SOAR：面向机器人操作的时间结构引导单步动作生成

Di Xin[1*], Mingfei Han[1], [2*†], Genming Cui[1], YuanxiangChen[1], XiaolongWu[1], SiyuNing[1], XiaojunChang[1,2], Zhihui Li[†1]
[1] 中国科学技术大学未来媒体计算实验室 [2] LeapWing AI *贡献相同 †项目负责人

arXiv:2611.30363v1 [cs.RO] 23 Nov 2026

摘要 高效的机器人操作需要时间协调的动作，并要求较低的推理延迟。多步扩散和流策略通过逐步细化动作轨迹来提升效果，但需要耗费大量计算的迭代推理；而单步策略无需迭代细化轨迹，从而降低推理成本。我们提出**SOAR**，一种结构引导的单步生成策略，在保持单步效率的同时提升动作质量。SOAR预测以观测为条件的未来动作离散余弦变换（DCT）表示，捕捉多个频带上的时间变化，为关键时间结构信息提供显式引导。不同于通过预测速度场间接恢复干净动作的速度参数化单步流策略，SOAR直接在动作空间中生成干净动作轨迹。这使模型输出与可执行的控制目标保持一致，并为结构校准提供直接接口。预测结构还会根据谱差异进一步校正生成轨迹，且无需额外评估生成主干网络。训练期间，从校准后的干净动作推导出相应的MeanFlow 速度，并用于一致性学习；同时，端点目标使训练与单步推理所用的纯噪声条件相一致。在53 个 Adroit 和 Meta-World任务上，SOAR取得了 83.9% 的总体成功率，推理延迟仅为 8.7 毫秒。将SOAR 应用于视觉-语言-动作模型 $\pi_{0.5}$ 后，其在 LIBERO 上的成功率进一步达到 97.2%，在 RoboTwin 2.0 上达到 71.7%。真实世界实验和大量消融实验进一步验证了其有效性与效率。

1 引言

通过专家演示数据学习视觉运动策略，机器人无需手动设计特定任务控制器，便能掌握复杂的操作技能。然而，面对相同的观测和任务目标，可能存在多种可行的动作轨迹，它们在接近方向、运动时序、

或接触策略上各不相同。与可能对不同动作模式取平均的确定性行为克隆不同，扩散策略和基于流的策略通过对动作分布本身进行建模来处理这种歧义，并已在机器人操作中展现出优异性能 [8, 60]。

尽管生成式策略效果显著，但它们往往依赖迭代推理。扩散策略逐步对动作轨迹进行去噪，而基于流的策略通常通过多个步骤对学习得到的速度场进行积分。这种迭代策略需要反复进行网络评估，导致推理延迟显著增加，降低了对动力学变化的响应能力，并可能使预测动作在场景演变时变得过时 [4]。近期一些研究尝试通过加速采样、蒸馏、一致性学习或缩短流轨迹来减少推理步数 [19, 36, 39, 43, 46], ，甚至将其降至仅需一次网络评估 [11, 13, 41]。然而，这种效率提升是以牺牲迭代优化为代价的。多步生成会反复调整动作轨迹，逐步修正其时间组织；而单步策略必须一次性确定整个轨迹结构，之后不再有机会进行修正。

为在不依赖迭代优化的情况下提供明确的时间结构引导，我们提出**SOAR**，这是一种面向机器人**操**作的结构引导**单**步**动**作生成框架，**如**图1所示。与通过速度参数化单步流策略间接恢复干净动作不同，SOAR 直接在动作空间中生成干净动作轨迹，使模型输出与可执行控制目标保持一致，并支持直接进行结构校准。给定当前观测，SOAR 会预测未来动作的多频带离散余弦变换（DCT）表示，以提供时间

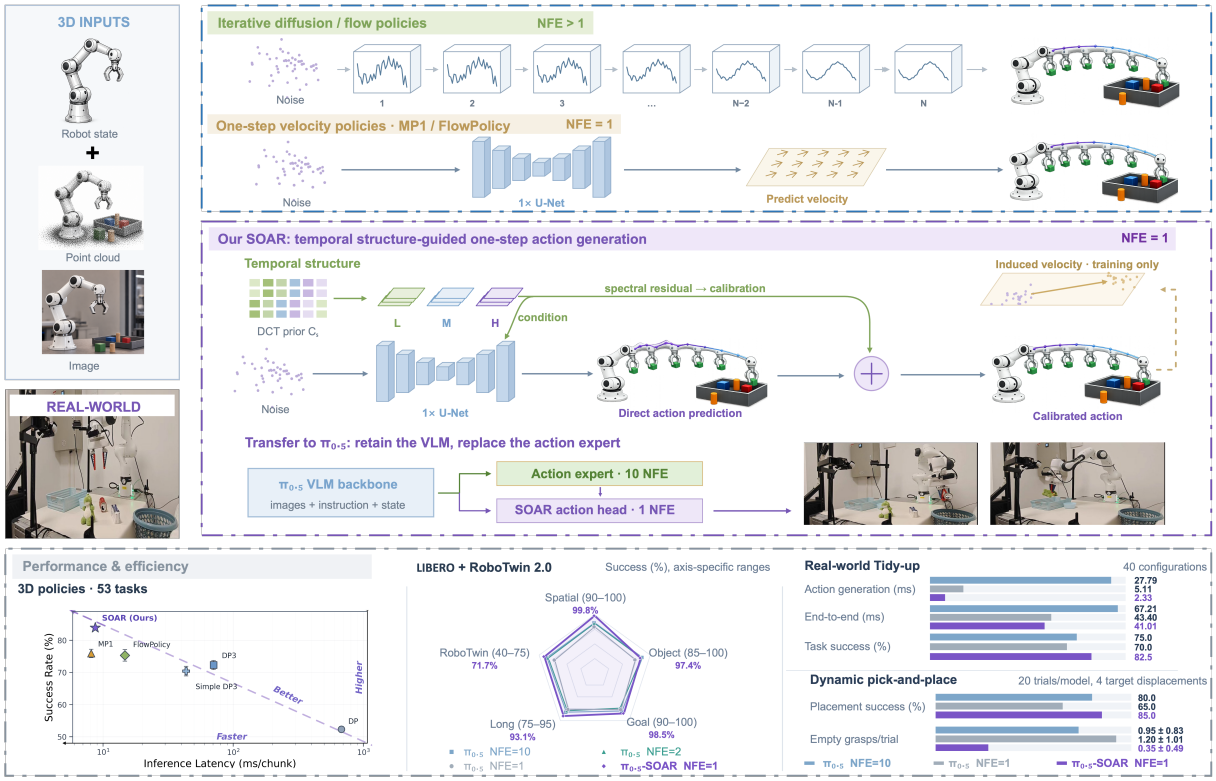

图 1. SOAR 示意图。 迭代扩散策略和流策略需要经过多步细化，而现有的一步策略通常通过学习得到的速度场生成动作。相比之下，SOAR 通过单次 U-Net 评估直接预测干净动作块，并利用观测条件频率表示进行引导和校准；相应的输运速度仅用于训练。同样的设计以单步方案取代了 $\pi_{0.5}$ 的多步动作专家，同时保留其视觉语言模型骨干网络。在 Adroit [37],Meta-World [56],LIBERO [29],RoboTwin2.0 [7], 和真实世界步骤中，SOAR 在 NFE 为 $=1$ 时取得了有竞争力的成功率，同时大幅降低了动作生成延迟。

结构引导 [1, 65]。预测结构被纳入观测条件，以引导基础动作生成；随后，依据生成轨迹的谱差异对其进行校准，无需额外评估生成式骨干网络。训练期间，校准后的动作会映射到相应的 MeanFlow 速度，以实现一致性监督和端点监督。因此，SOAR 在保持 NFE $=1$ 推理和毫秒级延迟的同时，提升了动作质量。

我们的贡献如下：

- 我们提出了**SOAR**，这是一种直接的一步动作生成策略，可在动作空间中以 NFE $=1$ 生成干净动作轨迹，并实现毫秒级推理。
- 我们提出了时间结构引导生成与校准方法，显式建模动作轨迹的关键时间结构特征，并在无需额外评估骨干网络的情况下校正生成动作中的结构差异。
- SOAR 在 Adroit 和 Meta-World 上的表现始终优于所比较的单步策略，同时保持 NFE $=1$；此外，在 LIBERO 和 RoboTwin2.0 的视觉-语言-动作设置中也展现出具有竞争力的性能。

2 相关工作

机器人学习的视觉输入。 视觉运动策略通常以图像观测或显式三维场景表示为输入。Behavior Transformer、ACT 和 RT-1 等基于图像的方法利用学习得到的视觉表示进行多模态动作预测、动作分块和大规模机器人控制 [5, 40, 62]。由于图像仅通过投影编码几何信息，一些研究转而在三维空间中进行显式推理。PerAct、Act3D、RVT 和 GNFactor 使用体素化特征表示场景，

连续三维特征、虚拟视图或神经特征场 [14, 15, 57]。显式三维表示也与生成式动作模型相结合，包括 3D Diffuser Actor、DP3 和 EquiBot [22,52, 58]。这些研究凸显了三维几何对于实现精确且数据高效的操作的价值，尽管仅凭观测模态并不能决定动作生成的成本。

基于扩散的视觉运动策略。 扩散策略通过渐进式去噪表示多模态动作分布 [17]。Diffusion Policy 及 ChainedDiffuser、HDP 等后续方法采用迭代优化来生成结构化动作轨迹 [8, 33, 50]。降低推理成本的需求推动了加速采样器和蒸馏方法的发展。DDIM 缩短采样轨迹，而渐进式蒸馏则将多个去噪步骤压缩为更少的评估次数 [39, 43]。一致性模型进一步通过约束生成轨迹的一致性，实现单步或少步采样 [44, 45]。在机器人学习中，Consistency Policy、ManiCM、SDM Policy 和 OneDP 采用相关的一致性、分布匹配或蒸馏目标来加速动作生成 [12, 20, 36, 49]。这些方法降低了迭代去噪的成本，而许多单步变体则通过压缩或蒸馏多步生成过程来获得单步映射。

基于流的视觉运动策略。流匹配将生成建模为在学习得到的时变速度场下进行传输 [28]。标准流推理通过数值积分求解该速度场，因此需要重复进行网络评估。Rectified Flow 通过鼓励更直的传输路径来降低积分难度，而 AdaFlow 则自适应地确定积分步数 [19, 31]。相比之下，一致性流匹配通过约束流轨迹上的一致性来支持高效生成 [53]。在机器人操作中，FlowPolicy 将基于一致性的流生成应用于点云条件控制，而 FreqPolicy 则引入频域约束，以确保动作序列的一致性 [46, 60]。MeanFlow 通过直接预测区间平均速度，免去了推理时的数值积分 [13], ，为 MP1、OMP 和 OFP 等原生单步策略奠定了基础 [11, 27, 41]。SOAR 基于这一单步形式，同时引入显式的轨迹结构条件和直接的动作空间校正，无需额外的采样迭代即可改善生成轨迹。

3 方法

3.1 概述

给定最近的T_o个观测$\mathbf{o}_{1:T_o} = \{\mathbf{P}^{1:T_o}$和$\mathbf{q}_{1:T_o}\}$，我们将归一化后的示范动作块记为$\mathbf{x} = \mathrm{Norm}(\mathbf{A}_1^{\mathrm{gt}}:_T) \in \mathbb{R}^{B\times T\times D_a}$。观测编码器$E\phi$ 将观测映射为全局条件$\mathbf{c} = E\phi(\mathbf{o}^{1:T_o})$。

如图2所示，SOAR 直接在动作空间中生成干净动作轨迹，而非通过预测速度场恢复轨迹。观测条件频率表示为单步生成提供时间引导，随后根据动作与预测结构之间的谱差异进一步校准所得动作。训练期间，校准后的动作会导出相应的 MeanFlow 传输速度，用于一致性学习和推理端点对齐；而推理只需对动作生成器进行一次评估。

总体目标函数为

$$\mathcal{L}_{\mathrm{SOAR}} = \lambda_{\mathrm{struct}}\mathcal{L}_{\mathrm{struct}} + \lambda_{\mathrm{act}}\mathcal{L}_{\mathrm{act}} + \lambda_{\mathrm{dis}}\mathcal{L}_{\mathrm{dis}} + \mathcal{L}_{\mathrm{MF}} + \lambda_{\mathrm{end}}\mathcal{L}_{\mathrm{end}}. \tag{1}$$

其中， $\mathcal{L}_{\mathrm{struct}}$ 用于监督轨迹结构， $\mathcal{L}_{\mathrm{act}}$ 用于约束校准后的动作， $\mathcal{L}_{\mathrm{MF}}$ 是标准的 MeanFlow 目标函数， $\mathcal{L}_{\mathrm{dis}}$ 是继承自 MP1 [41], 的弥散特征正则项，而 $\mathcal{L}_{\mathrm{end}}$ 则使训练与单步推理条件对齐。

3.2 频域轨迹结构学习

单步生成器必须在不进行迭代优化的情况下，确定整个动作块的时间组织。为在这一约束下提供明确的结构引导，我们在频域中预测未来动作的时间结构。使用离散余弦变换（DCT） [1],$\mathbf{C}^{\mathrm{gt}} = \mathrm{DCT}(\mathbf{x})$ 和 $\mathbf{C}^s = H\psi(\mathbf{c})$，其中$\mathbf{C}^{\mathrm{gt}}$ 仅用于训练监督，而$\mathbf{C}^s$ 则根据观测进行预测。结构头作为观测条件估计的辅助部分，用于

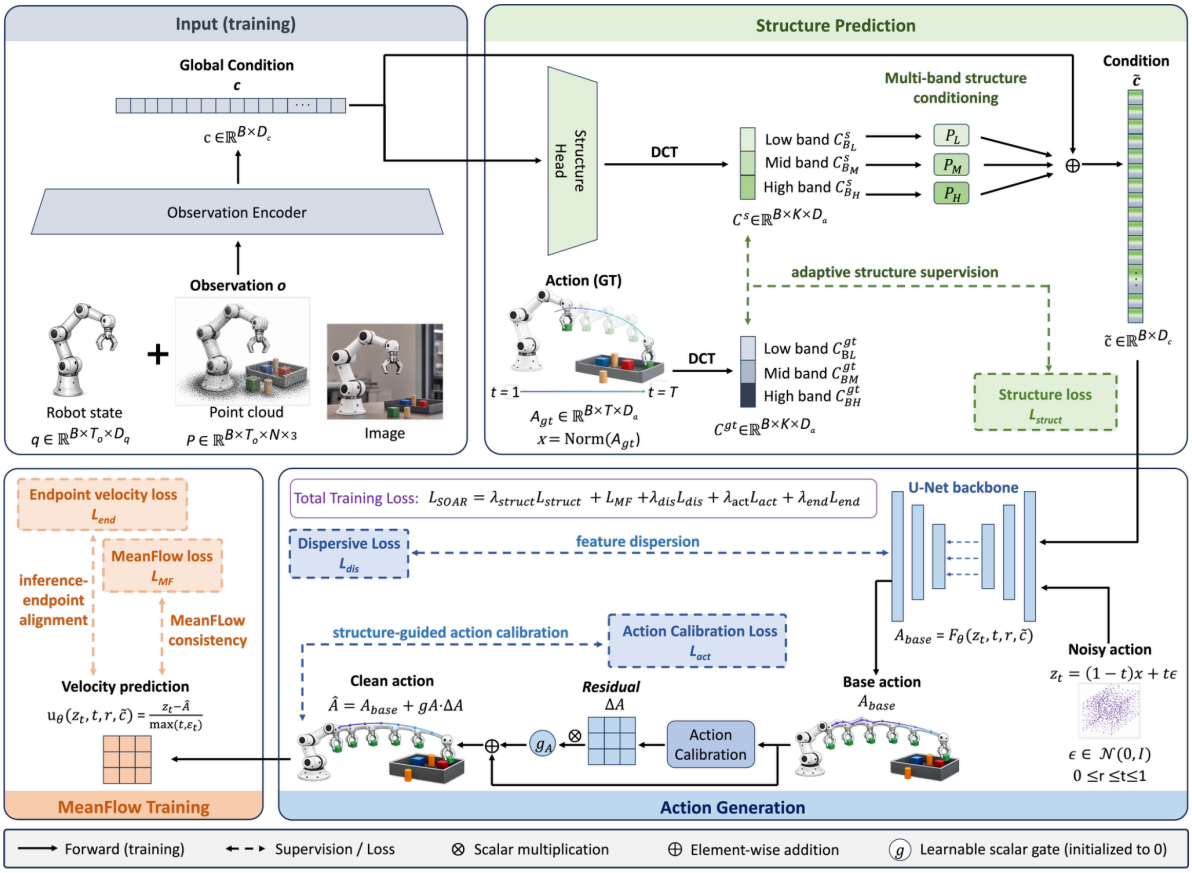

图2. SOAR概述。给定观测，SOAR 会预测未来动作轨迹的频域结构，并将其多频带特征注入动作生成器。条件U-Net直接生成干净动作轨迹，再利用预测结构对其进行校准。在训练过程中，校准后的动作会产生相应的MeanFlow 输运速度，用于一致性学习和推理端点对齐。推理只需进行一次 U-Net 评估。

引导动作生成，而非作为直接执行预测轨迹的独立策略。残差融合在保留原始观测条件的同时，融入互补的时间结构信息，以供后续动作生成使用。

多频带结构条件调制。 为捕捉不同尺度的时间变化，我们将 $\mathbf{C}^s$分解为低频、中频和高频分量，并将它们投影到观测特征空间。随后，将得到的多频带特征与观测条件融合，得到结构增强表示 $\tilde{\mathbf{c}}$，并以此为后续的单步动作生成器提供条件。频率划分和融合机制详见附录A.1.1。

