Pannag Sanketi[4] Quan Vuong[5,†] Thomas Kollar[3] Benjamin Burchfiel[3]
Russ Tedrake[3,6] Dorsa Sadigh[1] Sergey Levine[2] Percy Liang[1] Chelsea Finn[1]

https://openvla.github.io

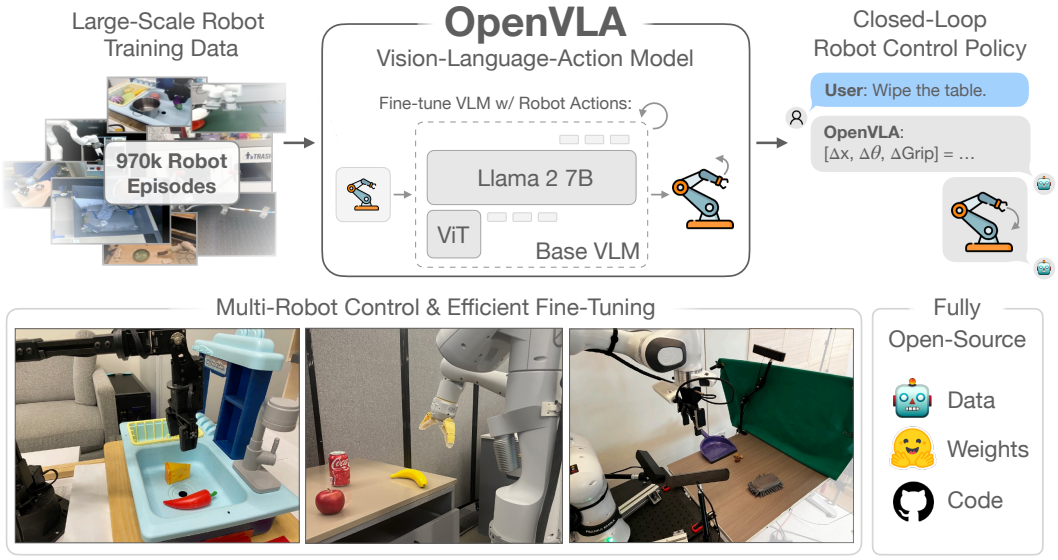


图 1：本文提出 OpenVLA，一个拥有 70 亿参数的开源视觉—语言—动作模型（Vision-Language-Action Model，VLA），使用 Open X-Embodiment 数据集中的 97 万段机器人轨迹训练 [1]。OpenVLA 在通用机器人操作策略上刷新了当前最佳性能，能够直接控制多种机器人，并通过参数高效微调快速适配新的机器人领域。OpenVLA 的模型检查点与 PyTorch 训练流程均已完全开源，模型可从 Hugging Face 下载并进行微调。

摘要： 在互联网规模的视觉—语言数据与多样化机器人示范上预训练的大型策略，有望改变机器人技能教学方式：无需从头训练新行为，只需微调视觉—语言—动作模型（Vision-Language-Action Model，VLA），便可得到鲁棒且具泛化能力的视觉运动控制策略。然而，VLA 尚未在机器人领域得到广泛采用，原因有二：1）现有 VLA 大多闭源，公众无法使用；2）以往工作没有研究如何面向新任务高效微调 VLA，而这正是推广应用的关键。为解决这些问题，本文提出 OpenVLA，一个拥有 7B 参数的开源 VLA，使用多样化的 97 万段真实机器人示范训练。OpenVLA 以 Llama 2 语言模型为基础，并结合融合 DINOv2 与 SigLIP 预训练特征的视觉编码器。得益于更丰富的数据和新的模型组件，OpenVLA 在通用机器人操作上表现出色：面对 29 项任务和多种机器人本体，其参数量仅为 RT-2-X（55B）的七分之一，任务绝对成功率却高出 16.5%。本文还表明，OpenVLA 能够有效微调以适应新环境，在包含多个物体、强调语言落地能力的多任务环境中展现出较强泛化性，并以 20.4% 的幅度超过从头训练的高表达能力模仿学习方法 Diffusion Policy。本文还研究了计算效率，并证明通过现代低秩适配方法可在消费级 GPU 上微调 OpenVLA，同时借助量化高效部署，且不降低

[*]：同等贡献

通讯作者：moojink@stanford.edu, pertsch@berkeley.edu, skaramcheti@stanford.edu
[1] 斯坦福大学，[2] 加州大学伯克利分校，[3] 丰田研究院，[4]Google DeepMind，[5]Physical Intelligence，[6] 麻省理工学院，[†] 部分工作于任职 Google DeepMind 期间完成

下游任务成功率。最后，本文发布模型检查点、微调 Notebook 与 PyTorch 代码库，内置对在 Open X-Embodiment 数据集上规模化训练 VLA 的支持。

1 引言

机器人操作学习策略的一个主要弱点，是无法泛化到训练数据之外。针对单项技能或语言指令训练的现有策略，虽能将行为外推到物体位置、光照等新的初始条件 [2, 3]，但面对场景干扰物或新物体时鲁棒性不足 [4, 5]，也难以执行未见过的任务指令 [6, 7]。机器人领域之外，CLIP [8]、SigLIP [9] 和 Llama 2 [10] 等视觉与语言基础模型却具备这类乃至更强的泛化能力，这源于其互联网规模预训练数据所蕴含的先验知识。机器人领域仍难以复现如此规模的预训练 — 即便最大的机器人操作数据集 [1, 11] 也仅有 10 万至 100 万个样本 — 但这种数据规模失衡也带来了机会：将现有视觉与语言基础模型作为训练机器人策略的核心构件，使策略能够泛化到训练数据之外的物体、场景和任务。

为实现这一目标，已有工作尝试将预训练语言模型和视觉—语言模型用于机器人表征学习 [12–14]，或作为模块化任务规划与执行系统的组成部分 [15, 16]。近期研究进一步使用这些模型直接学习用于控制的 VLA [VLAs; 1, 7, 17, 18]。VLA 是将预训练视觉与语言基础模型用于机器人的一种直接实现：对 PaLI [19, 20] 等视觉条件语言模型（Visually-Conditioned Language Model，VLM）直接微调，使其生成机器人控制动作。依托互联网规模数据训练的强大基础模型，RT-2 [7] 等 VLA 展现出较强鲁棒性，并能泛化到新物体和新任务，为通用机器人策略树立了新标准。然而，现有 VLA 难以普及主要有两个原因：1）当前模型 [1, 7, 17, 18] 并未开放，模型架构、训练流程与数据混合方案均缺乏透明度；2）已有工作没有提供将 VLA 部署并适配到新机器人、环境和任务的最佳实践，尤其缺乏面向消费级 GPU 等通用硬件的方案。本文认为，为未来研究与开发奠定充分基础，机器人领域需要支持有效微调和适配的开源通用 VLA，形成类似开源语言模型的生态系统 [21–24]。

为此，本文提出拥有 7B 参数的开源 VLA OpenVLA，在通用机器人操作策略上刷新了当前最佳性能。[1]OpenVLA 由能够捕获多粒度视觉特征的预训练视觉条件语言模型主干构成，并在 Open X-Embodiment [1] 中包含 970k 条机器人操作轨迹的大规模多样化数据集上微调；该数据集涵盖多种机器人本体、任务和场景。得益于更丰富的数据与新的模型组件，OpenVLA 在 WidowX 和 Google Robot 本体的 29 项评估任务上，绝对成功率比此前最佳的 550 亿参数 VLA RT-2-X [1, 7] 高 16.5%。本文还首次研究面向 *VLA* 的高效微调策略，覆盖从物体拾放到清洁桌面的 7 项操作任务。微调后的 OpenVLA 策略显著优于 Octo [5] 等微调后的预训练策略。与从头训练的扩散策略模仿学习 [3] 相比，微调后的 OpenVLA 在多物体、多任务环境中需要将语言落地为行为的任务上提升明显。在此基础上，本文首次证明低秩适配（Low-Rank Adaptation，LoRA） [26] 与模型量化 [27] 等计算高效微调方法的有效性，使 OpenVLA 无需大型服务器节点即可在消费级 GPU 上适配，且不损失性能。最后，本文开源全部模型、部署与微调 Notebook，以及用于规模化训练 VLA 的 OpenVLA 代码库，希望这些资源能推动机器人 VLA 的后续研究与适配。

2 相关工作

视觉条件语言模型 视觉条件语言模型（VLM）在互联网规模数据上训练，根据输入图像与语言提示生成自然语言，已广泛用于从视觉问答 [28–31] 到物体定位 [32, 33] 的众多应用。推动近期 VLM 发展的关键进展之一，是将预训练视觉编码器 [8, 9, 25] 的特征与预训练语言模型 [10, 23, 34–36] 连接起来的模型架构；这类架构直接结合计算机视觉与自然语言建模的进展，构建出强大的多模态模型。早期工作研究了视觉与语言特征交叉注意的多种架构 [37–41]，而新的开源 VLM [20, 42–44] 逐渐统一为更简单的“图像块即词元”方法：将预训练视觉 Transformer 输出的图像块特征视为词元，再投影至语言模型输入空间。这种简洁设计使大规模语言模型训练工具能够方便地复用于 VLM 训练。本文使用这些工具扩展 VLA 训练，并选用 Karamcheti et al. [44] 的 VLM 作为预训练主干。该模型在多分辨率视觉特征上训练，融合 DINOv2 [25] 的低层空间信息与 SigLIP [9] 的高层语义，以增强视觉泛化能力。

[1]OpenVLA 采用多个预训练模型组件：SigLIP [9] 与 DinoV2 [25] 视觉编码器，以及 Llama 2 [10] 语言模型主干。这三个模型均开放权重，但不开放训练数据或代码。本文发布用于在这些组件上复现 OpenVLA 的训练数据、代码和模型权重。

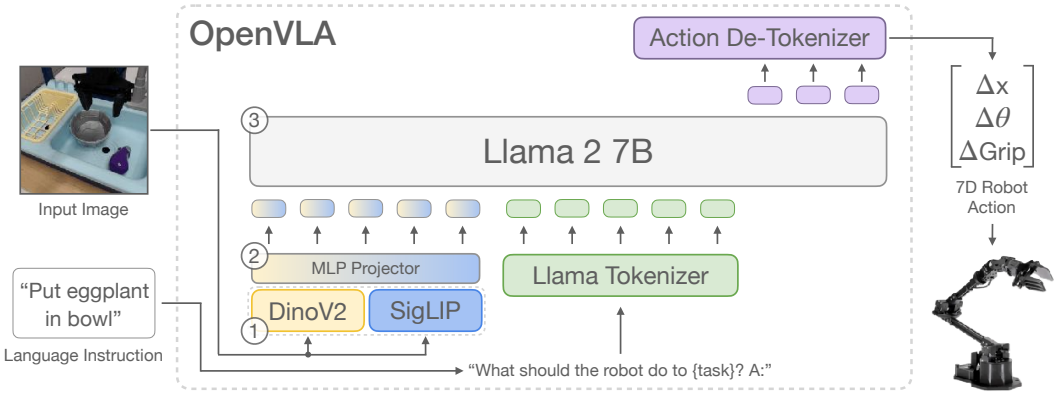


图 2: **OpenVLA 模型架构**。给定图像观测和语言指令，模型预测 7 维机器人控制动作。该架构包含三个关键组件：(1) 拼接 DINO V2 [25] 与 SigLIP [79] 特征的**视觉编码器**；(2) 将视觉特征映射到语言嵌入空间的**投影器**；(3) **大语言模型主干**，即拥有 7B 参数的 Llama 2 大语言模型 [10]。

通用机器人策略 机器人领域近期的一项趋势，是在大规模、多样化且涵盖多种机器人本体 [1, 5, 45–55] 的数据集 [1, 2, 6, 11, 46, 56–63] 上训练多任务"通用"机器人策略 [2, 6, 58, 63–66]。其中，Octo [5] 训练了一种可直接控制多种机器人、并能灵活微调到新机器人配置的通用策略。这些方法与 OpenVLA 的关键差异在于模型架构。Octo 等以往工作通常组合语言嵌入、视觉编码器等预训练组件与从头初始化的额外模型组件 [2, 5, 6]，在策略训练过程中学习将它们"拼接"起来。OpenVLA 则采用更端到端的方法：把机器人动作视为语言模型词表中的词元，直接微调 VLM 以生成动作。实验表明，这一简洁且可扩展的流程显著优于以往通用策略的性能与泛化能力。

视觉—语言—动作模型 许多工作探索了 VLM 在机器人领域的应用，例如视觉状态表征 [12, 13]、物体检测 [67]、高层规划 [16] 和反馈信号生成 [68–71]。另一些工作将 VLM 直接集成到端到端视觉运动操作策略中 [14, 15]，但在策略架构中加入大量先验结构，或要求相机经过标定，因而限制了适用范围。近期多项研究采用与本文相似的方案，直接微调大型预训练 VLM 以预测机器人动作 [1, 7, 17, 18, 72–74]。这类模型将机器人控制动作直接融入 VLM 主干，通常称为视觉—语言—动作模型（VLA）。这一设计有三项主要优势：（1）在互联网规模视觉—语言数据集上对齐预训练视觉与语言组件；（2）采用非专为机器人控制定制的通用架构，从而利用现代 VLM 训练背后的可扩展基础设施 [75–77]，仅需少量代码修改即可扩展到十亿参数级策略训练；（3）为机器人领域直接利用 VLM 的快速进步提供路径。现有 VLA 工作要么只关注单机器人或仿真配置下的训练与评估 [72–74, 78]，缺乏通用性；要么保持闭源，不支持高效微调到新的机器人配置 [1, 7, 17, 18]。

与本文最相关的是 RT-2-X [1]：它在 Open X-Embodiment 数据集上训练了拥有 550 亿参数的 VLA 策略，并在通用操作策略上达到此前最佳性能。不过，本文与 RT-2-X 存在多项重要差异：（1）OpenVLA 将强大的开放 VLM 主干与更丰富的机器人预训练数据集结合，模型规模小一个数量级，却在实验中优于 RT-2-X；（2）本文全面研究将 OpenVLA 微调到新目标配置，而 RT-2-X 未研究微调场景；（3）本文首次证明现代参数高效微调与量化方法对 VLA 的有效性；（4）OpenVLA 是首个开源通用 VLA，可支持后续研究 VLA 的训练、数据混合方案、目标函数与推理。

3 OpenVLA 模型

本文提出 OpenVLA 模型，这是一个拥有 7B 参数的视觉—语言—动作模型（Vision-Language-Action Model，VLA），在 Open X-Embodiment 数据集的 970k 个机器人演示上训练 [1]。开发 VLA 模型的最佳实践仍存在许多尚未充分探索的问题，e.g., 应选择何种模型主干、数据集和超参数进行训练。下文将详述 OpenVLA 的开发方法并总结关键经验。具体而言，首先简要概述构成 OpenVLA 主干的现代视觉条件语言模型（Vision-Language Model，VLM）（章节 3.1）；随后介绍基本训练方案和数据集（章节 3.2和章节 3.3）；讨论关键设计决策（章节 3.4）；最后说明训练和推理所用的基础设施（章节 3.5）。

3.1 预备知识：视觉条件语言模型

近期大多数 VLM [20, 42–44] 的架构由三个主要部分组成（见图 2）：(1) 将图像输入映射为若干“图像块嵌入”的**视觉编码器**；(2) 接收视觉编码器输出嵌入并将其映射到语言模型输入空间的**投影器**；(3) **大语言模型（Large Language Model，LLM）主干**。训练 VLM 时，模型在从各类互联网来源整理的配对或交错式视觉—语言数据上，以预测下一个文本词元为目标进行端到端训练。

本文以 Prismatic-7B VLM [44] 为基础。Prismatic 采用上述标准架构，包括拥有 600M 参数的**视觉编码器**、小型两层 MLP **投影器**，以及拥有 7B 参数的 Llama 2 **语言模型主干** [10]。Prismatic 采用双组件视觉编码器，由预训练的 SigLIP [79] 和 DINOv2 [25] 模型组成。输入图像块分别通过两个编码器，所得特征向量沿通道维拼接。相比更常用的 CLIP [80] 或仅使用 SigLIP 的视觉编码器，加入 DINOv2 特征已被证明有助于提升空间推理能力 [44]，这对机器人控制尤为有益。

SigLIP、DINOv2 和 Llama 2 均未公开其训练数据细节；这些数据很可能分别包含数万亿个源自互联网的图文、纯图像和纯文本词元。Prismatic VLM 在这些组件之上使用 LLaVA 1.5 数据混合 [43] 进行微调，其中包含来自开源数据集 [29, 42, 81–83] 的约 1M 个图文和纯文本数据样本。

