arXiv:2603.01469v1 [cs.RO] 2 Mar 2026

基于 Mean-Flow 的单步视觉-语言-动作

Yang Chen、Xiaoguang Ma、Bin Zhao

摘要——**近期，基于 FlowMatching 的视觉-语言-动作（VLA）框架取得了显著进展，在生成高频动作块方面展现出突出优势，尤其适用于高度灵巧的机器人操作任务。尽管取得了这些显著成果，其实际应用仍受制于较长的生成延迟，这源于其固有的迭代采样需求和架构限制。为解决这一关键瓶颈，我们提出了一种基于 Mean-Flow 的单步VLA方法。具体而言，我们解决了动作生成过程中的噪声引起的问题，从而消除了传统流匹配方法固有的一致性约束。这显著提高了生成效率，并实现了单步动作生成。真实世界机器人实验表明，所提出的基于 Mean-Flow 的单步VLA生成速度分别比 SmolVLA 和扩散策略快 8.7 倍和 83.9 倍。这些结果展现了其作为基于视觉-语言-动作的机器人操作高效骨干模型的巨大潜力。**

I. 引言

视觉-语言-动作（VLA）模型将语义 grounding、空间推理和序列动作生成整合到统一框架中，使机器人能够在多种场景下执行具有泛化性和适应性的行为 [1]。根据所采用的策略学习范式，VLA 模型大致可分为三类：基于Transformer的视觉-语言-动作模型、基于扩散的模型和基于 FlowMatching 的 VLA [2]。

首个 VLA 模型 RT-1 [3]由谷歌开发，采用基于 Transformer 的架构，并利用 130k+ 机器人演示数据进行训练，实现了视觉输入到动作的端到端映射，在家庭任务上取得了 90%+ 的成功率。RT-2 [4] 融合了预训练视觉-语言模型（VLM），整合网络规模的世界知识，无需重新训练即可泛化至未见物体和抽象指令，使跨任务性能较 RT-1 [5]提升了 30%+。尽管基于 Transformer 的 VLA 模型在长序列建模和统一多模态表征方面表现出色，但它们也存在明显缺陷，包括离散化伪影、运动不连续性和动作多样性有限。

不同于基于Transformer的视觉-语言-动作模型将动作生成视为基于词元的序列预测任务，并输出离散动作词元，基于扩散的视觉-语言-动作模型将机器人视觉运动控制建模为去噪扩散过程，并通过随机朗之万动力学学习动作分数梯度，在此过程中，连续动作空间通过逐步降低噪声直接建模 [6]。开源项目 Octo [7]为通用机器人策略奠定了基础，并表明基于扩散的动作生成能够显著提升零样本性能，例如比 RT-1-X 高出 33% [8]。

基于 FlowMatching 的 VLA 在基于扩散的方法之上进一步发展，通过基于流的动力学直接生成连续轨迹。与依赖迭代去噪的扩散模型不同，FlowMatching 学习数据分布与噪声分布之间的确定性传输映射，从而实现近乎即时的采样并降低延迟，优于扩散方法的迭代去噪 [9]。π_0 [10] 中的基于 FlowMatching 的动作专家将生成过程重新表述为从噪声分布到数据分布的连续流，并通过满足确定性常微分方程（ODE）的时变向量场进行参数化。FlowMatching 中的去噪过程通过在较小的离散时间区间上进行欧拉积分来实现。当推理步数减少时，每个欧拉步骤覆盖的时间跨度会增大，从而放大偏差，使生成样本进一步偏向数据集均值 [11]。因此，FlowMatching 面临实时效率与动作生成准确性之间固有的权衡：减少推理步数可以降低延迟，却会损害轨迹保真度；若要保持质量，则需要更多积分步数，进而提高推理成本。

与 FlowMatching 学习瞬时向量场、并依赖多步数值积分来重构轨迹不同，MeanFlow 学习均值去噪向量场，直接对从噪声端点到数据端点的整体映射进行建模，从而减轻数值误差的累积，并避免多步迭代去噪推理的需求 [12]。得益于这一设计，它无需预训练、蒸馏或额外的一致性启发式方法。本文提出基于 Mean-Flow 的单步 VLA，以同时实现高生成质量、低推理成本和强训练稳定性。事实上，它只需单步或少数几步推理，即可生成高质量的连续动作，在保持训练流程简洁的同时，显著减少推理步数和延迟。

本文的贡献如下：

- 我们提出了基于 Mean-Flow 的单步 VLA，这是一种新颖的视觉-语言-动作框架，将 MeanFlow 生成方法应用于机器人操作。
- 我们通过以均值去噪向量场预测器取代传统的 FlowMatching 机制，实现了显著的效率提升。与最先进方法相比，生成速度快 8.7 倍。
- 我们通过真实世界机器人实验验证了该框架，证明基于 Mean-Flow 的单步 VLA 在以下两种模式下均能实现稳健性能：