自适应结构监督。 离散余弦变换系数的幅值差异显著，导致统一的频谱目标过度强调高能量分量。因此，我们根据每个系数相对于其目标能量的预测误差对其进行加权：

$$\mathcal{L}_{\text{struct}} = \frac{1}{B}\sum_{b=1}^{B}\sum_{k=0}^{K-1}\omega_{b,k}e_{b,k}. \tag{2}$$

在此，$e_{b,\,k}$ 表示频谱预测误差，而$\omega_{b,\,k}$ 表示其自适应权重。相对于自身尺度而言，仍难以预测的频率会获得更强的监督。其精确定义见附录A.1.1。随后，预测频谱$\mathbf{C}^s$ 将用于动作条件化和校准。

3.3 结构引导的动作生成与校准

在训练过程中，我们遵循基于流的生成建模中使用的标准线性概率路径 [13,28] 来构造示范动作与高斯噪声之间的中间噪声动作状态$\mathbf{z}_t$ 。详细信息见附录A.1.2。

在以结构增强表示 $\tilde{\mathbf{c}}$为条件时，FiLM条件化一维U-Net [8, 35] 直接预测干净基础动作

$$\mathbf{A}_{\text{base}} = F_\theta(\mathbf{z}_t, t, r, \tilde{\mathbf{c}}). \tag{3}$$

与从预测的传输场中恢复动作的速度参数化策略不同，F_θ 直接在动作空间中输出干净动作轨迹。

结构引导的动作校准。 仅进行结构条件化并不能明确保证生成轨迹遵循预测结构。因此，我们根据谱差异对基础动作进行校准：

$$\widehat{\mathbf{A}} = \mathbf{A}_{\text{base}} + \tanh(\rho)\,\text{IDCT}\left[\mathbf{C}^s - \text{DCT}(\mathbf{A}_{\text{base}})\right]. \tag{4}$$

频谱残差衡量由观测条件化的结构预测与基础动作的频率表示之间的差异。通过逆离散余弦变换（IDCT），这一差异被映射回动作空间，从而为生成轨迹提供轻量级修正。可学习标量 ρ控制修正幅度，并初始化为零，因此在训练开始时得到 $\mathbf{A} = \mathbf{A}_{\text{base}}$。因此，结构头既为动作生成提供引导，也为后续校准提供引导，同时无需引入额外的动作生成网络。

校准后的动作由以下损失直接监督

$$\mathcal{L}_{\text{act}} = \text{SmoothL1}(\widehat{\mathbf{A}}, \mathbf{x}). \tag{5}$$

我们还保留了弥散特征正则项 $\mathcal{L}_{\text{dis}}$，来自 MP1 [41],，作用于中间U-Net表示。其公式见附录 A.1.4。

3.4 MeanFlow一致性与推理端点对齐

训练期间，MeanFlow 进一步约束校准后的动作。SOAR 不预测独立速度场，而是直接从动作推导出相应的输运速度：

$$\mathbf{u}_\theta(\mathbf{z}_t, t, r, \tilde{\mathbf{c}}) = \frac{\mathbf{z}_t - \widehat{\mathbf{A}}}{\max(t, \varepsilon_t)}, \qquad \varepsilon_t = 10^{-5}. \tag{6}$$

其中，u_θ de 表示生成输运速度，而非机器人的物理速度。我们调整了
数据预测
参数化方式，将像素 MeanFlow [32] 应用于以观测为条件的动作生成 on.

MeanFlow 一致性。我们遵循标准的随机区间 MeanFlow 训练目标 $\mathcal{L}_{\text{MF}}$ [13,41] 来约束诱导的输运场。区间采样、自引导目标、JVP 计算和 AdaptiveL2 回归均遵循标准形式，详见附录 A.1.2。

推理端点对齐。MeanFlow 在随机的(t、r) 区间上训练，而单步推理始终在固定边界(t、r) $= (\mathbf{1}, \mathbf{0})$处执行。因此，我们在此条件下评估同一个生成器，并将其预测与示范动作显式对齐：

$$\mathcal{L}_{\text{end}} = \text{AdaptiveL2}\left(\mathbf{x} - \widehat{\mathbf{A}}_{\text{end}}\right), \tag{7}$$

其中，$\widehat{\mathbf{A}}_{\text{end}}$ 是根据实际单步推理条件预测的校准后动作。该目标等价于在 $t = 1$处匹配相应的端点输运速度；推导详见附录A.1.3。

在推理时，SOAR 根据观测预测 $\mathbf{C}^s$，构造 $\tilde{\mathbf{c}}$，采样一个高斯动作潜变量，并仅计算一次 $F_\theta(\epsilon, 1, 0, \tilde{\mathbf{c}})$。随后使用公式 4校准得到的基础动作。无需进行 JVP 计算、迭代去噪或数值 ODE 积分，因此 NFE $= 1$。

4 实验

4.1 仿真实验：三维输入

实验设置。 我们按照先前工作的评估协议 [11, 46]，在两个广泛使用的三维机器人操作基准 Adroit [37] 和 Meta-World [56]，上评估 SOAR。我们考虑

表1. 使用三维输入在37项任务基准上的性能。我们在三个随机种子上评估了37项具有挑战性的任务，并报告平均成功率（%）和标准差。标记为 ∗ 的结果为复现结果。**粗体**表示最佳结果。

		Adroit（3）				Meta-World（34）					
方法	NFE	锤子	Door	Pen	平均值	简单 (21)	中等 (4)	困难 (4)	非常困难 (5)	平均值	总体
DP [8]	10	16.0 ± 10.0	34.0 ± 11.0	13.0 ± 2.0	21.0	50.7 ± 6.1	11.0 ± 2.5	5.25 ± 2.5	22.0 ± 5.0	36.5	35.2 ± 5.3
DP3 [58]	10	100.0 ± 0.0	56.0 ± 5.0	46.0 ± 10.0	67.3	87.3 ± 2.2	44.5 ± 8.7	32.7 ± 7.7	39.4 ± 9.0	68.8	68.7 ± 4.7
简单 DP3 [58]	10	98.0 ± 2.0	40.0 ± 17.0	36.0 ± 4.0	58.0	86.8 ± 2.3	42.0 ± 6.5	38.7 ± 7.5	35.0 ± 11.6	68.3	67.4 ± 5.0
AdaFlow [19]	–	45.0 ± 11.0	27.0 ± 6.0	18.0 ± 6.0	30.0	49.4 ± 6.8	12.0 ± 5.0	5.75 ± 4.4	24.0 ± 4.8	36.1	35.6 ± 6.1
CP [36]	1	45.0 ± 4.0	31.0 ± 10.0	13.0 ± 6.0	29.7	69.3 ± 4.2	21.2 ± 6.0	17.5 ± 3.9	30.0 ± 4.9	51.8	50.1 ± 4.7
FlowPolicy [60]	1	98.0 ± 1.0	61.0 ± 2.0	54.0 ± 4.0	71.0	84.8 ± 2.2	58.2 ± 7.9	40.2 ± 4.5	52.2 ± 5.0	71.6	71.6 ± 3.5
MP1 [41]	1	100.0 ± 0.0	69.0 ± 2.0	58.0 ± 5.0	75.7	88.2 ± 1.1	68.0 ± 3.1	58.1 ± 5.0	67.2 ± 2.7	79.2	78.9 ± 2.1
OMP [11]	1	100.0 ± 0.0	68.0 ± 3.0	60.0 ± 4.0	76.0	89.7 ± 0.7	77.4 ± 2.2	62.5 ± 3.1	**77.8 ± 3.0**	83.3	82.3 ± 1.6
SOAR（我们的方法）	**1**	**100.0 ± 0.0**	**77.5 ± 2.4**	**61.9 ± 7.3**	**79.8 ± 2.9**	**92.3 ± 0.5**	76.8 ± 0.7	**63.5 ± 2.2**	76.9 ± 1.8	**84.8 ± 0.7**	**84.4 ± 0.5**

表 2. 采用三维输入时在53项任务基准上的性能。我们在三个随机种子下评估了53项具有挑战性的任务，并报告平均成功率（%）和标准差。平均推理延迟以毫秒为单位报告。标有 ∗ 的结果为复现结果。**粗体** 表示最佳结果。

		Adroit（3）				Meta-World（50）					
方法	NFE	锤子	Door	Pen	平均值	简单（28）	中等（11）	困难（6）	非常困难（5）	平均值	总体推理延迟（毫秒）
DP* [8]	100	75.1 ± 3.5	23.7 ± 3.7	19.3 ± 1.8	39.4 ± 1.5	69.0 ± 1.6	32.8 ± 2.4	34.3 ± 2.9	31.5 ± 5.7	53.1 ± 1.1	52.3 ± 0.9	678.4 ± 6.9
DP3* [58]	10	100.0 ± 0.0	73.5 ± 2.1	56.7 ± 2.6	76.7 ± 1.6	83.1 ± 0.7	58.7 ± 3.6	54.1 ± 3.9	60.7 ± 1.5	72.0 ± 1.5	72.3 ± 1.4	70.7 ± 2.6
简单 DP3* [58]	10	100.0 ± 0.0	56.8 ± 1.2	56.1 ± 0.3	71.0 ± 0.4	82.0 ± 1.1	53.8 ± 3.1	54.3 ± 5.3	61.7 ± 4.5	70.4 ± 1.6	70.4 ± 1.5	43.4 ± 0.4
ManiCM [12]	1	100.0 ± 0.0	68.0 ± 1.0	49.0 ± 4.0	72.3	83.6	55.6	33.3	67.0	69.7	69.0 ± 4.6	–
SDM [20]	1	100.0 ± 0.0	73.0 ± 2.0	49.0 ± 4.0	74.0	86.5	65.8	35.8	71.6	74.4	74.8 ± 4.5	–
FlowPolicy* [60]	1	95.2 ± 2.4	57.3 ± 6.2	47.8 ± 11.0	66.8 ± 6.4	90.6 ± 0.5	56.0 ± 1.0	54.4 ± 11.3	62.7 ± 2.4	75.9 ± 1.6	75.3 ± 1.8	14.7 ± 0.0
FreqPolicy [46]	1	98.0 ± 2.0	68.0 ± 5.0	52.0 ± 4.0	72.7	90.6	65.8	46.8	**79.4**	78.8	78.5 ± 2.6	–
MP1* [41]	1	96.1 ± 2.3	69.2 ± 3.1	57.1 ± 6.1	74.1 ± 2.0	88.5 ± 0.8	59.5 ± 2.9	56.1 ± 3.2	65.7 ± 1.3	76.0 ± 1.5	75.9 ± 1.3	**8.1 ± 0.0**
SOAR（我们的方法）	**1**	**100.0 ± 0.0**	**77.5 ± 2.4**	**61.9 ± 7.3**	**79.8 ± 2.9**	**93.7 ± 0.7**	**74.3 ± 0.6**	**63.5 ± 1.8**	76.9 ± 1.8	**84.1 ± 0.5**	**83.9 ± 0.3**	8.7 ± 0.1

两种设置：包含 3 个 Adroit 任务和 34 个 Meta-World任务的 37 个任务，以及包含 3 个 Adroit 任务和全部 50 个 Meta-World任务的 53 个任务。每个任务使用 10 组专家演示进行训练，并将点云下采样至 512 个点。每个 Adroit 任务独立训练 3,000 个训练轮次，每个 Meta-World 任务训练 1,000 个训练轮次，并使用三个随机种子（0、1 和 2）。所有训练和评估均在单张 NVIDIA RTX 4090 GPU 上进行。批量大小设为 128，并使用 AdamW 以 1×10^{-4} 的学习率优化策略。观测窗口和预测时域分别设为 2 和 4。每训练 200 轮评估一次策略。对于每个随机种子，选取评估成功率最高的五次结果并取平均值；最终任务表现以三个随机种子结果的均值和样本标准差报告。SOAR 在推理时以 NFE $= 1$ 执行单步动作生成。

结果。 表 1 和 2 总结了仿真结果。SOAR 在 37 项任务和 53 项任务基准测试上的总体成功率分别达到 84.4% 和 83.9%，同时保持 NFE $= 1$。将评估范围从 37 项任务扩展到 53 项任务后，性能变化很小，表明 SOAR 在涵盖更广泛任务时依然稳健。在完整的 53 项任务基准测试中，SOAR 在 Meta-World 上达到 84.1%，并且在中等和困难子集上表现尤为出色，分别比最佳基线方法高出 8.5% 和 7.4%。这表明 SOAR 的优势并不局限于接近饱和的简单任务，也适用于更具挑战性的操作场景。

表 2 和图 1 中的延迟结果表明，在保留单步推理的同时也能获得这些提升。SOAR 的推理延迟仅为 8.7 ms，与速度最快的一步基线方法 MP1 相近，同时总体成功率提高了 8.0%。与 DP3 相比，SOAR 将推理延迟从 70.7 ms 降至 8.7 ms，同时将总体成功率从 72.3% 提高到 83.9%。这些结果表明，SOAR 无需依赖迭代优化即可提升单步动作生成效果，从而实现良好的性能与效率权衡。各任务的详细结果和复现结果见附录 A.2。

4.2 VLA 设置下的仿真实验

实验设置。 我们进一步研究 SOAR 从三维视觉运动策略迁移到大规模视觉-语言-动作（VLA）模型的迁移能力。具体而言，我们将 $\pi_{0.5}$ [2] 的原始动作专家替换为 SOAR，并在 LIBERO [29] 和 RoboTwin 2 0 [7]上评估所得策略。在这两个基准测试中，视觉-语言骨干网络均采用 OpenPI `pi05_base` 检查点进行初始化。在 LIBERO 上，我们针对四个标准套件中的全部 40 项任务联合训练单个策略。SOAR 预测 10 步动作块，并在重新规划前执行五个动作。在 RoboTwin 2.0 上，我们使用干净示范数据针对全部 50 项任务联合训练单个策略。该 $\pi_{0.5}$-SOAR 经过 100K 优化步数的训练。SOAR 预测 50 步关节位置动作块。在这两种设置下，SOAR 均通过单次动作专家评估（NFE $= 1$）生成动作块。为进行直接比较，我们还复现了具有不同动作生成评估次数的$\pi_{0.5}$ 基线方法。这些复现结果在表格中以带上标星号（*）标记，未标记的基线结果则取自相应参考文献。详细的训练与评估配置见附录 A.3。**结果**。 表 3 展示了 LIBERO 上的结果。在 NFE $= 1$ 条件下， $\pi_{0.5}$-SOAR 的平均成功率达到 97.2%，其中空间任务为 99.8%，物体任务为 97.4%，目标任务为 98.5%，长程任务为 93.1%。我们复现的 $\pi_{0.5}$ 基线方法在 NFE $= 10$分别为 10、2 和 1 时，平均成功率为 96.5%、96.1% 和 95.5%。与复现的 10 步基线相比，SOAR 的平均成功率提高了 0.7 个百分点；与对应的一步基线相比，则提高了 1.7 个百分点，同时仅需进行一次动作专家评估。在所比较的方法中，SOAR 的平均成功率最高，且与 NFE ≤ 2 更高的 VLA 策略相比仍具竞争力。

我们进一步在 RoboTwin 2.0 上评估 SOAR。该基准包含 50 个联合训练的操作任务，且动作块显著更长。如表4所示， $\pi_{0.5}$-SOAR 在干净到干净设置下的平均成功率达到 71.7%，NFE 为 $= 1$。我们复现的 $\pi_{0.5}$ 基线方法在 NFE 分别为 $= 10$、2 和 1 时，成功率分别为 70.0%、69.1% 和 65.1%。因此，SOAR 比复现的一步基线高出 6.6 个百分点，比 10 步基线高出 1.7 个百分点。在 LIBERO 和 RoboTwin 2.0 上，SOAR 始终优于复现的一步 $\pi_{0.5}$ 基线方法，同时在多步生成方面保持了有竞争力的表现。这些结果证明了 SOAR 的有效性和迁移能力，表明其结构引导的一步生成能够成功集成到不同操作基准测试中的视觉-语言-动作策略，同时保持较高的任务表现。

4.3 真实世界实验

实验设置。 我们在两项真实世界操作任务上评估 $\pi_{0.5}$ [2] 和 $\pi_{0.5}$-SOAR。第一项是长时程整理任务，机器人需要将垃圾丢入垃圾桶，并将其他物体放入收纳篮，从而对多个桌面物体进行分类。我们评估了 40 种物体配置，每种配置包含两件垃圾物品和两件非垃圾物品。第二项是动态抓取与放置任务：抓取过程中，目标物体会被移动四次；每种模型进行 20 次试验。

所有模型使用相同的观测、动作表示和部署平台。我们将$\pi_{0.5}$ 的 10 步和 1 步推理配置与 $\pi_{0.5}$-SOAR 进行比较；后者通过一次网络评估（NFE $= 1$）生成完整的 16 步动作块。我们报告整理任务的任务成功率和推理延迟，以及动态抓取与放置任务的放置成功率和每次试验的空抓取次数。更多详情见附录A.4。

结果。 图 3 展示了真实世界的结果。在长时程整理任务中， $\pi_{0.5}$-SOAR 的成功率达到 82.5%，比 10 步和 1 步 $\pi_{0.5}$ 基线分别高出 7.5 和 12.5 个百分点。与 10 步基线相比，SOAR 将动作生成延迟从 27.79 毫秒降至 2.33 毫秒，并将端到端推理延迟从 67.21 毫秒降至 41.01 毫秒。在相同的一步设置下，SOAR 进一步将动作生成延迟降低了 54.4%，同时将任务成功率从 70.0% 提升至 82.5%。尽管直接将原始 $\pi_{0.5}$ 的推理步数从 10 步减少到 1 步会降低其成功率，但 SOAR 在推理延迟更低的情况下，取得了高于两种基线方法的成功率。

在动态抓取与放置任务中， $\pi_{0.5}$-SOAR 的放置成功率达到 85%，而 10 步和 1 步基线的放置成功率分别为 80% 和 65%。此外，其每次试验的空抓次数更少（0.35 ± 0.49），而两种基线方法分别为 0.95 ± 0.83 和 1.20 ± 1.01。对应的空抓取总次数分别为 7、19 和 24。与 10 步基线相比，SOAR 降低了空抓的平均次数