3.2 OpenVLA 训练流程

为训练 OpenVLA，本文对预训练的 Prismatic-7B VLM 主干进行微调，使其能够预测机器人动作（见图 2）。动作预测被表述为“视觉—语言”任务：将输入观测图像和自然语言任务指令映射为预测机器人动作的字符串 [7]。为了使 VLM 的语言模型主干能够预测机器人动作，本文将连续机器人动作映射为该语言模型分词器使用的离散词元，从而在 LLM 输出空间中表示动作。遵循 Brohan et al. [7]，每个机器人动作维度分别离散为 256 个区间之一。对于每个动作维度，区间宽度设为均匀划分训练数据中动作第 1^{st} 与第 99^{th} 分位数之间的范围。Brohan et al. [7]使用最小值—最大值边界，而本文采用分位数，因而可以忽略数据中的离群动作；否则这些动作可能大幅扩张离散化区间并降低动作离散化的有效粒度。

通过这种离散化方式，可得到 N 个离散整数 $\in [0 \dots 255]$，用于表示一个 N 维机器人动作。遗憾的是，OpenVLA 语言主干使用的分词器，即 Llama 分词器 [10]，仅为微调期间新引入的词元预留 100 个“特殊词元”，不足以容纳动作离散化所需的 256 个词元。为保持简洁，本文再次沿用 Brohan et al. [7]的方法，直接以动作词元覆盖 Llama 分词器词表中使用频率最低的 256 个词元（即最后 256 个词元）。动作被处理为词元序列后，使用标准的下一词元预测目标训练 OpenVLA，且仅在预测的动作词元上计算交叉熵损失。章节 3.4讨论了实现该训练流程时的关键设计决策。下文介绍用于训练 OpenVLA 的机器人数据集。

3.3 训练数据

构建 OpenVLA 训练数据集的目标是覆盖丰富多样的机器人本体、场景和任务，使最终模型既能开箱即用地控制多种机器人，又能高效微调到新的机器人配置。本文以 Open X-Embodiment 数据集 [1]（OpenX）为基础整理训练数据集。截至本文撰写时，完整 OpenX 数据集由 70 多个独立机器人数据集组成，包含超过 2M 条机器人轨迹；大型社区协作将其汇集为统一且易用的数据格式。为了使这些数据上的训练切实可行，本文对原始数据集执行多步数据整理。

整理的目标是确保：(1) 所有训练数据集具有统一的输入与输出空间；(2) 最终训练混合中的本体、任务和场景保持均衡。[2]为实现目标 (1)，本文遵循 [1, 5]，将训练数据限制为至少配有一个第 3^{rd} 人称相机的操作数据集，并采用单臂末端执行器控制。对于目标 (2)，所有通过首轮筛选的数据集均采用 Octo [5] 的数据混合权重。Octo 采用启发式方法降低权重或移除多样性较低的数据集，并提高任务和场景更多样的数据集权重；详见 Octo Model Team et al. [5]。

本文还尝试将 Octo 发布后新增至 OpenX 的少数数据集纳入训练混合，其中包括 DROID 数据集 [11]，但为其设置了较为保守的 10% 混合权重。实践中，DROID 上的动作词元准确率在整个训练期间始终较低，表明未来可能需要更高的混合权重或更大模型才能拟合其多样性。为避免损害最终模型质量，本文在最后三分之一训练阶段将 DROID 从数据混合中移除。章节 A完整列出了所用数据集及其混合权重。

[2]Octo [5] 展示了如何在具有异构感知输入的数据集上训练。尽管这一方向很有前景，本文将跨异构传感器模态与动作空间的 VLA 训练留待未来研究。

3.4 OpenVLA 设计决策

在开发 OpenVLA 模型时，本文先通过小规模实验探索多项设计决策，再启动最终模型训练。具体而言，为提高迭代速度并降低计算成本，初步实验在 BridgeData V2 [6] 上训练和评估 OpenVLA 模型，而非使用完整 OpenX 数据混合。下文总结这些探索所得的关键经验。

VLM 主干。初期实验测试了多个 VLM 主干。除 Prismatic [44] 外，还测试了对 IDEFICS-1 [84] 和 LLaVA [85] 进行机器人动作预测微调。在场景仅包含一个物体的任务上，LLaVA 与 IDEFICS-1 表现相当；但在场景含多个物体、且策略需要操作正确物体时，LLaVA 表现出更强的语言定位能力，i.e., 需操作语言指令指定的物体。具体而言，在 BridgeData V2 水槽环境中的五项语言定位任务上取平均后，LLaVA 的绝对成功率比 IDEFICS-1 高 35%。经过微调的 Prismatic VLM 策略进一步提升了性能：在简单单物体任务和多物体语言定位任务上，其绝对成功率均比 LLaVA 策略高约 10%。本文将这种性能差异归因于融合 SigLIP-DINOv2 主干带来的更强空间推理能力（见章节 3.1）。除性能优势外，Prismatic 还提供模块化且易用的代码库，因此最终选择其作为 OpenVLA 模型的主干。

图像分辨率。输入图像分辨率显著影响 VLA 训练的计算需求，因为更高分辨率会产生更多图像块词元，进而增加上下文长度，使训练计算量呈二次增长。本文比较了输入为 224×224px 和 384×384px 的 VLA，评估中未发现性能差异，但后者训练耗时为前者的 3 倍。因此，最终 OpenVLA 模型采用 224×224px 分辨率。尽管提高分辨率能改善许多 VLM 基准上的性能 [44, 86, 87]，本文尚未在 VLA 上观察到这一趋势。

微调视觉编码器。既往 VLM 研究发现，在 VLM 训练期间冻结视觉编码器通常会获得更高性能 [44]。直观而言，冻结视觉编码器可能更有利于保留其从互联网规模预训练中学到的稳健特征。然而，本文发现，在 VLA 训练期间微调视觉编码器对获得良好 VLA 性能至关重要。本文推测，预训练视觉主干可能无法捕获场景重要部分足够精细的空间细节，因而难以实现精确机器人控制。

训练轮数。典型 LLM 或 VLM 训练至多遍历训练数据集一至两轮。相比之下，VLA 训练需要显著增加数据集遍历次数；在训练动作词元准确率超过 95% 之前，真实机器人性能会随训练持续提升。最终训练在训练数据集上完成 27 轮。

学习率。本文在多个数量级范围内扫描 VLA 训练学习率，结果表明固定学习率 2e-5 效果最佳（与 VLM 预训练所用学习率相同 [44]）。实验未发现学习率预热带来收益。

3.5 训练与推理基础设施

最终 OpenVLA 模型在由 64 A100 块 GPU 组成的集群上训练 14 天，合计 21,500 个 A100 小时，批大小为 2048。推理时，以 `bfloat16` 精度加载（i.e., 不量化）的 OpenVLA 需要 15GB GPU 显存，在单块 NVIDIA RTX 4090 GPU 上的运行频率约为 6Hz（未采用编译、推测解码或其他推理加速技巧）。如章节 5.4所示，通过量化可进一步降低 OpenVLA 推理时的内存占用，且不会损害真实机器人任务中的性能。图 6报告了多种消费级和服务器级 GPU 上的推理速度。为方便使用，本文实现了远程 VLA 推理服务器，可将动作预测实时远程流式传输至机器人，因而无需强大的本地计算设备即可控制机器人。该远程推理方案随开源代码一同发布（章节 4）。

4 OpenVLA 代码库

除模型外，本文还发布了 OpenVLA 代码库，这是用于训练 VLA 模型的模块化 PyTorch 代码库（见`https://openvla.github.io`）。该代码库既支持在单块 GPU 上微调 VLA，也能在多节点 GPU 集群上训练拥有数十亿参数的 VLA，并支持自动混合精度（Automatic Mixed Precision，AMP；PyTorch [75]）、FlashAttention [76] 和完全分片数据并行（Fully Sharded Data Parallel，FSDP；Zhao et al. [77]）等现代大型 Transformer 模型训练技术。OpenVLA 代码库原生完整支持 Open X 数据集训练，集成 HuggingFace [21] 的 `AutoModel` 类，并支持 LoRA 微调 [26] 和量化模型推理 [27, 88]。

5 实验

实验评测旨在检验 OpenVLA 能否直接作为强大的多机器人控制策略，以及能否作为微调至新机器人任务的良好初始化。具体而言，本文试图回答以下问题：

1. 在多个机器人和多种泛化类型上评测时，OpenVLA 与以往通用机器人策略相比表现如何？

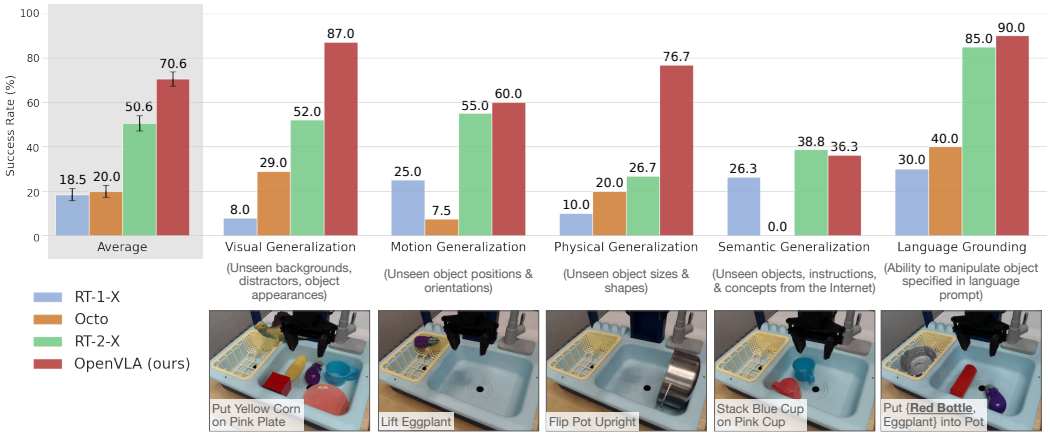


图 3: **BridgeData V2 WidowX 机器人评测任务与结果**。我们在一套涵盖多个泛化维度的综合任务以及专门评估语言条件控制能力的任务上，对 OpenVLA 和以往最先进的通用机器人策略进行评测。OpenVLA 的总体性能最高，除语义泛化外，在所有类别中均优于闭源模型 RT-2-X。每种方法的平均成功率 ± 标准误（StdErr）根据总计 170 条试验轨迹计算。详细结果见 表 4。

2. OpenVLA 能否在新的机器人配置和任务上有效微调？与当前最先进的数据高效模仿学习方法相比表现如何？
3. 能否通过参数高效微调和量化降低 OpenVLA 模型的训练与推理计算需求，使其更易于使用？性能与算力之间存在怎样的权衡？

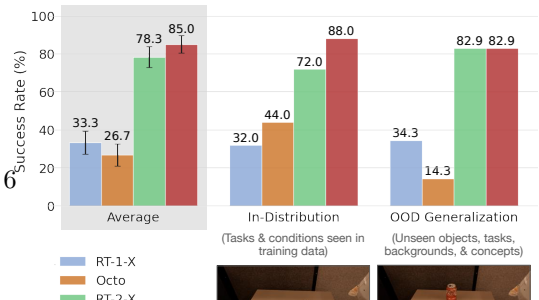
5.1 在多个机器人平台上的直接评测

机器人配置与任务。我们在两个机器人本体上评估 OpenVLA 的“开箱即用”性能：BridgeData V2 评测所用的 WidowX 机器人 [6]（见 图 1 左图），以及 RT-1 和 RT-2 评测所用的移动操作机器人 [2, 7]（“Google robot”；见 图 1 中图）。以往工作广泛使用这两个平台评估通用机器人策略 [1, 2, 5, 7]。本文在每个环境中定义了一套综合评测任务，覆盖多个泛化维度，包括**视觉**泛化（未见过的背景、干扰物体，以及物体颜色/外观）、**运动**泛化（未见过的物体位置/朝向）、**物理**泛化（未见过的物体尺寸/形状）和**语义**泛化（未见过的目标物体、指令和互联网概念）。此外，本文还在包含多个物体的场景中评估语言条件控制能力，检验策略能否按照用户提示操作正确的目标物体。BridgeData V2 和 Google robot 评测的任务示例图像分别见 图 3 和 图 4 的底行。总体而言，在 BridgeData V2 实验中，每种方法评测 170 条试验轨迹（17 个任务，每个任务 10 次试验）；在 Google robot 实验中，每种方法评测 60 条试验轨迹（12 个任务，每个任务 5 次试验）。所有任务及其与训练数据差异的详细划分见 章节 B。为确保公平比较，本节及后续章节的所有评测均采用 A/B 评测方式，在相同任务上使用相同的机器人和物体初始状态集合。

对比方法。本文将 OpenVLA 与三种以往的通用操作策略进行比较：**RT-1-X** [1]、**RT-2-X** [1] 和 **Octo** [5]。**RT-1-X**（35M 参数）和 **Octo**（93M 参数）均为在 OpenX 数据集子集上从头训练的 Transformer 策略；Octo 是开源操作策略中当前最先进的模型。**RT-2-X**（55B 参数）是当前最先进的闭源视觉—语言—动作模型（VLA），采用经互联网数据预训练的视觉和语言骨干网络。

BridgeData V2 和 Google robot 的评测结果分别汇总于 图 3 和 图 4（各任务的详细结果见附录 表 4 和 表 6）。RT-1-X 和 Octo 在测试任务上均表现不佳，常常无法操作正确的物体，尤其是在存在干扰物体时；某些情况下，机器人甚至会漫无目的地挥动机械臂。为检验经互联网数据预训练的 VLA 模型，本文评测所考察的泛化程度甚至高于这些以往工作。因此，未经互联网数据预训练的模型性能较低符合预期。RT-2-X 明显优于 RT-1-X 和 Octo，表明大型预训练视觉—语言模型（Vision-Language Model, VLM）有利于机器人控制。

尽管 OpenVLA 的规模小一个数量级（7B 参数对 55B 参数），但它在 Google robot 评测中与 RT-2-X 表现相当，在 BridgeData V2 评测中则显著优于 RT-2-X。从定性结果看，



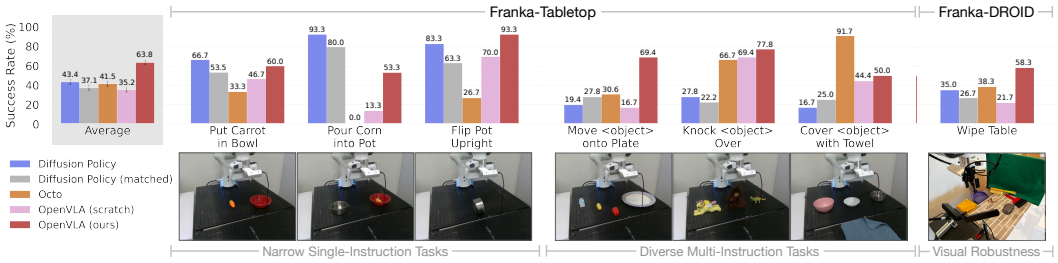


图 5: **适配新机器人配置**。本文评测在七个 Franka Emika Panda 任务（每个任务 10–150 个演示）上从头训练的当前最先进 Diffusion Policy，以及在相同数据上微调的通用机器人策略 Octo 和 OpenVLA。Diffusion Policy 在范围较窄的单指令任务上表现强劲，而 Octo 和 OpenVLA 在涉及多条指令和干扰物体的多样化微调任务上表现更好。总体而言，OpenVLA 在两种配置上的综合性能最高，表明它可作为下游任务策略学习的有效默认选择。每种方法的平均成功率 ± 标准误根据 129 条试验轨迹计算（Franka-Tabletop 任务 99 条，Franka-DROID 任务 30 条）。详细结果见 表 7。

RT-2-X 和 OpenVLA 的行为均明显比其他受测模型更稳健，例如在存在干扰物体时接近正确物体、适当调整机器人末端执行器使其与目标物体朝向对齐，甚至能从抓取不牢等错误中恢复（定性试验轨迹示例见 https://openvla.github.io）。如 图 3 所示，RT-2-X 在语义泛化任务上的性能更高。这符合预期，因为它使用了规模更大的互联网预训练数据，并在机器人动作数据和互联网预训练数据上联合微调，从而更好地保留预训练知识；相比之下，OpenVLA 仅在机器人数据上微调。然而，在 BridgeData V2 和 Google robot 评测的所有其他任务类别中，OpenVLA 的表现均相当或更好。性能差异可归因于多项因素：本文为 OpenVLA 整理了包含 970k 条轨迹的更大训练数据集（RT-2-X 为 350k 条）；对训练数据集进行了更细致的清洗，e.g., 滤除 Bridge 数据集中的全零动作（详见 章节 C）；此外，OpenVLA 使用融合视觉编码器，结合了预训练的语义与空间特征。这些组件的消融分析见 章节 D。