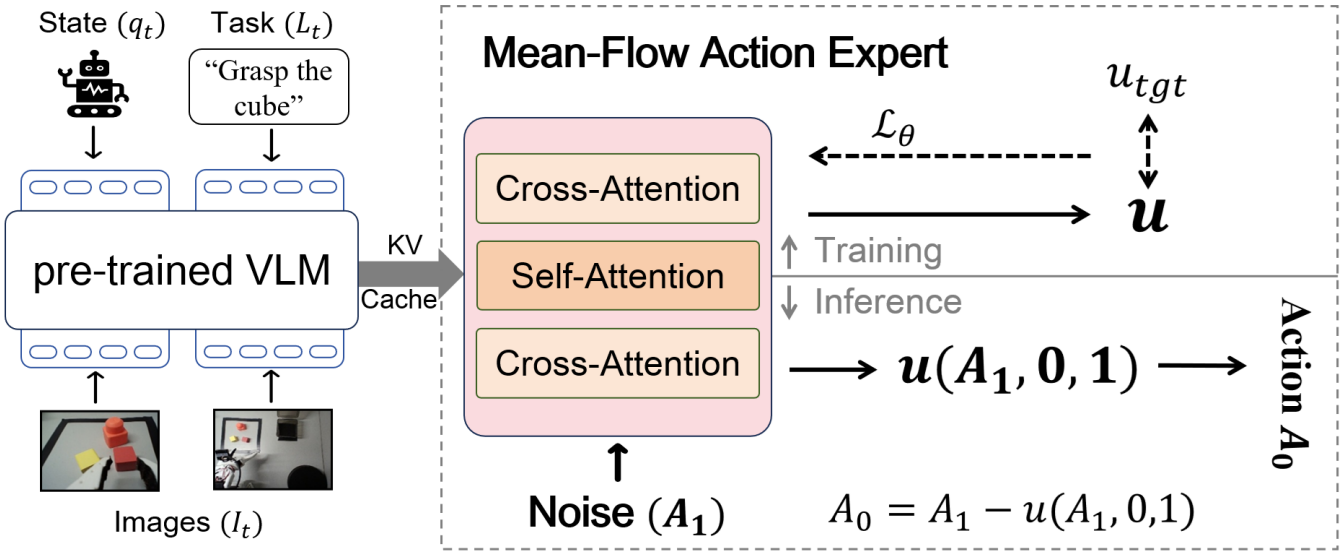

图 1. 基于 Mean-Flow 的单步 VLA 框架概览。预训练 VLM 处理多模态输入。训练期间，Mean-Flow 动作专家根据 VLM 特征估计均值去噪向量场；推理期间则预测平均向量场，以获得动作 A_0 。

单步生成和多步生成模式。

II. 相关工作

A. 基于扩散的视觉-语言-动作模型

将基于扩散的策略整合到视觉-语言-动作系统中，使机器人动作生成从确定性回归转向多模态生成式策略 [13]。Octo [7] 是首个采用基于扩散的策略进行连续动作控制的视觉-语言-动作（VLA）模型。它采用 Transformer 主干网络，并搭配扩散头，通过去噪扩散过程解码动作块，从而显著提升动作解码的性能和时间连贯性。尽管基于扩散的策略能够生成平滑且高频的控制信号，但缺少解决复杂任务所需的推理机制。传统的自回归视觉-语言-动作模型具有很强的推理能力，却往往难以生成精准且稳定的动作。Diffusion-VLA（DiVLA） [14]结合大规模预训练视觉语言模型（VLM）进行任务分解和解释，并利用扩散头实现稳健、连续的动作生成，从而兼具上述优势。在一项涉及 102 种此前未见物体的零样本抓取分拣基准测试中，DiVLA 的任务成功率达到 63.7%，展现出对新颖视觉场景的强大泛化能力。

B. 基于流匹配的视觉-语言-动作模型

FlowMatching 是一类生成模型，它构造流路径，并将先验分布传输到数据分布 [15]。 π_0 模型采用 FlowMatching 框架，通过训练学习向量场。在推理过程中，模型随时间对所学向量场进行积分，从噪声计算出通往目标状态的轨迹。与自回归预测相比，FlowMatching 更适用于连续动作生成，因为它能够在一次前向传播中并行预测整个动作块，从而降低推理延迟 [16]。通过缩小模型规模并减少可训练参数，SmolVLA[17]在保持与大型 VLA 性能相当的同时，提高了计算效率。该模型旨在使用单个 GPU 进行训练，并部署在消费级硬件上，包括 CPU，同时采用替代的自注意力层和交叉注意力层。

尽管如此，基于 FlowMatching 的 VLA 模型在减少推理步数时仍会出现质量下降。这一固有限制是当前可用的基于流的 VLA 系统实现高效实时生成式控制的关键瓶颈。

C. 生成效率提升

在扩散模型中，函数评估次数（NFE）表示数值求解器调用神经网络的次数。这直接决定了推理速度和计算成本。现有降低 NFE 的方法主要依赖基于常微分方程的公式 [18],，因为确定性动力学更易于数值求解器处理。蒸馏 [19], [20], [21] 或一致性模型 [22], [23], [24] 等替代策略可以进一步降低 NFE，但需要额外训练或复杂的调度。近年来，FlowMatching [25] 作为一种连续时间替代方法应运而生，它无需进行分数估计，即可构建从噪声分布到数据分布的概率路径，从而实现更快速、更平滑的生成。然而，FlowMatching 只能将 NFE 降至约 10，因为进一步减少 NFE 会因累积的离散化误差而严重降低生成质量。这一局限源于：学习得到的瞬时向量场必须沿高度非线性的概率路径进行紧密积分，因此粗略的数值更新并不准确。

在确定性常微分方程框架的基础上，Mean-Flow 方法 [12] 通过学习近似某一时间区间内的均值去噪向量场，而非瞬时向量场，解决了 FlowMatching 的内在局限。MeanFlow 预测两个时间状态之间的平均流动方向，从而捕捉概率路径的整体趋势。均值去噪向量场可视为瞬时向量场的积分，为连续时间动力学提供了严谨的数学表述。通过关注轨迹的粗粒度演化，MeanFlow 缩小了单步扩散/流模型与其多步前身之间的差距。因此，它显著提升了推理效率，并简化了训练流程，无需额外蒸馏或一致性正则化。

III. 方法

本节介绍如何将 MeanFlow 原理融入视觉-语言-动作框架，以提高生成的效率和稳定性。我们首先推导 MeanFlow 的数学表述，随后详细说明其在 VLA 流程中的具体实现，最后介绍所提出的动作生成策略。图 1 展示了整体模型架构，并描绘了 Mean-Flow动作专家的工作逻辑。

