KDPE：一种用于扩散策略轨迹选择的核密度估计策略

Andrea Rosasco[1,2] Federico Ceola[1] Giulia Pasquale[1] Lorenzo Natale[1]
[1]Istituto Italiano di Tecnologia, [2]University of Genoa
{ andrea.rosasco, federico.ceola, giulia.pasquale, lorenzo.natale } @iit.it

Abstract: 学习能够捕捉训练数据中的多模态性的机器人策略一直是行为克隆中的一个长期开放挑战。近期的方法通过使用生成模型来建模条件动作分布从而解决这个问题。这些方法之一是扩散策略，它依赖扩散模型将随机点去噪成为机器人动作轨迹。虽然实现了最先进的性能，但它有两个主要缺点，这可能导致机器人在策略执行期间偏离数据分布。首先，去噪过程的随机性可能极大影响生成动作轨迹的质量。其次，作为一种监督学习方法，它可能从用于训练的数据集中学习到数据异常。最近的工作集中于通过将扩散策略与大规模训练或经典行为克隆算法结合来缓解这些限制。相反，我们提出了 KDPE，一种基于核密度估计的策略，它过滤掉扩散策略产生的潜在有害轨迹，同时保持低测试时计算开销。对于核密度估计，我们提出了一种流形感知核来建模由末端执行器笛卡尔位置、方向和夹爪状态组成的动作的概率密度函数。总体而言，KDPE 在模拟单臂任务和真实机器人实验中比扩散策略表现更好。有关更多材料和代码，请访问我们的项目页面：https://hsp-iit.github.io/KDPE/ 。

Keywords: Behavior Cloning, Manipulation, Trajectory Selection.

1 引言

扩散策略 (DP) [1] 最近由于其处理多模态行为的能力而成为一种强大的机器人策略表示。DP 将机器人策略建模为去噪扩散概率模型 (DDPM) [2]，在策略执行期间，将一组随机采样的轨迹点去噪为机器人被请求执行的轨迹，并通过退化视野控制 [3]。虽然这种方法使机器人能够捕捉用于训练的演示中的多模态性，但在推理时选择随机点来初始化去噪过程可能会导致机器人在相同观察下执行不同的轨迹。如果采样的轨迹相对于训练数据中最具代表性的模式是异常值，则这可能会导致机器人进入一个可能超出演示数据集分布的状态。

为了克服这一限制，我们提出了 KDPE：一种抽样策略，用以抽取能够代表策略所学模式的轨迹。我们建议计算一组 N 轨迹，通过并行执行 N 去噪过程，从相同的当前观察条件下的 N 个不同起始噪声样本生成轨迹。然后，使用核密度估计（KDE）估计由扩散策略生成的轨迹中最后一个动作的概率密度函数（PDF），并选择与密度最高的动作相关联的轨迹。

我们考虑的动作由末端执行器姿态和夹持器状态组成。在一组动作上建模核密度估计（KDE）需要一个能关联末端执行器位置、方向和夹持器状态的核函数。虽然之前已经提出了关于末端执行器姿态的概率分布 [4, 5]，但它们都没有整合夹持器状态。我们修改了在 [4] 中引入的概率分布以处理夹持器状态组件。

原则上，由 DP 预测的轨迹的多模态性可以用非参数聚类算法来建模 [6]。然而，这些方法在用于闭环控制任务中的异常值拒绝 [7] 时，需要不可忽视的收敛时间。相反，KDPE 能够在 DP 预测的轨迹子集之间进行过滤和采样，以显著降低计算开销，使其适合目标视觉运动闭环控制任务。虽然 KDPE 使用 DP 进行轨迹生成，但通过 KDE 对多个轨迹进行采样和过滤的核心思想可以应用于任何概率生成策略。

我们在使用 [1] 对 DP 进行基准测试的四个 RoboMimic [8] 单臂任务以及 MimicGen 基准测试 [9] 中的三个任务上评估 KDPE 与 DP 的性能。我们在这些任务上训练 DP，并使用

9th Conference on Robot Learning (CoRL 2025), Seoul, Korea.

www.xueshuxiangzi.com

KDPE 评估模型，取得了平均成功率的整体提升。然后，我们通过更改任务环境中物体的颜色来评估 KDPE 对视觉扰动的鲁棒性，并显示 KDPE 在性能上比 DP 更接近非偏移设置。

最后，我们使用 Franka Emika Panda [10] 机械臂，在三个实际任务中测试 KDPE：Pick-Plush，这是一个台面抓取任务，机器人需要抓取并提升毛绒玩具；CubeSort，这是一个多步骤多模式任务，机器人需要挑选两个立方体并将它们放入对应颜色的杯子中；以及 CoffeeMaking，这是一个精细的多步骤任务，机器人需要用咖啡胶囊填装咖啡机。总的来说，KDPE 在成功率方面优于 DP，并在质量上表现出更流畅的行为。

总之，本文的贡献有：

- KDPE：一种基于 KDE 的方法，用于选择具有代表性不同动作模式的 DP 动作轨迹。
- 一种流形感知核用于机器人动作的 KDE，其中包括欧几里得末端执行器位置和夹爪开口，以及非欧几里得末端执行器方向。
- 对 KDPE 进行扩展的定量评估，包括来自两个不同基准的七个模拟任务和三个现实世界任务。
- 用于操控策略定性分析的轨迹可视化工具。

2 相关工作

机器人领域正在投入大量精力通过大规模数据集 [11, 12] 创建通用机器人策略 [13, 14, 15]，可以是基于大型 transformer 模型 [16, 15, 13, 17, 18]，也可以将机器人策略参数化为语言符号 [19, 14, 20]。然而，对于单一任务，有证据表明 [14]，DP [1] 比大型通用模型 [13, 14] 具有更好的性能。

DP [1]，与其流匹配变体 [21] 一起，目前是用于机器人操作任务行为克隆的最广泛使用的方法之一。将其策略参数化为去噪扩散概率模型（DDPM）可以对训练数据中的多模态进行建模。最近的一些工作扩展了原始的 DP 问题公式，以学习目标条件 [22] 或语言条件 [23, 24] 任务。相反，Octo [13] 使用 DP 来在一个基于大型变压器模块化骨干的任务和体现间传递性上参数化一个通用的机器人操作策略。诸如 [25] 中所展示的方法，将 DP 与更经典的行为克隆算法如 DAgger [26] 结合，以减少纯模仿学习算法典型的累积误差。

