

机器人模仿学习：BC、Dagger、ACT、VLA 与 RTC

从监督学习到 VLA 动作生成与实时连续执行

2026-10-20

目录

1. 模仿学习基础：BC 与 Dagger	2
2. 动作块与生成式模仿学习	4
3. 从模仿学习到视觉-语言-动作模型（VLA）	6
4. 实时动作块执行与 RTC	8
参考文献	10

[课程首页](#) · [English](#) · [PDF](#)

项目	说明
主题	BC、Dagger、ACT、VLA 与 RTC
修订日期	2026-10-21
许可	CC BY-NC-SA 4.0 · Xiong Zeng and contributors

1. 模仿学习基础：BC 与 DAgger

定位：本讲义讨论模仿学习算法、动作块建模、VLA 的动作监督训练与实时动作块执行。数据采集硬件、视频重定向、相机标定、完整部署流程和实验步骤不再独立成章。数学表达旨在解释方法及其假设，并非所有实现的逐行复刻。

阅读路线：BC 的监督目标 $\rightarrow$ 分布偏移与 DAgger $\rightarrow$ ACT / Diffusion / Flow Matching $\rightarrow$ VLA 架构与动作监督 $\rightarrow$ Temporal Ensembling $\rightarrow$ 推理时 RTC 与训练时 RTC。

适用对象：希望理解机器人操作策略、VLA 动作头和实时控制接口的研究者与研究生。

1.1 学习对象：从观测历史预测可执行动作

模仿学习 (Imitation Learning, IL) 旨在从专家行为学习闭环策略，而不必预先构造奖励函数或显式求解机器人动力学。令 s_t 为物理隐状态， o_t 为图像等外感受观测， q_t 为关节/末端/手部本体反馈， g 为任务条件。实际策略通常以历史而非单帧为输入：

$$h_t = (o_{\leq t}, q_{\leq t}, g), \quad a_t \sim \pi_\theta(\cdot | h_t), \quad s_{t+1} \sim P(\cdot | s_t, u_t).$$

a_t 是策略希望输出的动作， u_t 是经过限幅、滤波或低层控制器后实际执行的命令；二者不必相等。动作应明确指定语义，例如关节目标 q^{target} 、关节增量 Δq 、末端位姿 T_{EE}^{target} 或末端增量 Δx_{EE} 。不能把训练中的绝对位置在部署时当作速度或相对位移。

图像遮挡、未知接触、网络延迟造成部分可观测性，因此“当前图像相似”不一定意味着“正确动作相同”。历史、任务条件与本体反馈能缓解歧义，但不会自动补全不可见的接触状态。

设 T 为回合长度，策略 π 的状态占据分布为

$$d_t^\pi(s) = \Pr(s_t = s | \pi), \quad d^\pi(s) = \frac{1}{T} \sum_{t=0}^{T-1} d_t^\pi(s).$$

模仿学习的核心差别：BC 在专家访问的分布 d^{π_E} 上拟合动作；DAgger 试图让监督覆盖当前策略自身访问的分布 d^{π_θ} 。

1.2 Behavior Cloning：确定性回归与概率建模

专家数据集为 $\mathcal{D}_E = \{(h_i, a_i^E)\}_{i=1}^N$ 。最简单的确定性 BC 是加权平方误差回归：

$$\theta^* = \arg \min_{\theta} \frac{1}{N} \sum_{i=1}^N \|f_\theta(h_i) - a_i^E\|_W^2, \quad \|x\|_W^2 = x^\top W x, \quad W \succeq 0.$$

当条件动作分布近似单峰、噪声较小时，回归目标通常有效。若假设 $p_\theta(a | h) = \mathcal{N}(a; \mu_\theta(h), \Sigma)$ 且 Σ 固定，最大似然训练相当于最小化

$$\mathcal{L}_{\text{NLL}} = \mathbb{E}_{\mathcal{D}_E} \left[\frac{1}{2} (a^E - \mu_\theta(h))^\top \Sigma^{-1} (a^E - \mu_\theta(h)) \right] + \text{const.}$$

若同时学习 $\Sigma_\theta(h)$ ，高斯负对数似然还需包含 $\frac{1}{2} \log \det \Sigma_\theta(h)$ ，否则模型可以通过无界增大方差降低二次项。 L_1 回归对应另一种误差假设；离散抓取模式可采用交叉熵，连续夹爪位置则不应随意二值化。