5.2 数据高效地适配新机器人配置

以往工作主要关注直接评测 VLA 的“开箱即用”能力 [1, 7, 16]，而如何将 VLA 模型有效微调至新任务和机器人配置仍缺乏研究，但这对其广泛应用至关重要。本节研究 OpenVLA 快速适配新的真实世界机器人配置的能力。（仿真环境中的微调实验见 章节 E。）

机器人配置与任务。本文测试一种简单的 OpenVLA 模型微调方案：使用包含目标任务 10–150 个演示的小型数据集，对全部模型参数进行完整微调（见 图 5；参数高效微调方法见 章节 5.3）。本文在两种配置中测试 OpenVLA：**Franka-Tabletop**，即固定安装于桌面的 Franka Emika Panda 7 自由度机械臂；以及 **Franka-DROID**，即近期发布的 DROID 数据集 [11] 中的 Franka 机械臂配置，安装在可移动升降桌上。两种配置分别采用 5Hz 和 15 Hz 非阻塞控制器。Franka 机械臂在机器人学习社区中应用广泛，很可能成为 OpenVLA 微调的目标机器人本体，因此本文选择它作为微调实验的目标本体。通过测试不同控制频率的配置，检验 OpenVLA 对多种使用场景的适用性。

对比方法。本文与从头训练的当前最先进数据高效模仿学习方法 **Diffusion Policy** [3] 进行比较。此外，还比较了输入和输出规格与 OpenVLA 一致的 **Diffusion Policy (matched)** 版本。[3] 此外，本文评测在目标数据集上微调的 **Octo** [5]，因为它是当前支持微调的最佳通

[3]完整的 Diffusion Policy 使用包含图像和本体感知状态的两步观测历史，并通过预测由 T 个未来动作组成的动作块执行滚动时域控制：先以开环方式执行前 X 个动作，再预测下一个动作块（对于 15Hz 控制，与 DROID 以往工作 [11] 相同，设置 $T = 16, X = 8$；对于 5Hz 控制，将动作块大小减小至 $T = 8, X = 3$）。它也是 章节 5.2 中唯一通过预测绝对笛卡尔坐标控制机器人的方法；其他所有

用策略（RT-2-X 的推理 API 不支持微调）。本文也在相同目标数据集上微调 OpenVLA，所得策略记为 **OpenVLA**。最后，作为消融实验，本文与 **OpenVLA (scratch)** 进行比较。该方法直接在目标机器人配置上微调底层基础 Prismatic VLM，而非微调经过 OpenX 预训练的 OpenVLA 模型，以评估大规模机器人预训练的收益。

结果见 图 5（各任务的详细结果见附录 表 7）。在“将胡萝卜放入碗中”和“将玉米倒入锅中”等范围较窄的单指令任务上，两个版本的 Diffusion Policy 均可与通用策略 Octo 和 OpenVLA 竞争或优于二者；但在场景中包含多个物体且需要语言条件控制的更多样化微调任务上，预训练通用策略表现更好。对 Octo 和 OpenVLA 进行 OpenX 预训练，使模型能更好地适应这些注重语言落地的多样化任务；OpenVLA (scratch) 较低的性能为此提供了证据。

总体而言，OpenVLA 的平均性能最高。多数以往方法仅在范围较窄的单指令任务或多样化的多指令任务中的一类任务上表现强劲，因而成功率差异很大。OpenVLA 是唯一在所有受测任务上均达到至少 50% 成功率的方法，表明它可作为模仿学习任务的有力默认选择，尤其适合包含多样化语言指令的任务。对于范围较窄但要求高度灵巧操作的任务，Diffusion Policy 的轨迹仍更平滑、更精确；引入 Diffusion Policy 所采用的动作分块和时序平滑，可能帮助 OpenVLA 达到同等灵巧程度，是值得探索的未来方向（当前局限的详细讨论见 章节 6）。

5.3 参数高效微调

上一节中 OpenVLA 的完整微调为每个任务使用 8 块 A100 GPU 训练 5-15 小时（取决于数据集规模），取得了较高性能。尽管所需算力远低于 VLA 预训练，本节仍进一步探索算力和参数效率更高的微调方法，并研究其有效性。

具体而言，本文比较以下微调方法：**完整微调**，即如 章节 5.2 所述，在微调期间更新全部权重；**仅最后一层**，即仅微调 OpenVLA 的 Transformer 骨干网络最后一层和词元嵌入矩阵；**冻结视觉编码器**，即冻结视觉编码器并微调其他所有权重；**夹心式微调**，即解冻视觉编码器、词元嵌入矩阵和最后一层；以及**低秩适配（LoRA）**，即采用 Hu et al. [26] 提出的常用低秩适配技术，在模型所有线性层上应用多个秩值 r。

表 1: **参数高效微调评测**。LoRA 微调实现了最佳的性能—算力权衡，仅训练模型 1.4% 的参数即可达到完整微调的性能。在选定的 Franka-Tabletop 任务上，每种方法的平均成功率 ± 标准误根据 33 条试验轨迹计算（详见 表 8）。
*：使用 FSDP [77] 分片至 2 块 GPU。

策略	成功率	训练参数量 ($\times 10^6$)	显存（批量 16）
完整微调	**69.7 ± 7.2 %**	7,188.1	163.3 GB*
仅最后一层	30.3 ± 6.1 %	465.1	51.4 GB
冻结视觉编码器	47.0 ± 6.9 %	6,760.4	156.2 GB*
夹心式微调	62.1 ± 7.9 %	914.2	64.0 GB
LoRA，秩 =32	**68.2 ± 7.5%**	**97.6**	**59.7 GB**
秩 =64	**68.2 ± 7.8%**	195.2	60.5 GB

表 1 给出了多个 Franka-Tabletop 任务上的微调成功率、训练参数量和 GPU 显存需求。[4]仅微调网络最后一层或冻结视觉编码器都会导致性能不佳，表明进一步使视觉特征适配目标场景至关重要。相比之下，“夹心式微调”会微调视觉编码器，因此性能更好；由于不微调整个大语言模型（Large Language Model, LLM）骨干网络，其 GPU 显存占用也更少。LoRA 在性能与训练显存占用之间实现了最佳权衡：它优于“夹心式微调”，且仅微调 1.4% 的参数便达到完整微调的性能。LoRA 的秩对策略性能影响可忽略不计，因此建议使用默认秩 $r = 32$。借助 LoRA，仅用一块 A100 GPU 即可在 10-15 小时内将 OpenVLA 微调至新任务，算力需求较完整微调降低 8 倍。

方法均采用相对位置控制。Diffusion Policy (matched) 以单张图像作为输入，不使用本体感知信息或观测历史，且不采用动作分块，仅预测一个相对位置控制动作。

[4]在 章节 5.3 和 章节 5.4 中，本文实验使用的 OpenVLA 模型版本采用较小的机器人数据混合进行预训练（与 Octo 使用相同的 OpenX 数据集混合），其架构也略小，仅使用 SigLIP [79] 视觉骨干网络，而非融合的 DinoSigLIP 编码器。实验发现，这一较简单的架构在微调任务和“开箱即用”任务上仍表现强劲。

5.4 通过量化实现显存高效推理

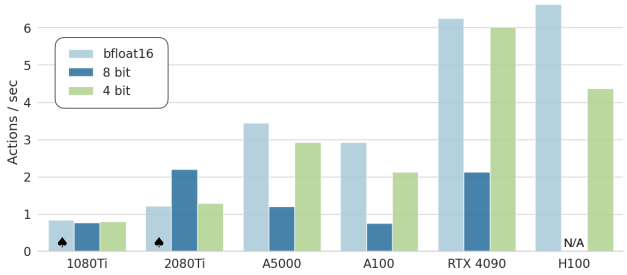


图 6: **OpenVLA 在不同 GPU 上的推理速度**。bfloat16 和 int4 量化均可实现高吞吐量，在采用 Ada Lovelace 架构的 GPU（RTX 4090、H100）上尤为明显。使用 TensorRT-LLM [89] 等现代 LLM 推理框架还可进一步提速。♠：模型分片至两块 GPU 以容纳模型。

精度	Bridge 成功率	显存
bfloat16	71.3 ± 4.8%	16.8 GB
int8	58.1 ± 5.1%	10.2 GB
int4	71.9 ± 4.7%	7.0 GB

表 2: **量化推理性能**。4 位量化达到 bfloat16 推理（本文默认方法）的性能，同时将 GPU 显存占用减少一半以上。在 8 个代表性 BridgeData V2 任务 [6] 上，每种方法的平均成功率 ± 标准误根据 80 条试验轨迹计算（详见 表 5）。

OpenVLA 是一个 7B 参数模型，其推理显存占用高于 Octo 等以往开源通用策略；Octo 的参数量为 <100M。遵循 LLM 服务的最佳实践，本文以 `bfloat16` 精度保存并加载 OpenVLA 进行推理（默认方法），将显存占用减半，使 OpenVLA 可在仅有 16GB 显存的 GPU 上运行。本节使用为 LLM 服务开发的现代量化技术 [27, 88]，测试能否进一步降低策略推理所需显存并提高 VLA 策略的可用性。这些方法以较低精度加载网络权重，以潜在的推理速度和准确率下降换取更低的显存需求。

具体而言，本文在 8 个代表性 BridgeData V2 任务上研究以 8 位和 4 位精度部署 OpenVLA 模型。显存占用和试验轨迹性能见 表 2；不同消费级和服务器级 GPU 可实现的控制频率见 图 6。由于额外量化操作带来的开销，8 位量化会降低多数 GPU 上的推理速度。4 位推理的吞吐量更高，因为 GPU 显存传输量的减少抵消了量化开销。

推理速度降低使 8 位量化的性能显著下降：在评测所用的 A5000 GPU 上，模型只能以 1.2Hz 运行；与 BridgeData V2 任务所用 5Hz 非阻塞控制器的训练数据集相比，这显著改变了系统动力学。[5]尽管 4 位量化所需 GPU 显存不到一半，其性能仍与 bfloat16 半精度推理相近。4 位量化模型可在 A5000 上以 3Hz 运行，因此更接近数据采集时的系统动力学。

6 讨论与局限性

本文提出了当前最佳的开源视觉—语言—动作模型 OpenVLA，无需额外适配即可在跨本体机器人控制中取得较强性能。本文还证明，参数高效微调技术能够轻松将 OpenVLA 适配到新的机器人配置。

当前 OpenVLA 模型存在若干局限。首先，它仅支持单图像观测，而真实世界中的机器人配置具有异构性，可能包含多种感知输入 [5]。未来的重要方向是扩展 OpenVLA，使其支持多图像、本体感知输入与观测历史。采用在图像与文本交错数据上预训练的 VLM，或有助于实现这种灵活输入形式的 VLA 微调。

其次，提高 OpenVLA 的推理吞吐量，是将 VLA 用于 ALOHA [90] 等 50Hz 高频控制平台的关键。这也将支持在比本文任务更灵巧的双臂操作任务上测试 VLA。动作分块或推测解码 [91] 等推理时优化技术可能解决这一问题。

模型性能仍有提升空间。尽管 OpenVLA 优于以往通用策略，但在测试任务上尚未达到很高的可靠性，成功率通常低于 90%。

最后，由于计算资源有限，许多 VLA 设计问题仍未得到充分研究：基础 VLM 的规模如何影响 VLA 性能？在机器人动作预测数据与互联网规模视觉—语言数据上联合训练，能否显著提升 VLA 性能？哪些视觉特征最适合 VLA？本文希望 OpenVLA 模型与代码库的发布能够推动社区共同研究这些问题。

致谢

感谢丰田研究院为本研究提供大量资金与计算资源。感谢斯坦福基础模型研究中心提供额外计算资源，并感谢 Google DeepMind 为评估工作提供 RT-2-X API 的内测访问权限。本

[5]本文将性能下降归因于较低的推理速度，因为在训练数据上离线评测时，8 位和 4 位量化的词元准确率均与 bfloat16 推理相当。支持性细节见 章节 D.4。

研究还获得 Volkswagen、Physical Intelligence、美国海军研究办公室项目 N00014-22-1-2621 与 N00014-22-1-2293、美国国家科学基金会项目 IIS-2246811 以及 DARPA ANSR 的支持。

参考文献

[1] Open X-Embodiment Collaboration, A. Padalkar, A. Pooley, A. Jain, A. Bewley, A. Herzog, A. Irpan, A. Khazatsky, A. Rai, A. Singh, A. Brohan, A. Raffin, A. Wahid, B. Burgess-Limerick, B. Kim, B. Schölkopf, B. Ichter, C. Lu, C. Xu, C. Finn, C. Xu, C. Chi, C. Huang, C. Chan, C. Pan, C. Fu, C. Devin, D. Driess, D. Pathak, D. Shah, D. Büchler, D. Kalashnikov, D. Sadigh, E. Johns, F. Ceola, F. Xia, F. Stulp, G. Zhou, G. S. Sukhatme, G. Salhotra, G. Yan, G. Schiavi, H. Su, H.-S. Fang, H. Shi, H. B. Amor, H. I. Christensen, H. Furuta, H. Walke, H. Fang, I. Mordatch, I. Radosavovic, I. Leal, J. Liang, J. Kim, J. Schneider, J. Hsu, J. Bohg, J. Bingham, J. Wu, J. Wu, J. Luo, J. Gu, J. Tan, J. Oh, J. Malik, J. Tompson, J. Yang, J. J. Lim, J. Silvério, J. Han, K. Rao, K. Pertsch, K. Hausman, K. Go, K. Gopalakrishnan, K. Goldberg, K. Byrne, K. Oslund, K. Kawaharazuka, K. Zhang, K. Majd, K. Rana, K. Srinivasan, L. Y. Chen, L. Pinto, L. Tan, L. Ott, L. Lee, M. Tomizuka, M. Du, M. Ahn, M. Zhang, M. Ding, M. K. Srirama, M. Sharma, M. J. Kim, N. Kanazawa, N. Hansen, N. Heess, N. J. Joshi, N. Suenderhauf, N. D. Palo, N. M. M. Shafiullah, O. Mees, O. Kroemer, P. R. Sanketi, P. Wohlhart, P. Xu, P. Sermanet, P. Sundaresan, Q. Vuong, R. Rafailov, R. Tian, R. Doshi, R. Martín-Martín, R. Mendonca, R. Shah, R. Hoque, R. Julian, S. Bustamante, S. Kirmani, S. Levine, S. Moore, S. Bahl, S. Dass, S. Song, S. Xu, S. Haldar, S. Adebola, S. Guist, S. Nasiriany, S. Schaal, S. Welker, S. Tian, S. Dasari, S. Belkhale, T. Osa, T. Harada, T. Matsushima, T. Xiao, T. Yu, T. Ding, T. Davchev, T. Z. Zhao, T. Armstrong, T. Darrell, V. Jain, V. Vanhoucke, W. Zhan, W. Zhou, W. Burgard, X. Chen, X. Wang, X. Zhu, X. Li, Y. Lu, Y. Chebotar, Y. Zhou, Y. Zhu, Y. Xu, Y. Wang, Y. Bisk, Y. Cho, Y. Lee, Y. Cui, Y. hua Wu, Y. Tang, Y. Zhu, Y. Li, Y. Iwasawa, Y. Matsuo, Z. Xu, and Z. J. Cui. Open X-Embodiment: Robotic learning datasets and RT-X models. https://arxiv.org/abs/2310.08864, 2023.

[2] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, T. Jackson, S. Jesmonth, N. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, K.-H. Lee, S. Levine, Y. Lu, U. Malla, D. Manjunath, I. Mordatch, O. Nachum, C. Parada, J. Peralta, E. Perez, K. Pertsch, J. Quiambao, K. Rao, M. Ryoo, G. Salazar, P. Sanketi, K. Sayed, J. Singh, S. Sontakke, A. Stone, C. Tan, H. Tran, V. Vanhoucke, S. Vega, Q. Vuong, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitkovich. Rt-1: Robotics transformer for real-world control at scale. In *arXiv preprint arXiv:2212.06817*, 2022.

[3] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

[4] A. Xie, L. Lee, T. Xiao, and C. Finn. Decomposing the generalization gap in imitation learning for visual robotic manipulation. *arXiv preprint arXiv:2307.03659*, 2023.

[5] Octo Model Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, C. Xu, J. Luo, T. Kreiman, Y. Tan, D. Sadigh, C. Finn, and S. Levine. Octo: An open-source generalist robot policy. https://octo-models.github.io, 2023.