最近的工作探索了故障检测和缓解策略，以增强机器人策略的可靠性。Sentinel [27] 通过分析在不同时间步采样的重叠轨迹段的一致性来识别潜在故障，并利用视觉-语言模型检测动作与视觉内容之间的不一致。尽管有效，这种方法通常在策略已经进入分布外状态后才检测到故障，使得恢复变得具有挑战性。RoboMD [28] 通过在多样化环境条件下进行 DRL 引导探索系统性地识别故障模式来解决鲁棒性问题。RoboMD 在这些识别的故障模式上微调策略。然而，这种训练-部署-分析-再训练的流程在现实世界部署中可能不切实际。V-GPS [29] 通过使用价值函数网络对动作进行排名来在推理时改善机器人策略。然而，这需要使用离线强化学习训练一个价值函数网络并设计奖励函数。相比之下，KDPE 在推理时无需任何额外的训练。通过利用 DP 的轨迹采样过程的固有多样性，KDPE 主动减轻故障，选择与学习分布的统计模式对齐的轨迹，而不是依赖事后检测或昂贵的再训练。

自然语言处理领域中越来越多的文献研究推理或测试时缩放方法，这些方法从生成的多个输出样本中选择一个（例如，通过大型语言模型生成）以提高训练模型的性能 [30, 31, 32]。此外，最近的研究关注于推理时缩放在比如文本到图像扩散模型中的应用 [33, 34]。在 [35] 中，实验验证了最简单的从 N 个中选最优（通过生成大量随机样本并使用外部评估器选择最佳样本）的方法的有效性。然而，据我们所知，KDPE 是首个基于轨迹分布的统计性质在推理时改进 DP 的方法。

www.xueshuxiangzi.com

3 方法论

3.1 扩散策略

DP 在 DDPM 的基础上进行改进，DDPM 是一种常用于图像生成的生成扩散模型，DP 将其适应为一种行为克隆，以退缩视界方式应用于机器人操作任务。DP 在给定环境观察的情况下，输出形状为 $T \times D$ 的动作轨迹，其中 T 是时间步数，D 是动作的维度。这些轨迹是通过一个去噪过程生成的，该过程通过迭代地减去一个学习到的梯度场来逐步优化高斯噪声。该过程允许策略建模多模态轨迹分布，并维持任务演示的多模态性。生成轨迹的去噪过程定义为：

$$\boldsymbol{A}^{k-1} = \alpha(\boldsymbol{A}^k - \gamma\epsilon_\theta(\boldsymbol{A}^k, k) + \mathcal{N}(0, \sigma^2 I)), \tag{1}$$

，其中 ϵ_θ 是噪声预测网络，参数为 θ ，$\boldsymbol{A}^k$ 是经过去噪步骤的噪声样本。该过程重复进行 K 步，初始从 $\boldsymbol{A}^K$ 随机采样高斯噪声，输出轨迹 $\boldsymbol{A}^0$ 。在训练期间，轨迹从数据集中采样，并通过加入与随机去噪步骤 k 对应的适量噪声进行扰动。噪声预测网络 ϵ_θ 以噪声样本作为输入，并被优化以预测已添加到真实轨迹 $\boldsymbol{A}^0$ 上的噪声 ϵ^k ，具有以下损失函数：

$$\mathcal{L} = MSE(\epsilon^k, \epsilon_\theta(\boldsymbol{A}^0 + \epsilon^k, k)). \tag{2}$$

。

3.2 键区能量

KDPE 通过利用核密度估计（KDE）对轨迹群进行评分，并通过 N 最优抽样选择最佳轨迹，来增强扩散策略。我们的方法将多重假设抽样与扩散过程相结合，并应用对流形敏感的核。给定观测 $\boldsymbol{o}_t$ ，我们从扩散过程中对独立同分布的动作轨迹 $\{\boldsymbol{A}_i\}_{i=1}^N \sim p_{\boldsymbol{\theta}}(\boldsymbol{A}|\boldsymbol{o}_t)$ 进行抽样，其中 $\boldsymbol{A}_i \in \mathbb{R}^{T\times D}$ ，并通过 KDE 估计轨迹最后动作的 PDF，此后简称为 $\boldsymbol{a}_i \in \mathbb{R}^D$ 。这使得我们可以获得每个动作的概率密度，并利用它来剔除异常轨迹。

为了对 PDF 建模，KDE 需要一个统一的核覆盖所有动作组件，这些组件代表末端执行器的位置和方向，以及夹具开口。虽然多元高斯核是欧几里得领域的自然选择，但它不能直接处理位于不同流形上的数据，例如 SO(3) 中的旋转，这是一个非欧几里得空间。

流形感知核密度估计我们提出了一个关于 [4] 中概率分布的修改，以同时处理末端执行器的姿态和夹具状态。虽然 DP 在 [36] 中介绍的 6D 矩阵表示中输出方向，为了清晰起见，我们考虑它们在 SO(3) 中的矩阵表示。我们的核函数由以下方程定义

$$k(\boldsymbol{a}_i, \boldsymbol{a}_j) = \frac{1}{\sqrt{(2\pi)^D|\boldsymbol{H}|}} \exp\left(-\frac{1}{2}\boldsymbol{\Delta}_{ij}^\top \boldsymbol{H}^{-1}\boldsymbol{\Delta}_{ij}\right), \tag{3}$$

其中 $\boldsymbol{a}_i = [\boldsymbol{t}_i; \boldsymbol{R}_i; \boldsymbol{g}_i]$ 和 $\boldsymbol{a}_j = [\boldsymbol{t}_j; \boldsymbol{R}_j; \boldsymbol{g}_j]$ 是由位置、旋转和夹具开口组件构成的动作。$\boldsymbol{H}$ 是协方差矩阵，其定义为 $\boldsymbol{H} = \mathrm{diag}\big(\sigma_{\text{pos}}^2\boldsymbol{I}_3, \sigma_{\text{rot}}^2\boldsymbol{I}_3, \sigma_{\text{grip}}^2\boldsymbol{I}_1\big)$ ，而 $\boldsymbol{\Delta}_{ij} \in \mathbb{R}^D$ 表示流形特定组件之间的差异定义为：

$$\boldsymbol{\Delta}_{ij} = [\underbrace{\boldsymbol{t}_i - \boldsymbol{t}_j}_{\text{Euclidean}}; \underbrace{\log(R_j^T R_i)^\vee}_{\text{SO(3)}}; \underbrace{\boldsymbol{g}_i - \boldsymbol{g}_j}_{\text{Euclidean}}] \tag{4}$$

对于位置和夹具组件，标准欧几里得差异就足够了。对于表示为旋转矩阵 $R_i, R_j \in \mathrm{SO}(3)$ 的旋转，我们计算从 R_j 到 R_i 的变换，并使用 Lie 群对数映射 $\log : \mathrm{SO}(3) \to \mathfrak{so}(3)$ 将其转换为轴角表示，随后使用 vee 操作符 $(\cdot)^\vee : \mathfrak{so}(3) \to \mathbb{R}^3$ 以获得切空间中的表示。根据上述定义，我们可以将 $-\frac{1}{2}\boldsymbol{\Delta}_{ij}^\top \boldsymbol{H}^{-1}\boldsymbol{\Delta}_{ij}$ 重写为

$$-\frac{1}{2}\left(\frac{\|\boldsymbol{t}_i - \boldsymbol{t}_j\|^2}{\sigma_{\text{pos}}^2} + \frac{\|\log(R_j^\top R_i)^\vee\|^2}{\sigma_{\text{rot}}^2} + \frac{(\boldsymbol{g}_i - \boldsymbol{g}_j)^2}{\sigma_{\text{grip}}^2}\right), \tag{5}$$