对于长度为 H 的动作块，接近 episode 末尾的不足长度部分通常被填充，应使用有效掩码而非把填充值当成专家标签：

$$\mathcal{L}_{\text{masked}} = \frac{\sum_{i,k} m_{i,k} \ell(\hat{a}_{i,k}, a_{i,k}^E)}{\sum_{i,k} m_{i,k} + \varepsilon}, \quad m_{i,k} \in \{0, 1\}.$$

理解要点：归一化必须使用训练集统计量；观测与标签须满足因果时间对应；动作单位、参考坐标系和控制模式要在训练与执行中一致。这三项属于学习问题的必要定义，无须另设“数据工程”章节。

1.3 为什么单峰 BC 会在多模态示教中失败

假设面对障碍物，专家可以选择左绕或右绕；在同一条件 h 下，两个动作都是合理选择。平方误差的 Bayes 最优估计为条件均值：

$$f^*(h) = \mathbb{E}[a^E | h].$$

如果两种示教对应 $a^{(L)}$ 和 $a^{(R)}$ ，预测 $(a^{(L)} + a^{(R)})/2$ 可能恰好撞上障碍物。这里的问题不是网络不够大，而是用单峰点估计表示多峰条件分布不合适。

解决途径依次是：增加任务指令或历史以消除意图歧义；使用包含潜变量的轨迹模型（如 ACT）；或通过 Diffusion/Flow Matching 直接拟合多模态动作块分布。需要注意，生成式 BC 提升的是分布表示能力，不等于自动解决策略在新状态上的纠错问题。

1.4 Distribution Shift：为什么离线误差小仍会失败

BC 的训练样本来自 d^{π_E} ，但训练后的策略实际访问 d^{π_θ} 。一次小误差可能改变接触位置，使之后的图像、手指位姿和摩擦状态都偏离训练数据；后续预测会进一步失真。

经典有限时域分析中的直观结论是：若在专家访问状态上的单步关键错误概率至多为 ϵ ，最坏情形下先前错误导致未来离轨的概率随时间积累，于是累计代价差可以呈现

$$J(\pi_\theta) - J(\pi_E) = O(T^2\epsilon).$$

如果能在策略自身访问的状态上持续保持小的监督错误，在合适的损失与专家代价假设下，典型的上界标度可改进为 $O(T\epsilon)$ 。这只是说明分布偏移的理论框架：不能把机器人连续动作的 MSE 当作上述错误概率，更不能直接由该标度算出任务成功率。

1.5 DAgger：从策略访问状态中聚合专家标签

DAgger (Dataset Aggregation) 是交互式模仿学习的数据更新协议，而不是一种指定网络结构的新动作头。第 k 轮的核心过程是：当前策略访问状态 $\rightarrow$ 专家对这些状态给出正确动作 $\rightarrow$ 聚合数据 $\rightarrow$ 重训/更新策略。

$$\begin{aligned}\pi_k^{\text{roll}} &= \beta_k \pi_E + (1 - \beta_k) \pi_k, \\ \mathcal{D}_{k+1} &= \mathcal{D}_k \cup \{(h, \pi_E(h)) : h \sim d^{\pi_k^{\text{roll}}}\}, \\ \pi_{k+1} &\approx \arg \min_{\pi \in \Pi} \sum_{(h, a^E) \in \mathcal{D}_{k+1}} \ell(\pi(h), a^E).\end{aligned}$$

其中 β_k 表示按状态/时间步选择由专家还是学习器控制的概率，而不是把专家与机器人动作向量按系数直接相加。与“额外采集一批成功示教”相比，DAgger 的关键是：采样的状态由学习策略造成，动作标签仍由专家提供。

从在线学习看，第 k 轮分布上的监督风险可写作

$$L_k(\pi) = \mathbb{E}_{h \sim d^{\pi_k}} [\ell(\pi(h), \pi_E(h))].$$

若学习器对这些变化的损失具有较小遗憾值 R_N ，则

$$\frac{1}{N} \sum_{k=1}^N L_k(\pi_k) \leq \underbrace{\min_{\pi \in \Pi} \frac{1}{N} \sum_{k=1}^N L_k(\pi)}_{\epsilon_N} + \frac{R_N}{N}.$$