A. 将 *MeanFlow* 引入 *VLA*

MeanFlow 方法是一种基于流的生成方法，旨在通过去噪技术提升生成模型的稳定性和效率。其核心思想是引入一种新颖的真实场，表示为全局平均去噪向量场 $u(z,\ r,\ t)$，定义在区间 $[r,t]$ [26] 上。均值向量场刻画了在给定区间内从噪声分布到数据分布的平均传输过程，它通过汇总不同时刻的瞬时去噪向量场来实现这一点。这与 FlowMatching 有本质区别，后者对瞬时去噪场本身进行建模。这一原理自然构成了指导神经网络训练的基本准则。为此，我们引入相应的损失函数，促使神经网络满足均值向量场与瞬时向量场之间这一内在关系。由于真实目标场已有明确的定义，因此无需额外的一致性约束，且理论上最优解与具体的网络架构无关。实际上，这使训练更加稳定、鲁棒。

我们将时间步 r 与 t 之间的平均向量场定义为该区间内的平均变换，通过对瞬时去噪向量场随时间进行聚合得到。形式化地，平均向量场 u 定义为

$$u(z_t,r,t) \triangleq \frac{1}{t-r}\int_r^t v(z_\tau,\tau)\,d\tau \tag{1}$$

z_t 表示在数据样本 x 与噪声样本 e 之间连续插值的中间潜变量。其定义为 $z_t = a_t x + b_t e$，其中 $e \sim \mathcal{N}(0, I)$。平均向量场记为 u，瞬时向量场记为 v。u 是对 v应用泛函得到的结果。

$$u = \mathcal{F}[v] \triangleq \frac{1}{t-r}\int_r^t v\,d\tau \tag{2}$$

直接采用式 (2) 中定义的平均向量场作为网络训练的真实标签并不可行，因此我们将式 (2) 重新表述为

$$(t-r)u(z_t,r,t) = \int_r^t v(z_\tau,\tau)\,d\tau \tag{3}$$

对等式两边关于 t 求导，可得一个相同的关系式。

$$u(z_t,r,t) + (t-r)\frac{d}{dt}u(z_t,r,t) = v(z_t,t) \tag{4}$$

因此，

$$u(z_t,r,t) = v(z_t,t) - (t-r)\frac{d}{dt}u(z_t,r,t) \tag{5}$$

该等式称为“MeanFlow 恒等式”，它描述了 v 与 u 之间的关系。式 (5) 中的最后一项可以改写为

$$\frac{d}{dt}u(z_t,r,t) = \frac{dz_t}{dt}\partial_z u + \frac{dr}{dt}\partial_r u + \frac{dt}{dt}\partial_t u \tag{6}$$

Let

$$\frac{dz_t}{dt} = v(z_t,t), \quad \frac{dr}{dt} = 0, \quad \frac{dt}{dt} = 1 \tag{7}$$

式 (7) 可以改写为

$$\frac{d}{dt}u(z_t,r,t) = v(z_t,t)\partial_z u + \partial_t u \tag{8}$$

式（8）可改写为矩阵乘法形式:

$$\frac{d}{dt}u(z_t,r,t) = [\partial_z u, \partial_r u, \partial_t u][v(z_t,t),0,1]^T \tag{9}$$

第一项是输出对各输入变量的导数，称为雅可比矩阵。第二项是瞬时向量场和常数。两者的乘积称为雅可比向量积（JVP），在 Jax 和 Torch 中都可以直接使用。

我们对神经网络 u_θ 进行参数化，使其满足 MeanFlow 恒等式，从而能够预测将样本从噪声分布输运至数据分布的平均向量场。

$$\mathcal{L}(\theta) = \mathbb{E}\left\|u_\theta(z_t,r,t) - sg(u_{\text{tgt}})\right\|_2^2 \tag{10}$$

在式（10）中，u_{tgt} 表示从真实轨迹获得的目标平均向量场

$$u_{\text{tgt}} = v(z_t,t) - (t-r)(v(z_t,t)\partial_z u_\theta + \partial_t u_\theta) \tag{11}$$

并应用停止梯度算子 $sg()$，以阻止梯度通过雅可比向量积项进行反向传播，从而避免高阶梯度计算。若不执行此操作，梯度将通过 u_θ 的导数传播，并进行二次反向传播。这会带来高昂的计算开销，也容易导致不稳定[12]。

MeanFlow 方法可以看作是采用了修改后目标的 FlowMatching。当 r 始终设为 t 时，它就退化为标准的 FlowMatching。每当 $r \neq t$，模型便获得单步生成的能力。在实践中，训练期间需要神经网络同时拟合瞬时向量场和平均向量场。

我们的目标是对数据分布 $p(A|O_t)$进行建模，其中 $A = [\alpha_t, \alpha_{t+1}, \ldots, \alpha_{t+H}]$ 表示未来动作序列， O 是观测特征。每个观测由多张 RGB图像、一条语言指令和机器人的本体感知状态组成，记为 $O_t = [I_t^1, \ldots, I_t^n, L_t, q_t]$，其中 I_t^i是第 i 张图像（每台机器人有两张或三张）， L_t 是语言词元序列， q_t是关节角度向量。图像 I 和状态 q 分别由相应的编码器编码，然后线性投影到与语言词元相同的嵌入维度。所得嵌入标记为（t， $t - r$），表示时间和区间信息。

对于动作块 A中的每个动作 α_t，我们都会获得一个由动作专家提供、以观测为条件的对应动作词元。在训练期间，这些词元由 MeanFlow损失进行监督，该损失取代了以往工作中采用的传统 FlowMatching损失。

$$\mathcal{L}(\theta) = \mathbb{E}\left\| u_\theta(A_t|O_t) - sg(u_{\text{tgt}}(A|O_t)) \right\|_2^2 \tag{12}$$

