Institute of Automation
Chinese Academy of Sciences

ByteDance | Seed

BridgeVLA: 基于视觉-语言模型的高效 3D 操作学习的输入输出对齐

Peiyan Li[1,2,3,†], Yixiang Chen[1,3], Hongtao Wu[2,†,*], Xiao Ma[2,†], Xiangnan Wu[1]
Yan Huang[1,3,4], Liang Wang[1,3], Tao Kong[2], Tieniu Tan[1,3,5,*]
[1]CASIA [2]ByteDance Seed [3]UCAS [4]FiveAges [5]NJU

[†]Project Lead, [*]Corresponding Author

Abstract

Recently, leveraging pre-trained vision-language models (VLMs) for building vision-language-action (VLA) models has emerged as a promising approach to effective robot manipulation learning. However, only few methods incorporate 3D signals into VLMs for action prediction, and they do not fully leverage the spatial structure inherent in 3D data, leading to low sample efficiency. In this paper, we introduce BridgeVLA, a novel 3D VLA model that (1) projects 3D inputs to multiple 2D images, ensuring input alignment with the VLM backbone, and (2) utilizes 2D heatmaps for action prediction, unifying the input and output spaces within a consistent 2D image space. In addition, we propose a scalable pre-training method that equips the VLM backbone with the capability to predict 2D heatmaps before downstream policy learning. Extensive experiments show the proposed method is able to learn 3D manipulation efficiently and effectively. BridgeVLA outperforms state-of-the-art baseline methods across three simulation benchmarks. In RLBench, it improves the average success rate from 81.4 % to 88.2 % . In COLOSSEUM, it demonstrates significantly better performance in challenging generalization settings, boosting the average success rate from 56.7 % to 64.0 % . In GemBench, it surpasses all the comparing baseline methods in terms of average success rate. In real-robot experiments, BridgeVLA outperforms a state-of-the-art baseline method by 32 % on average. It generalizes robustly in multiple out-of-distribution settings, including visual disturbances and unseen instructions. Remarkably, it is able to achieve a success rate of 96.8 % on 10+ tasks with only 3 trajectories per task, highlighting its extraordinary sample efficiency.

Project Page: https://bridgevla.github.io/
Corresponding Email: wuhongtao.123@bytedance.com; tnt@nlpr.ia.ac.cn

1 引言

利用预训练的视觉-语言模型（VLMs） [? ? ? ?] 开发大型视觉-语言-动作（VLA）模型已成为学习具有广泛适用性和鲁棒操控策略的一种有前景的方法 [? ? ? ? ?] 。然而，大多数 VLA 模型仅结合 2D 图像输入，并且需要在数据收集上投入大量精力。另一方面，3D 机器人策略在模型设计中利用 3D 结构先验，并在学习复杂 3D 机器人操控任务中表现出卓越的样本效率 [? ? ? ? ?] 。我们能否开发一种统一的 3D VLA 模型，将 VLA 模型的有效性与 3D 策略的效率结合起来呢?

尽管已经有一些工作在将 3D 信息整合到 VLMs 中以开发 3D VLA 模型 [? ?] ，但这些工作通常将动作转换为没有空间结构的标记序列，并使用下一个标记预测来预测动作。与先前高效的 3D 策略 [? ? ? ? ?]