表 3. 与 LIBERO 上具有代表性的视觉-语言-动作方法进行比较。结果为成功率（%）。* 表示本研究复现并评估的基线结果；未标注的基线结果取自相应参考文献。NFE 表示每次策略推理所需的动作专家评估次数。**粗体** 表示每个比较组中的最佳结果。

方法	发表信息	NFE	空间	物体	Goal	Long	平均值
UniVLA [6]	RSS，2025	AR	96.5	96.8	95.6	92.0	95.2
OpenVLA-OFT [23]	RSS，2025	AR	97.6	98.4	97.9	94.5	97.1
VLA-Cache [51]	NeurIPS，2025	AR	**98.3**	97.5	**98.3**	**95.4**	**97.4**
UnifiedVLA [48]	ICLR，2026年	AR	95.4	**98.8**	93.6	94.0	95.5
Dita [18]	国际计算机视觉会议，2025	20	84.2	96.3	85.4	63.8	82.4
CronusVLA [26]	AAAI，2026年	100	**97.3**	**99.6**	**96.9**	**94.0**	**97.0**
DreamVLA [61]	NeurIPS，2025	10	97.5	94.0	89.5	89.5	92.6
$\pi_{0.5}$* [2]	CoRL，2025	10	98.6	97.9	98.0	91.5	96.5
FLOWER [38]	CoRL，2025	8	97.5	99.1	96.1	94.9	96.9
$\pi_{0.5}$ [2]	CoRL，2025	10	**98.8**	98.2	98.0	92.4	96.9
InstructVLA [54]	ICLR，2026年	10	97.3	99.6	96.5	89.8	95.8
MemoryVLA [42]	ICLR，2026年	10	98.4	98.4	96.4	93.4	96.5
X-VLA [64]	ICLR，2026年	10	98.2	98.6	97.8	**97.6**	98.1
Cosmos 策略 [24]	ICLR，2026年	10	98.1	**100.0**	**98.2**	**97.6**	**98.5**
UniAct [63]	计算机视觉与模式识别会议，2025	1	77.0	87.0	77.0	70.0	77.8
MolmoAct-7B-D [25]	arXiv，2025	1	87.0	95.4	87.6	77.2	86.6
NinA（多层感知机） [47]	神经信息处理系统大会研讨会，2025	1	87.8	98.2	90.2	92.8	90.9
EveryDayVLA [9]	国际机器人与自动化会议投稿，2026年	1	96.8	95.6	91.0	82.0	91.4
$\pi_{0.5}$* [2]	CoRL，2025	1	97.9	96.0	97.7	90.3	95.5
$\pi_{0.5}$* [2]	CoRL，2025	2	98.5	97.6	97.4	90.9	96.1
PI-VLA [21]	Symmetry，2026年	1	79.5	73.4	73.3	66.6	73.2
MIP（π_0 架构） [34]	ICLR，2026年	2	97.6	95.8	95.2	82.2	92.7
CF-VLA [10]	ACM MM，2026年	2	98.0	**99.2**	96.6	92.0	96.5
$\pi_{0.5}$-SOAR	–	**1**	**99.8**	97.4	**98.5**	**93.1**	**97.2**

表4. 在干净到干净设置下与 RoboTwin 2.0 上的代表性方法进行比较。结果为 50 项任务的平均成功率（%）。* 表示本研究复现并评估的基线结果；未标注的基线结果取自相应参考文献。NFE 表示每次策略推理的动作专家评估次数。**粗体** 表示各比较组中的最佳结果。

方法	发表信息	参数	NFE	成功率
Diffusion Policy [8]	RSS，2023	0.1B	100	28.0
SeedPolicy [16]	arXiv，2026年	0.2B	100	**42.8**
DP3 [58]	RSS，2024	0.3B	10	55.2
π_0 [3]	RSS，2025	3.2B	10	46.4
$\pi_{0.5}$* [2]	CoRL，2025	3.4B	10	70.0
$\pi_{0.5}$ [2]	CoRL，2025	3.4B	10	**70.7**
ACT [62]	RSS，2023	0.1B	1	29.7
RDT [30]	ICLR，2025	1.7B	5	34.5
FlowPolicy [60]	AAAI，2025	0.3B	1	41.0
UP-VLA [59]	ICML，2025	1.6B	1	52.9
$\pi_{0.5}$* [2]	CoRL，2025	3.4B	1	65.1
$\pi_{0.5}$* [2]	CoRL，2025	3.4B	2	69.1
StarVLA-α [55]	ECCV，2026年	3.8B	1	50.3
$\pi_{0.5}$-SOAR	–	2.9B	**1**	**71.7**

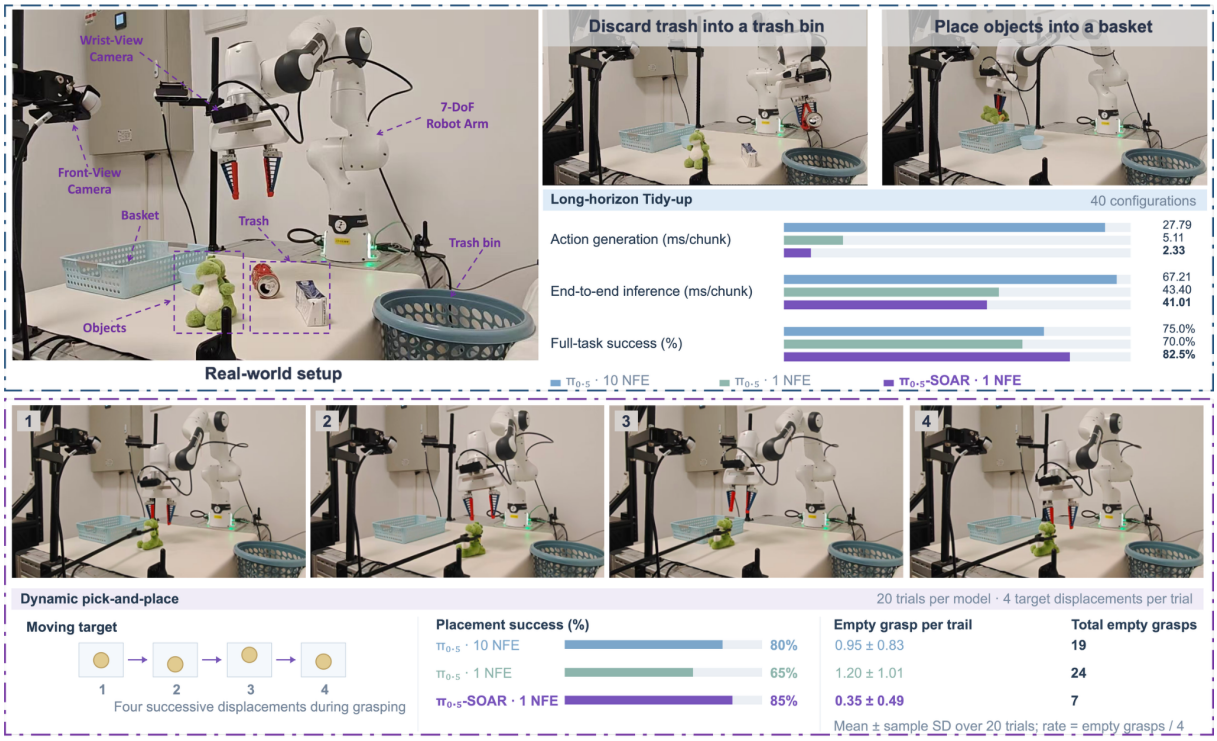

图 3。 $\pi_{0.5}$ 和 $\pi_{0.5}$-SOAR 的真实世界评估。**左上：** 实验设置包括一个 7自由度机械臂、腕部视角摄像头和正面视角摄像头、桌面物体、一个收纳篮和一个垃圾桶。**右上：** 长时程整理任务的执行示例，在 40 个配置上评估完整任务成功率和推理延迟。**底部：** 动态抓取与放置过程中进行四次连续的目标位移，每个模型进行 20 次试验评估。结果包括放置成功率、每次试验的空抓（均值 ± 样本标准差）以及空抓总数。

抓取次数减少了 63.2%，同时获得了更高的放置成功率。这些结果表明，尽管动作生成仅使用一次网络评估，SOAR 在反复目标位移下仍能保持可靠的操作性能。

4.4 消融研究

消融设置。 我们在 20 个操作任务上评估 SOAR，包括 3 个 Adroit任务（锤子、门和笔）以及 17 个 Meta-World任务（11 个中等难度和 6 个困难难度）。每个任务均使用 10 条专家演示和 3 个随机种子独立训练，其中 Adroit 训练 3,000 个训练轮次，Meta-World 训练 1,000 个训练轮次。按照主要评估协议，我们对每个任务和随机种子的 5 个最高定期评估成功率取平均值，并报告跨随机种子的均值和样本标准差。

我们考察四个组成部分：结构监督（SS）、多频带条件化（MC）、动作校准（AC）和推理端点对齐（EA）。SS 训练一个辅助头，根据观测预测未来动作轨迹的关键结构信息，而非将其用作独立的动作预测器。MC 将预测出的频率分量纳入生成条件，AC 则利用相同的结构估计来校正生成的动作。EA 使训练与单步推理条件对齐。我们逐步将这些组成部分加入直接行动基线，并进一步评估不含 MC 或 AC 的变体。我们还以直接动作空间校正替代谱校准，并比较直接干净动作预测（x预测）与直接 MeanFlow 速度预测（u预测）；在校准之前，根据预测速度还原基础动作。所有变体均保留单步推理（NFE $= 1$）。

结果。 表 5 报告了 20 项任务的结果。直接行动基线的平均成功率为 69.3%。仅加入结构监督（SS）后，平均值略微提升至 69.5%；Meta-World 中等和困难难度的表现没有改善，这与结构监督的作用相符：它用于学习关键的时间结构信息，而非直接预测可执行动作。通过多频带条件化（MC）融入这些信息后，平均值升至 70.5%；中等难度的成功率从 71.7% 提高到 73.0%，困难难度则从 60.9% 提高到 61.9%。

表5。 在 Adroit 和 Meta-World 上进行的组件消融。成功率（%）以均值 ± 样本标准差的形式报告，统计基于三个训练随机种子，并对每个任务和随机种子取五次最高的周期性评估结果。SS：结构监督；MC：多频带条件化；AC：动作校准；EA：推理端点对齐。在 AC 列中，✓ 表示谱校准，而直接表示动作空间校正。u预测变体直接预测 MeanFlow 输运速度，而非干净动作，并在校准前恢复基础动作。平均值是对 20 项任务按任务加权计算所得，基于表中显示的组均值。“–”表示缺少相应组件。粗体表示门、笔、中等、困难和平均值列中的最高均值。

					Adroit (3)			Meta-World (17)		
变体	**结构信息**	**多频带条件**	**AC**	**EA**	**锤子**	**Door**	**Pen**	**中等 (11)**	**困难 (6)**	**平均值**
直接行动基线	–	–	–	–	100.0 ± 0.0	73.6 ± 4.1	58.2 ± 4.3	71.7 ± 1.3	60.9 ± 2.6	69.3
+ SS	✓	–	–	–	100.0 ± 0.0	74.1 ± 4.3	61.1 ± 5.5	71.7 ± 0.2	60.9 ± 1.9	69.5
+ 结构信息 + 多频带条件	✓	✓	–	–	100.0 ± 0.0	75.4 ± 5.1	59.7 ± 5.3	73.0 ± 0.4	61.9 ± 2.9	70.5
+结构信息 +多频带条件 +动作校准	✓	✓	✓	–	100.0 ± 0.0	77.1 ± 2.0	59.3 ± 6.2	73.6 ± 0.5	62.4 ± 2.3	71.0
SOAR	✓	✓	✓	✓	100.0 ± 0.0	77.5 ± 2.4	61.9 ± 7.3	**74。3 ± 0。6**	63.5 ± 1.8	**71.9**
不含 MC 的 SOAR	✓	–	✓	✓	100.0 ± 0.0	76.7 ± 5.0	58.7 ± 4.9	72.9 ± 1.3	62.9 ± 1.9	70.7
SOAR w/o 动作校准	✓	✓	–	✓	100.0 ± 0.0	**77.7 ± 3.3**	59.5 ± 6.3	72.3 ± 0.9	62.2 ± 3.0	70.3
SOAR（直接校正）	✓	✓	直接	✓	100.0 ± 0.0	75.6 ± 4.4	62.7 ± 9.1	72.3 ± 0.3	60.9 ± 3.3	70.0
SOAR，带有u预测	✓	✓	✓	✓	99.2 ± 0.7	75.3 ± 2.9	**68.3 ± 5.6**	70.7 ± 2.0	**65。7 ± 2。1**	70.7

加入动作校准后，平均成功率进一步提升至 71.0%，表明多频带条件和动作校准有效利用了通过结构信息学习获得的结构信息。最后，推理端点对齐将平均成功率提升至 71.9%，其中中等和困难任务分别达到 74.3% 和 63.5%。这一进一步提升印证了使训练与单步推理条件保持一致的益处。

从完整模型中移除多频带条件或动作校准后，平均成功率分别降至 70.7% 和 70.3%；而将谱校准替换为直接动作空间校正后，平均成功率为 70.0%。这些结果进一步证实了结构条件化和谱校准对动作生成的贡献。我们还比较了动作参数化方式。尽管u预测在笔（68.3% 对 61.9%）和 Meta-World困难任务（65.7% 对 63.5%）上的表现更好，但在门（75.3% 对 77.5%）和 Meta-World Medium（70.7% 对 74.3%）上的表现较差。总体而言，直接干净动作预测取得了更高的平均成功率（71.9% 对 70.7%），但其相对优势因任务而异。

5 结论

我们提出了 SOAR，这是一种结构引导的单步策略，可从观测中预测关键的时间结构信息，以指导和校准干净动作的生成。通过纠正生成的动作中的结构差异，SOAR 无需对生成式骨干网络进行额外评估，即可提升动作质量。SOAR 在 53 项 Adroit 和 Meta-World 任务上的总体成功率达到 83.9%，推理延迟为 8.7 毫秒。与在 LIBERO 和 RoboTwin 2.0 上复现的一步基线相比，SOAR 表现更优；再结合其在真实世界实验中的提升，这些结果证明了所提方法的有效性及其向视觉-语言-动作策略迁移的能力。

人工智能使用声明

生成式人工智能工具协助进行了实验结果解读、方法反馈、稿件撰写与编辑，以及文献综述。作者对最终内容及所有经人工智能辅助生成的材料承担全部责任。

可复现性声明

第3 节介绍 SOAR架构和训练目标。附录 A.1提供频域结构预测与校准、MeanFlow 训练、推理端点对齐和色散特征正则化的实现细节。附录A.1.5 报告 DCT配置、损失权重以及学习得到的校准与条件参数。第 4 节介绍实验

设置和评估协议，包括 3D 仿真的训练随机种子和成功率汇总流程。附录 A.2 报告 Adroit 和 Meta-World 上各任务的结果，并说明推理延迟测量协议。附录A.3 介绍 LIBERO 和 RoboTwin 2.0 的模型初始化、训练配置和评估协议。附录A.4 介绍真实机器人数据、训练和部署设置以及评估指标。表 9 和 10 分别报告每种整理配置和每次动态抓取与放置试验的结果。

参考文献

[1] Nasir Ahmed、T. Natarajan 和 K. R. Rao。离散余弦变换。*IEEE*计算机汇刊，C-23(1):90–93，1974。doi: 10.1109/T-C.1974.223784。

[2] Kevin Black、Noah Brown、James Darpinian、Karan Dhabalia、Danny Driess、Adnan Esmail、Michael Equi、Chelsea Finn、Niccolo Fusai、Manuel Y. Galliker、Dibya Ghosh、Lachy Groom、Karol Hausman、Brian Ichter、Szymon Jakubczak、Tim Jones、Liyiming Ke、Devin LeBlanc、Sergey Levine、Adrian Li-Bell、Mohith Mothukuri、Suraj Nair、Karl Pertsch、Allen Z. Ren、Lucy Xiaoyang Shi、Laura Smith、Jost Tobias Springenberg、Kyle Stachowicz、James Tanner、Quan Vuong、Homer Walke、Anna Walling、Haohuan Wang、Lili Yu 和 Ury Zhilinsky。 $\pi_{0.5}$：一种具有开放世界泛化能力的视觉-语言-动作模型。载于机器人学习会议，2025。

[3] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, Laura Smith, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang 和 Ury Zhilinsky。 π_0：一种用于通用机器人控制的视觉-语言-动作流模型。载于机器人学：科学与系统，2025。doi：10.15607/RSS.2025.XXI.010。

[4] Kevin Black、Manuel Y. Galliker 和 Sergey Levine。动作分块流策略的实时执行。载于神经信息处理系统进展，第 38 卷，2025。

[5] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alexander Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Man- junath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael S. Ryoo, Grecia Salazar, Pannag R. Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan H. Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu 和 Brianna Zitkovich。RT-1：用于大规模真实世界控制的机器人变换器。发表于机器人学：科学与系统，2023。doi：10.15607/RSS.2023.XIX.025。

[6] Qingwen Bu、Yanting Yang、Jisong Cai、Shenyuan Gao、Guanghui Ren、Maoqing Yao、Ping Luo 和 Hongyang Li。《UniVLA：通过以任务为中心的潜在动作学习随处行动》。发表于机器人学：科学与系统，2025。

[7] Tianxing Chen、Zanxin Chen、Baijun Chen、Zijian Cai、Yibin Liu、Zixuan Li、Qiwei Liang、Xian- liang Lin、Yiheng Ge、Zhenyu Gu 等。《RoboTwin 2.0：一种具有强大领域随机化能力、用于稳健双手机器人操作的可扩展数据生成器和基准测试》。*arXiv* 预印本 *arXiv:2506.18088*，2025。