[6] H. Walke, K. Black, A. Lee, M. J. Kim, M. Du, C. Zheng, T. Zhao, P. Hansen-Estruch, Q. Vuong, A. He, V. Myers, K. Fang, C. Finn, and S. Levine. Bridgedata v2: A dataset for robot learning at scale, 2023.

[7] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, P. Florence, C. Fu, M. G. Arenas, K. Gopalakrishnan, K. Han, K. Hausman, A. Herzog, J. Hsu, B. Ichter, A. Irpan, N. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, L. Lee, T.-W. E. Lee, S. Levine, Y. Lu,

H. Michalewski, I. Mordatch, K. Pertsch, K. Rao, K. Reymann, M. Ryoo, G. Salazar, P. Sanketi, P. Sermanet, J. Singh, A. Singh, R. Soricut, H. Tran, V. Vanhoucke, Q. Vuong, A. Wahid, S. Welker, P. Wohlhart, J. Wu, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitkovich. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *arXiv preprint arXiv:2307.15818*, 2023.

[8] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, volume 139, pages 8748–8763, 2021.

[9] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer. Sigmoid loss for language image pre-training. In *International Conference on Computer Vision (ICCV)*, 2023.

[10] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[11] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K. Srirama, L. Y. Chen, K. Ellis, P. D. Fagan, J. Hejna, M. Itkina, M. Lepert, Y. J. Ma, P. T. Miller, J. Wu, S. Belkhale, S. Dass, H. Ha, A. Jain, A. Lee, Y. Lee, M. Memmel, S. Park, I. Radosavovic, K. Wang, A. Zhan, K. Black, C. Chi, K. B. Hatch, S. Lin, J. Lu, J. Mercat, A. Rehman, P. R. Sanketi, A. Sharma, C. Simpson, Q. Vuong, H. R. Walke, B. Wulfe, T. Xiao, J. H. Yang, A. Yavary, T. Z. Zhao, C. Agia, R. Baijal, M. G. Castro, D. Chen, Q. Chen, T. Chung, J. Drake, E. P. Foster, J. Gao, D. A. Herrera, M. Heo, K. Hsu, J. Hu, D. Jackson, C. Le, Y. Li, K. Lin, R. Lin, Z. Ma, A. Maddukuri, S. Mirchandani, D. Morton, T. Nguyen, A. O'Neill, R. Scalise, D. Seale, V. Son, S. Tian, E. Tran, A. E. Wang, Y. Wu, A. Xie, J. Yang, P. Yin, Y. Zhang, O. Bastani, G. Berseth, J. Bohg, K. Goldberg, A. Gupta, A. Gupta, D. Jayaraman, J. J. Lim, J. Malik, R. Martín-Martín, S. Ramamoorthy, D. Sadigh, S. Song, J. Wu, M. C. Yip, Y. Zhu, T. Kollar, S. Levine, and C. Finn. Droid: A large-scale in-the-wild robot manipulation dataset. 2024.

[12] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta. R3m: A universal visual representation for robot manipulation. In *CoRL*, 2022.

[13] S. Karamcheti, S. Nair, A. S. Chen, T. Kollar, C. Finn, D. Sadigh, and P. Liang. Language-driven representation learning for robotics. *ArXiv*, abs/2302.12766, 2023. URL https://api.semanticscholar.org/CorpusID:257205716.

[14] M. Shridhar, L. Manuelli, and D. Fox. Cliport: What and where pathways for robotic manipulation. In *Conference on robot learning*, pages 894–906. PMLR, 2022.

[15] A. Stone, T. Xiao, Y. Lu, K. Gopalakrishnan, K.-H. Lee, Q. Vuong, P. Wohlhart, B. Zitkovich, F. Xia, C. Finn, et al. Open-world object manipulation using pre-trained vision-language models. *arXiv preprint arXiv:2303.00905*, 2023.

[16] D. Driess, F. Xia, M. S. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023.

[17] A. S. et al. Introducing rfm-1: Giving robots human-like reasoning capabilities, 2024. URL https://covariant.ai/insights/introducing-rfm-1-giving-robots-human-like-reasoning-capabilities/.

[18] Wayve. Lingo-2: Driving with natural language. 2024. URL https://wayve.ai/thinking/lingo-2-driving-with-language/.

[19] X. Chen, X. Wang, S. Changpinyo, A. J. Piergiovanni, P. Padlewski, D. M. Salz, S. Goodman, A. Grycner, B. Mustafa, L. Beyer, A. Kolesnikov, J. Puigcerver, N. Ding, K. Rong, H. Akbari, G. Mishra, L. Xue, A. V. Thapliyal, J. Bradbury, W. Kuo, M. Seyedhosseini, C. Jia, B. K. Ayan, C. Riquelme, A. Steiner, A. Angelova, X. Zhai, N. Houlsby, and R. Soricut. Pali: A jointly-scaled multilingual language-image model. *ArXiv*, abs/2209.06794, 2022. URL https://api.semanticscholar.org/CorpusID:252222320.

[20] X. Chen, X. Wang, L. Beyer, A. Kolesnikov, J. Wu, P. Voigtlaender, B. Mustafa, S. Goodman, I. M. Alabdulmohsin, P. Padlewski, D. M. Salz, X. Xiong, D. Vlasic, F. Pavetic, K. Rong, T. Yu, D. Keysers, X.-Q. Zhai, and R. Soricut. PaLI-3 vision language models: Smaller, faster, stronger. *arXiv preprint arXiv:2310.09199*, 2023.

[21] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, and ... Transformers: State-of-the-art natural language processing. In *Proceedings of the 6th International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/1910.03771.

[22] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[23] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. d. l. Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

[24] G. Team, T. Mesnard, C. Hardin, R. Dadashi, S. Bhupatiraju, S. Pathak, L. Sifre, M. Rivière, M. S. Kale, J. Love, et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

[25] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[26] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[27] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36, 2024.

[28] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.

[29] D. A. Hudson and C. D. Manning. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *Computer Vision and Pattern Recognition (CVPR)*, 2019.

[30] A. Singh, V. Natarajan, M. Shah, Y. Jiang, X. Chen, D. Batra, D. Parikh, and M. Rohrbach. Towards VQA models that can read. In *Computer Vision and Pattern Recognition (CVPR)*, 2019.

[31] J. P. Bigham, C. Jayant, H. Ji, G. Little, A. Miller, R. C. Miller, R. Miller, A. Tatarowicz, B. White, S. White, and T. Yeh. VizWiz: nearly real-time answers to visual questions. In *User Interface Software and Technology (UIST)*, pages 333–342, 2010.

[32] S. Kazemzadeh, V. Ordonez, M. Matten, and T. Berg. ReferItGame: Referring to objects in photographs of natural scenes. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 787–798, 2014.

[33] L. Yu, P. Poirson, S. Yang, A. C. Berg, and T. L. Berg. Modeling context in referring expressions. In *European Conference on Computer Vision (ECCV)*, 2016.

[34] T. Mesnard, C. Hardin, R. Dadashi, S. Bhupatiraju, S. Pathak, L. Sifre, M. Rivière, M. S. Kale, J. Love, P. Tafti, L. Hussenot, P. G. Sessa, A. Chowdhery, A. Roberts, A. Barua, A. Botev, A. Castro-Ros, A. Slone, A. Héliou, A. Tacchetti, A. Bulanova, A. Paterson, B. Tsai, B. Shahriari, C. L. Lan, C. A. Choquette-Choo, C. Crepy, D. Cer, D. Ippolito, D. Reid, E. Buchatskaya, E. Ni, E. Noland, G. Yan, G. Tucker, G.-C. Muraru, G. Rozhdestvenskiy, H. Michalewski, I. Tenney, I. Grishchenko, J. Austin, J. Keeling, J. Labanowski, J.-B. Lespiau, J. Stanway, J. Brennan, J. Chen, J. Ferret, J. Chiu, J. Mao-Jones, K. Lee, K. Yu, K. Millican, L. L. Sjoesund, L. Lee, L. Dixon, M. Reid, M. Mikuła, M. Wirth, M. Sharman, N. Chinaev, N. Thain, O. Bachem, O. Chang, O. Wahltinez, P. Bailey, P. Michel, P. Yotov, R. Chaabouni, R. Comanescu, R. Jana, R. Anil, R. McIlroy, R. Liu, R. Mullins, S. L. Smith, S. Borgeaud, S. Girgin, S. Douglas, S. Pandya, S. Shakeri, S. De, T. Klimenko, T. Hennigan, V. Feinberg, W. Stokowiec, Y. hui Chen, Z. Ahmed, Z. Gong, T. Warkentin, L. Peran, M. Giang, C. Farabet, O. Vinyals, J. Dean, K. Kavukcuoglu, D. Hassabis, Z. Ghahramani, D. Eck, J. Barral, F. Pereira, E. Collins, A. Joulin, N. Fiedel, E. Senter, A. Andreev, and K. Kenealy. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

[35] Y. Li, S. Bubeck, R. Eldan, A. D. Giorno, S. Gunasekar, and Y. T. Lee. Textbooks are all you need ii: phi-1.5 technical report. *arXiv preprint arXiv:2309.05463*, 2023.

[36] J. Bai, S. Bai, Y. Chu, Z. Cui, K. Dang, X. Deng, Y. Fan, W. Ge, Y. Han, F. Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

[37] J. Li, D. Li, C. Xiong, and S. C. H. Hoi. BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning (ICML)*, 2022.

[38] J. Li, D. Li, S. Savarese, and S. C. H. Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International Conference on Machine Learning (ICML)*, 2023.

[39] W. Dai, J. Li, D. Li, A. M. H. Tiong, J. Zhao, W. Wang, B. A. Li, P. Fung, and S. C. H. Hoi. InstructBLIP: Towards general-purpose vision-language models with instruction tuning. *arXiv preprint arXiv:2305.06500*, 2023.

[40] H. H. Tan and M. Bansal. LXMERT: Learning cross-modality encoder representations from transformers. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2019.

[41] H. Laurençon, L. Saulnier, L. Tronchon, S. Bekman, A. Singh, A. Lozhkov, T. Wang, S. Karamcheti, A. M. Rush, D. Kiela, M. Cord, and V. Sanh. OBELICS: An open web-scale filtered dataset of interleaved image-text documents. In *Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS Datasets and Benchmarks)*, 2023.

[42] H. Liu, C. Li, Q. Wu, and Y. J. Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[43] H. Liu, C. Li, Y. Li, and Y. J. Lee. Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744*, 2023.

[44] S. Karamcheti, S. Nair, A. Balakrishna, P. Liang, T. Kollar, and D. Sadigh. Prismatic vlms: Investigating the design space of visually-conditioned language models. *arXiv preprint arXiv:2402.07865*, 2024.

[45] C. Devin, A. Gupta, T. Darrell, P. Abbeel, and S. Levine. Learning modular neural network policies for multi-task and multi-robot transfer. In *Proceedings of IEEE International Conference on Robotics and Automation*, 2017.

[46] S. Dasari, F. Ebert, S. Tian, S. Nair, B. Bucher, K. Schmeckpeper, S. Singh, S. Levine, and C. Finn. Robonet: Large-scale multi-robot learning. *CoRL*, 2019.

[47] E. S. Hu, K. Huang, O. Rybkin, and D. Jayaraman. Know thyself: Transferable visual control policies through robot-awareness. In *International Conference on Learning Representations*, 2022.

[48] J. H. Yang, D. Sadigh, and C. Finn. Polybot: Training one policy across robots while embracing variability. In *7th Annual Conference on Robot Learning*, 2023. URL https://openreview.net/forum?id=HEIRj51lcS.

[49] S. Reed, K. Zolna, E. Parisotto, S. G. Colmenarejo, A. Novikov, G. Barth-maron, M. Giménez, Y. Sulsky, J. Kay, J. T. Springenberg, T. Eccles, J. Bruce, A. Razavi, A. Edwards, N. Heess, Y. Chen, R. Hadsell, O. Vinyals, M. Bordbar, and N. de Freitas. A generalist agent. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856.

[50] G. Salhotra, I.-C. A. Liu, and G. Sukhatme. Bridging action space mismatch in learning from demonstrations. *arXiv preprint arXiv:2304.03833*, 2023.

[51] I. Radosavovic, B. Shi, L. Fu, K. Goldberg, T. Darrell, and J. Malik. Robot learning with sensorimotor pre-training. In *Conference on Robot Learning*, 2023.

[52] D. Shah, A. Sridhar, A. Bhorkar, N. Hirose, and S. Levine. Gnm: A general navigation model to drive any robot. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7226–7233. IEEE, 2023.

[53] K. Bousmalis, G. Vezzani, D. Rao, C. Devin, A. X. Lee, M. Bauza, T. Davchev, Y. Zhou, A. Gupta, A. Raju, et al. Robocat: A self-improving foundation agent for robotic manipulation. *arXiv preprint arXiv:2306.11706*, 2023.

[54] D. Shah, A. Sridhar, N. Dashora, K. Stachowicz, K. Black, N. Hirose, and S. Levine. ViNT: A foundation model for visual navigation. In *7th Annual Conference on Robot Learning*, 2023. URL https://arxiv.org/abs/2306.14846.

[55] J. Yang, C. Glossop, A. Bhorkar, D. Shah, Q. Vuong, C. Finn, D. Sadigh, and S. Levine. Pushing the limits of cross-embodiment learning for manipulation and navigation. *arXiv preprint arXiv:2402.19432*, 2024.

[56] L. Pinto and A. Gupta. Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 3406–3413. IEEE, 2016.

[57] A. Mandlekar, Y. Zhu, A. Garg, J. Booher, M. Spero, A. Tung, J. Gao, J. Emmons, A. Gupta, E. Orbay, et al. Roboturk: A crowdsourcing platform for robotic skill learning through imitation. In *Conference on Robot Learning*, pages 879–893. PMLR, 2018.

[58] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, et al. QT-Opt: Scalable deep reinforcement learning for vision-based robotic manipulation. *arXiv preprint arXiv:1806.10293*, 2018.

[59] A. Gupta, A. Murali, D. P. Gandhi, and L. Pinto. Robot learning in homes: Improving generalization and reducing dataset bias. *Advances in neural information processing systems*, 31, 2018.

[60] S. Cabi, S. G. Colmenarejo, A. Novikov, K. Konyushkova, S. Reed, R. Jeong, K. Zolna, Y. Aytar, D. Budden, M. Vecerik, O. Sushkov, D. Barker, J. Scholz, M. Denil, N. de Freitas, and Z. Wang. Scaling data-driven robotics with reward sketching and batch reinforcement learning. *RSS*, 2019.

[61] E. Jang, A. Irpan, M. Khansari, D. Kappler, F. Ebert, C. Lynch, S. Levine, and C. Finn. Bc-z: Zero-shot task generalization with robotic imitation learning. In *Conference on Robot Learning*, pages 991–1002. PMLR, 2022.

[62] H.-S. Fang, H. Fang, Z. Tang, J. Liu, C. Wang, J. Wang, H. Zhu, and C. Lu. Rh20t: A comprehensive robotic dataset for learning diverse skills in one-shot. *Towards Generalist Robots: Learning Paradigms for Scalable Skill Acquisition@ CoRL2023*, 3:5, 2023.

[63] H. Bharadhwaj, J. Vakil, M. Sharma, A. Gupta, S. Tulsiani, and V. Kumar. Roboagent: Generalization and efficiency in robot manipulation via semantic augmentations and action chunking. *arXiv preprint arXiv:2309.01918*, 2023.

[64] D. Kalashnkov, J. Varley, Y. Chebotar, B. Swanson, R. Jonschkowski, C. Finn, S. Levine, and K. Hausman. Mt-opt: Continuous multi-task robotic reinforcement learning at scale. *arXiv*, 2021.

[65] F. Ebert, Y. Yang, K. Schmeckpeper, B. Bucher, G. Georgakis, K. Daniilidis, C. Finn, and S. Levine. Bridge data: Boosting generalization of robotic skills with cross-domain datasets. *arXiv preprint arXiv:2109.13396*, 2021.

[66] K. Ehsani, T. Gupta, R. Hendrix, J. Salvador, L. Weihs, K.-H. Zeng, K. P. Singh, Y. Kim, W. Han, A. Herrasti, et al. Imitating shortest paths in simulation enables effective navigation and manipulation in the real world. *arXiv preprint arXiv:2312.02976*, 2023.

[67] S. Y. Gadre, M. Wortsman, G. Ilharco, L. Schmidt, and S. Song. Cows on pasture: Baselines and benchmarks for language-driven zero-shot object navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23171–23181, 2023.