在损失有界、专家标注可靠、探索分布适当覆盖且代价到达关系满足前提时，这种自访问分布上的监督误差才能与顺序决策性能相关联。DAgger 并没有普适的“必然比 BC 好”保证，也不能替代安全约束。

1.6 失败纠错：保留有用状态，而不是模仿错误动作

真实机器人中的失败样本至少应区分三种语义：

- **失败前的正确动作**：可保留为正例，即使整条 episode 最终失败。
- **策略造成的错误动作**：可用于诊断或为状态分布采样，但不能直接变成专家 BC 标签。
- **专家接管、纠错和恢复动作**：若观测和控制语义匹配，可为恢复策略提供新的监督。

专家监督有三种常见接口：对模型访问状态离线重标注、实时人类接管，以及在当前状态下单独查询专家动作。离线视频若缺少接触力或遮挡区域信息，专家标签可能不可靠；不可恢复或危险状态应停止、重置或交由安全模块，而非强求“必须示教一个动作”。

$$\mathcal{D}_0 \xrightarrow{\text{train}} \pi_0 \xrightarrow{\text{rollout}} \mathcal{H}_0 \xrightarrow{\text{expert label}} \Delta \mathcal{D}_0 \xrightarrow{\text{aggregate}} \mathcal{D}_1.$$

实际整理时应明确区分 a_t^{executed} 与 a_t^{expert} ，记录策略版本、专家接管标记及标签可信度。只给失败回合一个奖励或成功/失败标志而没有专家动作监督，并不是标准 DAgger；那更接近 RL、偏好学习或离线数据筛选的接口。

1.7 BC、DAgger 与生成式策略并不互斥

方法/协议	监督数据来自哪里	核心作用
单步 BC	固定专家数据	回归或建模当前专家动作
ACT / Chunked BC	固定专家数据	联合预测连续动作块
Diffusion / Flow BC	固定专家数据	表达多模态动作块分布
DAgger + 任意可监督动作头	当前策略访问的状态及其专家标签	修复分布偏移、补足纠错状态

因此可以采用 **DAgger + ACT** 或 **DAgger + Flow Policy**：前者表示动作块分布，后者决定哪些新状态需要专家监督。但若模型输出完整动作块，专家只标一个单步动作并不能直接监督整个 H 步输出；需要调整损失掩码、补全可行标签或改变训练目标。

2. 动作块与生成式模仿学习

2.1 Action Chunking：预测长度不等于开环执行长度

给定当前观测历史 h_t ，策略预测未来 H 个控制时刻的动作块：

$$A_t = (a_{t|t}, a_{t+1|t}, \dots, a_{t+H-1|t}) \in \mathbb{R}^{H \times D_a}, \quad A_t \sim p_\theta(\cdot | h_t).$$

这里 H 是 **prediction horizon（预测长度）**，实际一次连续执行 $K \leq H$ 个动作后可以重新观测、重新生成。 K 较大往往降低推理频率，但也会拉长对抗动的响应间隔； K 较小可以更快反馈，却增加推理和块间衔接压力。动作块是**输出接口**，而 ACT、Diffusion、Flow Matching 是学习其分布的不同模型。

2.2 ACT：Transformer + Conditional VAE + Temporal Ensembling

ACT（Action Chunking with Transformers）最初用于细粒度双臂操作。其两条关键路径是：

训练时的风格编码器根据当前关节状态 q_t 和未来专家动作块 A_t^E 推断潜变量分布：

$$q_\phi(z | q_t, A_t^E) = \mathcal{N}(\mu_\phi(q_t, A_t^E), \text{diag } \sigma_\phi^2(q_t, A_t^E)), \quad z = \mu_\phi + \sigma_\phi \odot \epsilon.$$

动作解码器融合多视角视觉特征、关节状态和风格变量 z ，经 Transformer 编码/解码结构一次性输出 $H \times D_a$ 维动作块：

$$\hat{A}_t = f_\theta(h_t, z), \quad p(z) = \mathcal{N}(0, I).$$

视觉 backbone 将图像转为 token，Transformer encoder 融合视觉、本体感知与潜变量；decoder 使用 H 个动作查询读取融合特征，并行预测各个未来步，而不是必须逐步自回归采样。原始 ACT 的 VAE 编码器主要使用关节状态与示教动作块，并不直接使用视觉图像作为编码器输入。