其中 $\|\log(R_j^\top R_i)^\vee\|^2$ 是表示最小旋转角 θ_{ij} 的测地距离，所需的旋转角度用于使 R_i 和 R_j 对齐：

$$d_{SO(3)}(R_i, R_j) = \|\log(R_j^\top R_i)^\vee\|^2 = \|\boldsymbol{\theta_{ij}}\| = \theta_{ij}.$$

www.xueshuxiangzi.com

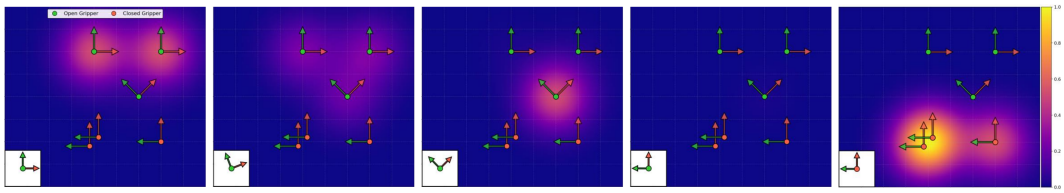

Figure 1: 使用所提出的流形感知核函数进行 KDE 估计 PDF 的可视化。我们对 6 个以参考框架表示为图中的平面末端执行器动作群体进行 KDE。其中三个代表开夹具（绿色圆圈），而另外三个代表闭夹具（红色圆圈）。热图中每个点的颜色表示某动作在对应二维位置上的密度值，该位置的旋转和夹具状态由每个图中的白色正方形内的指示框显示。从左到右，我们将指示框的旋转从 0 度变化到 90 度，并观察到在不同位置由 KDE 返回的密度相应地变化，在探测动作接近用于 PDF 建模的点时出现峰值。图中显示 KDE 如何通过为最具代表性的样本提供最高密度值来正确处理多模态性。最右边的两个图展示了 KDE 如何正确处理夹具状态。

在图 1 中，我们展示了所提出的流形感知核如何允许对包括末端执行器位置和方向以及夹具开口的动作进行 PDF 建模。

密度估计和动作选择我们通过在所有动作中结合核密度，使用 KDE 来对由 DP 预测的 N 动作 $\{\boldsymbol{a}_i\}_{i=1}^N$ 的基础概率密度函数进行建模。具体地，我们计算通用动作 $\tilde{\boldsymbol{a}}$ 的密度为：

$$\rho(\tilde{\boldsymbol{a}}) = \frac{1}{N}\sum_{i=1}^{N} k(\tilde{\boldsymbol{a}}, \boldsymbol{a}_i) \tag{6}$$

最后，我们为 DP 预测的每一个 N 个最终动作计算密度 $\rho_i = \rho(\boldsymbol{a}_i)$ ，并选择与具有最高 ρ_i 的动作 $\boldsymbol{a}_i$ 对应的轨迹 $\boldsymbol{A}_i$ 。

4 实验设置

4.1 模拟环境

我们在来自两个不同基准的七个任务上评估 KDPE：RoboMimic [37] 和 MimicGen [9] 。我们考虑了四个 RoboMimic 任务：Lift、Can、Square 和 ToolHang（如图 2 所示）。此 RoboMimic 任务子集对应于用于评估 DP 的单臂基于图像的任务 [1] 。RoboMimic 为 Lift、Can 和 Square 提供了两个数据集：一个由熟练的人类操作员（ph）收集，另一个由熟练和非熟练操作员混合（mh）收集，后者的数据质量较低。对于 ToolHang，则仅有 ph 数据集可用。我们还在三个 MimicGen 任务上测试了 KDPE：Coffee、Stack Three（为简洁起见称为 Stack）和 Three Piece Assembly（Assembly）。我们选择这个基准是因为它与 RoboMimic 共享相同的结构，使其能与评估框架无缝集成，并允许我们在其他具有挑战性的任务上测试 KDPE。

为了评估 KDPE 在视觉领域偏移下的鲁棒性，我们将其性能与 DP 进行比较，DP 是在原始环境扰动变体上稍微修改对象的颜色。具体来说，我们从特定对象上移除原有纹理，并将其新颜色设为原始纹理的平均值，亮度值在 HSL 色彩方案下减少了 10 %。为此实验而修改的特定对象在附录 A 中显示。

为了将 KDPE 与扩散策略 (DP) 进行比较，我们对 DP 进行了 $80k$ 次训练。在每个实验设置中，我们对环境进行 100 次随机重置以运行这些方法。在实验过程中，环境初始化的随机序列和 DDPM 噪声计划保持不变。在策略展开过程中，我们在每个推理步骤中采集 100 条轨迹样本，从中为每种方法不同地选择一条轨迹。DP 基线方法从样本族中均匀采样输出轨迹。对于 KDPE，我们对这些轨迹的第八个动作执行 KDE，因为在 DP 中使用的是八步动作执行范围。如果选择方法是非确定性的，比如 DP 的情况，我们将完整的轨迹集运行三次并报告其平均结果。

除了与 DP 的比较之外，我们还报告了两个 KDPE 修改版本的性能：KDPE-OOD 和 Tr-KDPE。KDPE-OOD 是 KDPE 的一种修改，在其中我们选择与代表性最少的动作相关的轨迹，即具有最小密度 $\{\rho_i\}_{i=1}^N$ 的动作（参见 Sec. 3.2 ）。我们评估这种方法以进一步支持在