[68] Y. Du, K. Konyushkova, M. Denil, A. Raju, J. Landon, F. Hill, N. de Freitas, and S. Cabi. Vision-language models as success detectors. *arXiv preprint arXiv:2303.07280*, 2023.

[69] Y. J. Ma, V. Kumar, A. Zhang, O. Bastani, and D. Jayaraman. Liv: Language-image representations and rewards for robotic control. In *International Conference on Machine Learning*, pages 23301–23320. PMLR, 2023.

[70] X. Zhang, Y. Ding, S. Amiri, H. Yang, A. Kaminski, C. Esselink, and S. Zhang. Grounding classical task planners via vision-language models. *arXiv preprint arXiv:2304.08587*, 2023.

[71] S. Sontakke, J. Zhang, S. Arnold, K. Pertsch, E. Bıyık, D. Sadigh, C. Finn, and L. Itti. Roboclip: One demonstration is enough to learn robot policies. *Advances in Neural Information Processing Systems*, 36, 2024.

[72] J. Huang, S. Yong, X. Ma, X. Linghu, P. Li, Y. Wang, Q. Li, S.-C. Zhu, B. Jia, and S. Huang. An embodied generalist agent in 3d world. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2024.

[73] X. Li, M. Liu, H. Zhang, C. Yu, J. Xu, H. Wu, C. Cheang, Y. Jing, W. Zhang, H. Liu, et al. Vision-language foundation models as effective robot imitators. *arXiv preprint arXiv:2311.01378*, 2023.

[74] H. Zhen, X. Qiu, P. Chen, J. Yang, X. Yan, Y. Du, Y. Hong, and C. Gan. 3d-vla: 3d vision-language-action generative world model. *arXiv preprint arXiv:2403.09631*, 2024.

[75] PyTorch. Automatic mixed precision. URL https://pytorch.org/docs/stable/amp.html.

[76] T. Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023.

[77] Y. Zhao, A. Gu, R. Varma, L. Luo, C.-C. Huang, M. Xu, L. Wright, H. Shojanazeri, M. Ott, S. Shleifer, et al. Pytorch fsdp: experiences on scaling fully sharded data parallel. *arXiv preprint arXiv:2304.11277*, 2023.

[78] N. Dorka, C. Huang, T. Welschehold, and W. Burgard. What matters in employing vision language models for tokenizing actions in robot control? In *First Workshop on Vision-Language Models for Navigation and Manipulation at ICRA 2024*.

[79] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11975–11986, 2023.

[80] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[81] P. Sharma, N. Ding, S. Goodman, and R. Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, 2018.

[82] C. Schuhmann, R. Vencu, R. Beaumont, R. Kaczmarczyk, C. Mullis, A. Katta, T. Coombes, J. Jitsev, and A. Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021.

[83] O. Sidorov, R. Hu, M. Rohrbach, and A. Singh. Textcaps: a dataset for image captioning with reading comprehension. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 742–758. Springer, 2020.

[84] H. Face. Introducing idefics: An open reproduction of state-of-the-art visual langage model. *Hugging Face Blog*, 2024.

[85] H. Liu, C. Li, Q. Wu, and Y. J. Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024.

[86] B. McKinzie, Z. Gan, J.-P. Fauconnier, S. Dodge, B. Zhang, P. Dufter, D. Shah, X. Du, F. Peng, F. Weers, et al. Mm1: Methods, analysis & insights from multimodal llm pre-training. *arXiv preprint arXiv:2403.09611*, 2024.

[87] J. Lin, H. Yin, W. Ping, Y. Lu, P. Molchanov, A. Tao, H. Mao, J. Kautz, M. Shoeybi, and S. Han. Vila: On pre-training for visual language models. *arXiv preprint arXiv:2312.07533*, 2023.

[88] T. Dettmers, M. Lewis, Y. Belkada, and L. Zettlemoyer. Gpt3. int8 (): 8-bit matrix multiplication for transformers at scale. *Advances in Neural Information Processing Systems*, 35:30318–30332, 2022.

[89] NVIDIA. Tensorrt-llm. URL https://github.com/NVIDIA/TensorRT-LLM.

[90] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.

[91] Y. Leviathan, M. Kalman, and Y. Matias. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*, pages 19274–19286. PMLR, 2023.

[92] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

[93] E. Rosete-Beas, O. Mees, G. Kalweit, J. Boedecker, and W. Burgard. Latent plans for task agnostic offline reinforcement learning. In *Proceedings of the 6th Conference on Robot Learning (CoRL)*, 2022.

[94] O. Mees, J. Borja-Diaz, and W. Burgard. Grounding language with visual affordances over unstructured data. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, London, UK, 2023.

[95] S. Dass, J. Yapeter, J. Zhang, J. Zhang, K. Pertsch, S. Nikolaidis, and J. J. Lim. CLVR jaco play dataset, 2023. URL https://github.com/clvrai/clvr_jaco_play_dataset.

[96] J. Luo, C. Xu, X. Geng, G. Feng, K. Fang, L. Tan, S. Schaal, and S. Levine. Multi-stage cable routing through hierarchical imitation learning. *arXiv preprint arXiv:2307.08927*, 2023.

[97] A. Mandlekar, Y. Zhu, A. Garg, J. Booher, M. Spero, A. Tung, J. Gao, J. Emmons, A. Gupta, E. Orbay, S. Savarese, and L. Fei-Fei. RoboTurk: A crowdsourcing platform for robotic skill learning through imitation. *CoRR*, abs/1811.02790, 2018. URL http://arxiv.org/abs/1811.02790.

[98] Y. Zhu, A. Joshi, P. Stone, and Y. Zhu. Viola: Imitation learning for vision-based manipulation with object proposal priors, 2023.

[99] L. Y. Chen, S. Adebola, and K. Goldberg. Berkeley UR5 demonstration dataset. https://sites.google.com/view/berkeley-ur5/home.

[100] G. Zhou, V. Dean, M. K. Srirama, A. Rajeswaran, J. Pari, K. Hatch, A. Jain, T. Yu, P. Abbeel, L. Pinto, C. Finn, and A. Gupta. Train offline, test online: A real robot learning benchmark, 2023.

[101] C. Lynch, A. Wahid, J. Tompson, T. Ding, J. Betker, R. Baruch, T. Armstrong, and P. Florence. Interactive language: Talking to robots in real time. *IEEE Robotics and Automation Letters*, 2023.

[102] S. Belkhale, Y. Cui, and D. Sadigh. Hydra: Hybrid robot actions for imitation learning. *arxiv*, 2023.

[103] Y. Zhu, P. Stone, and Y. Zhu. Bottom-up skill discovery from unsegmented demonstrations for long-horizon robot manipulation. *IEEE Robotics and Automation Letters*, 7(2):4126–4133, 2022.

[104] Z. J. Cui, Y. Wang, N. M. M. Shafiullah, and L. Pinto. From play to policy: Conditional behavior generation from uncurated robot data. *arXiv preprint arXiv:2210.10047*, 2022.

[105] M. Heo, Y. Lee, D. Lee, and J. J. Lim. Furniturebench: Reproducible real-world benchmark for long-horizon complex manipulation. In *Robotics: Science and Systems*, 2023.

[106] G. Yan, K. Wu, and X. Wang. ucsd kitchens Dataset. August 2023.

[107] S. Nasiriany, T. Gao, A. Mandlekar, and Y. Zhu. Learning and retrieval from prior data for skill-based imitation learning. In *Conference on Robot Learning (CoRL)*, 2022.

[108] H. Liu, S. Nasiriany, L. Zhang, Z. Bao, and Y. Zhu. Robot learning on the job: Human-in-the-loop autonomy and learning during deployment. In *Robotics: Science and Systems (RSS)*, 2023.

[109] G. Quere, A. Hagengruber, M. Iskandar, S. Bustamante, D. Leidner, F. Stulp, and J. Vogel. Shared Control Templates for Assistive Robotics. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, page 7, Paris, France, 2020.

[110] S. Saxena, M. Sharma, and O. Kroemer. Multi-resolution sensing for real-time control with vision-language models. In *7th Annual Conference on Robot Learning*, 2023. URL https://openreview.net/forum?id=WuBv9-IGDUA.

[111] R. Shah, R. Martín-Martín, and Y. Zhu. MUTEX: Learning unified policies from multimodal task specifications. In *7th Annual Conference on Robot Learning*, 2023. URL https://openreview.net/forum?id=PwqiqaaEzJ.

[112] X. Zhu, R. Tian, C. Xu, M. Ding, W. Zhan, and M. Tomizuka. Fanuc manipulation: A dataset for learning-based manipulation with fanuc mate 200id robot. 2023.

[113] R. Mendonca, S. Bahl, and D. Pathak. Structured world models from human videos. *CoRL*, 2023.

[114] J. Luo, C. Xu, F. Liu, L. Tan, Z. Lin, J. Wu, P. Abbeel, and S. Levine. Fmb: a functional manipulation benchmark for generalizable robotic learning. *arXiv preprint arXiv:2401.08553*, 2024.

[115] N. M. M. Shafiullah, A. Rai, H. Etukuru, Y. Liu, I. Misra, S. Chintala, and L. Pinto. On bringing robots home, 2023.

[116] B. Liu, Y. Zhu, C. Gao, Y. Feng, Q. Liu, Y. Zhu, and P. Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36, 2024.

[117] V. Sanh, L. Debut, J. Chaumond, and T. Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

A 数据混合细节

表 3列出了本文使用的数据混合方案。该方案大体遵循 [5]，并增加了少量数据集。

OpenVLA 训练数据集混合方案	
Fractal [92]	12.7%
Kuka [58]	12.7%
Bridge[6, 65]	13.3%
Taco Play [93, 94]	3.0%
Jaco Play [95]	0.4%
Berkeley Cable Routing [96]	0.2%
Roboturk [97]	2.3%
Viola [98]	0.9%
Berkeley Autolab UR5 [99]	1.2%
Toto [100]	2.0%
Language Table [101]	4.4%
Stanford Hydra Dataset [102]	4.4%
Austin Buds Dataset [103]	0.2%
NYU Franka Play Dataset [104]	0.8%
Furniture Bench Dataset [105]	2.4%
UCSD Kitchen Dataset [106]	<0.1%
Austin Sailor Dataset [107]	2.2%
Austin Sirius Dataset [108]	1.7%
DLR EDAN Shared Control [109]	<0.1%
IAMLab CMU Pickup Insert [110]	0.9%
UTAustin Mutex [111]	2.2%
Berkeley Fanuc Manipulation [112]	0.7%
CMU Stretch [113]	0.2%
BC-Z [61]	7.5%
FMB Dataset [114]	7.1%
DobbE [115]	1.4%
DROID [11]	10.0%[6]

表 3: OpenVLA 训练数据混合方案。数据取自 Open X-Embodiment 数据集 [1]，整体遵循 [5] 并加入少量额外数据集。

B 评估任务与详细结果

本节进一步介绍章节 5.1中的 BridgeData V2 WidowX 和 Google 机器人评估，以及章节 5.2中的 Franka-Tabletop 与 Franka-DROID 微调评估。

B.1 BridgeData V2 WidowX 评估细节

本节专门讨论章节 5.1中的 BridgeData V2 评估。

B.1.1 BridgeData V2 评估任务

如章节 5.1所述，我们在 17 项任务上评估各通用机器人操控策略，每项任务进行 10 次试验。本节介绍任务类别及各项任务的细节。

评估共包括 5 项视觉泛化任务、2 项运动泛化任务、3 项物理泛化任务、4 项语义泛化任务和 3 项语言落地任务。由于无法获得原始数据集使用的完全相同物体，所有评估任务都引入了某种分布偏移；在不同地点复现实物测试环境时还会自然产生其他分布偏移，详见章节 B.1.2。全部 17 项任务如图 7所示。每条试验轨迹记为失败（0）或成功（1）；部分较难任务还记录部分成功（0.5），其得分条件见下文任务说明。

下面按图 7中的顺序介绍 17 项任务：

1. **将茄子放入锅中（简单版）**：机器人需拿起茄子并将其放入锅中。这是一项**视觉泛化**任务，因为无法取得原始锅具，故使用外观不同于 BridgeData V2 原始训练数据集

[6] 由于学习进展缓慢（见章节 3.3），训练最后三分之一阶段移除 DROID，并将其混合权重重新分配给其他所有数据集。

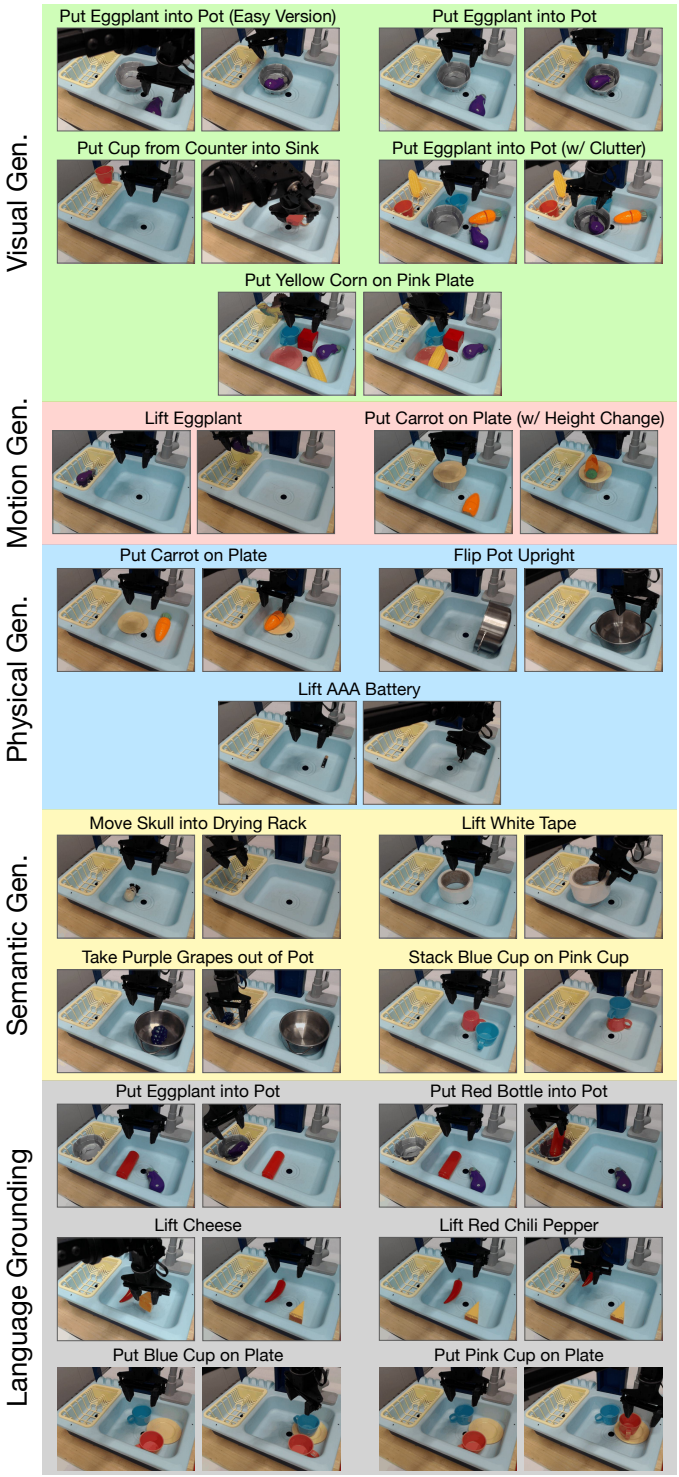


图 7: **BridgeData V2 WidowX 机器人评估任务**。我们在四类分布外（Out-of-Distribution，OOD）泛化任务上评估各通用机器人策略：**视觉**、**运动**、**物理**和**语义**泛化（定义见章节 5.1）。每对图像分别展示起始状态，以及机器人完成任务后的一个示例结束状态。最下方三行任务还严格评估**语言落地**能力：保持初始状态不变而改变提示，以检验策略能否接近正确的目标物体。