在基于 Mean-Flow 的单步 VLA 模型中，我们采用这一框架生成动作轨迹 A_t。随机采样时间步 r 和 t，以训练神经网络$u_\theta(A_t, r, t)$预测平均向量场，随后由该向量场将噪声 A_1 转换为动作 A_0 轨迹

$$\mathcal{L}(\theta) = \mathbb{E}\left\| u_\theta(A_t, r, t) - sg\left(u_{\text{tgt}}\right) \right\|_2^2 \tag{13}$$

B. 动作生成策略

在基于 Mean-Flow 的单步 VLA 模型中，动作生成策略的设计至关重要。为确保机器人能够在复杂环境中实时执行任务，我们采用由 MeanFlow 驱动的连续动作生成策略。如图 1 所示，视觉、语言和本体感知状态首先由视觉语言模型主干网络编码并融合，以生成统一多模态表征。随后，动作专家基于该表征执行单步去噪，直接生成完整的连续动作序列，以供立即执行。由此生成的轨迹既连续又符合动力学一致性。该设计的一大优势是能够根据任务条件的变化在线调整生成策略，从而快速响应新的视觉输入并提升实时控制性能。

在推理时，通过学习到的向量场从 $\tau = 0$ 到 $\tau = 1$ 生成动作。从随机噪声$A_1 \sim \mathcal{N}(0, I)$开始，单步生成得到

$$A_0 = A_1 - u_\theta(A_1|O_1) \tag{14}$$

如果采用多步生成，我们可以得到

$$A_{\gamma-\delta} = A_\gamma - \delta u_\theta(A_\gamma|O_\gamma) \tag{15}$$

其中 δ 是步长。

使用 MeanFlow模型进行采样，只需将时间积分替换为平均向量场

$$A_r = A_t - (t - r)u_\theta(A_t, r, t) \tag{16}$$

在单步采样的情况下，我们只需令

$$A_0 = A_1 - u_\theta(A_1, 0, 1) \tag{17}$$

IV. 实验

我们的实验分为两部分：超参数敏感性研究和消融研究。参数分析揭示了关键超参数对性能和稳定性的影响，以评估基于 Mean-Flow 的单步 VLA 模型背后的关键设计选择。在消融评估中，我们将我们的方法与 SmolVLA 和扩散策略 [13] 进行比较，涵盖三项真实操作任务。

模型架构：我们使用 SmolVLM-2 [27], ，这是一种高效的视觉-语言模型，专为多图像和视频输入而设计。它采用 SigLIP [28] 作为视觉编码器，采用 SmolLM-2 [29] 作为语言解码器，并通过视觉与文本词元之间的交叉注意力融合不同模态。

Mean-Flow动作专家：Mean-Flow动作专家u_θ 基于 Transformer [30] 架构构建，可根据视觉语言模型特征预测平均向量场，从而生成连续的动作轨迹 $A = [\alpha_t, \alpha_{t+1}, \ldots, \alpha_{t+H}]$。

环境设置：所有实验均使用一台自制的 SO-101机械臂进行，该机械臂具有六个自由度（DoF）和一个夹爪。我们使用两台 RGB相机覆盖工作空间：一台安装在机器人腕部，用于获取自我中心视角；另一台固定在工作台上方，用于获取俯视的第三人称视角，如图2所示。为确保数据集和推理条件一致，物理场景保持不变。经过 20 分钟的空闲期后测量动作生成速度。在此期间，CPU 和 GPU 温度均保持在 50 °C 以下。通过主动通风和稳定电源，将环境温度维持在约 15 °C。

任务设置：我们使用 SO-101 机械臂收集了三个特定于任务的数据集，每个数据集包含 100 个示范。对于每个示范，物体的初始位置和朝向均经过随机化，以确保场景配置的多样性。我们在三个真实操作任务上评估了我们的方法，即抓取放置、堆叠和分拣，如图2所示。

- 对于抓取放置任务，机器人被指示“抓取红色方块并将其放入盒子中”。立方体的初始位置和朝向在 10 cm× 10 cm 区域内随机化。评分标准为：成功抓取得 0.5 分，正确放置得 0.5 分。
- 对于堆叠任务，机器人被指示“将橙色玩具放在彩色方块上”。玩具的初始位置和朝向在相同的 10 cm × 10cm 区域内随机化。评分同样为：成功抓取得 0.5 分，成功堆叠得 0.5 分。
- 分拣任务涉及更长的时间跨度。机器人收到指令：“将红色立方体放入左侧盒子，并将