www.xueshuxiangzi.com

Figure 2: RoboMimic（提起、罐子、方块和工具挂）和 MimicGen（咖啡、堆叠和组装）任务被考虑用于 KDPE 的评估。

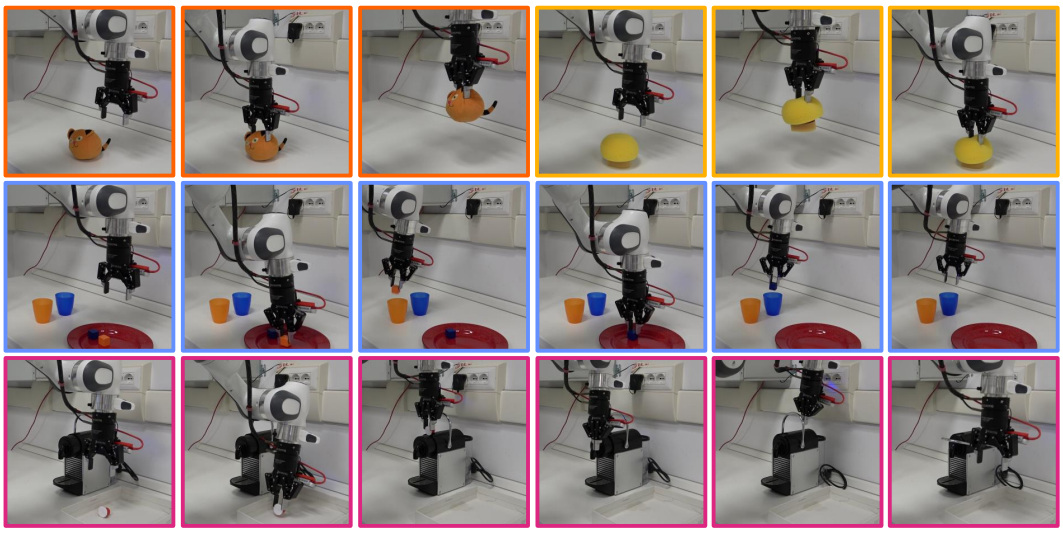

Figure 3: KDPE 自主执行真实机器人任务：PickPlush（橙色边框）及其变体 PickSponge（黄色边框），CubeSort（蓝色边框）和 CoffeeMaking（紫色边框）。

DP 输出轨迹中拒绝异常值的需求。而 Tr-KDPE 是 KDPE 的一种修改，它使用条件 KDE 和一阶 Markov 假设来估计整条轨迹群体的 PDF。我们在 App. B 中报告了 Tr-KDPE 的细节。

我们展示了基于 CNN（DP-C、KDPE-C、KDPE-OOD-C、Tr-KDPE-C）和基于 Transformer（DP-T、KDPE-T、KDPE-OOD-T、Tr-KDPE-T）的模型的结果。对于所有方法，我们使用相同的超参数，例如关联在附录 C 中的 KDE 核带宽。

4.2 真实机器人

我们使用 Meta Quest 3 VR 头显远程操作配备 Robotiq 2f-85 抓手的 Franka Emika Panda 机械臂来收集数据集以训练策略。我们为机器人提供来自安装在机器人手腕上的 Intel(R) RealSense D415 和外部 Intel(R) RealSense D405 的视觉 RGB 观测（图 3）。我们收集了 50 个 PickPlush 的演示、135 个 CubeSort 的演示以及 200 个 CoffeeMaking 的演示。

我们首先在 PickPlush（图 3）上测试 DP 和 KDPE，PickPlush 是一项抓取任务，包括从桌子上抓起和提起一个猫娃娃。此外，模仿在模拟中进行的视觉域转换实验，我们在一个版本的 PickPlush 上测试相同的检查点，其中娃娃被替换为一个黄色海绵，称为 PickSponge。然后，我们将 DP 和 KDPE 在 CubeSort 上进行 100 集进行比较。机器人需要从一个红色盘子上抓取一个蓝色和一个橙色的立方体，并将它们放入相应颜色的杯子中。这要求策略学习一个多模式高度的任务，无论是在立方体被挑选的顺序上还是在它们被操控的方式上。在任务演示中，这两个立方体是以随机顺序被抓取的，并且在模型评估期间，立方体在每个集开始时被随机投掷到盘子上。最后，我们在 CoffeeMaking 任务上评估这两种方法进行 10 集，这是一项现实世界的实验，用于了解 KDPE 在需要长时间规划能力和相当精确度的现实场景中的影响。咖啡制作任务要求机器人完成以下步骤：拿起咖啡胶囊，在视觉条件不佳的情况下将胶囊插入咖啡机，推动咖啡机前部以关闭它，然后拉下金属手柄。其中一些步骤需要高度精确，例如插入胶囊，可以从图 3 看出，任务由 KDPE 自主完成。

www.xueshuxiangzi.com

	Lift		Can		Square		ToolHang	Coffee	Stack	Assembly	Average
	ph	mh	ph	mh	ph	mh	ph				
DP-C	100	100	96.7	94.0	93.0	83.0	62.0	85.0	80.7	54.3	84.9
KDPE-OOD-C	100	100	96.0	84.0	83.0	49.0	7.0	85.0	72.0	40.0	71.6
Tr-KDPE-C	100	100	97.0	94.0	90.0	78.0	68.0	76.0	78.0	61.0	84.2
KDPE-C	100	100	98.0	96.0	92.0	86.0	76.0	88.0	85.0	61.0	88.2
DP-T	100	100	96.0	90.0	84.0	73.0	61.3	91.7	73.7	23.3	79.3
KDPE-OOD-T	100	99.0	95.0	91.0	84.0	65.0	2.0	93.0	61.0	23.0	71.3
Tr-KDPE-T	100	99.0	93.0	92.0	81.0	81.0	62.0	86.0	74.0	18.0	78.6
KDPE-T	100	100	97.0	92.0	88.0	83.0	66.0	91.0	72.0	23.0	81.2

Table 1: DP、KDPE-OOD、Tr-KDPE 和 KDPE 在 RoboMimic 和 MimicGen 上的平均成功率（%）。最后一列报告了所有任务的平均成功率。

	Lift		Can		Square		ToolHang	Coffee	Stack	Assembly	Average
	ph	mh	ph	mh	ph	mh	ph				
DP-C	100	99.6	92.3	91.3	93.7	79.7	59.0	92.3	75.3	44.7	82.8
KDPE-C	100	100	94.0	90.0	94.0	90.0	69.0	97.0	88.0	50.0	87.2
DP-T	98.7	100	89.0	84.0	79.3	68.3	58.0	73.3	67.7	15.6	73.4
KDPE-T	99.0	100	90.0	85.0	75.0	77.0	62.0	72.0	68.0	20.0	74.8

Table 2: DP 和 KDPE 在物体颜色扰动下的成功率（%）。

根据第 ?? 节中描述的方法，我们从大小为 480 × 480 的图像中训练基于 CNN 版本的 DP 模型 80k 步骤，并在同一检查点上评估 DP 和 KDPE，使用 DDIM [38] 作为噪声调度器以加快推理速度。

5 结果

5.1 RoboMimic 和 MimicGen 任务的基准测试

在表中报告的基准测试结果表明，KDPE 在两种架构上均优于 DP，并且 KDPE-C 整体上是表现最好的方法，成功率平均达到了 88.2%。当使用 CNN 骨干网络时，KDPE 比 DP 提高了 3.3%，而在基于 Transformer 的模型中提高了 1.9%。如果我们从分析中排除升降任务（其成功率在两种方法中都达到了 100%），这些差距更加明显（分别为 4.2% 和 2.4%）。还值得注意的是，在三个 DP 优于 KDPE 的实验中，性能差距很小，并且 KDPE 的成功率始终在 DP 三次试验的标准差范围内（见附录 D ）。