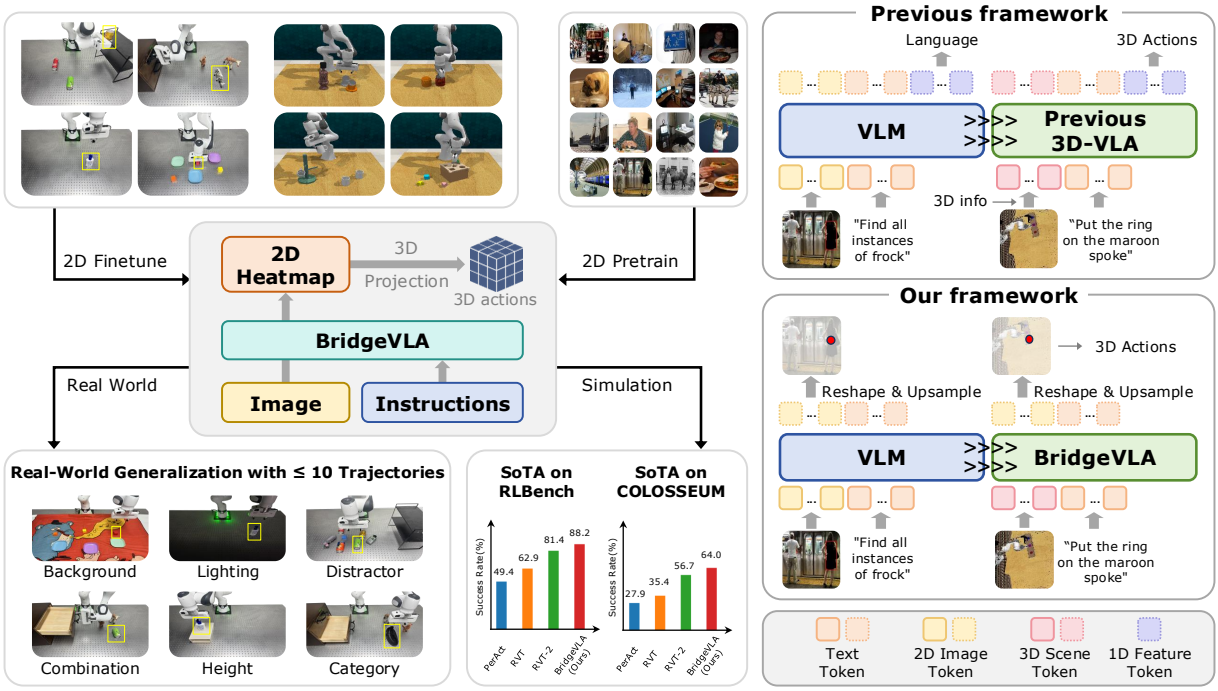

Figure 1 概述。BridgeVLA 是一种新颖的 3D VLA 模型，可以在统一的 2D 图像空间中对齐输入和输出。该模型在对象定位上通过 2D 热图进行预训练，并在 3D 操作预测上进行微调。模拟和现实世界的实验结果表明，该模型能够高效且有效地学习 3D 操作。

会将观察输入和动作输出对齐到统一空间的方法相比，这种策略未能利用 3D 结构先验，从而导致样本效率低下。在开发 3D VLA 模型中，另一个重要挑战在于动作微调中使用的 3D 输入与原始 VLM 预训练中使用的 2D 图像输入之间的不对齐，导致原始 VLM 预训练与微调之间发生大量的分布偏移。

为了解决上述挑战，如图 1 所示，我们提出了 BridgeVLA ，这是一种新颖的 3D VLA 模型，具有卓越的样本效率和强大的泛化能力。为了确保输入与预训练的 VLM 骨干相对齐，BridgeVLA 将 3D 点云观测转换为从不同正射投影视图捕获的多个 2D 图像 [? ?] 。为了利用 3D 输入的结构先验，BridgeVLA 被训练以预测用于平移动作预测的 2D 热图。这些由对应投影图像的标记生成的 2D 热图与这些图像共享相同的分辨率，在统一的空间结构内对齐输入观测和输出动作。鉴于原始 VLM 被预训练以预测标记序列，这与我们的 VLA 的 2D 热图输出不兼容，我们还引入了一种可扩展的预训练方法，该方法训练模型在文本输入条件下通过热图定位物体。这种预训练方法使 VLM 在下游策略学习微调之前具备预测热图的能力。总的来说，我们的设计在预训练和微调中将输入和输出对齐在一个共享的 2D 空间内。