原始 ACT 的典型损失是带掩码的 L_1 重建与 KL 正则：

$$\mathcal{L}_{\text{ACT}} = \mathbb{E} \left[\frac{\sum_{k=0}^{H-1} m_k \|\hat{a}_{t+k|t} - a_{t+k}^E\|_1}{\sum_k m_k + \varepsilon} + \beta D_{\text{KL}}(q_\phi(z | q_t, A_t^E) \| \mathcal{N}(0, I)) \right],$$

$$D_{\text{KL}} = \frac{1}{2} \sum_j (\mu_j^2 + \sigma_j^2 - \log \sigma_j^2 - 1).$$

β 太小可能导致模型依赖训练时才能得到的 z ，而过大可能使潜变量被忽略（posterior collapse）。**训练与推理不对称**：部署时没有真实未来动作块，不能再运行训练编码器；原始 ACT 通常固定使用先验均值 $z = 0$ 进行确定性预测。也可以设计随机采样变体，但那不是原论文默认方式。

Temporal Ensembling 用于融合在不同时刻产生、却指向同一物理执行时刻 t 的预测。令 $\mathcal{R}(t)$ 为预测到 t 的所有历史查询时刻：

$$\bar{a}_t = \frac{\sum_{r \in \mathcal{R}(t)} w_{r,t} \hat{a}_{t|r}}{\sum_{r \in \mathcal{R}(t)} w_{r,t}}, \quad w_{r,t} = \exp[-m j(r, t)].$$

如果 j 按旧预测到新预测从 0 起编号，这个写法强调较旧的动作计划；改变索引排序就会改变权重的含义。这里平均的是“对同一个 t 的多个预测”，不是简单对相邻时刻动作做移动平均。对于接触建立、释放和突然滑移，过强平均可能掩盖必要的动作模式切换。

2.3 Diffusion Policy：通过迭代去噪建模动作块分布

设真实动作块为 A_0 ，在扩散噪声步 n 中引入高斯噪声：

$$A_n = \sqrt{\bar{\alpha}_n} A_0 + \sqrt{1 - \bar{\alpha}_n} \epsilon, \quad \epsilon \sim \mathcal{N}(0, I).$$

学习条件噪声预测器 $\epsilon_\theta(A_n, n, h)$ ，其简化训练目标为

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{A_0, h, n, \epsilon} \|\epsilon - \epsilon_\theta(A_n, n, h)\|_2^2.$$

生成时从噪声动作块出发，通过若干次去噪逐渐得到满足视觉/任务条件的可执行序列。优势在于可表示多种合理操作轨迹，且具备局部重规划的能力；代价是采样迭代次数以及模型每步评估开销。**多模态能力 ≠ 无延迟实时控制**：采样之后如何对齐物理时钟仍需要执行机制。

2.4 Flow Matching：在动作生成空间学习速度场

将高斯噪声端点 $A_0 \sim p_0$ 与示教动作端点 $A_1 \sim p_{\text{data}}(\cdot | h)$ 相连接。以直线条件路径为例：

$$A_\tau = (1 - \tau)A_0 + \tau A_1, \quad \tau \sim \mathcal{U}[0, 1].$$

该路径的目标条件速度为 $A_1 - A_0$ ，对应条件流匹配目标：

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{h, A_0, A_1, \tau} \|v_\theta(A_\tau, \tau, h) - (A_1 - A_0)\|_2^2.$$

推理时数值积分

$$\frac{dA_\tau}{d\tau} = v_\theta(A_\tau, \tau, h), \quad A_{\tau=0} \sim p_0, \quad A_{\tau=1} \approx A^{\text{pred}}.$$

必须区分两种“速度”： v_θ 是由噪声动作流向示教动作的**生成空间速度场**， τ 是虚拟生成时间；它们不等于机械臂的关节速度 $\dot{q}$ ，也不等于机器人控制时钟 t 。流匹配不是“必然只需一步推理”，实时性依赖模型规模、求解器和积分步数。