KDPE 在高精度任务（例如 ToolHang）和从低质量数据学习的任务（例如 mh）上，相对于 DP 实现了更大的性能提升。第一个观察表明，高精度任务的成功率更容易受到异常轨迹的影响。第二点结合了训练时的直觉：在噪声数据上训练的策略倾向于生成数据集中包含的异常值，强调了我们过滤机制的重要性。这些发现得到了 KDPE-OOD 性能的进一步支持，在具有反应特性的情况下，它在 KDPE 表现最好的任务上经历了最大的性能下降。基于轨迹的 KDPE 版本实现了与 DP 相似的性能，导致两种架构都没有显著的改善。我们将此归因于这样一个事实：为了分析整个轨迹，Tr-KDPE 需要更高维的核。这可能导致维度诅咒和对总体分布的较差表征。虽然可以增加种群规模，但需要注意的是，所需样本量随着维度的增加呈指数增长。这将导致计算成本的指数增长。

5.2 视觉环境扰动下的分析

为了研究 DP 和 KDPE 对领域偏移的稳健性，我们通过第 4.1 节中描述的对象颜色修改扰动模型观察，并在第 5.1 节中考虑的同一组 RoboMimic 和 MimicGen 任务上测量成功率。表 2 中的结果显示，平均而言，KDPE 在使用 CNN 和基于 Transformer 的模型时都比 DP 表现更好，与在未扰动设置下的基准实验表 1 中呈现的差异相似。该结果进一步支持了 KDPE 在过滤轨迹异常值方面的有效性，即使在噪声更大的环境中也是如此。

www.xueshuxiangzi.com

	PickPlush	PickSponge	CubeSort	CoffeeMaking
DP-C	90	88	41	60
KDPE-C	96	90	44	70

Table 3: DP-C 和 KDPE-C 在实际任务中的成功率（%）。我们在 50 个回合中测试 PickPlush 和 PickSponge，在 100 个回合中测试 CubeSort，在 10 个回合中测试 CoffeeMaking。

5.3 真实机器人结果

我们将 KDPE 与 DP 在三个真实任务上进行比较，以研究在模拟任务中观察到的性能提升是否能够转化为在真实机器人上更好的性能。表 3 中的结果显示，在四个实验中，KDPE 在成功率方面优于 DP。此外，对于所有任务，我们观察到机器人末端执行器和手爪开口的行为更加平滑。这表明 KDPE 能够更一致地选择代表任务模式的轨迹。

我们在 Sec. 4.2 中描述的两种不同任务配置下评估 50 个回合的性能。我们在每个回合开始时随机初始化玩具的位置。然而，当其中一种方法抓取失败时，我们将从同一对象位置开始执行另一种方法的回滚。KDPE 解决了任务与橙色玩具相关的 PickPlush 48 次，而 DP 则是 45 次（96 % 与 90 % 的成功率），这与 Tab. 1 中 KDPE-C 的结果一致。在 DP 的三个失败案例中，选择一个合适的 KDPE 轨迹对于解决任务至关重要。我们还用黄色海绵（PickSponge ）测试相同的模型（仅训练抓取橙色玩具）。类似于在 Sec. 5.2 中提出的对象颜色扰动实验，KDPE 表现优于 DP，但与 PickPlush 相比，性能差距缩小。这可能表明在黄色海绵的失败回合中，DP（因此也就是 KDPE）无法预测合适的分布内轨迹。

CubeSort 我们通过在每个情节开始时随机将两个立方体抛到盘子上来评估这两种算法，共进行了 100 个情节。与 PickPlush 不同，我们无法在相同的物体位置测试这些方法，因为任务的失败状态通常是部分的（即，仅一个立方体被放置在匹配的杯子中）。KDPE 实现了更高的成功率（高出 3 %），这反映了 KDPE-C 在模拟中的结果，例如，在需要操纵大小相似的立方体的 Stack 任务中。

咖啡制作：我们报告了咖啡制作任务在 10 个回合中的表现。我们注意到，对于这个任务，桌子上胶囊的初始位置对任务的最终成功有巨大的影响。因此，为了公平的比较，我们在同样的胶囊初始化情况下测试了两种方法。KDPE 成功解决了七次任务，而 DP 成功了六次。

我们在用于实际实验的机器上（配备了 NVIDIA RTX 3080 GPU），使用 DDIM 采样测量 DP-C、KDPE-C 和 Tr-KDPE-C 的推理时间。原始实现中的 DP-C 生成单一轨迹需要 77 毫秒，而生成 100 条轨迹的人群则需要额外 13 毫秒，对其执行 KDE 仅增加 3 毫秒。因此，DP-C 可以在 $\sim$ 12.99 Hz 的控制频率下执行，而 KDPE-C 可以在 $\sim$ 10.75 Hz 下执行，这意味着 KDPE-C 仅增加了 $\sim$ 2.24 Hz 的计算开销。而对于 Tr-KDPE-C，执行 KDE 需要 30 毫秒，可以在 $\sim$ 8.33 Hz 下执行。

6 结论

DP 最近因其作为训练行为克隆策略的最有效方法之一而受到欢迎，这得益于其通过 DDPM 进行的策略参数化，从而能够对训练数据中的多模态性进行建模。然而，尽管其有效，但采样过程没有受到任何约束。这可能导致模型预测出将机器人带出分布范围的轨迹。我们建议通过 KDPE 来克服这一局限性，这是一种使用 KDE 选择由不同 DP 去噪过程计算出的最具代表性的轨迹的策略。

我们在四个 RoboMimic 和三个 MimicGen 模拟任务上，以及三个真实机器人实验中：(1) 桌面毛绒玩具抓取，(2) 多模态和多步骤立方体分类，(3) 一个需要高度精确才能完成的长周期咖啡制作任务中，对 KDPE 与 DP 进行了定量基准测试。

我们表明，KDPE 在较低的示范质量环境中以及在需要更高精度的任务中，甚至在存在视觉环境扰动的情况下，能够实现更好的性能。

未来研究的两个有趣方向是使用 KDPE 指导 DP 的去噪过程，以及将 KDPE 应用于更高维度的问题，如使用拟人化手的双手或灵巧操作任务。然而，这样的应用可能会带来额外的挑战，即解决 KDE 的维度诅咒，这是基于核的方法已知的挑战。

www.xueshuxiangzi.com

7 局限性

Limiting assumptions KDPE 从 DP 生成的轨迹中选择某些输出轨迹。如果后者未能生成具有代表性的轨迹分布，例如由于训练效果不佳，KDPE 无法提供任何性能改进。

DP 的优势之一在于轨迹生成的随机性，可以通过生成可能与较小概率密度相关的轨迹帮助机器人从分布外状态中恢复。KDPE 旨在降低机器人进入此类状态的概率。然而，检测到这种情况发生时（例如，从 KDE 估计的 PDF 中）并切换到不同的轨迹选择方法将是对 KDPE 的一个有趣的改进。

在本文中，我们提出了 KDPE 作为可以连接到 DP 顶部的一个组件。然而，将 KDPE 应用于其他生成模型，如 flow matching [21]，或应用于通用策略 [13, 18, 14]，也将是一个有趣的研究方向。

8

致谢 本研究得到了 PNRR MUR 项目 PE0000013-FAIR，国家事故保险研究所（INAIL）项目 ergoCub-2.0，以及意大利技术研究院的脑与机器旗舰计划的支持。

References

[1] 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.