我们在模拟和现实世界中进行了广泛的实验来评估所提出的方法。结果显示，BridgeVLA 能够高效且有效地学习 3D 操作。它在 RLBench [?] 中表现优于最先进的基线方法，将平均成功率从 81.4 % 提升到 88.2 %。在 COLOSSEUM [?] 中，它在具有挑战性的泛化环境中展示了强大的性能，将成功率从 56.7 % 提升到 64.0 %。在 GemBench [?] 中，它在平均成功率方面超过了所有比较的基线方法。在真实机器人实验中，我们评估了七种不同的设置，包括从视觉扰动到操控从未见类别中的物体。BridgeVLA 平均超越最先进方法 32 %，并在泛化到多个超分布设置中表现出色。值得注意的是，BridgeVLA 能够在 10 多个任务上达到 96.8 % 的成功率，每个任务仅使用 3 个轨迹进行训练，这突显了其卓越的样本效率。总之，本文的贡献有三个方面：

- 我们介绍了 BridgeVLA ，这是一种新颖的三维 VLA 模型，它通过与二维热图的输入输出对齐，利用视觉语言模型高效有效地学习三维机器人操作。
- 我们提出了一种可扩展的预训练方法，使模型具备通过物体定位根据文本输入预测热图的能力。
- 我们在仿真和真实环境中进行了大量实验，以全面评估所提出的方法。结果表明，BridgeVLA 在这两种环境中均优于最新的方法，并在真实机器人实验中实现了卓越的样本效率。

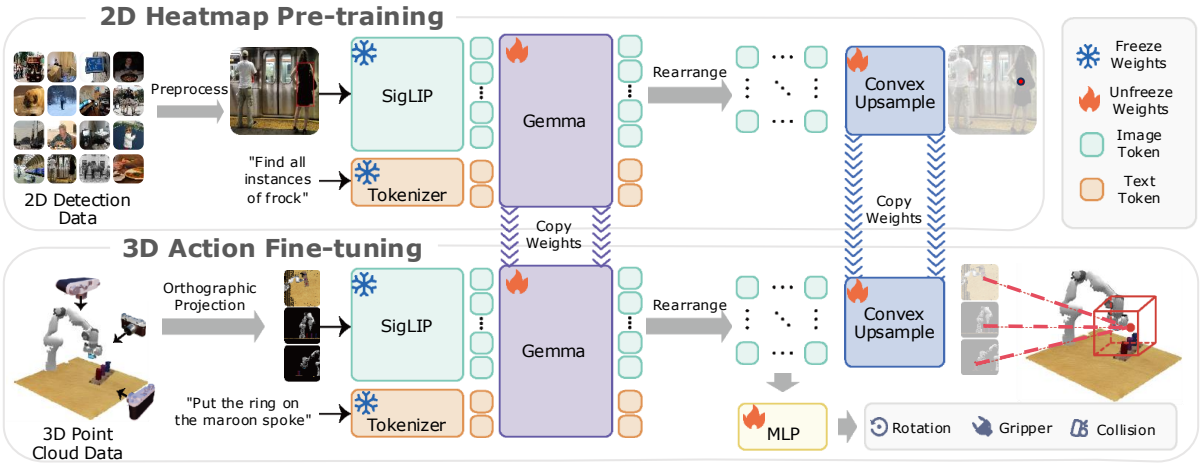

Figure 2 模型架构。(a) 2D 热图预训练：我们在 2D 物体检测数据集上训练 BridgeVLA 。该模型以图像和描述目标物体的语言作为输入，输出一个 2D 热图，突出显示对应于目标物体的感兴趣区域。请注意，这里展示的边界框仅用于说明目的；在输入模型时，图像中不存在它。(b) 3D 动作微调：该模型输入三张 3D 点云的正交投影图像和一条语言指令。它输出三个 2D 热图，突出显示所有三个视图中下一个关键帧中末端执行器的位置。对于剩余的动作组件，它使用 MLP 处理图像特征标记，以预测下一个关键帧的旋转动作、夹持器动作和碰撞标志。

大多数语言条件的视觉运动策略使用变压器处理二维视觉输入，并直接生成三维动作用于操作。在这些工作中，利用预训练的视觉语言模型（VLMs）来开发大型视觉语言动作（VLA）模型变得流行，因为它在学习复杂操作方面很有效。然而，这种基于二维图像的策略通常需要大量的数据收集工作，通常需要每个任务数百个轨迹才能有效学习。另一方面，三维操作策略通过利用三维输入中固有的空间结构，具有高效学习的巨大潜力。一种流行的工作是将点云数据作为输入。例如，Act3D 提出将图像特征提升到观测点云上创建三维特征云，并通过分类预测观测空间中三维点的平移动作。另一种工作利用体素来表示观测空间，并在体素空间中预测平移动作，将输入观测和输出动作统一在同一空间中。最近，RVT 和 RVT-2 提出利用三维点云的正交投影将三维信号转换为二维图像，以避免处理三维输入的高计算成本。与上述方法不同，我们的方法旨在将 VLA 模型的有效性和三维策略的效率统一在一个连贯的框架中，结合两者的优势。