[8] Cheng Chi、Siyuan Feng、Yilun Du、Zhenjia Xu、Eric Cousineau、Benjamin Burchfiel 和 Shuran Song。《Diffusion Policy：通过动作扩散进行视觉运动策略学习》。发表于机器人学：科学与系统，2023。doi：10.15607/RSS.2023.XIX.026。

[9] Samarth Chopra、Alex McMoil、Ben Carnovale、Evan Sokolson、Rajkumar Kubendran 和 Samuel Dickerson。EveryDayVLA：一种用于经济型机器人操作的视觉-语言-动作模型。*arXiv* 预印本 *arXiv:2511.05397*，2025。

[10] Fan Du、Feng Yan、Jianxiong Wu、Xinrun Xu、Weiye Zhang、Weinong Wang、Yu Guo、Bin Qian、Zhihai He、Fei Wang 和 Heng Yang。CF-VLA：面向视觉-语言-动作策略的高效由粗到细动作生成。载于 ACM 多媒体国际会议论文集，2026。doi：10.1145/3767308.3836126。[11] Han Fang、Yize Huang、Yuheng Zhao、Paul Weng、Xiao Li 和 Yutong Ban。OMP：具有方向对齐的一步 MeanFlow 策略。载于第 43 届国际机器学习会议论文集，2026。[12] Zifeng Gao、Guanxing Lu、Tianxing Chen、Wenxun Dai、Ziwei Wang、Chao Shang、Wenbo Ding 和 Yansong Tang。ManiCM：通过一致性模型实现实时三维扩散策略，用于机器人操作。载于 2026 年 IEEE 实时计算与机器人国际会议（RCAR），2026。[13] Zhengyang Geng、Mingyang Deng、Xingjian Bai、J. Zico Kolter 和 Kaiming He。用于一步生成建模的均值流。载于神经信息处理系统进展，第 38 卷，第 84032–84054 页，2025。doi：10.52202/085713-2534。[14] Theophile Gervet、Zhou Xian、Nikolaos Gkanatsios 和 Katerina Fragkiadaki。Act3D：用于多任务机器人操作的三维特征场 Transformer。载于第 7 届机器人学习会议论文集，机器学习研究论文集第 229 卷，第 3949–3965 页。PMLR，2023。[15] Ankit Goyal、Jie Xu、Yijie Guo、Valts Blukis、Yu-Wei Chao 和 Dieter Fox。RVT：用于三维物体操作的机器人视角 Transformer。载于第 7 届机器人学习会议论文集，机器学习研究论文集第 229 卷，第 694–710 页。PMLR，2023。[16] Youqiang Gui、Yuxuan Zhou、Shen Cheng、Xinyang Yuan、Haoqiang Fan、Peng Cheng 和 Shuaicheng Liu。SeedPolicy：通过自进化扩散策略扩展机器人操作的时域。arXiv 预印本 arXiv:2603.05117，2026。[17] Jonathan Ho、Ajay Jain 和 Pieter Abbeel。去噪扩散概率模型。载于神经信息处理系统进展，第 33 卷，第 6840–6851 页，2020。[18] Zhi Hou、Tianyi Zhang、Yuwen Xiong、Haonan Duan、Hengjun Pu、Ronglei Tong、Chengyang Zhao、Xizhou Zhu、Yu Qiao、Jifeng Dai 和 Yuntao Chen。Dita：面向通用视觉-语言-动作策略的扩散 Transformer 扩展。载于 IEEE/CVF 国际计算机视觉会议论文集，2025。[19] Xixi Hu、Bo Liu、Xingchao Liu 和 Qiang Liu。AdaFlow：采用方差自适应基于流的策略进行模仿学习。载于神经信息处理系统进展，第 37 卷，第 138836–138858 页，2024。doi：10.52202/079017-4405。[20] Bofang Jia、Pengxiang Ding、Can Cui、Mingyang Sun、Pengfang Qian、Siteng Huang、Zhaoxin Fan 和 Donglin Wang。分数与分布匹配策略：通过匹配蒸馏实现先进的加速视觉运动策略。arXiv 预印本 arXiv:2412.09265，2024。[21] Yina Jian、Di Tian、Xuan-Jing Chen、Zhen-Yuan Wei、Chen-Wei Liang 和 Mu-Jiang-Shan Wang。PI-VLA：用于长程视觉-语言-动作操作的自适应对称性感知决策。Symmetry，18(3)：394，2026。doi：10.3390/sym18030394。[22] Tsung-Wei Ke、Nikolaos Gkanatsios 和 Katerina Fragkiadaki。3D Diffuser Actor：基于三维场景表示的策略扩散，2024。URL https://arxiv.org/abs/2402.10885。[23] Moo Jin Kim、Chelsea Finn 和 Percy Liang。微调视觉-语言-动作模型：优化速度与成功率。载于机器人学：科学与系统，2025。doi：10.15607/RSS.2025.XXI.017。[24] Moo Jin Kim、Yihuai Gao、Tsung-Yi Lin、Yen-Chen Lin、Yunhao Ge、Grace Lam、Percy Liang、Shuran Song、Ming-Yu Liu、Chelsea Finn 和 Jinwei Gu。Cosmos 策略：微调视频模型以实现视觉运动控制与规划。载于国际学习表征会议，2026。

[25] Jason Lee, Jiafei Duan, Haoquan Fang, Yuquan Deng, Shuo Liu, Boyang Li, Bohan Fang, Jieyu Zhang, Yi Ru Wang, Sangho Lee, Winson Han, Wilbert Pumacay, Angelica Wu, Rose Hendrix, Karen Farley, Eli VanderBilt, Ali Farhadi, Dieter Fox, 和 Ranjay Krishna。MolmoAct：能够进行空间推理的动作推理模型。arXiv 预印本 arXiv:2508.07917，2025。[26] Hao Li, Shuai Yang, Yilun Chen, Xinyi Chen, Xiaoda Yang, Yang Tian, Hanqing Wang, Tai Wang, Dahua Lin, Feng Zhao, 和 Jiangmiao Pang。通过多帧视觉-语言-动作建模实现高效、稳健的操作。载于 AAAI人工智能会议论文集，第40卷，第18388–18396页，2026。doi: 10.1609/aaai.v40i22.38903。[27] Shaolong Li, Lichao Sun, 和 Yongchao Chen。单步流策略：通过自蒸馏实现快速视觉运动策略。arXiv 预印本 arXiv:2603.12480，2026。[28] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, 和 Matt Le。用于生成建模的流匹配。载于国际学习表征会议，2023。[29] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, 和 Peter Stone。LIBERO：终身机器人学习知识迁移基准。载于神经信息处理系统进展，第36卷，2023。[30] Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, 和 Jun Zhu。RDT-1B：用于双手操作的扩散基础模型。载于国际学习表征会议，2025。[31] Xingchao Liu, Chengyue Gong, 和 Qiang Liu。让流变直、变快：学习通过 Rectified Flow 生成和迁移数据。载于国际学习表征会议，2023。URL https://openreview.net/forum?id=XVjTT1nw5z。[32] Yiyang Lu, Susie Lu, Qiao Sun, Hanhong Zhao, Zhicheng Jiang, Xianbang Wang, Tianhong Li, Zhengyang Geng, 和 Kaiming He。利用像素均值流实现一步无潜变量图像生成，2026。URL https://arxiv.org/abs/2601.22158。[33] Xiao Ma, Sumit Patidar, Iain Haughton, 和 Stephen J ames。面向运动学感知多任务机器人操作的分层扩散策略。载于 IEEE/CVF计算机视觉与模式识别会议论文集，第18081–18090页，2024。doi: 10.1109/CVPR52733.2024. 01712。[34] Chaoyi Pan, Giridharan Anantharaman, Nai-Chieh Huang, Claire Jin, Daniel Pfrommer, Chenyang Yuan, Frank Permenter, Guannan Qu, Nicholas Boffi, Guanya Shi, 和 Max Simchowitz。大谈噪声：揭穿生成式机器人控制的迷思。载于国际学习表征会议，2026。[35] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, 和 Aaron C. Courville。FiLM：通过通用条件层进行视觉推理。载于AAAI人工智能会议论文集，第32卷，第3942–3951页，2018。doi: 10.1609/aaai.v32i1.11671。[36] Aaditya Prasad, Kevin Lin, Jimmy Wu, Linqi Zhou, 和 Jeannette Bohg。Consistency Policy：通过一致性蒸馏加速视觉运动策略。载于机器人学：科学与系统，2024。doi: 10.15607/RSS.2024.XX.071。[37] Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, 和 Sergey Levine。结合深度强化学习和演示数据学习复杂的灵巧操作。载于机器人学：科学与系统，2018。doi: 10.15607/RSS.2018.XIV.049。[38] Moritz Reuss, Hongyi Zhou, Marcel Rühle, Ömer Erdinç Yaˇgmurlu, Fabian Otto, 和 Rudolf Lioutikov。FLOWER：借助高效视觉-语言-流模型普及通用机器人策略。载于第9届机器人学习会议论文集，机器学习研究论文集第305卷，第3736–3761页。PMLR，2025。[39] Tim Salimans 和 Jonathan Ho。通过渐进式蒸馏加速扩散模型采样。CoRR，abs/2202.00512，2022。URL https://arxiv.org/abs/2202.00512。

[40] Nur Muhammad Mahi Shafiullah, Zichen Cui, Ariuntuya Altanzaya, 和 Lerrel Pinto。行为变换器：用一块石头克隆*k* 种模式。载于神经信息处理系统进展，第35卷，第22955–22968页，2022。doi: 10.52202/068431-1668。[41] Juyi Sheng, Ziyi Wang, Peiming Li, 和 Mengyuan Liu。MP1：MeanFlow 让机器人操作的单步策略学习更加得心应手。*AAAI*人工智能会议论文集，40(22)：18532–18539，2026。doi: 10.1609/aaai.v40i22.38919。[42] Hao Shi, Bin Xie, Yingfei Liu, Lin Sun, Fengrong Liu, Tiancai Wang, Erjin Zhou, Haoqiang Fan, Xiangyu Zhang, 和 Gao Huang。MemoryVLA：面向机器人操作的视觉-语言-动作模型中的感知认知记忆。载于国际学习表征会议，2026。[43] Jiaming Song, Chenlin Meng, 和 Stefano Ermon。去噪扩散隐式模型。载于国际学习表征会议，2021。URL https://openreview.net/forum?id=St1giarCHLP。[44] Yang Song 和 Prafulla Dhariwal。改进一致性模型的训练技术。载于国际学习表征会议，2024。URLhttps://openreview.net/forum?id=WNzy9bRDvG。[45] Yang Song, Prafulla Dhariwal, Mark Chen, 和 Ilya Sutskever。一致性模型。载于第*40*届国际机器学习会议论文集，机器学习研究论文集第*202*卷，第32211–32252页。PMLR，2023。[46] Yifei Su, Ning Liu, Dong Chen, Zhen Zhao, Kun Wu, Meng Li, Zhiyuan Xu, Zhengping Che, 和 Jian Tang。FreqPolicy：通过频率一致性实现高效的基于流的视觉运动策略。载于神经信息处理系统进展，第38卷，第31574–31602页，2025。doi: 10.52202/085713-0937。[47] Denis Tarasov, Alexander Nikulin, Ilya Zisman, Albina Klepach, Nikita Lyubaykin, Andrei Pol- ubarov, Alexander Derevyagin, 和 Vladislav Kurenkov。NinA：动作中的归一化流。使用归一化流训练视觉-语言-动作模型。*arXiv* 预印本 *arXiv:2508.16845*，2025。[48] Yuqi Wang, Xinghang Li, Wenxuan Wang, Junbo Zhang, Yingyan Li, Yuntao Chen, Xinlong Wang, 和 Zhaoxiang Zhang。统一视觉语言动作模型。载于国际学习表征会议，2026。[49] Zhendong Wang, Zhaoshuo Li, Ajay Mandlekar, Zhenjia Xu, Jiaojiao Fan, Yashraj Narang, Linxi Fan, Yuke Zhu, Yogesh Balaji, Mingyuan Zhou, Ming-Yu Liu, 和 Yu Zeng。一步扩散策略：通过扩散蒸馏实现快速视觉运动策略，2024。URL https://arxiv.org/abs/2410.21257。[50] Zhou Xian, Nikolaos Gkanatsios, Theophile Gervet, Tsung-Wei Ke, 和 Katerina Fragkiadaki。ChainedDiffuser：统一轨迹扩散与关键姿态预测，实现机器人操作。载于第*7*届机器人学习会议论文集，机器学习研究论文集第*229*卷，第2323–2339页。PMLR，2023。[51] Siyu Xu, Yunke Wang, Chenghao Xia, Dihao Zhu, Tao Huang, 和 Chang Xu。VLA-Cache：通过自适应令牌缓存实现高效的视觉-语言-动作操作。载于神经信息处理系统进展，第38卷，2025。doi: 10.52202/085713-5484。[52] Jingyun Yang, Ziang Cao, Congyue Deng, Rika Antonova, Shuran Song, 和 Jeannette Bohg。EquiBot：用于可泛化且数据高效学习的 SIM(3)-等变扩散策略。载于第*8*届机器人学习会议论文集，机器学习研究论文集第*270*卷，第1048–1068页。PMLR，2025。[53] Ling Yang, Zixiang Zhang, Zhilong Zhang, Xingchao Liu, Minkai Xu, Wentao Zhang, Chenlin Meng, Stefano Ermon, 和 Bin Cui。一致性流匹配：以速度一致性定义直流。*arXiv* 预印本 *arXiv:2407.02398*，2024。

[54] Shuai Yang、Hao Li、Bin Wang、Yilun Chen、Yang Tian、Tai Wang、Hanqing Wang、Feng Zhao、Yiyi Liao 和 Jiangmiao Pang。《InstructVLA：从理解到操作的视觉-语言-动作指令微调》。载于国际学习表征会议，2026年。

[55] Jinhui Ye、Ning Gao、Senqiao Yang、Jinliang Zheng、Zixuan Wang、Yuxin Chen、Pengguang Chen、Yilun Chen、Shu Liu 和 Jiaya Jia。《StarVLA-α：降低视觉-语言-动作系统的复杂度》。载于欧洲计算机视觉会议，2026年。

[56] Tianhe Yu、Deirdre Quillen、Zhanpeng He、Ryan Julian、Karol Hausman、Chelsea Finn 和 Sergey Levine。《Meta-World：多任务与元强化学习的基准和评估》。载于机器人学习会议论文集，第100卷机器学习研究论文集，第1094–1100页。PMLR，2020年。

[57] Yanjie Ze、Ge Yan、Yueh-Hua Wu、Annabella Macaluso、Yuying Ge、Jianglong Ye、Nicklas Hansen、Li Erran Li 和 Xiaolong Wang。GNFactor：利用可泛化神经特征场进行多任务真实机器人学习。发表于第七届机器人学习会议论文集，机器学习研究论文集第 229 卷，第 284–301 页。PMLR，2023。

[58] Yanjie Ze、Gu Zhang、Kangning Zhang、Chenyuan Hu、Muhan Wang 和 Huazhe Xu。三维扩散策略：通过简单的三维表征实现可泛化的视觉运动策略学习。发表于机器人学：科学与系统，2024。doi: 10.15607/RSS.2024.XX.067。

[59] Jianke Zhang、Yanjiang Guo、Yucheng Hu、Xiaoyu Chen、Xiang Zhu 和 Jianyu Chen。UP-VLA：面向具身智能体的统一理解与预测模型。发表于第 *42* 届国际机器学习会议论文集，机器学习研究论文集第 267 卷，第 74911–74922 页。PMLR，2025。

[60] Qinglun Zhang、Zhen Liu、Haoqiang Fan、Guanghui Liu、Bing Zeng 和 Shuaicheng Liu。Flow- Policy：通过一致性流匹配实现快速稳健的三维流策略，用于机器人操作。*AAAI*人工智能会议论文集，39(14):14754–14762，2025。doi: 10.1609/aaai.v39i14.33617。

[61] Wenyao Zhang、Hongsi Liu、Zekun Qi、Yunnan Wang、XinQiang Yu、Jiazhao Zhang、Runpei Dong、Jiawei He、He Wang、Zhizheng Zhang、Li Yi、Wenjun Zeng 和 Xin Jin。DreamVLA：一个基于全面世界知识构想而成的视觉-语言-动作模型。发表于神经信息处理系统进展，第 38 卷，2025。

[62] Tony Z. Zhao、Vikash Kumar、Sergey Levine 和 Chelsea Finn。利用低成本硬件学习细粒度双手操作。发表于机器人学：科学与系统，2023。doi: 10.15607/RSS. 2023.XIX.016。

[63] Jinliang Zheng、Jianxiong Li、Dongxiu Liu、Yinan Zheng、Zhihao Wang、Zhonghong Ou、Yu Liu、Jingjing Liu、Ya-Qin Zhang 和 Xianyuan Zhan。《通用动作：增强具身基础模型》。载于*IEEE/CVF*计算机视觉与模式识别会议论文集，第 22508–22519 页，2025 年。[64] Jinliang Zheng、Jianxiong Li、Zhihao Wang、Dongxiu Liu、Xirui Kang、Yuchun Feng、Yinan Zheng、Jiayin Zou、Yilun Chen、Jia Zeng、Tai Wang、Ya-Qin Zhang、Jingjing Liu 和 Xianyuan Zhan。《X-VLA：作为可扩展跨具身视觉-语言-动作模型的软提示变换器》。载于国际学习表征会议，2026年。
[65] Yiming Zhong、Yumeng Liu、Chuyang Xiao、Zemin Yang、Youzhuo Wang、Yufei Zhu、Ye Shi、Yujing Sun、Xinge Zhu 和 Yuexin Ma。《FreqPolicy：采用连续令牌的频域自回归视觉运动策略》。载于神经信息处理系统进展，第 38 卷，2025 年。