[2] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. Advances in neural information processing systems, 33:6840–6851, 2020.

[3] D. Q. Mayne and H. Michalska. Receding horizon control of nonlinear systems. In Proceedings of the 27th IEEE Conference on Decision and Control, pages 464–465. IEEE, 1988.

[4] T. D. Barfoot and P. T. Furgale. Associating uncertainty with three-dimensional poses for use in estimation problems. IEEE Transactions on Robotics, 30(3):679–693, 2014. doi:10.1109/TRO.2014.2298059.

[5] I. Gilitschenski, G. Kurz, S. J. Julier, and U. D. Hanebeck. A new probability distribution for simultaneous representation of uncertain position and orientation. In 17th International Conference on Information Fusion (FUSION), pages 1–7, 2014.

[6] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(5):603–619, 2002. doi:10.1109/34.1000236.

[7] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, KDD'96, page 226–231. AAAI Press, 1996.

[8] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. In Conference on Robot Learning, pages 1678–1690. PMLR, 2022.

[9] A. Mandlekar, S. Nasiriany, B. Wen, I. Akinola, Y. Narang, L. Fan, Y. Zhu, and D. Fox. Mimicgen: A data generation system for scalable robot learning using human demonstrations. In 7th Annual Conference on Robot Learning, 2023.

www.xueshuxiangzi.com

[10] S. Haddadin, S. Parusel, L. Johannsmeier, S. Golz, S. Gabl, F. Walch, M. Sabaghian, C. Jähne, L. Hausperger, and S. Haddadin. The franka emika robot: A reference platform for robotics research and education. IEEE Robotics & Automation Magazine, 29(2):46–64, 2022. doi:10.1109/MRA.2021.3138382.

[11] Open X-Embodiment Collaboration. Open X-Embodiment: Robotic learning datasets and RT-X models. In 2024 IEEE International Conference on Robotics and Automation (ICRA), 2024.

[12] 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. arXiv preprint arXiv:2403.12945, 2024.

[13] D. Ghosh, H. R. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, T. Kreiman, C. Xu, J. Luo, Y. L. Tan, L. Y. Chen, Q. Vuong, T. Xiao, P. R. Sanketi, D. Sadigh, C. Finn, and S. Levine. Octo: An Open-Source Generalist Robot Policy. In Proceedings of Robotics: Science and Systems, Delft, Netherlands, July 2024. doi:10.15607/RSS.2024.XX.090.

[14] M. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, Q. Vuong, T. Kollar, B. Burchfiel, R. Tedrake, D. Sadigh, S. Levine, P. Liang, and C. Finn. Openvla: An open-source vision-language-action model. arXiv preprint arXiv:2406.09246, 2024.

[15] K. Bousmalis, G. Vezzani, D. Rao, C. M. Devin, A. X. Lee, M. B. Villalonga, T. Davchev, Y. Zhou, A. Gupta, A. Raju, A. Laurens, C. Fantacci, V. Dalibard, M. Zambelli, M. F. Martins, R. Pevceviciute, M. Blokzijl, M. Denil, N. Batchelor, T. Lampe, E. Parisotto, K. Zolna, S. Reed, S. G. Colmenarejo, J. Scholz, A. Abdolmaleki, O. Groth, J.-B. Regli, O. Sushkov, T. Rothörl, J. E. Chen, Y. Aytar, D. Barker, J. Ortiz, M. Riedmiller, J. T. Springenberg, R. Hadsell, F. Nori, and N. Heess. Robocat: A self-improving generalist agent for robotic manipulation. Transactions on Machine Learning Research, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=vsCpILiWHu.

[16] 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.

[17] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, et al. π_0: A vision-language-action flow model for general robot control. arXiv preprint arXiv:2410.24164, 2024.

www.xueshuxiangzi.com

[18] P. Intelligence, K. Black, N. Brown, J. Darpinian, K. Dhabalia, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, et al. $\pi_{0.5}$: A vision-language-action model with open-world generalization. arXiv preprint arXiv:2504.16054, 2025.

[19] B. Zitkovich, T. Yu, S. Xu, P. Xu, T. Xiao, F. Xia, J. Wu, P. Wohlhart, S. Welker, A. Wahid, Q. Vuong, V. Vanhoucke, H. Tran, R. Soricut, A. Singh, J. Singh, P. Sermanet, P. R. Sanketi, G. Salazar, M. S. Ryoo, K. Reymann, K. Rao, K. Pertsch, I. Mordatch, H. Michalewski, Y. Lu, S. Levine, L. Lee, T.-W. E. Lee, I. Leal, Y. Kuang, D. Kalashnikov, R. Julian, N. J. Joshi, A. Irpan, B. Ichter, J. Hsu, A. Herzog, K. Hausman, K. Gopalakrishnan, C. Fu, P. Florence, C. Finn, K. A. Dubey, D. Driess, T. Ding, K. M. Choromanski, X. Chen, Y. Chebotar, J. Carbajal, N. Brown, A. Brohan, M. G. Arenas, and K. Han. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In J. Tan, M. Toussaint, and K. Darvish, editors, Proceedings of The 7th Conference on Robot Learning, volume 229 of Proceedings of Machine Learning Research, pages 2165–2183. PMLR, 06–09 Nov 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html.

[20] G. R. Team, S. Abeyruwan, J. Ainslie, J.-B. Alayrac, M. G. Arenas, T. Armstrong, A. Balakrishna, R. Baruch, M. Bauza, M. Blokzijl, et al. Gemini robotics: Bringing ai into the physical world. arXiv preprint arXiv:2503.20020, 2025.

[21] J. Zhang, J. Zhang, K. Pertsch, Z. Liu, X. Ren, M. Chang, S.-H. Sun, and J. J. Lim. Bootstrap your own skills: Learning to solve new tasks with large language model guidance. In 7th Annual Conference on Robot Learning, 2023.

[22] M. Reuss, M. Li, X. Jia, and R. Lioutikov. Goal-conditioned imitation learning using score-based diffusion policies. arXiv preprint arXiv:2304.02532, 2023.

[23] H. Ha, P. Florence, and S. Song. Scaling up and distilling down: Language-guided robot skill acquisition. In Conference on Robot Learning, pages 3766–3777. PMLR, 2023.

[24] S. Dasari, O. Mees, S. Zhao, M. K. Srirama, and S. Levine. The ingredients for robotic diffusion transformers. arXiv preprint arXiv:2410.10088, 2024.

[25] X. Zhang, M. Chang, P. Kumar, and S. Gupta. Diffusion meets dagger: Supercharging eye-in-hand imitation learning. arXiv preprint arXiv:2402.17768, 2024.