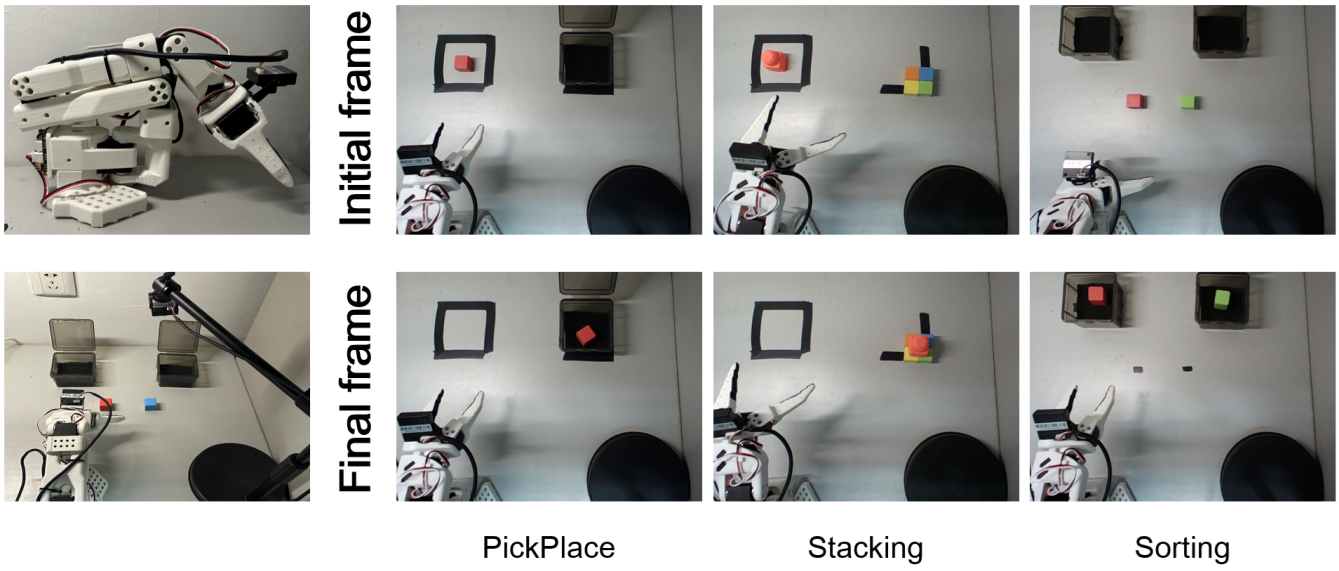

图2. 真实世界机械臂及三个真实操作任务的示意图。

绿色立方体放入右侧盒子”。盒子的位置保持固定，两个立方体的初始朝向均为随机。每成功抓取一次得 0.25 分，每次立方体与盒子匹配正确得 0.25 分。

训练设置：通过使用 Joy-Con 控制器手动远程操控机械臂，收集专家示范。每条轨迹持续至少 20 秒，以确保对任务执行过程有充分的时间覆盖。最终数据集包含 300 条轨迹。在每个时间步，模型接收单帧观测，其中包括多视角 RGB 图像、语言指令和机器人的本体感知状态。动作空间为 7 维，包括 6 自由度机械臂关节指令和 1 个夹爪控制信号。给定当前观测，模型以多步方式预测一段未来动作，从而实现时间连贯的运动生成。视觉语言模型主干网络被冻结，仅从头开始训练动作专家。在参数敏感性实验中，每个模型训练 60,000 步。在消融实验中，我们采用 200,000 步的多任务训练设置，而单任务训练则进行 60,000 步。训练在 NVIDIA RTX 4090 上进行。数据采集和在线推理在配备 Intel Core i5-10200H 和 NVIDIA GTX 1650 GPU 的笔记本电脑上执行。

A. 参数敏感性

我们使用一个简单的抓取放置任务，考察不同超参数对基于 Mean-Flow 的单步 VLA 性能的影响。对于每种超参数配置，我们进行 10 轮独立实验，每轮中模型连续尝试 20 次。实际上，基于 Mean-Flow 的单步 VLA 的最优超参数配置尚不明确，而直接在 1-NFE 设置下训练已被证明极不稳定，往往无法收敛。因此，我们首先在一种更易于拟合的 5-NFE 设置下开展实验，以分析流比例和损失度量对模型性能的影响。

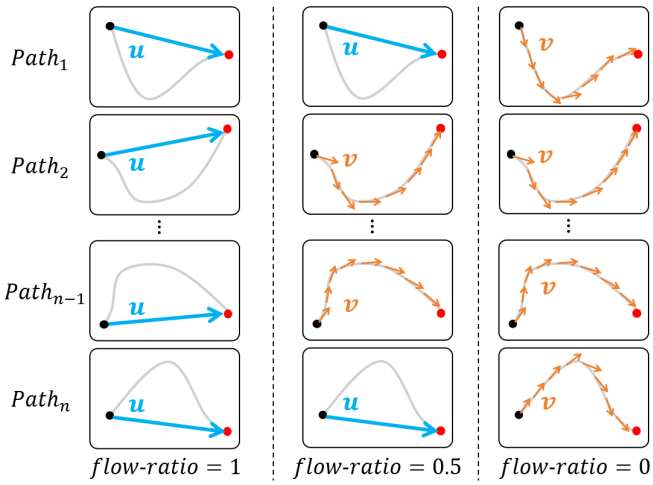

图3. 不同 $flow\text{-}ratio$ 下的 MeanFlow。

1) 流比例：在训练期间，随机采样时间步 r 和 t（$0 \le r < t \le 1$），使神经网络能够学习任意两个时刻之间的平均向量场。为加强时间连续性和向量场建模，我们引入一个流比例超参数，定义为

$$flow\text{-}ratio = 1 - \frac{N_{r=t}}{N_{total}} \tag{18}$$

$N_{r=t}$ 表示被强制设为相同值的样本数量，也就是说，训练目标退化为预测瞬时去噪向量场，与标准 FlowMatching 中的做法相同； N_{total} 是一个批次内采样的时间对总数。如图3所示。

事实上，当推理步数较少时，基于 FlowMatching 的 VLA 的动作生成质量会严重下降。

这一局限阻碍了函数评估次数的进一步减少。因此，在实验中，我们评估了 0.2、0.5 和 1.0 三种流比例，以考察在 NFE =5 设置下流比例对模型性能的影响。