2.5 如何选择 BC、ACT、Diffusion 和 Flow

策略族	主要输出及训练	更适合的场景	主要限制
单步 BC	a_t ; MSE / NLL	动作近单峰、控制频率高	长程一致性与多模态不足
ACT	H 步; CVAE + 重建 + KL	连续精细操作、监督示教	后验/先验差异、块间衔接
Diffusion	H 步; 噪声预测	多种合理轨迹	多步去噪延迟
Flow Matching	H 步; 生成速度回归	生成式 VLA 动作头	ODE 求解与采样延迟

选择策略时应首先明确：示教中是否存在多模态；输入是否存在隐状态歧义；动作语义是否固定；系统允许的端到端推理延迟有多大。**策略分布建模和数据分布覆盖是两个独立维度**：即使动作头是生成模型，也依然可能需要 DAgger、恢复示教或交互式纠错。

3. 从模仿学习到视觉-语言-动作模型（VLA）

3.1 VLA 是什么：在视觉和语言条件下学习机器人动作

视觉-语言-动作模型（Vision-Language-Action Model, VLA）通常将**视觉观测、语言指令与机器人本体状态**映射为可执行动作。沿用前文的动作块记号，一种通用接口为

$$A_t \sim p_\theta(\cdot \mid I_{\leq t}, q_{\leq t}, \ell, e), \quad A_t = (a_{t|t}, \dots, a_{t+H-1|t}),$$

其中 $I_{\leq t}$ 为相机观测历史， $q_{\leq t}$ 为关节、末端或手部本体反馈， ℓ 为自然语言任务指令， e 标识机器人本体及其动作空间。并非每一种 VLA 都必须显式输入这四项，或者一次预测 $H > 1$ 步；这个表达式说明**任务条件和可执行动作如何连接**，并不是对所有具体模型的统一实现要求。

核心区别在于问题层次：BC 是监督学习目标，DAgger 是交互式数据聚合协议，ACT / Diffusion / Flow 是动作分布的建模方法，而 VLA 是整合多模态表征与机器人动作预测的策略模型类别。VLA 的动作模块经常仍使用前两章讨论的模仿学习损失。给 BC 增加一条语言输入不自动等于具备跨任务泛化能力的基础模型；泛化还依赖预训练、数据覆盖与模型设计。

3.2 模型架构：VLM 表征与动作生成如何相连

一种常见但并非唯一的设计可写成两步：

$$Z_t = f_\phi(I_{\leq t}, \ell), \quad A_t \sim p_\psi(\cdot \mid Z_t, q_{\leq t}, e).$$

视觉编码器和语言模型提供场景、对象及指令的条件表征 Z_t ；动作模块再利用本体状态产生面向某一种动作接口的预测。两部分可以联合训练，也可以冻结或部分微调视觉语言主干。动作必须经过机器人专用的表示转换、限幅和底层控制才能执行；**VLM 理解指令不等于它已经学会接触动力学或安全的手指控制**。

VLA 的两个常见动作输出范式如下；它们继承的是第 2 章的不同建模思路。

动作接口	如何训练与输出	代表性模型	主要权衡
离散动作 token	将连续动作量化，作为序列预测；通过 token 交叉熵监督	OpenVLA	复用语言模型预测接口，但有量化误差和自回归延迟

动作接口	如何训练与输出	代表性模型	主要权衡
连续生成式动作块	以视觉语言表征为条件，使用 Flow Matching 等损失生成 H 步连续动作	π_0 、GR00T N1	连续高维控制更自然，但需考虑生成计算和动作块执行时延

对于离散动作序列 $z^E = (z_1^E, \dots, z_M^E)$ ，自回归动作建模可使用

$$\mathcal{L}_{\text{token}} = -\mathbb{E}_{\mathcal{D}_E} \sum_{j=1}^M \log p_{\theta}(z_j^E \mid z_{<j}^E, I, q, \ell, e).$$

对于连续动作块， $\mathcal{L}_{\text{FM}}$ 则可直接使用第 2.4 节的目标，只需将其条件 h 扩展为 $(Z_t, q_{\leq t}, e)$ 。这两种损失是不同架构的替代选择，不要求在同一个 VLA 上同时采用。也存在其他动作解码设计，不能把「VLA」直接等同于「Flow Matching」。