虽然二维 VLA 模型已经被广泛研究，但三维 VLA 模型仍然相对研究较少。Zhen 等人基于大型语言模型（LLM）构建了 3D-VLA，并训练该模型以执行三维推理、多模态目标生成和机器人规划。Lift3D 建议通过隐式和显式的三维机器人表示来增强二维基础模型（例如，DINOv2）以学习三维操作策略。FP3 利用一个转换器来融合点云、自感知状态和语言指令的信息。PointVLA 分别使用 VLM 和点云编码器来处理二维图像和三维点云。从两个编码器中提取的嵌入被注入到一个动作专家中进行动作预测。SpatialVLA 引入 Ego3D 位置编码，将三维信息注入到二维图像观察中，并通过自适应动作网格更可转移地表示机器人移动。我们的方法不同于上述方法，它是通过设计来利用三维输入的空间结构进行动作预测。此外，它通过将三维输入投影到多个二维图像中，而不是将三维信息注入 VLM 中，从而弥合了预训练 VLM 的二维图像输入与三维输入之间的差距。这样的设计使其能够同时利用 VLM 骨干中的广泛知识和三维输入中嵌入的空间结构先验。

2 BridgeVLA

2.1 预备知识

BridgeVLA 旨在学习一个多任务 3D 机器人操作策略 π ，它将观测 $\boldsymbol{o}$ 和语言指令 l 映射为动作 $\boldsymbol{a}$ ：

$$\pi : (\boldsymbol{o}, l) \mapsto \boldsymbol{a} \tag{1}$$

我们假设可以接触到包含 N 轨迹的专家演示集合 $\mathcal{D} = \{\tau^i\}_{i=1}^N$ 。每个轨迹包含一个语言指令和一个观测-动作对的序列，即 $\tau^i = \{l^i, (\boldsymbol{o}_1^i, \boldsymbol{a}_1^i), ..., (\boldsymbol{o}_H^i, \boldsymbol{a}_H^i)\}$ 。观测 $\boldsymbol{o}$ 是从一个或多个视点获取的一个或多个 RGB-D 图像。根据之前的工作 [? ? ?] ，动作 $\boldsymbol{a}$ 包含一个 6 自由度末端执行器姿态 $T \in SE(3)$ 、目标夹持器状态 $g \in \{0, 1\}$ 和下一个关键帧的碰撞标志 $c \in \{0, 1\}$ 。碰撞标志 c 指示运动规划器在移动到目标姿态时是否应该避免碰撞。关键帧通常捕捉轨迹中的重要或瓶颈步骤（详细信息见附录 ?? ） [?] 。BridgeVLA 通过迭代

过程操作：1）在当前观测 $\boldsymbol{o}_t$ 和指令 l 的条件下预测动作 $\boldsymbol{a}_t$ ，2）使用基于采样的运动规划器 [? ? ?] 移动至预测的下一个关键帧姿态 T_t ，3）更新观测并重复，直到任务完成或达到最大步骤 $H_{\max}$ 。

如图 2 所示，BridgeVLA 采用了双阶段训练方案。在预训练期间，它被训练以预测对象检测数据集上的二维热图。在微调期间，点云被投射到多个二维图像中作为 VLM 骨干网络的输入。模型被训练以预测二维热图，用于估算平移动作及其他动作组件。该设计使得输入和输出在预训练和微调中的共享二维空间内保持一致。