在表 I 中，我们发现成功率在比例为 0.2 时达到最佳。此设置平衡了瞬时向量场与平均向量场的学习。仅使用平均向量场进行训练会丢失瞬时信息，并导致局部轨迹误差；而仅使用瞬时向量场进行训练，则难以保证模型在单步或少步采样下的整体正确性。换言之，瞬时向量场刻画动作轨迹中细粒度、短期的变化，而平均向量场捕捉粗粒度、长期的运动趋势。这些互补的表示共同构成了视觉运动演化的统一多尺度描述，使动作生成过程中既能实现精确的局部控制，又能保持稳定的全局行为。MeanFlow 恒等式包含一个时间导数校正项 –($t-r$)du/dt，其中 du/dt 通过雅可比向量积（JVP）计算（见式 (5)）。尽管对于当前神经网络参数，雅可比向量积在数值上是精确的，但它可能会放大长区间样本的梯度方差，并在训练期间引入优化不稳定性。引入一定比例的 $r = t$ 样本后，这些样本的校正项会变为零，从而为目标向量场提供局部锚点，帮助神经网络在高方差的平均流样本中维持稳定的参考。这可以提升训练鲁棒性和收敛稳定性。此外，在少步生成中，模型必须执行大跨度的跨区间变换并保持逐步精度。适度的 $r = t$ 采样能够为局部精度提供监督，并补充对跨区间平均向量场的学习，从而共同减少数值积分误差并提高任务成功率。

表 I 不同流比例下的成功率（%）（函数评估次数 = 5）。

比例	成功率
0.2	84.5
0.5	80.5
1.0	4.5

2）损失度量：尽管前述公式 (12) 中使用了 L_2 损失，但采用这一目标进行训练的效果不佳，而且在许多次运行中无法收敛。因此，我们将 L_2 损失替换为自适应损失，此后优化趋于稳定，最终准确率也显著提高。

$$\mathcal{L} = \frac{1}{B}\sum_{i=1}^{B}\left[\frac{1}{(\| \Delta_i\|^2 + c)^{1-\gamma}}\| \Delta_i\|^2\right] \tag{19}$$

在该公式中，$\| \Delta_i\|^2$ 表示第 i 个样本的平方误差之和，γ 是一个用于控制自适应程度的超参数，c是一个防止分母为零的小常数。

在表 II 中，我们通过三种设置（0.3、0.5 和 1.0）评估自适应损失超参数 γ 的影响。当 γ 接近 1.0 时，损失函数对异常值的降权作用最弱；在极限情况下，它会退化为普通的 L_2 损失。结果表明，L_2 损失（$\gamma = 1$）会严重降低性能，而当 $\gamma = 0.5$ 时，模型的成功率最高。

这一趋势可以用 VLA 数据的多模态特性以及 MeanFlow 目标的方差特征来解释。对于相同的视觉-语言条件，VLA 数据通常对应多种合理的动作模式。纯粹的 L_2 损失倾向于回归这些模式的平均方向，导致预测模糊或减弱。相反，过小的 γ 会过度抑制大误差样本，从而削弱对主导模式的监督。此外，由于 MeanFlow 目标包含一个与 ($t-r$) 成正比的时间导数修正项，较长时间间隔的样本会表现出更高的目标方差。由 γ 控制的自适应缩放通过降低不稳定和高方差样本的权重来缓解这些影响，从而稳定优化。

设置 $\gamma = 0.5$在鲁棒性与精确度之间实现了均衡的权衡。它抑制异常值的梯度贡献，同时保持对内点样本的有效监督，使优化器能够专注于主要分布。因此，训练过程更加稳定、收敛更快，并取得了最高的整体成功率。

表 II 不同伽马值下的成功率（%），取值为$flow\text{-}ratio$=0.2（函数评估次数 = 5）。

伽马	成功率
0.3	79.5
0.5	86
1.0	9.5

3）函数评估次数（*NFE*）：我们在表 III 中评估函数评估次数对模型性能的影响。当函数评估次数为 = 1 时，模型可以通过单步生成动作。然而，生成的轨迹精度较低。随着函数评估次数从 1 增至 5，任务成功率显著提高。将函数评估次数提高到 5，可生成明显更准确、更稳定的运动；进一步增加到 10 仅带来微小改进，这表明适中的评估次数在质量和计算成本之间实现了最佳平衡。

这一趋势可以从理论和实证视角加以解释。当函数评估次数 = 1时，模型必须预测一个全局平均向量场，一次性将整个分布从噪声映射到数据。理想情况下，模型能够完美学会这一全局映射；但在实践中，数据集汇集了来自多个任务的多样化示范，而且预测跨度很长（即模型在一次前向传播中预测最多 50 个未来动作步数），因此这种单步估计本质上很困难。

增加函数评估次数会有效地将时间区间 [0,1] 划分为更小的区段，促使模型学习较短区间内的局部平均向量场。每个局部目标都更简单，非线性也更弱。因此，必须在模型拟合误差与求解器积分误差之间取得平衡。

从 NFE = 1 到 NFE = 5 的提升，主要源于底层向量场的固有特性。真实的边缘流具有高度非线性，会产生曲线轨迹。多次评估使模型能够捕捉更简单的局部条件流和模式特定动力学，从而生成更平滑的轨迹并提高成功率。事实上，NFE = 5 提供了有效的折中方案，在保持 MeanFlow 高效性的同时显著提升了运动保真度；而进一步细分由于模型容量、数据集多样性和数值精度的限制，收益微乎其微。

表 III 不同 NFE 值下的成功率（%），其中$flow\text{-}ratio$=0.2，伽马=0.5。

NFE	1	2	3	4	5
成功率	49	48	51.5	79	84.5
NFE	6	7	8	9	10
成功率	83.5	83.5	86	84.5	85

4) 动作块大小：为了提高模型在单步生成设置（NFE=1）下的成功率，我们研究了动作块大小对整体性能的影响。动作块大小（n）定义了每个预测动作片段中连续时间步的数量。