A 附录

A.1 补充方法细节

本附录补充介绍频域结构模块、MeanFlow 训练、推理端点对齐以及色散特征正则化的实现细节。为使正文更加清晰，这些细节未在正文中展开。

A.1.1 频域结构细节

结构头由三个线性层组成，其中有两个 512 维隐藏层。前两个层之后均使用 Mish 激活函数，输出层则为线性层。该头将观测条件映射为预测的离散余弦变换系数，并将其重塑为频率×动作通道格式。该头接收由观测得出的条件；演示中的未来动作仅用作训练目标。我们沿时间维度对每个纳入的动作通道分别应用正交离散余弦变换 II。逆变换使用正交变换矩阵的转置。我们保留完整频谱，系数数量等于预测时域：Adroit/Meta-World 为 4，LIBERO 为 10，RoboTwin 为 50，真实机器人为 16。对于真实机器人适配，结构监督、频谱条件调制和谱校准仅作用于七个连续关节通道，不包括二值夹爪。

多频带结构条件调制。 令K 表示结构分支所表示的离散余弦变换系数数量。我们将归一化频率索引定义为$\nu_k = {}_{kK-1}$，并将预测频谱划分为低频组、中频组和高频组：$\mathcal{B}_L = \{k \dashv \nu_k < \frac{1}{2}\}$、$\mathcal{B}_M = \{k \mid \frac{1}{2} \le \nu_k < \frac{3}{4}\}$和$\mathcal{B}_H = \{k \mid \nu_k \ge \frac{3}{4}\}$。低频系数主要描述缓慢变化的轨迹结构，而高频系数则捕捉时间尺度逐渐变短的时间变化。

三个频率组分别投影到观测特征空间，并与全局条件融合：

$$\tilde{\mathbf{c}} = \mathbf{c} + g_L P_L\left(\mathbf{C}^s_{\mathcal{B}_L}\right) + g_M P_M\left(\mathbf{C}^s_{\mathcal{B}_M}\right) + g_H P_H\left(\mathbf{C}^s_{\mathcal{B}_H}\right), \tag{8}$$

其中P_L、P_M和P_H 是可学习投影，g_L、g_M和g_H 是可学习标量门控。所有门控均初始化为零，因此在初始化时得到$\tilde{\mathbf{c}} = \mathbf{c}$。这种初始化在训练开始时保留原始观测条件，并逐步引入结构引导。

自适应结构监督。 离散余弦变换系数的尺度可能存在显著差异。因此，即使低能量频率仍然难以预测，统一的频谱目标也可能被高能量分量主导。我们改为根据每个系数自身的目标能量来衡量其预测难度。

对于样本 b 和频率系数 k，我们将预测误差和目标能量定义为 $e_{b,k} = {}^{1}D_a \sum^{D_a} d{=}1(C^s_{b,k,d} - C^{gt}$ $b,k,d)^2$, 和 $E_{b,k} = {}^{1}D_a \sum^{D_a} d{=}1(C^{gt}$ $b,k,d)^2$。相对预测难度为 $d_{b,k} = e_{b,k}$ $\max(E_{b,k},10^{-6})$ 。

我们将难度分数转换为归一化正权重：

$$\omega_{b,k} = \frac{\exp\left(\gamma_s \,\mathrm{sg}[d_{b,k}]\right)}{\sum_{j=0}^{K-1} \exp\left(\gamma_s \,\mathrm{sg}[d_{b,j}]\right)}, \qquad \gamma_s = 2, \tag{9}$$

其中 sg[·] 表示停止梯度。结构目标为 $\mathcal{L}_{\text{struct}} = {}_{B} \sum_{b=1}^{B} \sum^{K-1}$ $_{k=0}$ $\omega_{b,k}$ $_{b,}$ $_{k}$.

因此，相对于各自目标能量而言预测误差较大的系数会得到更强的监督。该目标沿用了 FreqPolicy [46], 的自适应谱重加权原理，但学习目标有所不同。FreqPolicy 对不同流时间下速度预测之间的频域差异进行加权。这里的权重则根据观测条件化结构预测 C^s 与示范频谱 C^{gt}之间的误差计算，并按目标系数能量进行归一化。此外，推理时还会通过多频带条件化和动作校准来使用预测频谱。因此，二者的区别不仅在于是否使用 DCT 或自适应加权，还在于结构分支学习什么，以及其预测如何参与动作生成。

A.1.2 MeanFlow 训练细节

线性概率路径。训练期间，我们采样高斯噪声 $\epsilon \sim \mathcal{N}(\mathbf{0},\mathbf{I})$，并构造中间动作状态 $\mathbf{z}_t = (1-t)\mathbf{x} + t\epsilon$，其中 $t \in [0,1]$。该路径在 $t = 0$ 时的示范干净动作 $\mathbf{x}$与 $t = 1$时的高斯噪声之间进行插值。

区间采样。遵循 MeanFlow [13, 41]，，我们采样 $\mathrm{a}_1, \mathrm{a}_2 \sim \mathcal{N}(-0.4, 1)$、$\mathrm{s}_i = \sigma(\mathrm{a}_i)$、$\mathrm{t} = \max(\mathrm{s}_1, \mathrm{s}_2)$ 和 $\mathrm{r} = \min(\mathrm{s}_1, \mathrm{s}_2)$，其中 $\sigma(\cdot)$ 表示 sigmoid 函数。这确保 $0 \le \mathrm{r} \le \mathrm{t} \le 1$。对于每个小批次中的一半样本，我们进一步设定 $\mathrm{r} = \mathrm{t}$，将有限区间的 MeanFlow 监督与零长度流匹配情形结合起来。

自引导 MeanFlow 目标物。对于线性概率路径，瞬时方向为 $\mathbf{v} = \epsilon - \mathbf{x}$。我们在 (t,t) 处评估当前输运场，且不进行梯度传播，并构造自引导方向

$$\widehat{\mathbf{v}} = w\mathbf{v} + (1-w)\,\mathrm{sg}\left[\mathbf{u}_\theta\left(\mathbf{z}_t, t, t, \widetilde{\mathbf{c}}\right)\right], \qquad w = 2. \tag{10}$$

输运场的方向导数通过雅可比-向量积计算：

$$\dot{\mathbf{u}}_\theta = \mathrm{JVP}_{(\mathbf{z},t,r)}\left[\mathbf{u}_\theta; (\widehat{\mathbf{v}}, \mathbf{1}, \mathbf{0})\right]_{(\mathbf{z}_t,t,r)}. \tag{11}$$

r 方向上的切向量设为零。

由此得到的 MeanFlow 目标为 $\mathbf{u}^{\mathrm{tgt}} = \mathrm{sg}[\widehat{\mathbf{v}} - (t-r)\dot{\mathbf{u}}_\theta]$。相应的目标函数为 $\mathcal{L}_{\mathrm{MF}} =$自适应 $\mathrm{L2}(\mathbf{u}_\theta - \mathbf{u}^{\mathrm{tgt}})$。

在 SOAR 中，u_θ 由校准后的动作通过式 (6) 得出，而非由独立的速度头预测。除数值截断之外，底层的数据到平均速度的转换与像素 MeanFlow [32], 具有相同的代数形式。SOAR 将这一公式适配到可执行动作块，并将其与观测条件化轨迹结构预测器相结合。预测结构对生成器进行条件化，并在评估 MeanFlow 目标函数之前校准其输出。因此，$\mathcal{L}_{\mathrm{MF}}$ 对策略生成的结构引导动作施加约束。结合附录 A.1.3 中的推理端点目标函数，该设计将预测的时间结构与最终动作的单步生成和训练相结合，同时在推理时仍只需对生成式骨干网络进行一次评估。

自适应回归。对于逐样本残差 δ，自适应 L2 指标的实现方式为

$$\begin{aligned} \mathrm{AdaptiveL2}(\delta) &= \frac{1}{B}\sum_{b=1}^{B} \mathrm{sg}\left[\left(m_b + 10^{-3}\right)^{-1/2}\right] m_b, \\ m_b &= \mathrm{mean}\left(\delta_b^2\right). \end{aligned} \tag{12}$$

分离出的因子会自适应地重新缩放样本级回归误差，同时避免梯度修改加权项本身。

A.1.3 推理端点对齐细节

随机区间 MeanFlow 训练不会显式优化单步推理时遇到的固定条件。因此，SOAR 在$(t, r) = (1, 0)$处引入了一个额外的目标。

我们独立采样 $\epsilon_{\mathrm{end}} \sim \mathcal{N}(\mathbf{0},\mathbf{I})$，并在此推理条件下评估同一个动作生成器。相应的真实端点速度和模型诱导的端点速度分别为$\mathbf{u}^{\mathrm{gt}}$ end $= \epsilon_{\mathrm{end}} - \mathbf{x}, \mathbf{u}^{\theta}_{\mathrm{end}} = \epsilon_{\mathrm{end}} - \widehat{\mathbf{A}}_{\mathrm{end}}$，其中 $\widehat{\mathbf{A}}_{\mathrm{end}}$ 表示在$(t, r) = (1, 0)$处预测的校准后的动作。

速度空间端点目标为

$$\mathcal{L}_{\mathrm{end}} = \mathrm{AdaptiveL2}\left(\mathbf{u}^{\theta}_{\mathrm{end}} - \mathbf{u}^{\mathrm{gt}}_{\mathrm{end}}\right). \tag{13}$$

在 $t = 1$时，两个端点速度共享同一个噪声样本，从而得到 $\mathbf{u}^{\theta}_{\mathrm{end}} - \mathbf{u}^{\mathrm{gt}}$ end $= \mathbf{x} - \widehat{\mathbf{A}}_{\mathrm{end}}$。因此，公式 13 在代数上等价于 $\mathcal{L}_{\mathrm{end}} = \mathrm{AdaptiveL2}(\mathbf{x} - \widehat{\mathbf{A}}_{\mathrm{end}})$，这也是正文中采用的紧凑形式。速度空间表述强调，该目标对齐的是实际的 MeanFlow 采样边界，而不是机器人的物理动力学。

表6。 跨越 9 个 Adroit 检查点和 150 个 Meta-World 检查点的学习得到的校准与条件参数。统计量涵盖全部三个 Adroit任务和全部 50 个 Meta-World任务，每个任务使用三个训练随机种子。除 α 的范围和均协方差缩放因子外，数值均为跨任务-随机种子检查点的均值 $\pm$ 样本标准差。其中，$\alpha := \tanh(\rho)$。

基准	α	α的范围	g_L	g_M	g_H	均值 $(1-\alpha)^2$
Adroit	0.118051 ± 0.081276	$[0.007725, 0.220070]$	-0.002653 ± 0.004563	0.000278 ± 0.005642	0.001601 ± 0.003492	0.7837
Meta-World	0.075216 ± 0.053296	$[0.015644, 0.273769]$	0.000244 ± 0.009540	-0.001454 ± 0.004646	-0.003051 ± 0.005112	0.8580

A.1.4 色散特征正则化

遵循 MP1 [41], ，我们在动作 U-Net 的中间表示上保留一个分散正则项。对于层 l 上的隐藏表示 $\mathbf{h}^{(l)}$，我们计算成对特征平方距离$d^{(l)}_{ij} = \|\mathbf{h}^{(l)}_i - \mathbf{h}^{(l)}_j\|_2^2$。

使用归一化距离 $\bar{d}^{(l)}_{ij}$，正则项为

$$\mathcal{L}_{\text{dis}} = \sum_l \log\left[\frac{1}{B^2}\sum_{i,j}\exp\left(-\frac{\bar{d}^{(l)}_{ij}}{\tau}\right)\right], \qquad \tau = 1. \tag{14}$$

最小化该目标会促使不同样本保持具有判别性的隐藏表示，并有助于减少单步生成中的表示坍塌或过度平均化预测。

A.1.5 学习到的校准与条件化参数

配置。 我们检查了全部三个 Adroit任务和全部 50 个 Meta-World任务的最佳 EMA检查点，每个任务使用三个训练随机种子。因此，共检查了 9 个 Adroit检查点和 150 个 Meta-World检查点。所有检查点均使用完整的 DCT表示，且 $K = T = 4$。低频组、中频组和高频组分别包含系数索引 $\{0,1\}$、$\{2\}$ 和 $\{3\}$。式 (8) 中的条件门控 g_L、g_M 和 g_H 是不受约束的可学习标量，直接作用于投影特征，不使用 Sigmoid函数或双曲正切激活。

式 (1) 中固定的损失权重为 $(\lambda_{\text{struct}}, \lambda_{\text{act}}, \lambda_{\text{dis}}, \lambda_{\text{end}}) = (0.05, 0.01, 0.5, 0.05)$，$\mathcal{L}_{\text{MF}}$ 的系数为 1。在各基准测试中，我们共享式 (1) 中的外部损失系数、正交归一时间DCT-II约定、全频谱表示以及宽度为 512 的结构头隐藏层。结构头深度和激活函数、观测编码器、预测长度与执行长度、初始化以及优化计划均因设置而异。附录 A.4 报告的额外时间正则化、夹爪加权、事件采样和 RTC概率描述了真实机器人适配，并不假定这些设置会在各仿真基准测试中共享。

学习参数统计。 表 6 报告了学习得到的校准系数和条件门控。均值和样本标准差根据任务-随机种子检查点计算。这些统计量描述的是训练后模型之间的差异，而非任务内训练随机种子的不确定性。

与确定性回归的关系。 对于此处使用的完整离散余弦变换，线性和可逆性使公式 (4) 与下式完全等价：

$$\hat{A} = (1-\alpha)A_{\text{base}} + \alpha\ \text{IDCT}(C^s(o)), \qquad \alpha := \tanh(\rho). \tag{15}$$

由于 $C^s(o)$ 仅取决于观测，因此随着 α 趋近于 1，输出也趋近于确定性结构预测。然而，测得的系数远未达到这一极限：Adroit 上的最大值为 0.220070，Meta-World 上的最大值为 0.273769。

Fo对于固定的观测和检查点，随机性来自高斯潜变量 ϵ，因此有：

$$\text{Cov}_\epsilon[\hat{A} \mid o] = (1-\alpha)^2\,\text{Cov}_\epsilon[A_{\text{base}} \mid o]. \tag{16}$$

检查点平均协方差缩放因子在 Adroit 上为 0.7837，在 Meta-World 上为 0.8580。这些结果表明，SOAR 保留了随机动作生成能力，而非在校准后趋近于确定性回归。

表 7。 Adroit 和 Meta-World 基准测试上的推理延迟对比。推理延迟衡量生成一个动作块所需的一次完整策略评估的实际运行时间。结果以三个随机种子上的平均延迟（毫秒） ± 标准差报告。对于 FlowPolicy，每个随机种子进行三次重复测量并取平均值。**粗体** 表示最佳结果。

			Adroit			Meta-World				
方法	发表信息	NFE	锤子	Door	Pen	简单 (28)	中等 (11)	困难 (6)	非常困难 (5)	推理延迟(ms)
DP [8]	RSS，2023	100	671.6 ± 8.7	693.9 ± 16.9	676.4 ± 2.7	676.5 ± 6.8	672.7 ± 5.8	681.0 ± 9.2	696.7 ± 23.4	678.4 ± 6.9
DP3 [58]	RSS，2024	10	69.0 ± 0.9	70.4 ± 1.8	80.6 ± 3.4	69.3 ± 1.7	69.9 ± 1.2	72.7 ± 2.3	76.5 ± 1.6	70.7 ± 2.6
简单 DP3 [58]	RSS，2024	10	43.7 ± 0.3	43.3 ± 0.1	43.6 ± 0.4	43.3 ± 0.4	43.4 ± 0.4	43.3 ± 0.4	43.4 ± 0.3	43.4 ± 0.4
FlowPolicy [60]	AAAI，2025	1	14.6 ± 0.1	14.7 ± 0.1	14.6 ± 0.0	14.7 ± 0.0	14.8 ± 0.0	14.7 ± 0.0	14.7 ± 0.1	14.7 ± 0.0
MP1 [41]	AAAI，2026年	1	**8.2 ± 0.0**	**8.2 ± 0.1**	**8.2 ± 0.0**	**8.1 ± 0.1**	**8.1 ± 0.1**	**8.0 ± 0.1**	**8.1 ± 0.1**	**8.1 ± 0.0**
SOAR（我们的方法）	–	**1**	8.7 ± 0.3	8.6 ± 0.6	8.6 ± 0.4	8.7 ± 0.2	8.6 ± 0.0	8.6 ± 0.0	8.9 ± 0.1	8.7 ± 0.1

A.2 使用 3D 输入进行仿真实验的更多详情

A.2.1 推理延迟比较的更多详情

表 7 报告了 Adroit 和 Meta-World 上的详细推理延迟。推理延迟是指完成一次策略评估并生成一个动作块所需的实际运行时间。所有结果均以均值 ± 标准差的形式报告，统计基于三个随机种子。对于 FlowPolicy，每个种子都进行了三次重复测量，并取其平均值后再计算最终统计量。动作生成所用的网络评估次数越多，延迟越高。DP 需要 678。4 ± 6.9 毫秒，NFE 为 = 100；DP3 和 Simple DP3 在 NFE 为 7 ± 2时分别需要 70。4 ± 0.6 毫秒和 43。 = 10.4 毫秒。在单步方法中，FlowPolicy 需要 14。7 ± 0.0 毫秒，MP1 需要 8。1 ± 0.0 毫秒，SOAR 需要 8。7 ± 0.1 毫秒。SOAR 在不同任务中的延迟相近，在 Adroit 和 Meta-World 上为 8.6 至 8.9 毫秒。与 MP1 相比，额外的结构预测和动作校准仅带来小幅延迟增加，同时生成式骨干网络仍只需评估一次。这些测量结果证实，SOAR 在 NFE 为 = 1时仍能保持毫秒级推理。