[26] S. Ross, G. Gordon, and D. Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In Proceedings of the fourteenth international conference on artificial intelligence and statistics, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.

[27] C. Agia, R. Sinha, J. Yang, Z. Cao, R. Antonova, M. Pavone, and J. Bohg. Unpacking failure modes of generative policies: Runtime monitoring of consistency and progress. In Proceedings of The 8th Conference on Robot Learning, volume 270 of Proceedings of Machine Learning Research, pages 689–723. PMLR, 2025.

[28] S. Sagar, J. Duan, S. Vasudevan, Y. Zhou, H. B. Amor, D. Fox, and R. Senanayake. From mystery to mastery: Failure diagnosis for improving manipulation policies. arXiv preprint arXiv:2412.02818, 2025.

[29] M. Nakamoto, O. Mees, A. Kumar, and S. Levine. Steering your generalists: Improving robotic foundation models via value guidance. In 8th Annual Conference on Robot Learning, 2024. URL https://openreview.net/forum?id=6FGlpzC9Po.

[30] Y. Wu, Z. Sun, S. Li, S. Welleck, and Y. Yang. Scaling inference computation: Compute-optimal inference for problem-solving with language models. In The 4th

www.xueshuxiangzi.com

Workshop on Mathematical Reasoning and AI at NeurIPS'24, 2024. URL https://openreview.net/forum?id=j7DZWSc8qu.

[31] B. Brown, J. Juravsky, R. Ehrlich, R. Clark, Q. V. Le, C. Ré, and A. Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling. arXiv preprint arXiv:2407.21787, 2024.

[32] C. Snell, J. Lee, K. Xu, and A. Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters. arXiv preprint arXiv:2408.03314, 2024.

[33] N. Ma, S. Tong, H. Jia, H. Hu, Y.-C. Su, M. Zhang, X. Yang, Y. Li, T. Jaakkola, X. Jia, and S. Xie. Inference-time scaling for diffusion models beyond scaling denoising steps. arXiv preprint arXiv:2501.09732, 2025.

[34] S. Li, K. Kallidromitis, A. Gokul, A. Koneru, Y. Kato, K. Kozuka, and A. Grover. Reflect-dit: Inference-time scaling for text-to-image diffusion transformers via in-context reflection. arXiv preprint arXiv:2503.12271, 2025.

[35] E. Xie, J. Chen, Y. Zhao, J. Yu, L. Zhu, C. Wu, Y. Lin, Z. Zhang, M. Li, J. Chen, H. Cai, et al. Sana 1.5: Efficient scaling of training-time and inference-time compute in linear diffusion transformer. arXiv preprint arXiv:2501.18427, 2025.

[36] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li. On the continuity of rotation representations in neural networks. In 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 5738–5746, 2019. doi:10.1109/CVPR.2019.00589.

[37] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. In 5th Annual Conference on Robot Learning, 2021.

[38] J. Song, C. Meng, and S. Ermon. Denoising diffusion implicit models. In International Conference on Learning Representations, 2021. URL https://openreview.net/forum?id=St1giarCHLP.

[39] R. J. Hyndman, D. M. Bashtannyk, and G. K. Grunwald. Estimating and visualizing conditional densities. 5(4):315–336. ISSN 1061-8600, 1537-2715. doi:10.1080/10618600.1996.10474715. URL http://www.tandfonline.com/doi/abs/10.1080/10618600.1996.10474715.

[40] Rerun Development Team. Rerun: A visualization sdk for multimodal data, 2024. URL https://www.rerun.io. Available from https://www.rerun.io/ and https://github.com/rerun-io/rerun.

www.xueshuxiangzi.com

A 视觉环境扰动下的分析

在第 4.1 节中展示的视觉环境扰动实验中，我们对每个环境中的物体颜色略作修改。具体来说，如果物体有纹理，我们计算其平均颜色，将其亮度（在 HSL 颜色模式下的第三通道）降低 10 % 并将其设为物体颜色。在模拟任务环境中被修改的物体如图 4 所示。

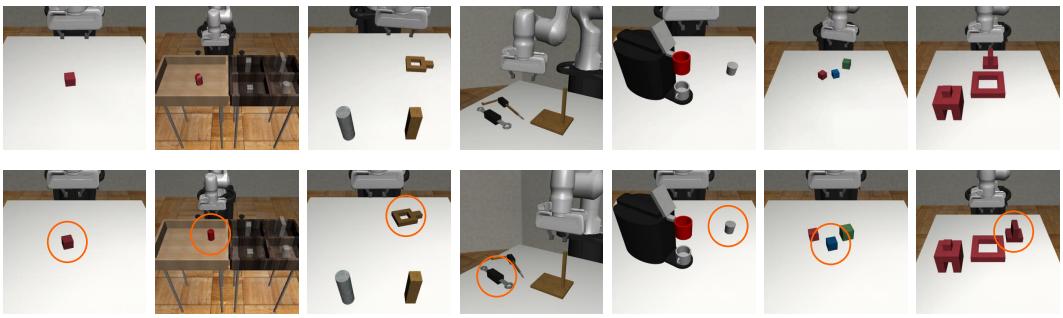

Figure 4: 第一行：未扰动的任务环境。第二行：扰动环境中，被颜色扰动实验修改的物体用橙色圆圈标出。

B Tr-KDPE 的定义

Tr-KDPE（Trajectory-KDPE）是对 KDPE 算法的一种修改，它在完整动作轨迹的群体中估计概率密度函数（PDF）。我们回顾一下，KDPE 只根据与最终动作相关联的密度来选择动作轨迹。因此，Tr-KDPE 减轻了由于偏向异常轨迹的风险，这些轨迹虽然在 KDPE 中最终处于分布内的状态，但总体上可能是分布外的轨迹。

为了对整个动作序列的联合 PDF $p(\boldsymbol{a}_1, ..., \boldsymbol{a}_T)$ 建模，Tr-KDPE 使用马尔可夫假设将这种联合概率分解为条件概率的乘积：$p(\boldsymbol{a}_1, ..., \boldsymbol{a}_T) = p(\boldsymbol{a}_1) \prod_{t=2}^{T} p(\boldsymbol{a}_t|\boldsymbol{a}_{t-1})$ 。估计依赖于使用条件核密度估计（Conditional Kernel Density Estimation [39] ）来建模转移概率 $p(\boldsymbol{a}_t|\boldsymbol{a}_{t-1})$：

$$\hat{f}(y|x) = \frac{\hat{g}(x, y)}{\hat{h}(x)}, \tag{7}$$

，其中 $\hat{g}(x, y)$ 是联合密度的 KDE $g(x, y)$ ，$\hat{h}(x)$ 是边际密度的 KDE $h(x)$ 。因此，我们可以计算 $p(\boldsymbol{a}_t|\boldsymbol{a}_{t-1})$ 的核密度估计为：

$$\rho(\boldsymbol{a}_t|\boldsymbol{a}_{t-1}) = \frac{\hat{g}(\boldsymbol{a}_{t-1}, \boldsymbol{a}_t)}{\hat{h}(\boldsymbol{a}_{t-1})}. \tag{8}$$