所用锅具的手工纸锅。与其余 16 项任务不同，该任务在运行策略前将机器人末端执行器直接初始化到茄子上方，因此称为“将茄子放入锅中”的“简单版”。

2. **将茄子放入锅中**：任务与上项相同，但末端执行器并不直接初始化在茄子上方，而是在所有试验轨迹中采用同一固定位置，因此机器人必须先水平伸向茄子才能进行操控（下述所有其他任务同样如此）。基于与上项相同的原因，这是一项**视觉泛化**任务。
3. **将台面上的杯子放入水槽**：机器人需从厨房台面或沥水架拿起粉色杯子，并放入右侧水槽。这是一项**视觉泛化**任务，因为此处使用粉色杯子而非原始 BridgeData V2 数据集中的蓝色杯子；实验发现所有受评方法都无法可靠操控蓝色杯子，最可能的原因是其颜色与水槽颜色混在一起。
4. **将茄子放入锅中（含杂物）**：任务与“将茄子放入锅中”相同，但多个干扰物体提高了难度。基于普通版本所述原因，这是一项**视觉泛化**任务；场景中未见过的干扰物进一步加剧了视觉分布偏移。当机器人朝正确目标物体移动时，计部分得分（*1* 分中的 *0.5* 分）。
5. **将黄色玉米放到粉色盘子上**：机器人需拿起黄色玉米并放到粉色盘子上。由于场景中存在未见过的干扰物体，例如水槽后部台面上的绿色恐龙，这是一项**视觉泛化**任务。当机器人朝正确目标物体移动时，计部分得分（*1* 分中的 *0.5* 分）。
6. **提起茄子**：机器人需抓住茄子并将其提到空中。这是一项**运动泛化**任务：茄子的初始位置和/或朝向未在训练中出现，机器人必须超出训练位置和/或朝向分布，且常需远距离伸展才能完成任务。原始 BridgeData V2 示范未在该环境中展示远距离伸展，详见章节 B.1.2。该任务看似简单，却对许多策略颇具挑战。当机器人接触到茄子时，计部分得分（*1* 分中的 *0.5* 分）。
7. **将胡萝卜放到盘子上（高度变化）**：机器人需拿起胡萝卜并放到黄色盘子上。这是一项**运动泛化**任务，因为盘子从水槽底部的常规位置被抬高，机器人必须调整轨迹，将胡萝卜正确放到抬高的平台上且不能碰倒盘子。当机器人抓住胡萝卜并用其接触盘子时，计部分得分（*1* 分中的 *0.5* 分）。
8. **将胡萝卜放到盘子上**：任务与上项相同，但盘子位于常规位置，即水槽或沥水架底部。由于所用胡萝卜的尺寸和形状不同于原始 BridgeData V2 数据集中更短、更细的胡萝卜，因此归为**物理泛化**任务。严格而言，上一个版本因使用相同胡萝卜也属于物理泛化，但其重点是运动泛化，故列入该类别。
9. **将锅翻正**：机器人需操控锅具，使其在回合结束时直立于水槽中。由于该锅具的尺寸和形状不同于 BridgeData V2 原始训练示范中的锅具（本文所用锅具更宽、更矮），这是一项**物理泛化**任务。
10. **提起 AAA 电池**：机器人只需抓住 AAA 电池并将其提到空中。该电池远小于且薄于 BridgeData V2 在此环境中的训练示范目标物体，因此归为**物理泛化**任务，详见章节 B.1.2。该目标物体也未出现在此环境的原始 BridgeData V2 示范中，因而也属于“语义泛化”，但此处重点是物理属性，故仅归为物理泛化。
11. **将骷髅移入沥水架**：机器人需抓住骷髅发条玩具，并将其放入水槽左侧的黄色沥水架。骷髅是未见过的目标物体，未出现在 BridgeData V2 训练示范中，因此这是一项**语义泛化**任务。
12. **提起白色胶带**：机器人需抓住白色胶带卷并将其提到空中。白色胶带卷是未见过的目标物体，未出现在 BridgeData V2 训练示范中，因此这是一项**语义泛化**任务。由于其形状不同于此环境训练示范中的物体，也可视为物理泛化；多数策略难以抓取这种环状结构，常将末端执行器直接移向中心区域。
13. **从锅中取出紫葡萄**：机器人需抓住钢锅内的紫葡萄并将其取出，即提起后放到锅外任意位置。由于该语言指令未曾出现，机器人在原始 BridgeData V2 训练数据集中从未见过此任务，因此这是一项**语义泛化**任务。
14. **将蓝色杯子叠放到粉色杯子上**：机器人需抓住蓝色杯子，并将其稳妥放到粉色杯子上。由于这条语言指令未曾出现，机器人在原始 BridgeData V2 训练数据集的该环境中从未见过此任务，因此这是一项**语义泛化**任务。当机器人抓住蓝色杯子并用其接触粉色杯子时，计部分得分（*1* 分中的 *0.5* 分）。

15. **将 {茄子、红瓶} 放入锅中**：这是一项**语言落地**任务。机器人需将指定目标物体放入锅中，场景中同时存在茄子和红瓶。采用成对评估：在相同初始状态下，一个回合提示策略以茄子为目标，下一个回合则以红瓶为目标。每种方法针对茄子和红瓶各测试 5 次，两种目标物体使用相同的 5 个初始状态。当机器人朝正确目标物体移动时，计部分得分（*1* 分中的 *0.5* 分）。
16. **提起 {奶酪、红辣椒}**：这是一项**语言落地**任务。机器人需抓住并提起指定目标物体，采用与上项任务相同的成对评估。当机器人朝正确目标物体移动时，计部分得分（*1* 分中的 *0.5* 分）。
17. **将 {蓝色杯子、粉色杯子} 放到盘子上**：这是一项**语言落地**任务。机器人需抓住指定目标物体并放到盘子上，采用与其他语言落地任务相同的成对评估。当机器人朝正确目标物体移动时，计部分得分（*1* 分中的 *0.5* 分）。

B.1.2 评估任务与原始 BridgeData V2 训练数据的比较

评估在原始 BridgeData V2 数据集 [6] 使用的水槽环境中进行。复现环境时，粗略估计机器人相对于水槽的位置以及相机相对于场景的位置，以匹配 BridgeData V2 中的原始环境。原始数据集未提供这些位置的精确测量值，因此无法完全复现环境配置；机器人、水槽和相机位置的细微差异会自然产生分布偏移。此外，机器人策略的评估地点不同于训练示范采集地点，也会产生其他自然分布偏移。例如，照明条件和背景（e.g., 水槽后方可见区域）必然不同于训练数据集。由于还无法获得原始 BridgeData V2 数据集所用的完全相同物体，训练与测试所用物体之间也存在分布偏移。

尽管存在这些挑战，OpenVLA 和 RT-2-X 等部分通用策略仍能泛化，并在无需额外调整的情况下较可靠地完成多种任务。RT-1-X 和 Octo 等其他通用策略也能完成部分任务，但面对 BridgeData V2 评估套件中难度更高的泛化任务时表现不佳。

原始 BridgeData V2 数据集在该水槽环境中包含以下七项任务的示范：“Flip Pot Upright”“Put Carrot on Plate”“Put Cup from Counter (or Drying Rack) into Sink”“Put Eggplant into Pot”“Put Knife on Cutting Board”“Put Spoon in Pot”和“Turn Lever Vertical to Front”。图 8展示了原始数据集中这些任务的示例图像。该环境中采集的所有训练示范都将机器人末端执行器初始化在目标物体正上方。不过，BridgeData V2 的其他环境并非全部如此；在某些环境中，机器人初始位置离目标物体较远，必须先水平伸向物体再进行操控。

图 8: **原始 BridgeData V2 水槽环境任务**。原始 BridgeData V2 数据集水槽环境中的示范样例表明，该环境所有示范的初始状态都将机器人末端执行器置于目标物体正上方。这些初始状态不同于图 7所示 BridgeData V2 评估任务所用状态。评估中始终将末端执行器初始化到水槽上方的固定位置，而非目标物体正上方；唯一例外是“Put Eggplant into Pot (Easy Version)”任务。

BridgeData V2 评估套件中，仅“Put Eggplant into Pot (Easy Version)”将机器人末端执行器初始化到目标物体正上方；其余 16 项任务均将末端执行器初始化到水槽上方的固定位

置，机器人必须水平伸向物体。该初始条件与评估套件各类 OOD 泛化所引入的分布偏移共同增加了通用策略的难度，策略需具备较强鲁棒性才能成功完成任务。因此，RT-1-X 和 Octo 等策略的成功率低于先前工作报告值。然而，面对这些分布偏移与挑战，RT-2-X 和 OpenVLA 等策略仍取得相对较强的性能。

B.1.3 BridgeData V2 详细评估结果

完整的 BridgeData V2 WidowX 评估结果见表 4。表中列出各方法在 17 项任务各 10 次试验中的成功次数。OpenVLA 在多数任务上表现最佳，并在通用策略中取得最高总体成功率。RT-2-X 也表现良好，优于 RT-1-X 和 Octo，但不及 OpenVLA；RT-1-X 和 Octo 通常难以应对这些泛化任务。

表 4: **BridgeData V2 WidowX 详细评估结果**。报告完整 17 项任务评估套件（见章节 5.1）的性能，包括视觉、运动、物理和语义泛化任务以及语言落地任务。部分任务允许部分成功（0.5 分），详见章节 B.1.1。OpenVLA 在多数任务上表现最佳，总体性能最高，RT-2-X 次之；RT-1-X 和 Octo 在评估中表现较弱，多项任务仅成功 0–2 次。全部任务图示见图 7。

类别	任务	试验次数	RT-1-X 成功次数	Octo 成功次数	RT-2-X 成功次数	OpenVLA （本文）成功次数
视觉泛化	Put Eggplant into Pot (Easy Version)	10	1	5	7	**10**
视觉泛化	Put Eggplant into Pot	10	0	1	5	**10**
视觉泛化	Put Cup from Counter into Sink	10	1	1	0	**7**
视觉泛化	Put Eggplant into Pot (w/ Clutter)	10	1	3.5	6	**7.5**
视觉泛化	Put Yellow Corn on Pink Plate	10	1	4	8	**9**
运动泛化	Lift Eggplant	10	3	0.5	6.5	**7.5**
运动泛化	Put Carrot on Plate (w/ Height Change)	10	2	1	**4.5**	**4.5**
物理泛化	Put Carrot on Plate	10	1	0	1	**8**
物理泛化	Flip Pot Upright	10	2	6	5	**8**
物理泛化	Lift AAA Battery	10	0	0	2	**7**
语义泛化	Move Skull into Drying Rack	10	1	0	**5**	**5**
语义泛化	Lift White Tape	10	**3**	0	0	1
语义泛化	Take Purple Grapes out of Pot	10	**6**	0	5	4
语义泛化	Stack Blue Cup on Pink Cup	10	0.5	0	**5.5**	4.5
语言落地	Put {Eggplant, Red Bottle} into Pot	10	2.5	4	**8.5**	7.5
语言落地	Lift {Cheese, Red Chili Pepper}	10	1.5	2.5	8.5	**10**
语言落地	Put {Blue Cup, Pink Cup} on Plate	10	5	5.5	8.5	**9.5**
		平均成功率	18.5±2.7%	20.0±2.6%	50.6±3.5%	**70.6±3.2%**

此外，表 5给出了表 2所概述量化推理实验的完整评估结果。这些评估在 8 项有代表性的 BridgeData V2 任务上测试策略，覆盖完整评估套件中的所有任务类别。

表 5: **完整量化推理结果**。此处给出表 2中结果的详细版本。

类别	任务	试验次数	bfloat16 成功次数	int8 成功次数	int4 成功次数
视觉泛化	Put Eggplant into Pot (Easy Version)	10	**9**	7	**9**
视觉泛化	Put Eggplant into Pot	10	**7**	**7**	**7**
视觉泛化	Put Cup from Counter into Sink	10	5	3	**7**
运动泛化	Lift Eggplant	10	6	4	**7.5**
物理泛化	Put Carrot on Plate	10	6	5	**7**
物理泛化	Lift AAA Battery	10	**7**	5	3
语义泛化	Take Purple Grapes out of Pot	10	8	8	**9**
语言落地	Put {Eggplant, Red Bottle} into Pot	10	**9**	7.5	8
		平均成功率	**71.3 ± 4.8%**	58.1 ± 5.1%	**71.9 ± 4.7%**

B.2 Google 机器人评估细节

本节进一步介绍章节 5.1中的 Google 机器人评估。

B.2.1 Google 机器人评估任务

在 Google 机器人上，每种通用机器人策略均评估 12 项任务，每项运行 5 条试验轨迹，共 60 条。前五项任务测试分布内条件，后七项测试更困难的分布外（OOD）条件。全部任务如图 9所示，每条试验轨迹记为失败（0）或成功（1）。

下面介绍 12 项任务：

1. **拿起可乐罐**（分布内）：机器人位于平台前方，平台上放有一罐可乐；目标是抓住并提起可乐罐。

图 9: **Google 机器人评估任务**。在分布内任务和分布外（OOD）泛化任务上评估各通用机器人策略。OOD 任务涉及未见过的背景、目标物体、指令/物体关系和语义概念（例如未出现在机器人动作数据中的互联网照片）。

2. **将苹果移到绿色罐旁**（分布内）：平台上放有一个苹果和一个绿色汽水罐；机器人需抓住苹果并将其移到绿色罐旁。
3. **将蓝色薯片袋移到苹果旁**（分布内）：平台上放有蓝色薯片袋和苹果；机器人需抓住薯片袋并将其移到苹果附近。
4. **将可乐罐立正**（分布内）：平台上的可乐罐侧卧；机器人需抓住可乐罐并将其调整为直立姿态。
5. **打开中间抽屉**（分布内）：机器人位于一组三层抽屉前，需抓住中间抽屉的把手并将其拉开。
6. **将橙子移到棕色薯片袋旁**（OOD）：平台上放有棕色薯片袋和橙子，物体下方铺着蓝天白云图案的桌布。机器人需抓住橙子并移到薯片袋旁。该任务属于 OOD，因为橙子是训练数据集中未见过的物体，桌布也是未见过的背景。[7]
7. **拿起百事可乐罐**（OOD）：平台上放有一罐百事可乐，其下铺着亮黄/棕色图案桌布。机器人需抓住并提起罐子。百事可乐罐和桌布背景均未见过，因此该任务属于 OOD。
8. **拿起香蕉**（OOD）：平台上放有苹果、可乐罐和香蕉；机器人需抓住并提起香蕉。香蕉是未见过的目标物体，因此该任务属于 OOD。
9. **拿起绿色杯子**（OOD）：平台上放有香蕉、百事可乐罐和绿色杯子；机器人需抓住并提起绿色杯子。场景中所有物体都未出现在训练数据中，因此该任务属于 OOD。

[7]Google 机器人评估中 OOD 条件的详细列表见 Brohan et al. [7]附录。

10. **将苹果放到盘子上**（OOD）：平台上放有盘子和苹果；机器人需抓住苹果并移到盘子上。该任务使用描述未见物体关系的新指令，因而属于 OOD：训练示范只包含将苹果移到盘子附近，而非放在盘子上方。
11. **将香蕉放入平底锅**（OOD）：平台上放有平底锅和香蕉；机器人需抓住香蕉并将其移入锅中。香蕉是未见过的目标物体，且该新指令描述了未见过的物体关系，因此该任务属于 OOD。
12. **将可乐罐移到 Taylor Swift 照片旁**（OOD）：平台上放有可乐罐和三位不同名人的照片，其中包括 Taylor Swift。机器人需抓住罐子并移到 Taylor Swift 的照片旁。名人照片未出现在机器人交互数据中，因此该任务属于 OOD。

B.2.2 Google 机器人详细评估结果

表 6: **Google 机器人详细评估结果**。报告章节 5.1所述 Google 机器人评估的完整结果。每种通用策略跨 12 项任务评估 60 条试验轨迹，涵盖分布内和分布外（OOD）测试条件。底行报告各策略的平均成功率 ± 标准误差。OpenVLA 和 RT-2-X 总体均显著优于 RT-1-X 和 Octo；由于两者误差条重叠，其平均成功率均以粗体标出。全部任务图示见图 9。

类别	任务	试验次数	RT-1-X 成功次数	Octo 成功次数	RT-2-X 成功次数	OpenVLA（本文）成功次数
分布内	Pick Coke Can	5	**5**	1	**5**	**5**
分布内	Move Apple near Green Can	5	3	3	3	**5**
分布内	Move Blue Chip Bag near Apple	5	0	3	4	**5**
分布内	Place Coke Can Upright	5	0	0	**4**	**4**
分布内	Open Middle Drawer	5	0	**4**	2	3
OOD	Move Orange near Brown Chip Bag	5	1	2	**5**	**5**
OOD	Pick Pepsi Can	5	3	0	**5**	4
OOD	Pick Banana	5	**5**	3	**5**	**5**
OOD	Pick Green Cup	5	1	0	**5**	**5**
OOD	Place Apple on Plate	5	0	0	**4**	**4**
OOD	Place Banana in Pan	5	0	0	2	**4**
OOD	Move Coke Can near Taylor Swift	5	2	0	**3**	2
		平均成功率	33.3±6.1%	26.7±5.8%	**78.3±5.4%**	**85.0±4.6%**