如表 IV 所示，块大小过小或过大都会导致性能下降。我们认为，这一现象源于时间一致性与观测新鲜度之间固有的权衡。当块大小过大时（例如，n = 50），模型根据单次观测预测过长的未来动作序列，从而降低了对环境变化的响应能力，导致误差累积、控制精度下降。相反，当块大小过小时（例如，n = 1），模型过于频繁地更新动作，使得连续步骤间预测的动作仅有细微变化。结果，机器人执行的微动作几乎重复，无法取得实质性进展，成功率也接近于零。

在 $n = 20$ 时，模型在单步生成设置（NFE=1）下取得了最高性能。该参数在时间一致性与环境响应性之间实现了最佳平衡。生成的动作序列在时间上连贯且动态平滑，而单步生成过程能够实现快速推理，同时保持较高的任务成功率。

B. 消融研究

单步VLA 和 SmolVLA 使用混合多任务数据进行训练，而 DiffusionPolicy 则在

表 IV 不同动作块大小下的成功率（%），$flow\text{-}ratio$=0.2，伽马=0.5（函数评估次数 = 1）。

块大小	成功率
1	0
10	82
20	84.25
50	50.5

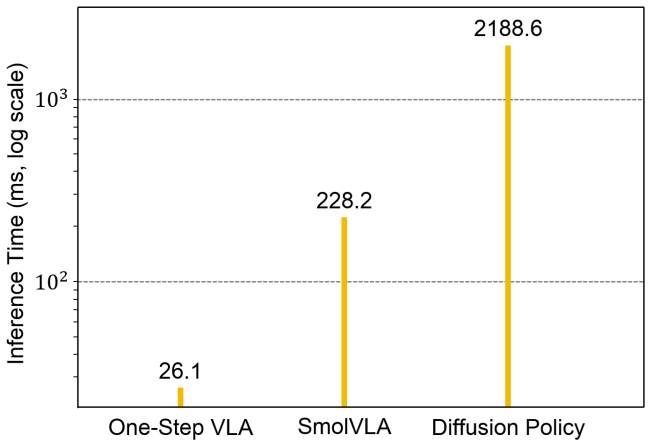

图 4. 扩散策略、SmolVLA 和单步VLA 在三个真实操作任务中的平均动作生成速度示意图。单步VLA 的速度是 SmolVLA 的 8.7 倍，是扩散策略的 83.9 倍。

单任务设置下进行训练。评估时，我们针对每项任务进行 10 轮独立测试，每轮包含 10 次试验，并报告平均成功率和平均动作生成时间。

在单步配置（函数评估次数=1）下，所提出的基于 Mean-Flow 的单步 VLA（$flow\text{-}ratio$=0.2，γ=0.5，块大小=20）展现出显著的效率优势。如图 4 和表 V 所示，单步VLA 的速度分别是 SmolVLA 和扩散策略的 8.7 倍和 83.9 倍，同时保持了有竞争力的任务成功率。

这种加速源于生成机制的根本差异。扩散策略依赖迭代去噪步骤，每个动作块都需要进行多次神经网络评估。基于 FlowMatching 的 VLA（例如 SmolVLA）降低了计算成本，但仍依赖多步欧拉积分，每一步都需要对模型进行一次前向传播。相比之下，单步VLA预测整个时间区间内的平均向量场，从而无需迭代细化即可通过单次前向传播直接生成动作。事实上，消除重复的求解器评估可大幅降低计算开销，实现真正的单步控制。

我们进一步发现，SmolVLA 无法在函数评估次数=1 下可靠运行，除非引入额外的一致性正则化；否则，轨迹幅度会变得不稳定，不适合部署。

在堆叠任务中，单步VLA的成功率略低于SmolVLA，这可能是因为任务要求较高的

精确度，而多步细化能够进行更细微的调整。不过，对于拾放和长时序分拣任务，单步VLA在保持相当性能的同时，显著加快了推理速度，有效缩小了单步与多步动作生成之间的差距。

表 V 三种模型在三种不同任务上的成功率（%）。

策略	拾放	堆叠	分拣	Avg.
单任务训练				
扩散策略	78.5	46	60	61.5
多任务训练				
SmolVLA	86.5	81.5	85.5	84.5
单步VLA	88	64	82	78

V. 结论

本文提出了基于 Mean-Flow 的单步 VLA，这是一种新型视觉-语言-动作框架，可大幅提高生成速度，并支持单步生成动作序列。该模型预测平均向量场，而非瞬时向量场，因此只需评估一次即可输出动作，实现单步推理。真实机器人实验表明，所提出的框架仅通过一次前向传播即可生成动作序列；与 SmolVLA 和扩散策略相比，其推理速度分别快 8.7 倍和 83.9 倍，同时保持了相当的成功率。

未来的工作中，我们将专注于提升基于 Mean-Flow 的单步 VLA 的泛化能力和鲁棒性，尤其是应对更复杂的任务。

参考文献

[1] R. Sapkota, Y. Cao, K. I. Roumeliotis 和 M. Karkee，“视觉-语言-动作模型：概念、综述、进展、应用与挑战”，*arXiv*预印本 *arXiv:2505.04769*，2025。[2] K. Kawaharazuka, J. Oh, J. Yamada, I. Posner 和 Y. Zhu，“面向机器人领域的视觉-语言-动作模型：迈向现实世界应用的综述”，*IEEE Access*，2025。[3] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu，等，“Rt-1：用于大规模现实世界控制的机器人Transformer”，*arXiv*预印本 *arXiv:2212.06817*，2022。[4] B. Zitkovich, T. Yu, S. Xu, P. Xu, T. Xiao, F. Xia, J. Wu, P. Wohlhart, S. Welker, A. Wahid，等，“Rt-2：视觉-语言-动作模型将网络知识迁移至机器人控制”，载于机器人学习会议，第2165–2183页，PMLR，2023。[5] R. Shao, W. Li, L. Zhang, R. Zhang, Z. Liu, R. Chen 和 L. Nie，“基于大型视觉语言模型的视觉-语言-动作机器人操作模型：综述”，*arXiv*预印本 *arXiv:2508.13073*，2025。[6] J. Ho, A. Jain 和 P. Abbeel，“去噪扩散概率模型”，神经信息处理系统进展，第33卷，第6840–6851页，2020。[7] O. M. Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, T. Kreiman, C. Xu，等，“Octo：一种开源通用机器人策略”，*arXiv*预印本 *arXiv:2405.12213*，2024。[8] Q. Vuong, S. Levine, H. R. Walke, K. Pertsch, A. Singh, R. Doshi, C. Xu, J. Luo, L. Tan, D. Shah，等，“Open X-Embodiment：机器人学习数据集与RT-X模型”，载于迈向通用机器人：可扩展技能习得的学习范式@ *CoRL 2023*，2023。