3.3 从示教数据到 VLA：预训练、动作监督与下游适配

一条用于动作监督的样本可以组织为

$$\mathcal{D}_{\text{robot}} = \{(I_{\leq t}^{(i)}, q_{\leq t}^{(i)}, \ell^{(i)}, e_i, A_t^{E,(i)}, m_i)\}_{i=1}^N,$$

其中 m_i 是时间步及动作维度的有效掩码。训练时必须保持语言目标与实际轨迹一致，并明确动作单位、坐标系、控制频率、时间戳和目标/增量定义。来自不同机器人的数据不能只在维度上补零就视为语义相同：同一数组位置可能表示不同关节、末端坐标或夹爪命令。跨本体训练需要本体标识、专用编码/解码器或统一且可验证的动作表示。

典型训练路径可分为三个逻辑阶段，但各论文会合并或改变顺序：

1. **视觉语言预训练**：利用图像文本知识获得对象、空间关系和指令表征；这一步不必具有机器人动作标签。
2. **机器人动作监督训练**：使用多任务、多场景或多本体的 (I, q, ℓ, A^E) 数据学习条件动作策略；离散动作头可用 token 交叉熵，连续生成头可用 Flow Matching。就机器人动作监督而言，这仍属于广义的 BC。
3. **下游任务适配**：用目标机器人示教进行监督微调（SFT），或引入新的纠错示教。是否需要后续强化学习取决于任务、模型与数据，不是成为 VLA 的必要条件。

需要区分预训练中的动作监督与在已有模型上进行 SFT 后训练：二者可使用相似损失，但起点、数据规模和需要更新的参数不同。本讲只建立接口；具体的 VLA/WAM SFT、RL 目标留给《具身基础模型后训练》。

3.4 三个代表性模型应该怎样理解

OpenVLA 使用预训练视觉语言表征，把机器人动作离散化后交给自回归 token 预测，说明了把动作视为语言模型可预测序列的路线。 π_0 在预训练 VLM 基础上采用连续动作的 Flow Matching 专家，说明了语义表征 + 高维连续动作生成的路线。GR00T N1 将视觉语言模块和生成式动作模块结合，并引入与本体相关的状态/动作接口，说明了跨机器人条件化与分层动作表示的路线。三者数据、动作频率、机器人平台和评测任务上不同，不能只凭单个成功率判断孰优孰劣。

这里无需重复 GR00T 的网络细节或完整训练配方；模型分层、状态动作编码和底层控制衔接见《GR00T 与 SONIC》。相对于直接输出动作的 VLA，同时显式学习未来视觉/世界状态的 WAM 则在《世界动作模型》中比较。

3.5 与 BC、DAgger、ACT、Flow Matching 和 RTC 的准确关系

层次	对应概念	在 VLA 系统中的作用
学习目标与数据协议	BC、DAgger	用专家动作训练；按需对策略访问的新状态补充纠错标签
动作表征与生成器	Action Chunking、ACT、Diffusion、Flow	决定一次生成什么动作、如何表示多种合理动作

层次	对应概念	在 VLA 系统中的作用
多模态策略架构	VLA	将视觉、语言、本体状态和动作预测放入同一策略
时序执行机制	Temporal Ensembling、RTC	缓解异步推理时动作块接续不平滑和等待问题

因此，「VLA + Flow Matching + RTC」完全可以是一条连贯的系统路线，但它们分别回答「理解哪个任务」「怎样生成动作」和「生成结果如何准时执行」。VLA 仍可能遇到数据分布偏移、接触状态不可见、目标机器人动作接口不匹配及失败后无法恢复的问题；大型视觉语言主干本身不会自动解决这些问题。

4. 实时动作块执行与 RTC

4.1 两条时间轴：机器人控制周期与模型推理延迟

动作块沿物理时间执行，而 Diffusion/Flow 等模型沿计算时间生成新动作。设底层控制周期为 Δt_c ，观测到新动作块返回的端到端延迟为 T_{lat} ：

$$d = \left\lceil \frac{T_{\text{lat}}}{\Delta t_c} \right\rceil.$$

例如 $\Delta t_c = 20 \text{ ms}$ ，若一次请求的端到端延迟为 110 ms ，则 $d = 6$ 个控制步。 T_{lat} 不只包含 GPU 前向传播，还可能包含曝光、预处理、排队、传输和调度；系统设计必须留出尾部延迟余量。