A.2.2 性能的更多详情

表8 展示了 53项任务基准中各任务的结果。每项结果均以三个随机种子下的平均成功率 ± 和样本标准差报告。对于多种方法，大多数简单任务的表现都接近饱和，而在中等、困难和非常困难的任务上，差异则更加明显。SOAR 在一系列尚未饱和的任务上表现出色，包括侧向插销、推墙、扫入、从孔中取出、搁板放置、拉杆和墙边拾取放置。它还提升了若干 Adroit 和简单任务的表现，例如门、拉动把手、侧向拉把手和拉杠杆。性能仍取决于具体任务，其他方法在若干任务上仍具有优势。因此，详细结果表明，SOAR 的总体提升主要来自对更多尚未饱和任务的改进。

A.3 VLA 实验的补充细节

本节补充介绍第 4.2 节中 VLA 实验的实现、训练和评估细节。在两个基准测试中，SOAR 替换了 $\pi_{0.5}$ 中的迭代动作专家，同时保留原有的视觉-语言骨干网络和数据处理流水线。

A.3.1 LIBERO

我们在全部四个标准 LIBERO 套件 [29] 上评估 $\pi_{0.5}$-SOAR：LIBERO-Spatial、LIBERO-Object、LIBERO-Goal 和 LIBERO-Long。每个套件包含 10 项任务，总计 40 项任务。所有任务共同训练一个策略。视觉-语言骨干网络以 OpenPI`pi05_base` 检查点为初始化。原有的迭代动作专家由 SOAR 替换，新引入的 SOAR 参数则从头初始化。我们使用 AdamW，在四块 NVIDIA A100-80GB GPU 上以 256 的全局批次大小对模型进行 30K 个优化步的微调。SOAR 在每次策略查询时预测一个包含 10 步的动作块。评估期间，执行前五个预测动作后，策略才会接收新的观测并重新规划。与原来的 $\pi_{0.5}$ 动作专家不同，后者需要多次网络评估来生成动作，而 SOAR 会生成完整的动作块，并使用一个

表8。 53项任务基准上各任务的详细表现。结果以三个随机种子下的平均成功率（%）±样本标准差报告。**粗体** 表示各列中的最佳结果。

方法	发表信息	NFE	Adroit 锤子	Adroit Door	Adroit Pen	Meta-World（简单） 按钮按压	Meta-World（简单） 咖啡按钮	Meta-World（简单） 向后滑动板	Meta-World（简单） 背面
DP	RSS, 2023	100	75.1 ± 3.5	23.7 ± 3.7	19.3 ± 1.8	79.7 ± 3.8	83.0 ± 3.5	99.3 ± 0.6	
DP3	RSS, 2024	10	**100.0 ± 0.0**	73.5 ± 2.1	56.7 ± 2.6	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**	
Simple DP3	RSS，2024	10	**100.0 ± 0.0**	56.8 ± 1.2	56.1 ± 0.3	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**	
FlowPolicy	AAAI，2025	1	95.2 ± 2.4	57.3 ± 6.2	47.8 ± 11.0	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**	
MP1	AAAI, 2026	1	96.1 ± 2.3	69.2 ± 3.1	57.1 ± 6.1	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**	
SOAR（我们的方法）	-	1	**100。0 ± 0。0**	**77。5 ± 2。4**	**61。9 ± 7。3**	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**	

方法	Meta-World（简单） 俯视按钮按压	墙面俯视按钮按压	墙面按钮按压	拔销	关闭侧门	门锁
DP	42.7 ± 7.6	39.3 ± 5.7	87.7 ± 5.7	32.0 ± 12.3	**100.0 ± 0.0**	66.3 ± 5.9
DP3	96.7 ± 5.8	88.0 ± 6.2	**100.0 ± 0.0**	71.3 ± 4.7	**100.0 ± 0.0**	73.7 ± 12.1
Simple DP3	96.0 ± 6.1	87.3 ± 7.8	**100.0 ± 0.0**	65.3 ± 14.2	**100.0 ± 0.0**	74.3 ± 21.1
FlowPolicy	**100。0 ± 0。0**	**100。0 ± 0。0**	**100。0 ± 0。0**	**85.0 ± 2.0**	**100。0 ± 0。0**	**100。0 ± 0。0**
MP1	**100。0 ± 0。0**	**100.0 ± 0.0**	99.7 ± 0.6	67.0 ± 4.6	**100.0 ± 0.0**	97.3 ± 2.9
SOAR（本文方法）	**100。0 ± 0。0**	**100。0 ± 0。0**	**100。0 ± 0。0**	83.7 ± 6.8	**100。0 ± 0。0**	**100。0 ± 0。0**

方法	Meta-World（简单） 开门	开锁	关闭抽屉	打开抽屉	关闭水龙头	打开水龙头	按压把手	拉动把手
DP	93.3 ± 2.5	98.7 ± 1.2	97.7 ± 2.1	55.7 ± 9.5	98.3 ± 1.5	89.0 ± 3.0	77.7 ± 2.5	9.0 ± 7.0
DP3	81.7 ± 3.5	99.3 ± 1.2	**100。0 ± 0。0**	97.0 ± 1.7	78.3 ± 11.4	**100。0 ± 0。0**	84.0 ± 1.0	21.3 ± 3.2
Simple DP3	86.7 ± 6.4	99.3 ± 1.2	**100.0 ± 0.0**	96.3 ± 4.0	90.7 ± 8.1	**100.0 ± 0.0**	84.0 ± 6.6	11.0 ± 5.3
FlowPolicy	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**	98.3 ± 2.9	13.7 ± 3.2
MP1	96.7 ± 0.6	98.7 ± 2.3	**100。0 ± 0。0**	**100。0 ± 0。0**	**100。0 ± 0。0**	**100。0 ± 0。0**	99.7 ± 0.6	41.3 ± 12.4
SOAR（我们的方法）	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**	**67.7 ± 15.0**

方法	Meta-World（简单） 按压把手	侧向拉把手	侧向拉杠杆	推盘	向后滑动板	旋转旋钮	到达	到达墙面
DP	60.3 ± 6.1	6.3 ± 1.2	44.0 ± 8.7	91.0 ± 1.0	**100。0 ± 0。0**	23.3 ± 3.8	37.7 ± 5.7	31.0 ± 10.5
DP3	48.3 ± 10.8	50.3 ± 9.9	77.3 ± 5.5	**100.0 ± 0.0**	**100。0 ± 0。0**	55.3 ± 5.5	44.0 ± 9.5	67.3 ± 3.1
Simple DP3	45.0 ± 7.0	35.7 ± 11.0	72.3 ± 12.9	**100.0 ± 0.0**	**100。0 ± 0。0**	56.0 ± 2.0	41.3 ± 11.0	59.7 ± 8.1
FlowPolicy	**100.0 ± 0.0**	54.0 ± 5.6	79.3 ± 3.5	**100.0 ± 0.0**	**100.0 ± 0.0**	85.3 ± 9.1	**45.3 ± 5.1**	**76.3 ± 3.1**
MP1	**100.0 ± 0.0**	56.7 ± 7.1	79.7 ± 2.1	**100.0 ± 0.0**	97.7 ± 1.5	70.0 ± 8.2	14.3 ± 2.1	62.0 ± 7.0
SOAR（我们的方法）	**100。0 ± 0。0**	**84.7 ± 8.5**	**96。0 ± 1。0**	**100。0 ± 0。0**	**100。0 ± 0。0**	**88.3 ± 1.5**	27.7 ± 4.0	74.7 ± 2.5

方法	Meta-World（简单） 推盘	关闭侧窗	打开窗户	篮球	料箱拣选	关闭箱子	Meta-World（中等） 拉咖啡	推咖啡
DP	98.0 ± 2.0	99.0 ± 1.7	92.0 ± 5.3	80.7 ± 2.5	20.3 ± 0.6	36.7 ± 5.7	22.0 ± 5.3	54.3 ± 2.1
DP3	**100.0 ± 0.0**	**100.0 ± 0.0**	94.0 ± 4.4	84.0 ± 5.0	23.3 ± 5.1	50.3 ± 11.0	58.0 ± 25.1	**74。0 ± 5。6**
Simple DP3	**100。0 ± 0。0**	**100。0 ± 0。0**	93.7 ± 5.7	66.3 ± 4.9	21.3 ± 6.1	46.3 ± 8.1	53.0 ± 18.1	69.3 ± 3.2
FlowPolicy	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**	97.7 ± 2.1	22.7 ± 4.7	49.7 ± 9.5	67.7 ± 4.0	64.0 ± 6.6
MP1	**100.0 ± 0.0**	**100.0 ± 0.0**	98.7 ± 2.3	80.0 ± 4.4	15.3 ± 9.1	49.3 ± 11.2	82.0 ± 6.0	59.3 ± 3.1
SOAR（我们的方法）	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**	**34。0 ± 7。5**	**65。7 ± 5。5**	**96.0 ± 2.6**	70.7 ± 3.2

方法	Meta-World（中等） 锤子	侧向插钉	推墙	足球	扫掠	Meta-World（困难） 扫入装配	手动插入	从孔中取出
DP	36.7 ± 15.3	31.0 ± 13.7	25.0 ± 15.6	8.0 ± 4.6	32.7 ± 10.6	13.0 ± 3.0	71.3 ± 7.0	10.3 ± 3.1	27.3 ± 8.6
DP3	93.7 ± 9.3	62.7 ± 4.5	74.3 ± 6.7	11.7 ± 5.0	98.7 ± 1.5	14.7 ± 4.2	94.0 ± 2.0	9.7 ± 2.3	15.7 ± 4.5
Simple DP3	89.3 ± 9.7	61.7 ± 11.2	58.3 ± 11.0	14.3 ± 4.9	91.0 ± 7.9	20.3 ± 4.5	93.7 ± 3.8	8.3 ± 0.6	14.7 ± 12.9
FlowPolicy	78.7 ± 6.0	63.7 ± 4.6	36.3 ± 7.0	14.7 ± 6.4	99.7 ± 0.6	21.0 ± 7.5	96.7 ± 5.8	9.7 ± 3.5	26.3 ± 14.6
MP1	98.3 ± 1.2	75.7 ± 3.5	72.7 ± 5.0	13.0 ± 5.6	93.7 ± 2.5	15.0 ± 1.7	98.0 ± 2.6	9.0 ± 2.6	11.7 ± 6.8
SOAR（我们的方法）	**100.0 ± 0.0**	**88.7 ± 4.0**	**99。7 ± 0。6**	**22。7 ± 6。0**	**100。0 ± 0。0**	**39。7 ± 12。9**	**100。0 ± 0。0**	**13.0 ± 4.0**	**42.3 ± 1.5**

方法	Meta-World（困难） 拾取放置	Push	向后推	搁板	Meta-World（非常困难） 放置	拆解棍子	拉杆	推、拾取、放置墙面	平均值	总体
DP	17.7 ± 8.1	24.3 ± 8.5	54.7 ± 13.1	4.3 ± 0.6	62.3 ± 3.5	21.0 ± 2.5	53.7 ± 10.8	16.3 ± 12.9	52.3 ± 0.9	
DP3	58.3 ± 3.5	72.0 ± 13.5	74.7 ± 12.7	15.7 ± 4.7	63.7 ± 6.1	45.0 ± 8.2	96.0 ± 4.4	83.3 ± 3.1	72.3 ± 1.4	
Simple DP3	60.0 ± 3.6	73.3 ± 15.3	75.7 ± 7.8	23.0 ± 7.2	59.0 ± 4.4	49.3 ± 14.6	96.7 ± 4.2	80.7 ± 5.5	70.4 ± 1.5	
FlowPolicy	50.7 ± 12.9	58.0 ± 25.6	**85.0 ± 13.2**	17.3 ± 3.8	**87.7 ± 8.5**	45.3 ± 11.5	98.7 ± 2.3	64.3 ± 3.1	75.3 ± 1.8	
MP1	60.7 ± 8.4	74.7 ± 11.5	82.7 ± 1.2	24.0 ± 11.4	62.7 ± 5.0	68.7 ± 1.5	95.7 ± 7.5	77.3 ± 3.1	75.9 ± 1.3	
SOAR（我们的方法）	**64.3 ± 8.1**	**76.7 ± 4.7**	84.7 ± 4.7	**43.3 ± 13.3**	79.7 ± 6.5	**73。0 ± 2。0**	**100。0 ± 0。0**	**88。7 ± 3。1**	**83.9 ± 0.3**	

单次动作专家评估（NFE $= 1$）。每项任务评估 50 个回合。

A.3.2 RoboTwin 2.0

我们进一步在 RoboTwin 2.0 [7] 上评估 $\pi_{0.5}$-SOAR。我们使用全部 50 项任务的干净示范数据联合训练单个多任务策略。仅使用干净数据的训练子集包含 2,500 条轨迹（每项任务 50 条）和 549,787 帧。我们采用任务均衡采样，在每项任务内均匀采样帧，并进行有放回采样。每个观测包含三个 RGB 视角，其中包括一个外部摄像头和两个腕部摄像头，以及一个 14 维本体感知状态。RGB 观测以 224×224 分辨率进行处理。视觉-语言骨干网络使用 OpenPI `pi05_base` 检查点进行初始化。原有的基于流的动作专家被 SOAR 动作模块取代，并使用预训练权重进行初始化。我们使用 FSDP，在四个 NVIDIA A100-80GB GPU 上对所有模型参数进行微调，共训练 100K 个优化步。全局微批次大小为 64，梯度累积步数为 2，因此有效批次大小为 128。我们使用 AdamW，参数为 $\beta_1 = 0.9$、$\beta_2 = 0.95$、$\epsilon = 10^{-8}$，权重衰减为 0.01，并将梯度范数裁剪设为 1.0。学习率在 1K 个预热步数内升至 2.5×10^{-5}，随后按照余弦调度降至第 60K 步时的 2.5×10^{-6}，并保持恒定直至第 100K 步。我们维护衰减率为 0.999 的指数移动平均。状态和动作使用根据仅含干净数据的训练子集计算出的统计量进行归一化。

策略预测长度为 50 步的动作块，每个动作由 14 维绝对关节位置和夹爪动作组成。状态和动作向量会补零至 32 维，但动作损失仅在 14 个有效维度上计算。每次调用策略时，SOAR 通过一次动作专家评估生成完整的动作块（NFE $= 1$）。所有 50 个动作均以开环方式执行，然后再重新规划。我们使用每项任务 100 个回合，在全部 50 项任务上评估最终检查点。

A.4 真实世界实验详情

数据集和任务。Tidy-up数据集包含 512 段以 30Hz 采集的真实机器人演示数据，其中 461 条轨迹用于训练，51 条留作模型选择和离线评估。每条观测由三个 RGB 视角、7自由度机器人关节状态和二元夹爪状态组成。动作由绝对关节目标和二值夹爪指令表示。演示数据包含多阶段分拣序列，机器人会将垃圾放入垃圾桶，并将其他物体放入收纳篮。

整理任务评估使用三种垃圾物品（铝罐、揉皱的纸和牛奶盒）以及六种非垃圾物品（恐龙玩具、蓝色积木、蓝色碗、茄子、粉色杯子和青椒）。每种配置包含两件垃圾物品和两件非垃圾物品。我们选取 40 种不同的组合进行评估，物体位置和摆放方式各不相同。

动态抓取与放置任务要求机器人抓取一个玩具，并将其放入固定的篮子中。每次试验都按照扰动计划进行，在抓取过程中依次进行四次目标位置偏移。所有目标位置均位于可达工作空间内。每种策略均在相同的扰动流程和任务时间限制下进行 20 次试验评估。

训练与策略。 所有三种受评估变体使用相同的观测和动作表示，并预测 16 步动作块。使用相同的训练检查点，对原始 $\pi_{0.5}$ 进行评估，分别采用 10 步和 1 步流积分。对于 $\pi_{0.5}$-SOAR，我们从已收敛的特定任务 $\pi_{0.5}$检查点初始化，冻结 PaliGemma 视觉-语言骨干网络，并训练 SOAR 动作模块。

使用 AdamW 在两张 80GB GPU 上训练 30K 个优化步，有效批次大小为 128。对连续关节动作通道应用基于频率的条件化，同时对二值夹爪通道进行单独监督。对关节轨迹应用时间正则化，并通过事件中心采样提高包含夹爪状态转换的样本比例。

我们使用 RTC，最大前缀延迟为八个动作。生成下一个动作块时，将先前已提交的动作作为时间前缀提供。SOAR 通过单次网络评估（NFE $= 1$）生成相应的动作块。

推理延迟是在相同部署硬件上对一个完整的 16 步动作块进行测量的。动作生成延迟衡量动作模块的计算时间，而端到端

推理延迟包括视觉语言处理和动作生成。我们以每块毫秒数（ms/chunk）报告延迟，而非每个动作的摊销延迟。

整理任务评估指标。 只有当四个物体都被正确分拣到指定容器中时，该回合才被视为成功。完整任务成功率计算如下：

$$\mathrm{SR}_{\mathrm{tidy}} = \frac{1}{40}\sum_{i=1}^{40} y_i \times 100\%, \tag{17}$$

其中，$y_i \in \{0,1\}$ 表示配置i的完整任务结果。所有策略均在相同的 40 种配置下进行评估。该标准衡量的是整个分拣序列是否完成，而非单次抓取或放置尝试。

动态评估指标。 只有当机器人在任务时间限制内抓取目标物体、将其运送至篮子、松开夹爪，且目标物体稳定地留在篮子内时，动态回合才被视为成功。超时、放置失败和安全终止均计为失败。放置成功率为：

$$\mathrm{SR}_{\mathrm{dyn}} = \frac{1}{20}\sum_{i=1}^{20} z_i \times 100\%, \tag{18}$$

其中，$z_i \in \{0,1\}$ 表示试验i中的放置成功情况。

当夹爪完成一次闭合尝试但未能抓住目标物体时，记录为空抓取。持续闭合计为一次尝试；成功抓取后掉落物体则单独记录。

令n_i表示试验i中的空抓取次数。我们报告空抓取总次数、每次试验的空抓取均值以及样本标准差：

$$\begin{aligned} N_{\mathrm{empty}} &= \sum_{i=1}^{20} n_i, \\ \bar{n}_{\mathrm{empty}} &= \frac{1}{20}\sum_{i=1}^{20} n_i, \\ s_{\mathrm{empty}} &= \sqrt{\frac{1}{19}\sum_{i=1}^{20}(n_i - \bar{n}_{\mathrm{empty}})^2}. \end{aligned} \tag{19}$$