[9] X. Liu, C. Gong 和 Q. Liu，“流动笔直且快速：通过整流流学习生成和迁移数据”，arXiv预印本 arXiv:2209.03003，2022。[10] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter 等，“π_0：一种用于通用机器人控制的视觉-语言-动作流模型”，arXiv预印本 arXiv:2410.24164，2024。[11] K. Frans, D. Hafner, S. Levine 和 P. Abbeel，“通过捷径模型实现单步扩散”，arXiv预印本 arXiv:2410.12557，2024。[12] Z. Geng, M. Deng, X. Bai, J. Z. Kolter 和 K. He，“用于单步生成建模的均值流”，arXiv预印本 arXiv:2505 13447，2025。[13] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake 和 S. Song，“扩散策略：通过动作扩散进行视觉运动策略学习”，国际机器人研究期刊，第44卷，第10–11期，第1684–1704页，2025。[14] J. Wen, M. Zhu, Y. Zhu, Z. Tang, J. Li, Z. Zhou, C. Li, X. Liu, Y. Peng, C. Shen 等，“Diffusion-VLA：通过自生成推理实现可泛化且可解释的机器人基础模型”，arXiv预印本 arXiv:2412.03293，2024。[15] M. S. Albergo 和 E. Vanden-Eijnden，“利用随机插值构建归一化流”，arXiv预印本 arXiv:2209.15571，2022。[16] D. Zhang, J. Sun, C. Hu, X. Wu, Z. Yuan, R. Zhou, F. Shen 和 Q. Zhou，“纯视觉语言动作（VLA）模型：全面综述”，arXiv预印本 arXiv:2509.19012，2025。[17] M. Shukor, D. Aubakirova, F. Capuano, P. Kooijmans, S. Palma, A. Zouitine, M. Aractingi, C. Pascal, M. Russi, A. Marafioti 等，“SmolVLA：一种经济高效的机器人视觉-语言-动作模型”，arXiv预印本 arXiv:2506.01844，2025。[18] C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li 和 J. Zhu，“DPM-Solver：一种用于扩散概率模型采样、约10步即可完成的快速ODE求解器”，神经信息处理系统进展，第35卷，第5775–5787页，2022。[19] M. Zhou, H. Zheng, Z. Wang, M. Yin 和 H. Huang，“分数恒等蒸馏：对预训练扩散模型进行指数级快速蒸馏，以实现单步生成”，载于第四十一届国际机器学习会议，2024。[20] T. Salimans 和 J. Ho，“用于扩散模型快速采样的渐进式蒸馏”，arXiv预印本 arXiv:2202.00512，2022。[21] W. Luo, T. Hu, S. Zhang, J. Sun, Z. Li 和 Z. Zhang，“Diff-Instruct：从预训练扩散模型迁移知识的通用方法”，神经信息处理系统进展，第36卷，第76525–76546页，2023。[22] Y. Song, P. Dhariwal, M. Chen 和 I. Sutskever，“一致性模型”，2023。[23] C. Lu 和 Y. Song，“简化、稳定并扩展连续时间一致性模型”，arXiv预印本 arXiv:2410.11081，2024。[24] L. Yang, Z. Zhang, Z. Zhang, X. Liu, M. Xu, W. Zhang, C. Meng, S. Ermon 和 B. Cui，“一致性流匹配：通过速度一致性定义直线流”，arXiv预印本 arXiv:2407 02398，2024。[25] Y. Lipman, R. T. Chen, H. Ben-Hamu, M. Nickel 和 M. Le，“用于生成式建模的流匹配”，arXiv预印本 arXiv:2210 02747，2022。[26] J. Sheng, Z. Wang, P. Li 和 M. Liu，“MP1：MeanFlow让策略学习在机器人操作中一步完成”，arXiv预印本 arXiv:2507.10543，2025。[27] A. Marafioti, O. Zohar, M. Farr´e, M. Noyan, E. Bakouch, P. Cuenca, C. Zakka, L. B. Allal, A. Lozhkov, N. Tazi 等，“SmolVLM：重新定义小型高效多模态模型”，arXiv预印本 arXiv:2504.05299，2025。[28] X. Zhai, B. Mustafa, A. Kolesnikov 和 L. Beyer，“用于语言-图像预训练的sigmoid损失”，载于IEEE/CVF国际计算机视觉会议论文集，第11975–11986页，2023。[29] L. B. Allal, A. Lozhkov, E. Bakouch, G. M. Bl´azquez, G. Penedo, L. Tunstall, A. Marafioti, H. Kydlí˘cek, A. P. Lajar´ın, V. Srivastav 等，“SmolLM2：Smol如何发展壮大——以数据为中心的小语言模型训练”，arXiv预印本 arXiv:2502.02737，2025。[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser 和 I. Polosukhin，“注意力就是一切”，神经信息处理系统进展，第30卷，2017。