Google 机器人评估的完整结果见表 6。总体而言，RT-1-X 和 Octo 难以应对评估任务，多项任务在五次试验中一次也未成功。相比之下，RT-2-X 和 OpenVLA 表现较强，每项任务至少在五次试验中成功两次；这两种 VLA 策略在该评估套件上的表现相当。

B.3 数据高效适配实验细节

本节进一步介绍章节 5.2中的数据高效适配实验，考察微调后的 OpenVLA 策略在 Franka-Tabletop 和 Franka-DROID 等新机器人配置上的有效性。

B.3.1 Franka-Tabletop 与 Franka-DROID 任务

针对七项任务分别采集 10–150 条示范。前六项对应称为 “Franka-Tabletop” 的机器人配置，即安装在桌面上的 Franka Emika Panda 机器人；最后一项对应 “Franka-DROID” 配置。

在 Franka-Tabletop 配置中，六项任务的前三项是范围较窄的单指令任务，后三项是多指令任务：场景中存在多个物体，机器人必须根据语言指令操控正确物体。

下面介绍图 10所示的六项 Franka-Tabletop 任务：

1. **将胡萝卜放入碗中**（单指令）：机器人需抓住胡萝卜并将其放入碗中。训练数据集包含该任务的 50 条示范，每个回合将胡萝卜和碗随机放在桌面不同位置，且胡萝卜始终初始化在碗左侧。评估时，每次试验记为成功（1）或失败（0），不设部分得分。
2. **将玉米倒入锅中**（单指令）：机器人需抓住红碗、移向钢锅，并将碗中物体（一根黄色玉米）倒入锅中。训练数据集包含该任务的 50 条示范，每个回合将碗和锅随机放在桌面不同位置，且碗始终初始化在锅右侧。评估时，每次试验记为成功（1）或失败（0），不设部分得分。
3. **将锅翻正**（单指令）：机器人需抓住初始竖置的钢锅，将其旋转至正立姿态并放回桌面。训练数据集仅包含该任务的 10 条示范，钢锅随机放置在桌面一小块区域内的不同位置。评估时，每次试验记为成功（1）、失败（0）或部分成功（0.5）。抓住锅

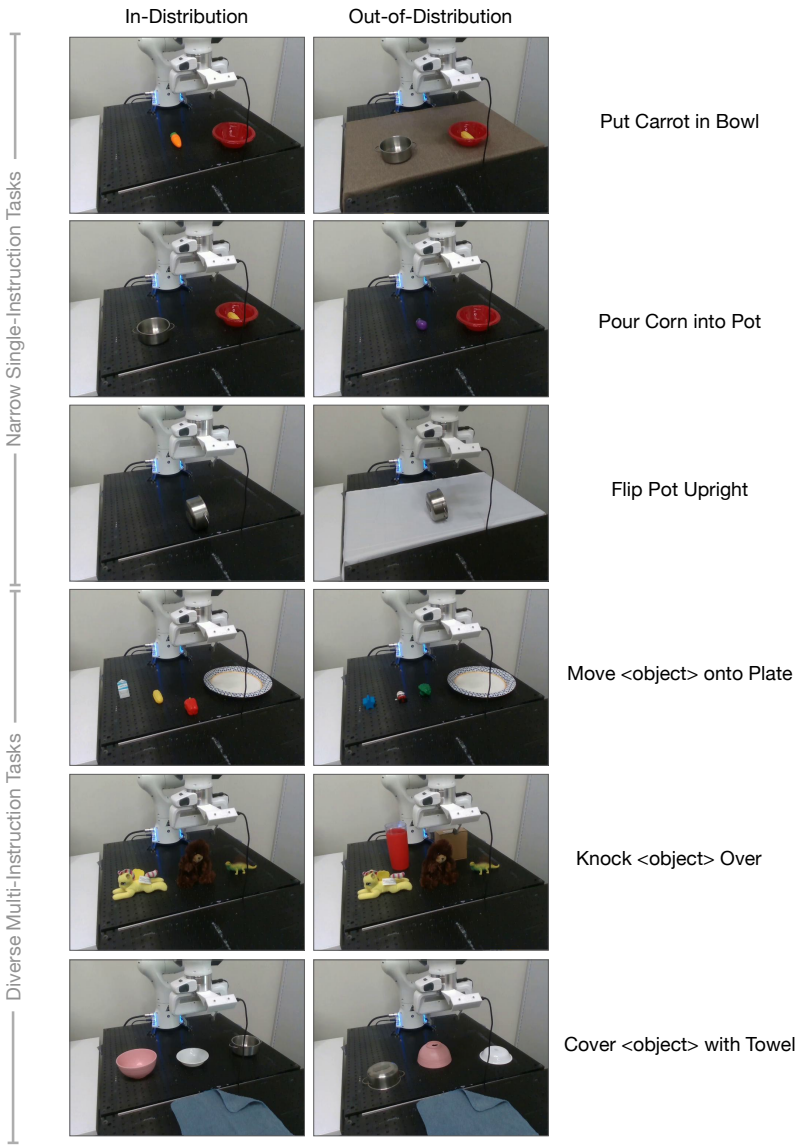


图 10: **Franka-Tabletop 微调任务**。图中展示章节 5.2数据高效适配实验所用的 Franka-Tabletop 任务，详细说明见图 10。六项任务的前三项位于上方三行，仅涉及单条指令；后三项位于下方三行，涉及多个物体与多条指令，指令指定目标物体或目标位置。第一列展示符合训练数据分布的初始状态样例，第二列展示分布外（OOD）初始状态（e.g., 未见过的背景、目标物体、干扰物以及物体位置/朝向）。章节 5.2中的每种策略均在分布内任务上评估 10–12 条试验轨迹，在 OOD 任务上评估 5–6 条。

但未将其翻正，或将其碰倒至正立姿态但未仔细引导，均计为部分成功；回合结束时机器人必须松开锅具才能获得满分。

4. **将 < 物体 > 移到盘子上**（多指令）：机器人需根据语言指令指定的目标，从三个物体中抓取一个，并放到桌面右侧的盘子上。训练数据集包含该任务的 150 条示范，在桌面随机摆放不同的三物体组合并选择其中一个作为目标；盘子始终初始化在桌面右侧。评估时，每次试验记为成功（1）、失败（0）或部分成功（0.5）。如果机器人首先接触的物体是语言指令指定的正确目标，但未完成任务，则记为部分成功。
5. **推倒 < 物体 >**（多指令）：机器人需根据语言指令指定的目标接近三个物体中的一个，并将其推倒。训练数据集包含该任务的 70 条示范，在桌面随机摆放不同的三物体组合并选择其中一个作为目标。评估时，每次试验记为成功（1）、失败（0）或

部分成功（0.5）。如果机器人首先接触的物体是语言指令指定的正确目标，但未完成任务，则记为部分成功。

6. **用毛巾覆盖 < 物体 >**（多指令）：机器人需抓住蓝色毛巾，并根据语言指令指定的目标将其放到三个物体中的一个上。训练数据集包含该任务的 45 条示范，在桌面随机摆放不同的三物体组合。评估时，每次试验记为成功（1）、失败（0）或部分成功（0.5）。如果毛巾首先接触的是语言指令指定的正确目标，但机器人未完成任务（e.g., 将毛巾掉在桌面而非目标物体上），则记为部分成功。只要毛巾任一部分搭在目标物体顶面即可获得满分，无需完全覆盖物体。

对每项 Franka-Tabletop 任务，各方法均进行 10–12 次分布内试验和 5–6 次 OOD 泛化试验。分布内与 OOD 测试条件见图 10第二列。

六项任务的 OOD 测试条件如下：

1. 将胡萝卜放入碗中（OOD）：用未见过的茄子替换胡萝卜。
2. 将玉米倒入锅中（OOD）：桌面铺上未见过的棕色桌布。
3. 将锅翻正（OOD）：桌面铺上未见过的白色桌布。
4. 将 < 物体 > 移到盘子上（OOD）：桌面放置三个未见过的物体。
5. 推倒 < 物体 >（OOD）：在三个已见物体后方放置两个未见干扰物体，即红色塑料杯和棕色盒子。
6. 用毛巾覆盖 < 物体 >（OOD）：将桌面三个物体倒置并放到未见过的位置。

最后，在 Franka-DROID 环境中实验一项任务及其变体：**擦拭桌面**（见图 11）。机器人需抓住刷子，将三个棕色小物体全部扫入簸箕。训练数据集包含该任务的 70 条示范，并改变所有物体的位置。

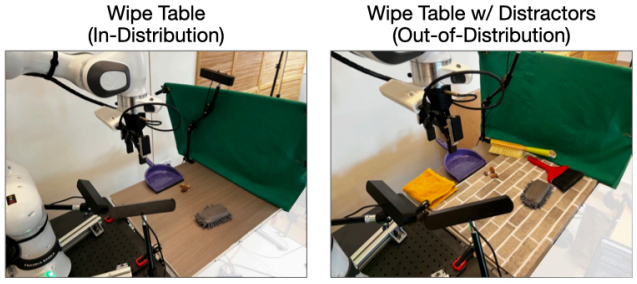


图 11: **Franka-DROID 微调任务**。图中的“擦拭桌面”是章节 5.2数据高效适配实验所用的最后一项任务。左图展示分布内试验的初始条件；右图展示桌面存在未见干扰物体的分布外试验。要完全完成任务，机器人必须抓住刷子，将三个物体全部扫入簸箕。

测试同时覆盖与训练数据匹配的分布内条件（图 11左）和桌面存在干扰物体的分布外（OOD）条件（图 11右）。由于每次试验可能产生多种结果，评分规则如下：每次试验满分 2 分；机器人将三个物体全部扫入簸箕得 2 分，成功扫入一个或两个物体得 1 分，否则得 0 分。每种策略进行 18 次分布内试验和 12 次 OOD 试验，总分为 60 分。

B.3.2 Franka-Tabletop 与 Franka-DROID 详细评估结果

Franka-Tabletop 和 Franka-DROID 的完整评估结果见表 7，评估对象为章节 5.2所述方法。Diffusion Policy 在单指令 Franka-Tabletop 任务（e.g., “Put Carrot in Bowl” 和 “Pour Corn in Pot”）上表现较强，优于其他方法；OpenVLA 和 Octo 则在更多样的多指令任务（“Move <object> onto Plate” “Knock <object> Over” 和 “Cover <object> with Towel”）上性能更高。在 Franka-DROID 环境中，OpenVLA 取得最佳结果。总体而言，OpenVLA 在两种环境上的平均性能最高。

此外，表 8给出表 1所概述参数高效微调实验结果的详细版本。实验选取两项有代表性的 Franka-Tabletop 任务，并同时设置分布内和 OOD 变体：一项范围较窄的单指令任务（“Put Carrot in Bowl”）和一项多样的多指令任务（“Move <object> onto Plate”）。训练示范数量与章节 5.2一致，分别为 50 和 150 条，详见章节 B.3.1。

表 7: **数据高效适配实验详细结果**。此处完整分解图 5概述的结果。报告在新机器人任务上从头训练的 Diffusion Policy，以及使用相同数据微调的通用策略性能。每种策略均在分布内与分布外（OOD）泛化条件下测试；Franka-Tabletop 任务见图 10，Franka-DROID 任务见图 11。没有一种策略在所有任务上都最佳：Diffusion Policy 在单指令任务上成功率较高，OpenVLA 和 Octo 在多样的多指令任务上表现良好；但就总体性能而言，OpenVLA 在两个环境上的平均成功率最高。

		试验次数	Diffusion Policy	Diffusion Policy（匹配）	Octo	OpenVLA（从头训练）	OpenVLA（本文）
Franka-Tabletop (5Hz)	“Put Carrot in Bowl” (in-distribution)	10	**90.0%**	80.0%	40.0%	70.0%	70.0%
	“Put Carrot in Bowl” (OOD)	5	20.0%	0.0%	20.0%	0.0%	**40.0%**
	“Pour Corn into Pot” (in-distribution)	10	**100.0%**	90.0%	0.0%	10.0%	50.0%
	“Pour Corn into Pot” (OOD)	5	**80.0%**	60.0%	0.0%	20.0%	60.0%
	“Flip Pot Upright” (in-distribution)	10	**100.0%**	85.0%	40.0%	85.0%	**100.0%**
	“Flip Pot Upright” (OOD)	5	50.0%	20.0%	0.0%	40.0%	**80.0%**
	“Move <object> onto Plate” (in-distribution)	12	25.0%	25.0%	41.7%	8.3%	**75.0%**
	“Move <object> onto Plate” (OOD)	6	8.3%	33.3%	8.3%	33.3%	**58.3%**
	“Knock <object> Over” (in-distribution)	12	33.3%	25.0%	**83.3%**	75.0%	75.0%
	“Knock <object> Over” (OOD)	6	16.7%	16.7%	33.3%	58.3%	**83.3%**
	“Cover <object> with Towel” (in-distribution)	12	16.7%	20.8%	**91.7%**	41.7%	50.0%
	“Cover <object> with Towel” (OOD)	6	16.7%	33.3%	**91.7%**	50.0%	50.0%
	平均值		48.5±4.9%	43.4±4.7%	43.4±4.4%	43.4±4.6%	**67.2±4.0%**
Franka-DROID (15Hz)	“Wipe Table” (in-distribution)	18	50.0%	27.8%	52.8%	25.0%	55.6%
	“Wipe Table” + Distractors (OOD)	12	12.5%	25.0%	16.7%	16.7%	62.5%
	平均值		35.0±8.0%	26.7±7.5%	38.3±8.5%	21.7±6.6%	**58.3±7.2%**

表 8: **参数高效微调实验详细结果**。此处给出表 1所概述的详细任务性能。

		试验次数	全量微调	仅最后一层	冻结视觉编码器	夹心式	LoRA, r=32	LoRA, r=64
Franka-Tabletop (5Hz)	“Put Carrot in Bowl” (in-distribution)	10	90.0	40.0	40.0	90.0	60.0	90.0
	“Put Carrot in Bowl” (OOD)	5	40.0	0.0	40.0	0.0	60.0	40.0
	“Move <object> onto Plate” (in-distribution)	12	79.2	33.3	50.0	75.0	75.0	62.5
	“Move <object> onto Plate” (OOD)	6	41.7	33.3	58.3	41.7	75.0	66.7
	平均值		**69.7±7.2%**	30.3±6.1%	47.0±6.9%	62.1±7.9%	**68.2±7.5%**	**68.2±7.8%**

C BridgeData V2 评估中的 RT-2-X 与 OpenVLA 比较

本节补充章节 5.1中 BridgeData V2 评估里 RT-2-X 与 OpenVLA 比较的细节。如前所述，OpenVLA 在比 RT-2-X 更大的 OpenX 数据子集上预训练，并使用融合的 SigLIP-DinoV2 视觉骨干网络而非单一视觉编码器。除这些因素外，OpenVLA 在 BridgeData V2 评估中显著优于 RT-2-X（见图 3），也源于对 Bridge 数据集更谨慎的预处理。

开发 OpenVLA 模型时，我们发现原始 BridgeData V2 数据集包含许多全零（无操作）动作转移。例如，每条示范的第一个时间步都将全零动作记录为真实动作。因此，若不进行数据预处理就直接在原始数据集上训练表达能力很强的 VLA 模型，策略会频繁预测全零动作并在评估时停滞。训练 OpenVLA 时只需滤除每条示范的第一个转移，即可在多数情况下缓解停滞行为。

RT-2-X 训练时未采用这种数据预处理，因此若不修改模型查询流程便直接部署，常会出现上述停滞行为，严重降低试验轨迹性能。RT-2-X 是专有模型，无法用预处理后的 BridgeData V2 数据集重新训练；为缓解该问题，我们直接查询模型的第二可能动作，因为最可能动作往往全为零，而第二可能动作通常不是。RT-2-X 开发者在 Open X-Embodiment 实验 [1] 所报告的 BridgeData V2 评估中也采用了相同的权宜方案。该方案显著增强了 RT-2-X 在 BridgeData V2 评估中的性能，但我们认为其仍不如在预处理数据集上重新训练模型。

我们还尝试动态查询 RT-2-X：先采样最可能动作；若该动作全为零，再采样第二可能动作。然而实验发现，动态查询的性能不如始终直接查询第二可能动作。原因可能是动态查询改变了机器人动力学：在轨迹中途暂停并重新查询模型时，查询流程不可忽略的延迟会轻微打断机器人运动，进而造成细微的性能下降。因此，与 Open X-Embodiment 项目 [1] 一致，本文报告始终查询第二可能动作时的 RT-2-X 性能。

D 其他实验与消融研究

本节进行多项补充实验，分析 OpenVLA 模型架构与训练方案各组成部分的作用，并为前文观点提供定量证据，旨在回答以下问题：

1. OpenX 训练有多重要，它如何影响 OpenVLA 性能（章节 D.1）?
2. 与仅使用 SigLIP 视觉编码器相比，融合的 SigLIP-DinoV2 视觉编码器如何影响 OpenVLA 性能（章节 D.2）?
3. 对 OpenVLA 而言，微调视觉编码器还是将其冻结更好（章节 D.3）?