。因此，估计完整的轨迹密度 $p(\boldsymbol{a}_1, ..., \boldsymbol{a}_T)$ 涉及计算一个标准的 KDE $p(\boldsymbol{a}_1)$ 以及 $T - 1$ 条件 KDE 用于项 $t = 2, ..., T$ 。

C 超参数

在训练和评估期间使用的超参数在表 4 中报告。

D DP 标准差

正如在 Sec. 3.2 中所报道的，由于基线性能（DP-C 或 DP-T）取决于轨迹的随机采样，我们使用三个不同的种子对其进行了三次评估。我们在 Tab. 5 中报告了所有方法在基准实验中的性能及其 DP 标准偏差，在 Tab. 6 中报告了在对象颜色扰动下实验的性能。

在开发和研究 KDPE 及比较基线的过程中，我们使用数据可视化库 [40] 开发了一个可视化工具。该工具有助于分析轨迹群体，包括动作位置、方向和抓取器状态。借助这个可视化工具，我们研究了核密度估计（KDE）带宽（在表 4 的最后三行中报告）如何影响动作密度，

www.xueshuxiangzi.com

Hyperparameter	Value
Optimizer	Adam
Learning Rate	1×10^{-4}
Betas	$(0.95, 0.999)$
Epsilon	1×10^{-8}
Weight Decay	1×10^{-6}
Noise scheduler	DDPM (simulation) / DDIM (real robot)
Diffusion Inference Steps	100 (simulation) / 10 (real robot)
Total Training Steps	80,000
Population Size (N)	100
Execution Horizon (T)	8
Action Dimensionality (D)	10
Position Bandwidth (σ_{pos})	0.05
Rotation Bandwidth (σ_{rot})	0.25
Gripper Bandwidth (σ_{grip})	1.0

Table 4: 用于训练和评估 KDPE 的超参数。括号中的符号对应于 Sec. 3 中使用的符号。

	Lift		Can		Square		ToolHang
	ph	mh	ph	mh	ph	mh	ph
DP-C	**100 ± 0**	**100 ± 0**	96.7 ± 1.00	94.0 ± 2.00	**93.0 ± 3.00**	83.0 ± 7.00	62.0 ± 2.00
KDPE-OOD-C	**100**	**100**	96.0	84.0	83.0	49.0	7.00
Tr-KDPE-C	**100**	**100**	97.0	94.0	90.0	78.0	68.0
KDPE-C	**100**	**100**	**98.0**	**96.0**	92.0	**86.0**	**76.0**
DP-T	**100 ± 0**	**100 ± 0**	96.0 ± 2.00	90.0 ± 3.00	84.0 ± 4.00	73.0 ± 7.00	61.3 ± 10.0
KDPE-OOD-T	**100**	99.0	95.0	91.0	84.0	65.0	2.00
Tr-KDPE-T	**100**	99.0	93.0	92.0	81.0	81.0	62.0
KDPE-T	**100**	**100**	**97.0**	**92.0**	**88.0**	**83.0**	**66.0**

	Coffee	Stack	Assembly
DP-C	85.0 ± 6.00	80.7 ± 1.00	54.3 ± 1.00
KDPE-OOD-C	85.0	72.0	40.0
Tr-KDPE-C	76.0	78.0	**61.0**
KDPE-C	**88.0**	**85.0**	**61.0**
DP-T	91.7 ± 6.00	73.7 ± 2.00	**23.3 ± 2.00**
KDPE-OOD-T	**93.0**	61.0	23.0
Tr-KDPE-T	86.0	**74.0**	18.0
KDPE-T	91.0	72.0	23.0

Table 5: 表 1 中展示的基准测试结果，同时报告了三次 DP 运行过程中成功率（%）的标准偏差。上表：RoboMimic（提升、罐头、方块、挂工具）。下表：MimicGen（咖啡、堆叠、装配）。

www.xueshuxiangzi.com

	Lift		Can		Square		ToolHang
	ph	mh	ph	mh	ph	mh	ph
DP-C	**100 ± 0**	99.6 ± 0.57	92.3 ± 4.16	**91.3 ± 3.21**	93.7 ± 2.52	79.7 ± 4.16	59.0 ± 1.53
KDPE-C	**100**	**100**	**94.0**	90.0	**94.0**	**90.0**	**69.0**
DP-T	98.7 ± 0.58	**100 ± 0**	89.0 ± 4.58	84.0 ± 3.61	**79.3 ± 1.53**	68.3 ± 2.51	58.0 ± 5.57
KDPE-T	**99.0**	**100**	**90.0**	**85.0**	75.0	**77.0**	**62.0**

	Coffee	Stack	Assembly
DP-C	92.3 ± 5.03	75.3 ± 0.58	44.7 ± 1.50
KDPE-C	**97.0**	**88.0**	**50.0**
DP-T	**73.3 ± 7.37**	67.7 ± 4.16	15.6 ± 4.16
KDPE-T	72.0	**68.0**	**20.0**

Table 6: 在对象颜色扰动下的实验结果如表格 2 所示，同时报告了三次 DP 运行的成功率（%）的标准差。上表：RoboMimic（Lift，Can，Square，ToolHang）。下表：MimicGen（Coffee，Stack，Assembly）。

并找到了合适的值，使得 KDE 能够捕捉到 DP 输出的多模态性。图 5 展示了该可视化工具的快照。

我们开源了可视化工具的代码，希望它能帮助其他人分析生成机器人策略。可视化工具的代码可以在项目页面的 https://hsp-iit.github.io/KDPE/ 上找到。

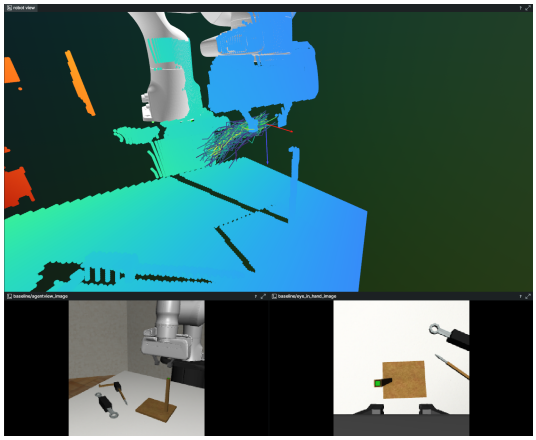

Figure 5: 用于分析 RoboMimic ToolHang 任务轨迹的轨迹可视化工具。由于真实世界环境中对象网格并不容易获取，3D 视图（机器人视图窗口）中的场景被表示为点云。该可视化工具支持为轨迹的集合分配不同的色彩图。在图片中，色彩图表示 KDPE 分配给每条轨迹的密度。

www.xueshuxiangzi.com