所有统计量均包含成功和失败的试验。空抓取次数未按目标位移数量归一化。

详细实验结果。 表 9和 10 报告了图 3中各项结果的具体数据。

在整理任务中，10 步 $\pi_{0.5}$、1 步 $\pi_{0.5}$和 $\pi_{0.5}$-SOAR分别完成了40种配置中的30种、28种和33种。与 10 步基线相比，SOAR在基线失败的6种配置上取得成功，而反过来的情况有3种。与1 步基线相比，相应的数量分别为7种和2种。

在动态抓取与放置任务中，三种策略分别在20次试验中的16次、13次和17次取得成功。它们的空抓取总次数分别为19、24和7次，每次试验的空抓取均值 ± 样本标准差分别为0.95 ± 0.83、1.20 ± 1.01和0.35 ± 0.49。这些按配置和试验统计的结果，详细呈现了任务成功率和未成功抓取尝试方面观察到的差异。

A.5 其他实验细节

A.5.1 时间结构分析

评估协议。 我们使用保存的最终检查点评估记录来评估时间轨迹差异，不进行额外的策略展开。我们根据记录的动作重建机器人运动，并将其与相应场景中的专家参考运动进行比较。对于 Adroit，我们使用手掌 XYZ 位置；对于 Meta-World，我们使用末端执行器 XYZ 位置。所有指标均基于完整轨迹计算。

表 9。 真实世界整理任务中各配置的结果。每种配置包含两个垃圾物品和两个非垃圾物品，以“/”分隔。列 10、1 和 SOAR 分别表示 $\pi_{0.5}$ (NFE=10)、 $\pi_{0.5}$ (NFE=1) 和 $\pi_{0.5}$-SOAR (NFE=1)。1 表示整个任务成功，0 表示失败。总体成功率分别为 75.0%、70.0% 和 82.5%。

识别垃圾/非垃圾物品	10	1	SOAR	识别垃圾/非垃圾物体	10	1	SOAR
01 铝罐、揉皱的纸/恐龙玩具、粉色杯子	0	0	1	21 揉皱的纸、牛奶盒/蓝色碗、粉色杯子	1	1	1
02 铝罐、牛奶盒/恐龙玩具、青椒	1	0	1	22 铝罐、揉皱的纸/蓝色积木、粉色杯子	1	1	1
03 揉皱的纸、牛奶盒/恐龙玩具、蓝色积木	1	1	1	23 铝罐、牛奶纸盒/蓝色积木、茄子	0	0	0
04 铝罐、揉皱的纸/蓝色碗、粉色杯子	0	0	1	24 揉皱的纸，牛奶盒 / 蓝色积木，青椒	1	1	1
05 铝罐，牛奶纸盒/蓝色积木，青椒	1	0	0	25 铝罐，揉皱的纸 /蓝色积木，茄子	1	1	1
06 揉皱的纸，牛奶盒/ 蓝色碗，青椒	1	1	1	26 铝罐，牛奶纸盒/恐龙玩具，蓝色碗	0	0	1
07 铝罐，揉皱的纸/ 蓝色积木，蓝色碗	1	1	1	27 揉皱的纸，牛奶盒/蓝色积木，粉色杯子	1	1	1
08 铝罐，牛奶纸盒/蓝色碗，茄子	0	0	0	28 铝罐，揉皱的纸/恐龙玩具，蓝色碗	0	1	0
09 揉皱的纸，牛奶盒/恐龙玩具，青椒	1	1	1	29 铝罐，牛奶盒/蓝色碗，粉色杯子	1	1	1
10 铝罐，揉皱的纸/茄子，青椒	1	1	1	30 揉皱的纸，牛奶盒/恐龙玩具，粉色杯子	1	1	1
11 铝罐，牛奶盒/茄子，青椒	1	0	1	31 铝罐，揉皱的纸/粉色杯子，青椒	1	1	1
12 揉皱的纸，牛奶盒/粉色杯子，青椒	1	1	1	32 铝罐，牛奶盒/茄子，粉色杯子	1	1	1
13 铝罐，揉皱的纸/恐龙玩具，青椒 1 1			1	33 揉皱的纸、牛奶纸盒/茄子、青椒	0	0	0
14 铝罐、牛奶纸盒/蓝色积木、粉色杯子	1	1	1	34 铝罐、揉皱的纸/蓝色积木、青椒 1 1			1
15 揉皱的纸、牛奶盒/蓝色碗、茄子	0	1	1	35 铝罐、牛奶纸盒/粉色杯子、青椒	1	1	1
16 铝罐、揉皱的纸/蓝色碗、青椒	1	1	1	36 揉皱的纸、牛奶盒/蓝色积木、茄子	1	0	1
17 铝罐、牛奶盒/恐龙玩具、蓝色积木	1	1	1	37 铝罐、揉皱的纸/恐龙玩具、茄子	1	1	0
18 揉皱的纸、牛奶盒/恐龙玩具、蓝色碗	1	0	0	38 铝罐、牛奶纸盒/蓝色积木、蓝色碗	0	1	1
19 铝罐、揉皱的纸/蓝色碗、茄子	0	1	1	39 压皱纸、牛奶盒/茄子、粉色杯子	1	1	1
20 铝罐、牛奶盒/恐龙玩具、粉色杯子	1	1	1	40 压皱纸、牛奶盒/恐龙玩具、茄子	1	0	1

表10. 动态抓取与放置的逐次试验结果。EG 表示空抓取次数，S 表示放置成功（1）或失败（0）。每种策略均在抓取过程中目标位移四次的条件下进行 20 次试验评估。

策略	指标	试验索引																				总计
		01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	
$\pi_{0.5}$ (NFE=10)	EG	2	1	1	1	0	0	1	2	1	1	1	0	0	1	1	1	0	2	3	0	**19**
	S	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	0	0	1	**16**
$\pi_{0.5}$ (NFE=1)	EG	2	0	1	1	2	0	1	3	1	0	1	0	3	2	2	2	1	0	2	0	**24**
	S	0	1	1	1	0	1	1	0	1	1	1	1	0	0	0	0	1	1	1	1	**13**
$\pi_{0.5}$-SOAR（NFE=1）	EG	0	0	1	0	1	1	0	1	0	0	0	0	0	1	1	0	0	0	1	0	**7**
	S	1	1	0	1	1	0	1	1	1	1	1	1	1	1	0	1	1	1	1	1	**17**

EG 均值 ± 标准差：0.95 ± 0.83（$\pi_{0.5}$，NFE=10）、1.20 ± 1.01（$\pi_{0.5}$，NFE=1）和 0.35 ± 0.49（$\pi_{0.5}$-SOAR）。

对于 DP3，我们比较了 53 项任务中的 NFE=1 和 NFE=10 ，其中包括三项 Adroit 任务和 50 项 Meta-World 任务。每种推理设置包含三个训练随机种子，并针对每个任务和随机种子记录一个闭环回合，总计得到 318 条轨迹。

对于 SOAR，我们使用保存的动作记录考察动作校准前后的运动。校准前的运动是根据记录的基础动作重建的，而非通过单独的闭环策略执行获得。

指标。 令 $P = \{p_t\}_{t=1}^{L}$ 和 $P^\star = \{p^\star{}_t\}_{t=1}^{L}$ 分别表示时间对齐的机器人轨迹和专家参考轨迹，其中 p_t、$p^\star{}_t \in \mathbb{R}^3$。我们减去每条轨迹的时间均值： $\tilde{P} = P - \bar{P}$， $\tilde{P}^\star = P^\star - \bar{P}^\star$。

时间形状误差（TSE）用于衡量中心化轨迹之间的位置均方误差：TSE =均方误差（$\tilde{P}$， $\tilde{P}^\star$）。

振幅谱误差（ASE）用于衡量它们的傅里叶幅度谱之间的差异：$\tilde{}$ $\tilde{}$ASE =均方误差（$|\mathcal{F}(P)|$， $|\mathcal{F}(P^\star)|$），其中 $\mathcal{F}$表示傅里叶变换。ASE 忽略谱相位，而 TSE 保留时间对应关系。数值越低，表示与专家参考运动的差异越小。

对于每个指标，任务组统计数据报告为三个训练随机种子上的均值 ± 样本标准差。总体均值根据报告的任务组均值重新计算，并按任务数量加权：

$$\begin{aligned}\overline{E}_{\text{all}} = \frac{1}{53}\big(&3\overline{E}_{\text{Adroit}} + 28\overline{E}_{\text{Easy}} \\ &+ 11\overline{E}_{\text{Medium}} + 6\overline{E}_{\text{Hard}} + 5\overline{E}_{\text{VeryHard}}\big).\end{aligned} \tag{20}$$

总体数值为近似值，因为它们是根据四舍五入后的任务组均值计算得出的。总体

表11. DP3 在 53 项任务中，采用 NFE=1 和 NFE=10时的时间轨迹误差。TSE 和推理端点对齐的单位为$10^{-3}m^2$。任务组结果为三个训练随机种子的均值 ± 样本标准差。总体值报告根据所展示的组均值重新计算的任务加权平均值。数值越低越好。

	TSE		ASE	
任务组	**NFE=1**	**NFE=10**	**NFE=1**	**NFE=10**
Adroit (3)	0.792 ± 0.060	0.757 ± 0.048	0.281 ± 0.045	0.248 ± 0.015
简单 (28)	2.373 ± 0.286	2.273 ± 0.184	0.984 ± 0.058	0.845 ± 0.037
中等 (11)	2.225 ± 0.638	1.918 ± 0.503	1.168 ± 0.305	0.998 ± 0.352
困难 (6)	0.404 ± 0.024	0.389 ± 0.052	0.260 ± 0.016	0.259 ± 0.031
非常困难 (5)	4.236 ± 1.734	3.930 ± 2.012	1.992 ± 1.395	1.645 ± 1.668
总体 (53)	**2.206**	**2.057**	**0.996**	**0.852**

表12。 SOAR 在动作校准前后的时间轨迹误差。TSE 和 ASE 的单位为$10^{-3}m^2$。任务组结果为三个训练随机种子的均值 ± 样本标准差。总体值为根据所示组均值重新计算的任务加权平均值。数值越低越好。

	TSE		ASE	
任务组	**动作校准前**	**动作校准后**	**动作校准前**	**动作校准后**
Adroit（3）	4.118 ± 3.235	1.665 ± 0.881	2.104 ± 1.618	0.525 ± 0.265
简单 (28)	2.579 ± 0.433	1.720 ± 0.136	1.153 ± 0.091	0.760 ± 0.014
中等 (11)	3.915 ± 0.458	1.563 ± 0.229	1.774 ± 0.412	1.116 ± 0.177
困难（6）	1.001 ± 0.058	0.878 ± 0.444	0.621 ± 0.027	0.497 ± 0.239
非常困难（5）	9.714 ± 1.216	7.047 ± 1.545	3.355 ± 0.628	2.298 ± 0.213
总体 (53)	**3.438**	**2.092**	**1.483**	**0.936**

标准差未列出，因为需要按随机种子划分的任务组均值才能重新计算。

DP3：单步与多步生成。 表 11 比较了 DP3 在不同 NFE 下的轨迹误差=1和 NFE=10。将 NFE 从 1 增加到 10，可将总体 TSE 从 2.206 降至 2.057，并将 ASE 从 0.996 降至 0.852，降幅分别为 6.8% 和 14.4%。在所有五个任务组中，NFE=10下的两项误差均较低。

这些测量结果表明，在此次 DP3评估中，推理步数越少，轨迹形状和频谱差异越大。

SOAR：动作校准的效果。表12 比较了动作校准前后的重建轨迹误差。校准使总体 TSE 从 3.438 降至 2.092，降幅为 39.2%。ASE 从 1.483 降至 0.936，降幅为 36.9%。

在全部五个任务组中，校准后 TSE 和 ASE 均有所下降。这些结果表明，在记录的评估条件下，校准后的动作相较于专家参考运动，具有更小的轨迹形状差异和频谱差异。

A.5.2 独立最终检查点评估

评估协议。 主要三维基准测试结果遵循先前工作（包括 MP1）所采用的周期性评估协议。模型每经过 200 个训练轮次评估一次，并对每个训练随机种子的五个最高评估分数取平均值。报告的分数来自用于检查点选择的同一批评估；原始前五名结果未使用额外的留出滚动测试。每次滚动测试都会重置仿真环境，但初始状态并未明确

表13. 对53项 Adroit 和 Meta-World 任务的最终训练检查点进行独立评估。评估回合未用于周期性检查点选择，且 SOAR 与 MP1 的初始观测配置相匹配。成功率（ % ）为三个训练随机种子的均值 ± 样本标准差。

方法	Adroit（3）	简单（28）	中等（11）	困难 (6)	非常困难 (5)
MP1	71.11 ± 3.85	91.90 ± 0.82	63.64 ± 1.82	66.67 ± 6.94	64.00 ± 9.62
SOAR	**77.11 ± 3.67**	**94。05 ± 2。06**	**70.91 ± 4.81**	**77.78 ± 9.62**	**72。00 ± 5。25**

在不同检查点之间进行匹配。

为评估检查点选择的影响，我们还在单独的评估回合中评估了 SOAR 和 MP1 的最终训练检查点，这些回合未用于检查点选择。没有根据周期性评估分数选择任何检查点。两种策略采用匹配的初始观测配置进行评估。我们使用三个独立训练种子，并报告跨种子的平均成功率和样本标准差。评估涵盖全部 53 项任务：3 项 Adroit任务和 50 项 Meta-World任务。

结果。表13 报告了独立最终检查点的结果。在 Adroit 上，SOAR 的成功率为 77。11 ±3.67%，而 MP1 为 71。11 ± 3.85%。在 Meta-World 上，SOAR 在简单（94。05 ± 2.06% vs. 91。90 ± 0.82%）、中等（70。91 ± 4.81% vs. 63。64 ± 1.82%）、困难（77。78 ± 9.62% vs. 66。67 ± 6.94%）和非常困难（72。00 ± 5.25% vs. 64。00 ± 9.62%）任务组中均取得更高的平均成功率。在这项最终检查点评估中，SOAR 在全部五个任务组中的平均值均更高。因此，即使不根据周期性评估分数选择检查点，SOAR 相较于 MP1 的优势依然存在。

A.5.3 仅结构预测与双分支评估

评估协议。 我们评估 SOAR 最终训练检查点生成的三种动作输出：（i）结构头预测的逆DCT重建结果；（ii）单步生成器生成的未校准基础动作；以及（iii）经过谱校准后的最终动作。在每个基准中，所有三种输出均采用相同的诊断协议进行评估。该比较考察结构预测、动作生成和校准各自所发挥的作用。

结构预测与校准。 观测条件结构头预测$\mathbf{C}^s = H^\psi(\mathbf{c})$，并以专家动作的离散余弦变换系数进行监督。在三维实验中，使用完整的离散余弦变换系数集合（$K = T = 4$），从而能够通过逆离散余弦变换（IDCT）重建动作序列$\mathbf{A}_{\text{struct}} = \text{IDCT}(\mathbf{C}^s)$。然而，结构头并非用于独立生成可靠的可执行轨迹。其主要作用是学习专家动作的时间结构，并为后续动作生成提供明确指导。

具体而言，预测的频带会嵌入观测条件中，以引导单步生成器生成基础动作 $\mathbf{A}_{\text{base}}$。相同的结构预测还会进一步用于校准生成的动作。对于完整正交离散余弦变换，公式4中的校准可准确写为

$$\widehat{\mathbf{A}} = (1-\alpha)\mathbf{A}_{\text{base}} + \alpha\mathbf{A}_{\text{struct}}, \qquad \alpha = \tanh(\rho). \tag{21}$$

该公式表明，校准对应于动作空间中的标量混合。不过，结构头和生成器的用途不同：前者提供时间引导，后者生成基础可执行动作。

结果。 表 14 比较了三种输出的直接执行性能。通过对预测系数应用逆离散余弦变换得到的仅结构重建，在 Adroit 和 Meta-World 上的成功率分别仅为 17.67% 和 24.03%。尽管完整的离散余弦变换系数能够重建完整的动作序列，但这些较低的成功率表明，仅凭结构预测并不能构成可靠的可执行策略。相反，其主要作用是捕捉支持动作生成的任务相关时间结构。

相比之下，单步生成器生成的基础动作可靠性显著更高，在 Adroit 和 Meta-World 上的成功率分别达到 59.22% 和 69.07%。这表明生成器在将

结构条件观测转换为可执行动作轨迹方面至关重要。应用谱校准后，成功率进一步提升至 74.56% 和 80.60%，相较于未校准基础动作分别提高了 15.34 和 11.53 个百分点。

这些结果凸显了三个组件各自互补的作用：结构头提供时间引导，单步生成器生成可执行的动作，谱校准则利用预测结构优化生成轨迹。

A.5.4 确定性回归与样本多样性分析

确定性回归基线。 我们将 SOAR 与采用均方误差训练的确定性策略进行比较，两者使用相同的观测编码器和动作 U-Net。该基线以最近两次观测为条件，并根据固定的零动作和时间输入预测四步动作序列。与 SOAR 不同，它不使用随机噪声、结构预测、多频带条件化、动作校准或 MeanFlow 训练。两种策略都只需进行一次网络评估（NFE $= 1$）。

我们使用三个训练随机种子评估 20 项任务（3 项 Adroit、11 项 Meta-World Medium 和 6 项困难任务）。Adroit 和 Meta-World 分别训练 3,000 和 1,000 个训练轮次，每 200 个训练轮次进行一次评估。每次评估在 Adroit 上使用 50 个回合，在 Meta-World 上使用 20 个回合。我们对每项任务和每个随机种子的五个最高评估分数取平均值，然后报告各随机种子间的均值和样本标准差。在 Meta-World 上，五次评估结果均计入。