4. 将策略性能与模型推理速度解耦后，章节 5.3中的量化推理结果如何变化（章节 D.4）?

下文依次讨论回答上述问题的实验配置与结果。

D.1 OpenX 训练数据消融实验

如章节 3.3所述，OpenVLA 在 Open X-Embodiment 数据集 [1]（OpenX）的大规模机器人本体、场景和任务数据上训练。本节消融 OpenX 混合数据，仅在单个机器人数据集上训练 VLA 策略，以评估 OpenX 训练对策略性能的影响。章节 5.2（见 OpenVLA (Scratch)）已显示在微调条件下移除 OpenX 训练的负面影响；本节在另一种机器人本体上开展补充实验，以提供更多证据。

实验配置与任务。比较原始 OpenVLA 模型与 **OpenVLA-Bridge**。后者采用与 OpenVLA 相同的预训练 VLM（Prismatic VLM [44]），但只在 BridgeData V2 [6] 上微调，而非使用章节 A中的完整 OpenX 训练混合数据。在章节 B.1.1所述 BridgeData V2 WidowX 机器人评估套件中，选取 8 项代表性任务评估 OpenVLA 与 OpenVLA-Bridge，任务见表 9。

结果。OpenX 训练混合数据消融结果见表 9。与 OpenVLA 相比，OpenVLA-Bridge 性能大幅下降，绝对成功率降低 30 个百分点，表明 OpenX 预训练对最终策略性能十分重要。语言落地性能虽未受影响，但所有泛化类别的性能都下降了。这说明 OpenX 训练混合数据中场景、物体和任务的高度多样性，是释放 OpenVLA 模型更强泛化能力的关键。

表 9: **BridgeData V2 WidowX 消融实验结果**。在 8 项代表性任务子集上评估不同方法，以衡量 OpenVLA 模型架构与训练方案各组成部分的重要性。OpenVLA-Bridge 是不进行 OpenX 训练的 OpenVLA 版本，仅在 BridgeData V2 上训练；OpenVLA-Bridge-SigLIP 进一步移除 DinoV2 编码器以消融融合视觉骨干网络，其视觉骨干仅包含 SigLIP 编码器。OpenX 训练和融合视觉编码器都能提升策略性能，但前者的作用远大于后者。

类别	任务	试验次数	OpenVLA 成功次数	OpenVLA-Bridge 成功次数	OpenVLA-Bridge-SigLIP 成功次数
视觉泛化	Put Eggplant into Pot (Easy Version)	10	10	8	8
视觉泛化	Put Eggplant into Pot	10	10	2	3
视觉泛化	Put Cup from Counter into Sink	10	7	4	2
运动泛化	Lift Eggplant	10	7.5	5.5	6.5
物理泛化	Put Carrot on Plate	10	8	4	1
物理泛化	Lift AAA Battery	10	7	2	2
语义泛化	Take Purple Grapes out of Pot	10	4	3	3
语言落地	Put {Eggplant, Red Bottle} into Pot	10	7.5	8	7
		平均成功率	76.3 ± 4.8%	45.6 ± 5.6%	40.6 ± 5.5%

D.2 双视觉编码器与单视觉编码器实验

OpenVLA 模型架构采用融合视觉骨干网络，结合 SigLIP [9] 与 DinoV2 [25] 编码器。本节消融 DinoV2 组件，以评估双视觉编码器的重要性。

实验配置与任务。构建 **OpenVLA-Bridge-SigLIP** 模型：该 OpenVLA 变体仅在 BridgeData V2 上训练，视觉骨干只含 SigLIP 编码器。将其与上一节（章节 D.1）的 OpenVLA-Bridge 模型比较；后者与原始 OpenVLA 架构相同，但也只在 Bridge 机器人数据上训练。因此，OpenVLA-Bridge-SigLIP 与 OpenVLA-Bridge 的唯一区别是前者从视觉骨干中移除了 DinoV2 编码器。两种模型在上一节所述同一组 8 项 Bridge 任务上评估。

结果。双视觉编码器消融结果见表 9。OpenVLA-Bridge-SigLIP 相较 OpenVLA-Bridge 性能下降，说明在视觉骨干中加入 DinoV2 编码器可提升策略性能。不过，此处 5 个百分点的性能降幅远小于移除 OpenX 训练时的 30 个百分点。DinoV2 所表征的低层空间特征似乎仅在部分情况下有助于泛化。

D.3 微调视觉编码器与冻结视觉编码器实验

如章节 3.4所述，先前 VLM 研究发现，冻结视觉编码器比微调其参数的性能更高 [44]。然而，训练 OpenVLA 时，我们微调了模型全部 7B 参数，包括 SigLIP-DinoV2 视觉骨干，因为开发早期发现微调视觉编码器能产生性能更高的 VLA；该结论在多种预训练 VLM 与模型架构上均成立。具体结果如下。

实验配置与任务。本节报告两种 VLA 策略的性能，它们分别由 Prismatic VLMs[44] 仓库中的两个不同预训练模型在 BridgeData V2 上微调而得。两个预训练模型名为 **SigLIP ViT-SO 224px** 和 **LLaVa v1.5 7B (Reproduction)**；架构与训练混合数据细节见 Karamcheti

et al. [44]。两种策略在表 10所示的多项 Bridge 任务上评估。此处评估配置不同于前述 Bridge 评估，故结果不能与其他类似实验直接比较。

结果。微调与冻结视觉编码器的实验结果见表 10。对于两种受测 VLA，微调视觉编码器都在多项任务上显著提高了成功率。从定性表现看，部署冻结视觉编码器的策略有时会导致明显次优且不稳定的机器人行为。因此，开发早期便决定不再进一步实验冻结视觉编码器。

表 10: **微调与冻结视觉编码器的实验结果**。在两种 VLA 策略中比较微调（"Fine-Tuned"）和冻结视觉编码器（"Frozen Vision"）的性能；两种策略分别基于 Prismatic VLMs[44] 仓库中的不同预训练 VLM 构建。表中 BridgeData V2 WidowX 任务与本文其他 Bridge 实验使用同一水槽环境，但初始环境配置不同，因为这些评估在项目早期进行。结果表明，微调视觉编码器对获得良好策略性能至关重要。部分冻结视觉编码器的评估因性能很差（接近零）且机器人行为不稳定而中止。在同时测试冻结和微调方法的评估中，微调视觉编码器的平均成功率为 80.0%，冻结时为 46.7%。

任务	试验次数	SigLIP ViT-SO 224px		LLaVa v1.5 7B (Reproduction)	
		冻结视觉编码器 成功次数	微调 成功次数	冻结视觉编码器 成功次数	微调 成功次数
Put Eggplant into Pot	10	7	10	5	9
Put Corn on Plate	10	10	9	0	9
	平均成功率	85	**95**	25	**90**
Put { Eggplant, Red Bottle } into Pot	4	2	4	–	3
Put { Blue Cup, Pink Cup } on Plate	4	0	0	–	0
Lift { Cheese, Red Chili Pepper }	4	0	3	–	2
Put { Strawberry, Lime } into Pot	4	1	0	–	3
Move { Sushi, Grapes }	4	3	4	–	3
	平均成功率	30	**55**	–	55

D.4 其他量化推理实验：解耦策略性能与模型推理速度

在章节 5.3中，我们以不同推理精度评估 OpenVLA：半精度（bfloat16）、8 位量化和 4 位量化。8 位量化在 BridgeData V2 上的性能低于另外两种方法；我们推测性能下降源于 8 位量化操作导致模型推理速度较低。本节通过实验检验该推测。

具体而言，再次以三种精度评估 OpenVLA，但改用*阻塞式控制*。每个动作在机器人上完全执行后，策略才预测下一动作并由控制器执行。该方案控制不同延迟方法的系统动力学，从而可独立于预测速度检验策略动作预测质量。实际效果是迫使吞吐量较高的 bfloat16 和 4 位量化以更低速度运行，从而匹配以 8 位精度部署 OpenVLA 时的动力学。因此，预计阻塞式控制下 8 位精度 OpenVLA 的性能将与 bfloat16 和 4 位精度相当。

实验配置与任务。在章节 D.1和章节 D.2使用的同一组 8 项 BridgeData V2 任务上，报告 OpenVLA 采用阻塞式控制与量化推理时的性能。

结果。采用阻塞式控制的量化推理实验结果见表 11。表 2中 8 位量化因推理速度较低而产生最差的试验轨迹性能；此处使用阻塞式控制消除推理速度差异对任务性能的影响后，8 位量化与 bfloat16 精度和 4 位量化表现相当。这证实了先前非阻塞式控制实验中关于推理速度影响 8 位量化性能的推测。与章节 5.3一致，最低的 4 位精度也未造成明显性能下降。

表 11: **采用阻塞式控制的量化推理实验结果**。报告 OpenVLA 在多项 BridgeData V2 WidowX 任务上采用 bfloat16 精度（默认方法）、8 位量化（int8）和 4 位量化（int4）推理时的成功率与标准误差。所有平均成功率的误差条均重叠，说明各方法表现相当。

类别	任务	试验次数	bfloat16 成功次数	int8 成功次数	int4 成功次数
视觉泛化	Put Eggplant into Pot (Easy Version)	10	10	10	10
视觉泛化	Put Eggplant into Pot	10	9	10	10
视觉泛化	Put Cup from Counter into Sink	10	5	5	3
运动泛化	Lift Eggplant	10	8	7	7.5
物理泛化	Put Carrot on Plate	10	10	10	10
物理泛化	Lift AAA Battery	10	3	6	4
语义泛化	Take Purple Grapes out of Pot	10	2	2	2
语言落地	Put {Eggplant, Red Bottle} into Pot	10	9	9.5	8.5
		平均成功率	70.0 ± 5.1%	74.4 ± 4.9%	68.8 ± 5.2%

E LIBERO 仿真实验

章节 5.2和章节 5.3主要讨论将 OpenVLA 适配到新的现实世界机器人配置与任务。本节利用 **LIBERO** 基准 [116]，探索将 OpenVLA 适配到仿真机器人配置与任务。仿真实验具有两项主要优势：

1. **展示通用性：**尽管 OpenVLA 仅在现实世界机器人数据上预训练，仍能有效适配仿真域，克服现实与仿真环境及动力学之间的潜在差异。
2. **提高可访问性与可复现性：**将 OpenVLA 集成到公开仿真平台，使其他研究人员更易使用该模型，尤其是无法获得机器人硬件的研究人员；仿真实验也比现实实验更容易复现。

实验配置见章节 E.1，结果见章节 E.2。复现实验所需材料随 OpenVLA 代码库一并发布。

E.1 LIBERO 仿真实验配置

仿真配置与任务。LIBERO 基准 [116] 包含四个任务套件，旨在研究机器人操控中的终身学习；原论文因此考察了面向多种任务的前向与后向迁移。本文实验仅关注在目标任务套件上的监督微调，衡量各策略通过行为克隆学习成功任务示范后的性能。

实验使用以下四个任务套件，每个套件包含 10 项任务，每项任务含 50 条人类遥操作示范：

- **LIBERO-Spatial** 使用相同物体集合但采用不同布局，检验模型对空间关系的理解。
- **LIBERO-Object** 使用相同场景布局但采用不同物体，检验模型对物体类型的理解。
- **LIBERO-Goal** 使用相同物体与布局但采用不同任务目标，检验模型对不同任务导向行为的掌握程度。
- **LIBERO-Long**（亦称 **LIBERO-10**）包含具有多样物体、布局与任务的长时程任务。

对上述各训练数据集进行以下修改：

1. 为适应需要较高分辨率图像（如 256×256px 或 224×224px）的方法，以 256×256px 的更高分辨率重新生成所有示范。基准原始数据集由 128×128px 图像组成，简单上采样至 256×256px 会导致图像质量较差。因此从高分辨率图像开始，并按需下采样，以满足不同分辨率要求时都能保持较高图像质量。生成高分辨率图像时，使用所提供人类采集示范中存储的动作逐步运行仿真环境，并保存仿真器渲染图像。
2. 从数据集中滤除所有“无操作”动作，即平移与旋转分量幅值接近零且不改变机器人夹爪状态的动作。该简单数据清洗步骤对 OpenVLA 等表达能力很强的单步策略至关重要，否则策略会学习模仿这些无操作动作，并在评估时于某些状态无限停滞。
3. 训练和测试时将所有第三人称图像旋转 180 度，因为 LIBERO 环境在本文硬件上返回的图像上下颠倒。
4. 由于策略通过模仿学习训练，而模仿学习假设示范成功，因此在对应仿真环境中重放所有示范，并根据环境成功判据滤除未完成任务的示范。最终从 500 条 LIBERO-Spatial 示范中移除 68 条，从 500 条 LIBERO-Object 示范中移除 46 条，从 500 条 LIBERO-Goal 示范中移除 72 条，从 500 条 LIBERO-Long 示范中移除 121 条。
5. 所有比较方法仅使用固定第三人称相机图像，不使用原始数据集额外提供的腕部相机图像。OpenVLA 的视觉输入仅包含第三人称相机图像，这样设置可确保公平比较。

比较方法。比较对象包括从头训练的 **Diffusion Policy**[8] [3]、在目标数据集上微调的 **Octo** [5]，以及按章节 5.3所述通过 LoRA（$r=32$）在目标数据集上微调的 **OpenVLA**。每种策略分别在各任务套件上独立训练，而非将四个套件合并训练单一策略。所有策略使用同一组示范训练，因此都受益于上述数据清洗步骤。

[8] 采用 DROID 数据集论文 [11] 所述 Diffusion Policy 实现，其动作生成以任务标签的 DistilBERT [117] 语言嵌入为条件。

评估细节。为降低实验结果方差，所有方法在每个任务套件上评估 500 次试验，报告性能为三个随机种子的平均成功率，即每项统计量共 1500 次试验。尽管按前述方式修改了训练数据集，但测试环境保持不变，仍使用原始 LIBERO 基准提供的相同初始环境配置。

E.2 LIBERO 仿真实验结果

LIBERO 实验结果见表 12。OpenVLA 可有效适配 LIBERO 仿真环境中的任务，在受测方法中取得最高平均成功率和最佳平均排名。不过，此处 OpenVLA 与其他方法的总体差距小于章节 5.2中的现实世界微调实验。原因可能是 OpenVLA 仅使用现实世界机器人数据预训练，未使用仿真数据；由于仿真与现实环境及动力学之间存在域差距，在仿真机器人任务上微调的效果可能不如现实任务。Octo 的结果也支持这一观点：它同样在大量现实世界机器人数据上预训练，相较从头训练的 Diffusion Policy 这一简单而强力的基线，总体性能也只获得小幅提升。若将仿真数据加入预训练混合数据，预计预训练后微调的方法会取得更大性能增益。

表 12: **LIBERO 仿真基准结果**。报告各方法在 LIBERO 四个任务套件上的成功率（SR）与标准误差；每个结果取三个随机种子、每个种子 500 次试验的平均值。另给出各方法在每个任务套件内的排名，排名 1 表示套件内最强方法，排名 3 表示最弱方法。平均排名表明哪种方法最适合作为多种任务的默认选择，因此值得关注；它比未按各任务套件难度归一化的平均成功率更具信息量。总体而言，微调后的 OpenVLA 取得最高平均成功率与最佳平均排名，其次是微调后的 Octo，最后是从头训练的 Diffusion Policy。

	LIBERO-Spatial		LIBERO-Object		LIBERO-Goal		LIBERO-Long		平均值	
	成功率（↑）	排名（↓）	成功率（↑）	排名（↓）	成功率（↑）	排名（↓）	成功率（↑）	排名（↓）	成功率（↑）	排名（↓）
从头训练的 Diffusion Policy	78.3 ± 1.1%	3	**92.5 ± 0.7%**	1	68.3 ± 1.2%	3	50.5 ± 1.3%	3	72.4 ± 0.7%	2.5
微调后的 Octo	78.9 ± 1.0%	2	85.7 ± 0.9%	3	**84.6 ± 0.9%**	1	51.1 ± 1.3%	2	75.1 ± 0.6%	2
微调后的 OpenVLA（本文）	**84.7 ± 0.9%**	1	88.4 ± 0.8%	2	79.2 ± 1.0%	2	**53.7 ± 1.3%**	1	**76.5 ± 0.6%**	**1.5**