2.2 二维热图预训练

VLM 骨干网络最初预训练是为了预测没有空间结构的标记序列。为了使其具备与下游策略学习相同的预测热图的能力，我们引入了一个预训练阶段，使模型通过热图定位目标对象。具体来说，我们利用 RoboPoint [?] 的 120K 目标检测集作为我们的预训练数据集。对于每张图像，我们从所有感兴趣对象的边界框构建真实热图 H^{gt} 。特别地，对于每个对象，我们构建一个具有空间截断的概率图：

$$H_i^{\text{gt}}(\boldsymbol{x}) = \begin{cases} p_i(\boldsymbol{x}) & \text{if } p_i(\boldsymbol{x}) \geq p_{\min} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

其中 $\boldsymbol{x} = (u, v)$ 表示像素位置，$p_i(\boldsymbol{x}) = \exp\left(-\|\boldsymbol{x} - \widehat{\boldsymbol{x}}_i\|^2 / 2\sigma^2\right)$ ，$\widehat{\boldsymbol{x}}_i$ 是对象边界框的中心，$p_{\min}$ 是一个概率阈值。对于所有感兴趣的对象，我们通过平均化和归一化来融合所有对象的概率图以获得 H^{gt} ：

$$H^{\text{gt}}(\boldsymbol{x}) = \frac{H_{\text{avg}}(\boldsymbol{x})}{\sum_{\boldsymbol{x} \in \Omega} H_{\text{avg}}(\boldsymbol{x})}, \text{ where } H_{\text{avg}}(\boldsymbol{x}) = \frac{1}{N} \sum_{i=1}^{N} H_i^{\text{gt}}(\boldsymbol{x}) \tag{3}$$

其中 Ω 表示像素空间。有关真实热图的样本，请参阅图 ?? 。

如图 2 所示，我们将描述感兴趣对象的文字提示与图像一起输入到 BridgeVLA 的 VLM 主干中。在本文中，我们使用 PaliGemma [?] 作为 VLM 主干，它由一个 SigLIP 视觉编码器 [?] 和一个 Gemma transformer 主干 [?] 组成。在其预训练期间，PaliGemma 接收一个或多个二维图像与前缀文本（例如，关于图像的一个问题）作为输入，并输出后缀文本（例如，问题的答案）。虽然模型使用因果注意力来预测后缀文本标记，但它采用双向注意力来处理图像标记和前缀文本标记。这使得图像标记可以融合来自前缀文本的信息。

为了预测热图，我们首先根据图像块的位置重新排列输出图像标记，以重建空间特征网格。一个凸包上采样块 [?] 然后将网格转换为与输入图像具有相同分辨率的热图。模型通过交叉熵损失来训练，以预测在图像中定位所有感兴趣对象位置的热图。我们强调，所提出的预训练策略输出一个空间感知的 2D 热图，而不是以往作品中使用的传统的下一个标记预测 [? ?] 。此外，这种方法具有高度的可扩展性，因为原则上它可以利用任何可以被表述为热图预测任务的视觉-语言数据集，例如关键点检测和语义分割。

在微调过程中，我们首先从经过校准的相机捕获的 RGB-D 图像中重建场景的点云。为了与 VLM 主干的 2D 图像输入对齐，我们从三个视点（顶部、正面和右侧）渲染点云的三个正交投影图像，并将这些图像用作 VLM 主干的输入图像，如同在 RVT [?] 和 RVT-2 [?] 中那样。这些图像与任务指令一起被输入到预训练的 VLM 主干中，以为每个视图生成热图。重要的是，我们在 VLM 前向过程中不加入任何额外信息（例如，机器人状态），以尽量减少预训练和微调之间的分布偏移。

对于平移动作，我们通过反投影三个视图的热图来估计均匀分布在机器人工作空间中的所有 3D 点网格的分数。得分最高的 3D 点的位置决定了下一个关键帧中末端执行器的平移。与之前的工作相似 [? ?] ，我们使用欧拉角来表示旋转动作，其中每个轴离散化为 72 个区间。为了预测旋转、二元夹爪动作和避撞标志，我们整合了来自全局和局部上下文的特征。对于全局特征，最大池化应用于每个输入正射投影图像的输出标记，总共产生三个标记——每个视图一个。对于局部特征，我们从每个视图的热图峰值中提取一个标记，同样产生三个标记。所有这些标记被连接并通过 MLP，以预测旋转动作、夹爪动作和避撞标志。