回归结果。 表 15 报告了对比结果。确定性回归在 20 项任务中的平均成功率为 63。$52 \pm 1.25\%$，而 SOAR 为 71.9%。总体提升了 8.38 个百分点，其中 Meta-World Medium 和困难任务分别提升了 9.91 和 9.28 个百分点。表5中的直接动作生成基线达到 69.3%。这些对比体现了确定性回归与单步生成预测之间的差异。相较于回归方法所带来的提升，反映的是 SOAR 的完整设计，而非任何单个组件。

条件采样协议。 我们进一步比较了 Meta-World `basketball` 上的确定性回归与 SOAR。使用以种子 0 训练的 SOAR，我们选取五个固定观测，并针对每个观测，从不同的高斯噪声初始化中生成 32 个样本。基础动作和校准后的动作使用相同的高斯噪声初始化。我们评估保留用于执行的三个动作步骤，不进行额外的环境滚动执行。对于重复输入，确定性回归会产生相同的预测。结果以五个观测的均值 $\pm$ 标准差表示。

多样性指标。 对于固定观测，设 $\{\mathbf{A}_i\}_{i=1}^{M}$ 表示$M = 32$ 个样本，其中 $\mathbf{A}_i \in \mathbb{R}^{T_e \times D^a}$、$T_e = 3$，动作维度为 D_a。总体多样性和时间形状多样性定义如下：

$$
\begin{aligned}
D_{\text{all}} &= \frac{2}{M(M-1)} \sum_{i<j} \frac{\|\mathbf{A}_i - \mathbf{A}_j\|_F^2}{T_e D_a}, \\
D_{\text{shape}} &= \frac{2}{M(M-1)} \sum_{i<j} \frac{\|\widetilde{\mathbf{A}}_i - \widetilde{\mathbf{A}}_j\|_F^2}{T_e D_a},
\end{aligned}
\tag{22}
$$

~其中 $\mathbf{A}_i = \mathbf{A}_i$ –均值$_i(\mathbf{A}_i)$。此外，我们还测量连续动作差值的成对变异。

相对于记录的专家动作$\mathbf{A}^\star$，预测误差为

$$
\begin{aligned}
E_{\text{mean}} &= \frac{1}{M} \sum_{i=1}^{M} \frac{\|\mathbf{A}_i - \mathbf{A}^\star\|_F^2}{T_e D_a}, \\
E_{\text{best}} &= \min_{i=1,\dots,M} \frac{\|\mathbf{A}_i - \mathbf{A}^\star\|_F^2}{T_e D_a}.
\end{aligned}
\tag{23}
$$

对于确定性回归，所有成对距离均为零，且$E_{\text{best}} = E$mean。

多样性结果。 表 16 比较了动作校准前后 SOAR 与确定性回归的表现。在 `basketball`，校准后总体成对多样性保留了 90.5%，时间形状多样性保留了 97.6%。专家参考均方误差均值从 46.197 降至 0.733，降幅为 98.4%。校准后的 SOAR 参考均方误差也低于确定性回归（0.733 对比 22.880）。

表14。 使用最终训练检查点直接执行三个 SOAR 输出。IDCT 表示逆离散余弦变换。IDCT（结构）在不使用生成器的情况下，直接根据预测结构系数重建动作。基础动作和校准后分别表示谱校准前后的输出。结果为成功率（%）。

动作输出	Adroit	Meta-World
IDCT（结构）	17.67	24.03
基础动作	59.22	69.07
校准后	**74.56**	**80.60**

表15. 在20项操作任务中，确定性回归与 SOAR 的对比。结果以成功率（%）表示。回归报告了三个随机种子下的均值 ± 标准差。

任务组	回归	SOAR
Adroit（3）	78.91 ± 2.70	**79.8**
中等难度 Meta-World (11)	64.39 ± 1.65	**74.3**
高难度 Meta-World (6)	54.22 ± 2.28	**63.5**
总体 (20)	63.52 ± 1.25	**71.9**

表16。 Meta-World `basketball` 上的条件样本多样性。结果为五次观测的均值 ± 标准差，按 10^{-3} 缩放。AC 表示动作校准。

指标	回归	动作校准前	动作校准后
总体成对距离	0 ± 0	4.238 ± 3.724	3.834 ± 2.827
时间形状距离	0 ± 0	1.993 ± 1.394	1.945 ± 1.276
动作差异多样性	0 ± 0	3.061 ± 2.128	2.991 ± 1.925
平均参考均方误差	22.880 ± 47.746	46.197 ± 81.401	$\mathbf{0.733 \pm 0.688}$
32选最佳均方误差	22.880 ± 47.746	45.196 ± 80.067	$\mathbf{0.570 \pm 0.600}$

这些 m 测量结果表明，SOAR 在校准后仍保留非零的条件样本变异，不同于确定性回归。

因此，校准会调整样本之间的差异，而不会取代随机动作生成。测得的结果表明，这种校正显著降低了参考动作误差，同时保留了大部分采样动作变异。

A.5.5 频带划分消融实验

评估设置。 我们评估频带划分对三个 Adroit 任务和 17 个 Meta-World 任务的影响，其中包括 11 个中等任务和 6 个困难任务。对于这些三维操作基准，动作时域为 $T = 4$，结构头预测全部四个离散余弦变换系数（$K = 4$）。我们比较三种频带划分：2 频带：$\mathcal{B}_1 = \{0,1\}$, $\mathcal{B}_2 = \{2,3\}$；3 频带：$\mathcal{B}_1 = \{0,1\}$, $\mathcal{B}_2 = \{2\}$, $\mathcal{B}_3 = \{3\}$；4 频带：$\mathcal{B}_1 = \{0\}$, $\mathcal{B}_2 = \{1\}$, $\mathcal{B}_3 = \{2\}$, $\mathcal{B}_4 = \{3\}$。

所有配置均保留相同的离散余弦变换系数。我们调整频率组及其对应的条件投影和门控，而结构预测和动作校准的公式保持不变。原始 SOAR 使用三频带划分。

我们遵循组件消融方案。每个任务均使用 10 个专家演示和三个随机种子独立训练。对于每个任务和随机种子，我们对定期评估成功率最高的五次取平均。任务组结果报告为跨随机种子的均值 ± 样本标准差。所有变体均保留单步推理（NFE = 1）。

结果。 表 17 比较了三种频带划分。原始的三频带 SOAR 在 Adroit 上的成功率为 79。$8 \pm 2.9\%$，在 Meta-World Medium 上为 74。$3 \pm 0.6\%$，在 Meta-World困难任务上为 63。$5 \pm 1.8\%$。其在 20 项任务上的任务加权平均为 71.9%，而 2 频带和 4 频带分别为 70.9% 和 71.2%。

三频带配置在全部三个任务组中均取得最高的平均成功率。在 Meta-World 中等难度任务上，相较于 2-band 和 4-band，它分别提升了 1.48 和 0.85 个百分点。在困难任务上的差异较小，相较于两种替代方案均提升了 0.22 个百分点。这些结果支持在所评估的短时域任务中采用三频带配置。

表17. 频带划分对 Adroit 和 Meta-World 的影响。所有配置均使用 $T = K = 4$，并保留相同的离散余弦变换系数。成功率（%）以均值 ± 样本标准差的形式报告，统计基于三个训练随机种子。总体是按任务加权计算的20项任务均值，依据报告的组均值计算得出。三频带配置对应原始 SOAR。每列中的最高均值以粗体标示。

	Adroit（3）	Meta-World（17）		
划分	**Adroit**	**中等 (11)**	**困难 (6)**	**总体 (20)**
2-频段	78.82 ± 1.55	72.82 ± 1.99	63.28 ± 2.30	70.9
3频带（SOAR）	$\mathbf{79.8 \pm 2.9}$	$\mathbf{74.3 \pm 0.6}$	$\mathbf{63.5 \pm 1.8}$	**71.9**
4频带	78.80 ± 1.05	73.45 ± 0.64	63.28 ± 2.67	71.2

A.5.6 MP1 增强基线比较 ns

评估设置。 为评估 SOAR 的提升是否主要源自 MP1/MeanFlow 训练组件，我们为 DP、DP3、CP 和 FlowPolicy 增添了本地 MP1 实现中使用的 MeanFlow 目标函数、引导目标和色散损失。

每个增强基线都保留其观测编码器和动作生成骨干网络。我们将原有的扩散或一致性训练目标替换为 MeanFlow 训练。给定专家动作序列 $\mathbf{a}$ 和高斯噪声 $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$，我们构造 $\mathbf{z}_t = (1-t)\mathbf{a} + t\boldsymbol{\epsilon}$。

网络预测平均速度 $u_\theta(\mathbf{z}_t, t, r, \mathbf{c})$，其中 $\mathbf{c}$ 表示观测特征。两个时间变量 t 和 r 分别进行编码，其嵌入在与观测特征结合之前相加。MeanFlow 目标使用雅可比-向量积（JVP）构造。

我们进一步采用本地 MP1 实现中的引导目标：$\mathbf{u}_{\text{ref}} = \text{sg}[u_\theta(\mathbf{z}_t, t, t, \mathbf{c})]$，$\hat{\mathbf{v}} = 2(\boldsymbol{\epsilon} - \mathbf{a}) - \mathbf{u}_{\text{ref}}$，其中 sg 表示停止梯度。我们不使用条件丢弃，也不额外采用无条件推理分支。

色散损失应用于 U-Net 下采样阶段的中间特征，但不包括最后一个阶段。训练目标为 $\mathcal{L} = \mathcal{L}_{\text{MeanFlow}} + 0.5\mathcal{L}_{\text{dispersive}}$。

推理时，通过一次速度预测恢复动作序列：$\hat{\mathbf{a}} = \boldsymbol{\epsilon} - u_\theta(\boldsymbol{\epsilon}, 1, 0, \mathbf{c})$。因此，所有增强基线方法都保留单步推理（NFE $= 1$）。

DP + MP1 保留了适配点云的 DP PointNet编码器、机器人状态编码器和 U-Net。DP3+ MP1 保留了完整的 DP3编码器和 U-Net，而 FlowPolicy +MP1 则保留了官方 FlowPolicy编码器和 U-Net。

对于 CP，我们将其学生网络适配为使用相同的点云输入和双时间条件。移除教师网络和蒸馏目标后，得到的纯学生网络实现等同于我们适配后的 DP3 + MP1 模型。因此，我们报告两者共同的结果，而不将它们视为独立的基线方法。

我们在相同的37项任务基准上评估所有方法，其中包括3项 Adroit 任务和34项 Meta-World 任务（21项简单、4项中等、4项困难和5项非常困难）。结果以三个训练随机种子的均值 ±样本标准差报告。

结果。表18 比较了 SOAR 与 MP1增强基线在相同单步推理设置下的表现。DP + MP1、DP3 + MP1 和 FlowPolicy + MP1 的总体成功率分别为79。$9 \pm 4.2\%$、81。$4 \pm 4.3\%$和81。$1 \pm 4.3\%$。SOAR 达到 84。$4 \pm 0.5\%$，分别比这些方法高出4.5、3.0和3.3个百分点。

在 Meta-World 上，SOAR 的成功率达到 $84.8 \pm 0.7\%$，而三种增强基线方法分别为 $81.6 \pm 4.4\%$、$81.9 \pm 4.6\%$ 和 $80.6 \pm 4.6\%$。SOAR 在简单、中等和极难任务上也取得了最高的平均成功率。不过，这一优势并未在所有任务组中都保持一致：FlowPolicy + MP1在 Adroit 上表现更好，而 DP + MP1 在 Meta-World困难任务上表现更好。

这些结果表明，额外的性能提升源于在动作生成和校准过程中使用预测的时间结构，而非仅仅来自共享的 MP1 训练组件。这些实验证明，SOAR 的结构引导单步动作生成具有有效性，且其效果不止于

表18。 SOAR 与 MP1增强基线在 37 个 Adroit 和 Meta-World 任务上的比较。增强基线保留各自的编码器和动作生成骨干网络，同时采用 MeanFlow/JVP训练、MP1引导目标和色散损失。CP 表示适配的仅学生模型，并与 DP3 + MP1 结果相同。所有方法均使用 NFE $= 1$。成功率（%）报告为三个训练随机种子下的均值 $\pm$ 样本标准差。每列中最高的均值以粗体标出。

		Adroit（3）	Meta-World（34）					
方法	NFE	平均值	简单（21）	中等（4）	困难（4）	非常困难（5）	平均值	总体（37）
DP + MP1	1	60.8 ± 9.8	91.0 ± 3.5	70.0 ± 18.9	$\mathbf{67.5 \pm 18.5}$	63.0 ± 12.3	81.6 ± 4.4	79.9 ± 4.2
DP3 +MP1 / CP	1	75.8 ± 10.2	91.2 ± 3.6	67.5 ± 21.5	62.5 ± 20.9	70.0 ± 10.5	81.9 ± 4.6	81.4 ± 4.3
FlowPolicy + MP1	1	86.7 ± 7.3	87.9 ± 4.7	72.5 ± 19.6	60.0 ± 20.1	73.0 ± 9.4	80.6 ± 4.6	81.1 ± 4.3
SOAR（我们的方法）	1	79.8 ± 2.9	$\mathbf{92.3 \pm 0.5}$	$\mathbf{76.8 \pm 0.7}$	63.5 ± 2.2	$\mathbf{76.9 \pm 1.8}$	$\mathbf{84.8 \pm 0.7}$	$\mathbf{84.4 \pm 0.5}$

表19。 在 LIBERO-LONG 上使用集成 SOAR 的 VLA策略进行组件消融。MC：多频带条件；AC：动作校准；EA：推理端点对齐。结果为成功率（%）。粗体表示最高结果。

指标	Full	不含 MC	不含 AC	不含 EA
H	10	10	10	10
执行。	5	5	5	5
NFE	1	1	1	1
成功率（%）	**96.0**	93.0	94.0	93.0

仅采用 MP1/MeanFlow 训练。

A.5.7 长时域VLA消融实验

评估设置。 我们进一步使用集成了 SOAR 的 VLA策略，在 LIBERO-LONG 上评估 SOAR 各组件的贡献。动作预测时域设置为$H = 10$，每次推理执行五个动作。我们将完整的 SOAR 模型与移除多频带条件化（MC）、动作校准（AC）或推理端点对齐（EA）的变体进行比较。每个变体都会移除相应组件，同时保留其余组件。所有配置均采用单步动作生成（NFE $= 1$）。

结果。 表 19 报告了 LIBERO-LONG 上的组件消融结果。完整的 SOAR 在 $H = 10$ 且执行五个动作的情况下，取得了 96.0% 的成功率。移除 MC、AC 或 EA 后，成功率分别降至 93.0%、94.0% 和 93.0%，对应下降 3.0、2.0 和 3.0 个百分点。

结合表格中关于组件消融的结果5，这些结果进一步阐明了结构条件化、动作校准和推理端点对齐各自的作用。结构头预测观测条件化时间信息，多频带条件（MC）将其融入动作生成，而动作校准（AC）则利用它来校准生成轨迹。

正文中的消融实验表明，通过多频带条件（MC）和动作校准（AC）利用预测的时间结构，能够改善动作生成。LIBERO-LONG上的结果进一步证实，移除MC或AC分别会使成功率降低3.0和2.0个百分点，而移除推理端点对齐（EA）则会导致成功率降低3.0个百分点。

总体而言，这些结果表明，结构条件化、动作校准和端点对齐带来的益处并不局限于短时域三维操作。将这些方法整合到具有更长预测时域的视觉-语言-动作策略后，它们的贡献依然显而易见。

A.5.8 在LIBERO上进行多种子评估

评估设置。 为评估视觉-语言-动作结果的稳健性，我们在LIBERO上使用三个独立训练随机种子，对具有不同NFE的 $\pi_{0.5}$ 和 $\pi_{0.5}$-SOAR进行了额外评估。

每个评 评估涵盖四个标准 LIBERO 套件，每项任务进行 50 个回合，每个套件进行 500 个回合 er

表20。 LIBERO 上的多种子评估。结果为成功率（%），以三个独立训练种子的均值 ± 标准差报告。* 表示本工作中复现的基线方法。NFE 表示每次策略推理的动作专家评估次数。

方法	NFE	空间	物体	Goal	Long	平均值
$\pi_{0.5}$*	10	98.60 ± 0.35	97.87 ± 0.31	98.00 ± 0.20	91.53 ± 0.81	96.50 ± 0.35
$\pi_{0.5}$*	2	98.47 ± 0.31	97.60 ± 0.20	97.40 ± 1.06	90.93 ± 0.42	96.10 ± 0.43
$\pi_{0.5}$*	1	97.90 ± 0.99	96.00 ± 3.39	97.70 ± 0.42	90.30 ± 0.14	95.48 ± 0.95
$\pi_{0.5}$-SOAR	**1**	**99.80 ± 0.20**	**97.43 ± 0.45**	**98。47 ± 0。15**	**93.13 ± 0.64**	**97.21 ± 0.34**

。我们报告不同训练随机种子下成功率的均值和样本标准差。总体成功率通过对每个随机种子的四项套件级结果取平均值计算得出。

结果。 如表 20所示， $\pi_{0.5}$-SOAR 的平均成功率为 97。21 ± 0.34%，NFE 为 = 1。复现的 $\pi_{0.5}$ 基线方法的成功率分别为 96。50 ± 0.35%、96。10 ± 0.43% 和 95。48 ± 0.95%，NFE 分别为 = 10、2 和 1。

SOAR t 因此，SOAR 的平均成功率比 th 高 0.71、1.11 和 1.73 个百分点 e
复现 10 步、两步和一步基线方法，同时只需进行一次动作专家评估 n.

这些多随机种子结果进一步表明，在单步推理设置下，SOAR 在独立训练运行中的任务表现具有竞争力。