三个量切勿混淆：预测长度 H 、计划每次使用的长度 K 和推理占用的控制周期 d 。 H 再长也不能保证机器人在新块返回前永远有动作可执行。

4.2 从同步执行、朴素异步到 RTC

同步执行若需要停下来等待模型，容易出现速度停顿；朴素异步执行让机器人一边沿旧块运动、一边请求新块，但新旧轨迹可能在切换时跳变。ACT 的 Temporal Ensembling 可以平滑针对同一时刻的预测，却不保证新生成的动作块尊重推理期间已经承诺执行的旧动作。

RTC (Real-Time Chunking) 的主张是：旧块在继续执行时，新块的生成应以已经承诺的动作前缀为条件，再利用最新观测生成剩余自由后缀。这将问题从“两个独立块如何平均”转变为“有条件地生成连续的新块”。

4.3 RTC 的时间几何：承诺前缀与自由后缀

假设旧块从物理时刻 $t - K$ 开始，预测覆盖 $[t - K, t - K + H)$ ；时刻 t 发起新推理，预计时刻 $t + d$ 收到结果。在这 d 个控制步里，机器人必须继续执行旧块。因此新块以 t 为原点的前 d 项已经被旧计划占用：

$$b_i = A_{K+i}^{\text{old}}, \quad 0 \leq i < d, \quad A_{0:d-1}^{\text{new}} \approx b_{0:d-1}.$$

旧块能够覆盖等待窗口的必要条件是

$$0 \leq d \leq H - K.$$

示意：设 $H = 16$ 、 $K = 6$ 、 $d = 3$ 。在 $t, t+1, t+2$ 执行旧块第 6, 7, 8 项；新块直到 $t+3$ 才可用，从新块的第 3 项继续执行，而不是把第 0 项当作 $t+3$ 的控制量。若实际延迟超出旧块覆盖范围，必须安全减速/保持、使用独立后备控制器，或改用更长的预测/更低的更新频率，RTC 不会凭空补全未知命令。

4.4 Inference-Time RTC: 通过生成时 Inpainting 对齐旧计划

Real-Time Execution of Action Chunking Flow Policies 的推理时 RTC 允许不重新训练兼容的 diffusion/flow 策略。将旧块投影到新块的物理时间轴得到参考 Y_i ；对等待期间已经承诺的前缀赋强权重，对其后仍有重叠的未来动作赋逐渐衰减的**软权重**：

$$\mathcal{L}_{\text{match}}(A) = \frac{1}{2} \sum_{i=0}^{H-1} M_i \|A_i - Y_i\|^2, \quad M_i = \begin{cases} 1, & 0 \leq i < d, \\ \rho_i, & d \leq i < H - K, \\ 0, & H - K \leq i < H. \end{cases}$$

其中 $0 \leq \rho_i \leq 1$ 随重叠位置衰减。这个式子是**教学用的一致性目标**，不是原论文全部 guidance 系数的复现。

以从噪声 $\tau = 0$ 流向数据 $\tau = 1$ 的表示为例，流模型对终点的局部估计可写为

$$\hat{A}_1(x_\tau) = x_\tau + (1 - \tau)v_\theta(x_\tau, \tau, h_t).$$

引导的直观形式是沿着降低重叠误差的方向调整生成更新：

$$\tilde{v}_\theta \sim v_\theta - \lambda_\tau \nabla_{x_\tau} \mathcal{L}_{\text{match}}(\hat{A}_1(x_\tau)).$$

原方法使用更具体的伪逆引导与稳定性处理；这里仅表达“**生成速度场 + 基于终点一致性的引导**”这一结构。梯度常需经向量-雅可比积（VJP）求出，给每个去噪/积分步引入额外计算。推理时 RTC 的优势是兼容已有模型；代价是引导计算、更复杂的延迟与掩码配置，以及一致性通常是近似而非任意精度硬保证。对不支持迭代生成/引导的普通 ACT 解码器，不应直接套用该公式。