遵循先前工作中的方法 [? ?] ，BridgeVLA ，采用了一种由粗到细的细化策略，以实现准确的动作预测。在对原始点云进行初步预测之后，我们使用以预测平移为中心的长方体放大并裁剪点云。在裁剪和放大的点云上执行第二次前向传递。第二次传递中的预测动作将用于执行。

微调期间的训练损失由四个部分组成:

$$L = L_{\text{trans}} + L_{\text{rot}} + L_{\text{gripper}} + L_{\text{collision}} \tag{4}$$

Models	Overall		Task Success Rate (%)							
	Avg. SR (%) ↑	Avg. Rank ↓	Close Jar	Drag Stick	Insert Peg	Meat off Grill	Open Drawer	Place Cups	Place Wine	Push Buttons
Image-BC (CNN) [? ?]	1.3	9.3	0.0	0.0	0.0	0.0	0.0	4.0	0.0	0.0
Image-BC (ViT) [? ?]	1.3	9.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
C2F-ARM-BC [? ?]	20.1	8.7	24.0	24.0	4.0	20.0	20.0	0.0	8.0	72.0
HiveFormer [?]	45.3	6.9	52.0	76.0	0.0	100.0	52.0	0.0	80.0	84.0
PolarNet [?]	46.4	6.5	36.0	92.0	4.0	100.0	84.0	0.0	40.0	96.0
PerAct [?]	49.4	6.3	55.2 ± 4.7	89.6 ± 4.1	5.6 ± 4.1	70.4 ± 2.0	88.0 ± 5.7	2.4 ± 3.2	44.8 ± 7.8	92.8 ± 3.0
Act3D [?]	65.0	4.3	92.0	92.0	27.0	94.0	93.0	3.0	80.0	99.0
RVT [?]	62.9	4.4	52.0 ± 2.5	99.2 ± 1.6	11.2 ± 3.0	88.0 ± 2.5	71.2 ± 6.9	4.0 ± 2.5	91.0 ± 5.2	100.0 ± 0.0
3D Diffuser Actor [?]	81.3	2.5	96.0 ± 2.5	100.0 ± 0.0	65.6 ± 4.1	96.8 ± 1.6	89.6 ± 4.1	24.0 ± 7.6	93.6 ± 4.8	98.4 ± 2.0
RVT-2 [?]	81.4	2.5	100.0 ± 0.0	99.0 ± 1.7	40.0 ± 0.0	99.0 ± 1.7	74.0 ± 11.8	38.0 ± 4.5	95.0 ± 3.3	100.0 ± 0.0
BridgeVLA (Ours)	88.2	1.9	100.0 ± 0.0	100.0 ± 0.0	88.0 ± 2.8	100.0 ± 0.0	100.0 ± 0.0	58.4 ± 10.0	88.0 ± 2.8	98.4 ± 2.2

Models	Put in Cupboard	Put in Drawer	Put in Safe	Screw Bulb	Slide Block	Sort Shape	Stack Blocks	Stack Cups	Sweep to Dustpan	Turn Tap
Image-BC (CNN) [? ?]	0.0	8.0	4.0	0.0	0.0	0.0	0.0	0.0	0.0	8.0
Image-BC (ViT) [? ?]	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	16.0
C2F-ARM-BC [? ?]	0.0	4.0	12.0	8.0	16.0	8.0	0.0	0.0	0.0	68.0
HiveFormer [?]	32.0	68.0	76.0	8.0	64.0	8.0	8.0	0.0	28.0	80.0
PolarNet [?]	12.0	32.0	84.0	44.0	56.0	12.0	4.0	8.0	52.0	80.0
PerAct [?]	28.0 ± 4.4	51.2 ± 4.7	84.0 ± 3.6	17.6 ± 2.0	74.0 ± 13.0	16.8 ± 4.7	26.4 ± 3.2	2.4 ± 2.0	52.0 ± 0.0	88.0 ± 4.4
Act3D [?]	51.0	90.0	95.0	47.0	93.0	8.0	12.0	9.0	92.0	94.0
RVT [?]	49.6 ± 3.2	88.0 ± 5.7	91.2 ± 3.0	48.0 ± 5.7	81.6 ± 5.4	36.0 ± 2.5	28.8 ± 3.9	26.4 ± 8.2	72.0 ± 0.0	93.6 ± 4.1
3D Diffuser Actor [?]	85.6 ± 4.1	96.0 ± 3.6	97.6 ± 2.0	82.4 ± 2.0	97.6 ± 3.2	44.0 ± 4.4	68.3 ± 3.3	47.2 ± 8.5	84.0 ± 4.4	99.2 ± 1.6
RVT-2 [?]	66.0 ± 4.5	96.0 ± 0.0	96.0 ± 2.8	88.0 ± 4.9	92.0 ± 2.8	35.0 ± 7.1	80.0 ± 2.8	69.0 ± 5.9	100.0 ± 0.0	99.0 ± 1.7
BridgeVLA (Ours)	73.6 ± 4.6	99.2 ± 1.8	99.2 ± 1.8	87.2 ± 6.6	96.0 ± 2.8	60.8 ± 7.7	76.8 ± 8.7	81.6 ± 3.6	87.2 ± 1.8	92.8 ± 3.3

Table 1 RLBench 的结果。“Avg. Rank” 列报告了每种方法在所有 18 个任务中的平均排名，较低的值表示整体表现更好。BridgeVLA 在 18 个任务中有 10 个获得了最佳表现。

类似于预训练，L_{trans} 是一个监督平移动作热图预测的交叉熵损失。每个正射视图的真实热图是方程 2 中定义的归一化单对象概率图，其中 $\widehat{\boldsymbol{x}}_i$ 表示下一个关键帧中真实末端执行器位置的投影像素位置。在将旋转的欧拉角离散化到多个箱时，我们也在 L_{rot} 中应用交叉熵损失来监督旋转预测。对于夹持器动作和碰撞避免，我们在 L_{gripper} 和 $L_{\text{collision}}$ 中使用二元交叉熵损失作为监督。为了增强几何鲁棒性，在训练过程中随机刚体变换会被同时施加到点云和真实动作上。附加的训练细节可以在附录 **??** 中找到。

3 实验

在本节中，我们在仿真和现实世界中进行了广泛的实验来评估所提出的方法。通过这些实验，我们旨在回答四个问题：

Q1: BridgeVLA 在学习三维机器人操作方面，与最先进的方法相比有多高效?

Q2: BridgeVLA 能够从非常有限的数据中高效学习吗（例如，每个任务 3 条轨迹)?

Q3: BridgeVLA 在处理视觉干扰（例如，干扰物、背景和光照）方面有多稳健?

Q4: BridgeVLA 如何推广到新的对象-技能组合以及从未见过类别的对象?

3.1 模拟实验

3.1.1 在 RLBench 上的实验

设置。 RLBench [?] 在 CoppeliaSim [?] 中使用安装了平行钳夹的 Franka Panda 机器人来执行任务。观察包含从前方、左肩、右肩和手腕位置的四台校准摄像机捕获的四幅 RGB-D 图像。根据之前的研究 [? ? ? ? ?]，我们在 RLBench 中的 18 个任务上进行实验。这些任务包括 1）非抓取操作（例如，将方块滑动到目标)，2）拾取和放置（例如，堆叠杯子)，以及 3）高精度插入（例如，插入销钉)。每个任务都提供 100 个专家演示。而每个演示都配有语言指令和多个关键帧。通过每个任务进行 25 次试验中的二进制成功率评价模型，每次试验最多允许 25 个动作步骤。

基线。 我们比较了 BridgeVLA 与多个基准方法。(1) Image-BC (CNN) 和 Image-BC (ViT) [?] 是两种二维基准方法，它们分别使用 CNN 和 ViT 作为骨干网络，直接从二维图像预测动作。(2) C2F-ARM-BC [?] 在体素空间中采用由粗到细的策略预测下一个关键帧动作。(3) PerAct [?] 同样在体素空间中操作，并使用感知转换器 [?] 预测动作。(4) HiveFormer 结合了历史信息，采用统一的多模态转换器架构。(5) PolarNet