4.5 Training-Time RTC: 在训练中学习“已知前缀 → 未来后缀”

Training-Time Action Conditioning for Efficient Real-Time Chunking 将延迟模拟移到训练阶段。对一条专家动作块 $A^E = (a_0^E, \dots, a_{H-1}^E)$ ，随机选取模拟延迟前缀长度 d ，使前 d 项保持为已知动作，后续项作为需要生成的目标。

对于逐 token 的生成时间 r_i ，训练输入可以表示成：

$$x_i^\tau = \begin{cases} a_i^E, & i < d, \\ (1 - \tau)\epsilon_i + \tau a_i^E, & i \geq d, \end{cases} \quad r_i = \begin{cases} 1, & i < d, \\ \tau, & i \geq d. \end{cases}$$

只对后缀应用流匹配监督：

$$\mathcal{L}_{\text{TT-RTC}} = \mathbb{E} \left[\frac{1}{H - d} \sum_{i=d}^{H-1} \|v_\theta(x^\tau, r, h)_i - (a_i^E - \epsilon_i)\|^2 \right], \quad d < H.$$

部署时输入当前已经承诺的前缀，使相应 token 的生成时间固定为 1，只积分生成未知后缀；不再在推理循环中为 inpainting 额外计算 VJP。训练时 RTC 的代价是需要相应训练/微调与**逐动作 token 的生成时间或等价的前缀掩码机制**，并且只对训练覆盖的延迟分布有可靠适应性。训练使用专家前缀而实际部署前缀来自旧模型输出时，仍可能出现新的前缀分布偏移。

4.6 Inference-Time 与 Training-Time RTC 的适用边界

维度	Inference-Time RTC	Training-Time RTC
是否必须再训练	通常不需要	需要相应条件训练/微调
一致性来源	生成时前缀/软重叠引导	已知前缀条件 + 后缀生成
额外推理成本	VJP / guidance	无额外的 inpainting 反向传播
对变化延迟的处理	可调整对齐和掩码	受训练覆盖的延迟范围限制
模型侧前提	允许生成步骤中施加引导	支持前缀掩码与逐 token 时间

一个常见误解是“使用了 RTC 就消除了所有控制延迟”。实际上 RTC 解决的是等待期间不暂停与新旧块的条件一致性；输出可能仍基于延迟观测，物理接触、滑移、状态估计误差与策略分布外失败不会自动消失。命令变化小也不等于接触力安全；底层控制器仍须负责限幅、碰撞/力矩约束及推理超时的安全处理。

4.7 关键概念辨析

BC 是监督拟合专家行为；Dagger 是从学习策略访问状态中获取新专家监督的协议；Action Chunking 指一次预测未来 H 个动作；ACT 是 Transformer + CVAE 动作块策略；Diffusion/Flow 是两类生成式动作块模型；VLA 将视觉、语言与机器人状态条件化为动作预测；Temporal Ensembling 融合同一执行时刻的多次预测；RTC 在异步推理中保证新块尊重已承诺的旧动作前缀。这些方法属于不同层次，可组合而不能相互替代。

参考文献

1. Ross, Gordon, & Bagnell. A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning, AISTATS 2011. (Dagger)
2. Zhao et al. Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware, 2023. (ACT / ALOHA)
3. Chi et al. Diffusion Policy: Visuomotor Policy Learning via Action Diffusion, 2023.
4. Lipman et al. Flow Matching for Generative Modeling, 2023.
5. Black, Galliker, & Levine. Real-Time Execution of Action Chunking Flow Policies, 2025. (Inference-Time RTC)
6. Black, Ren, Equi, & Levine. Training-Time Action Conditioning for Efficient Real-Time Chunking, 2025. (Training-Time RTC)
7. Kim et al. OpenVLA: An Open-Source Vision-Language-Action Model, 2024.
8. Black et al. π_0 : A Vision-Language-Action Flow Model for General Robot Control, 2024.
9. NVIDIA et al. GR00T N1: An Open Foundation Model for Generalist Humanoid Robots, 2025.

来源与许可：基于 *Foundation of Embodied AI* 原《数据采集与模仿学习》讲义重构，删去了原第 1、2、6、7 章的独立框架，并将动作语义、纠错示教与 VLA 基础连接到核心章节。原创内容 CC BY 4.0；第三方论文及图文仍适用各自许可。本讲义不包含实验和已运行验证的声明。