从 Motus 出发：一台机器人如何学会看、想与行动

> 一条主线串起多模态、VLA、Diffusion、世界模型、数据与强化学习

桌面上有一只蓝色杯子和一个篮子。人对机器人说：“把蓝杯放进篮子。”

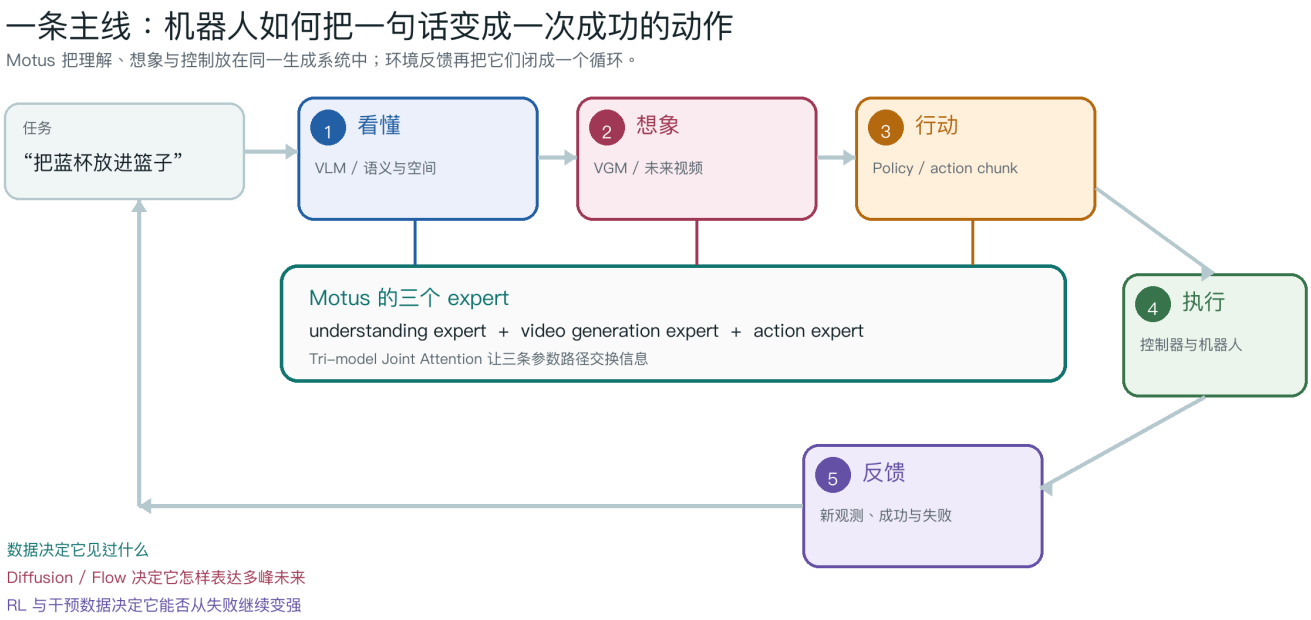


图 0–1 任务从理解开始，经由未来想象和动作生成进入真实世界，再由反馈返回模型。

---

1. 从一个WAM入手：Motus

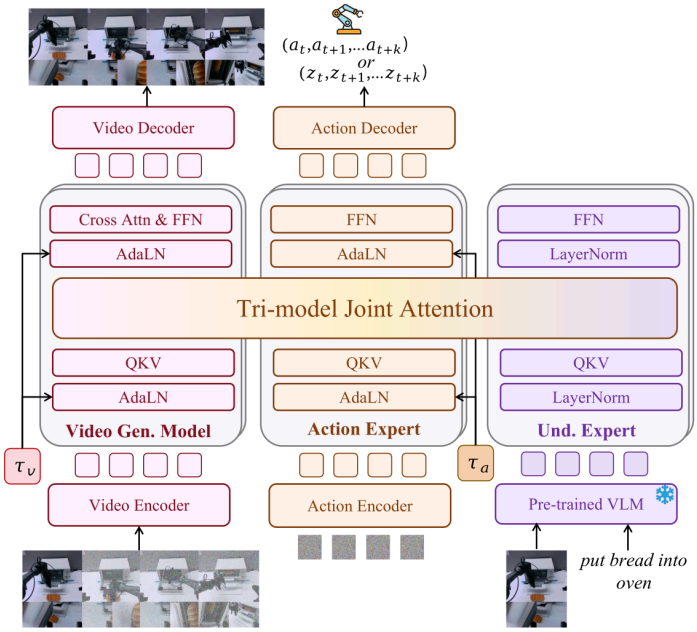
1.1 五个问题

设当前视觉观测为 o_t，语言指令为 ℓ，未来 k 步动作为 $a_{t+1:t+k}$，未来视频为 $o_{t+1:t+k}$。围绕“把蓝杯放进篮子”，系统可以提出五个不同的概率问题：

机器人正在问什么	建模形式	传统名称	在任务中的含义
现在应该怎样动	$p(a_{t+1:t+k} \mid o_t, \ell)$	VLA / policy	从图像和指令直接生成 action chunk
执行这组动作会看到什么	$p(o_{t+1:t+k} \mid o_t, a_{t+1:t+k})$	World Model	预测动作造成的视觉后果
已知物体这样移动，动作可能是什么	$p(a_{t+1:t+k} \mid o_{t:t+k})$	IDM	从状态变化反推动作
指令成功时，未来画面可能是什么	$p(o_{t+1:t+k} \mid o_t, \ell)$	VGM	生成语言条件的视觉计划
能否让未来画面与动作一起生成	$p(o_{t+1:t+k}, a_{t+1:t+k} \mid o_t, \ell)$	Joint video–action	联合约束“看起来合理”和“真的可执行”

早期系统往往为这些问题训练不同模型，再用接口连接。分模块的优点是清楚，代价是视频模型不一定理解机器人动作，动作模型不一定拥有视频生成模型学到的物理运动先验，IDM 还会把视频预测误差继续传给控制。

1.2 三个 expert



Motus Architecture. Here, $a_t \ldots a_{t+k}$ are actions, $z_t \ldots z_{t+k}$ are latent actions, and τ_v and τ_a are the rectified flow generation model and the action expert, respectively.

图 1-1 Motus 架构：video generation、action、understanding 三个 expert 通过 Tri-model Joint Attention 交换信息。来源：Motus 论文。

Motus 采用 Mixture-of-Transformers，而不是把所有 token 粗暴塞进一条完全共享的 Transformer：

Expert	论文采用的基础	输入与输出	它为抓杯任务提供什么
Understanding expert	Qwen3-VL-2B 的末层对应 token，再接若干 Transformer block	图像、语言到语义 token	物体、指令、空间关系和场景语义
Video generation expert	Wan 2.2 5B	noisy future-video latent 到 video velocity	物体如何运动、接触后画面如何变化
Action expert	与 Wan 相同深度的 Transformer action branch	state/noisy action 到 action velocity	生成连续 action chunk

每条路径保留自己的归一化、Q/K/V 投影和 FFN 等参数，以免“理解语言”“生成视频”“控制机械臂”被迫共享完全相同的特征变换；三条路径又在 joint attention 中交换信息。将其抽象为一层计算，可写成：

$$Q_m = H_m W_Q^m, \qquad K_m = H_m W_K^m, \qquad V_m = H_m W_V^m,$$

$$K_{\text{all}} = [K_v; K_a; K_u], \qquad V_{\text{all}} = [V_v; V_a; V_u],$$

$$\tilde{H}_m = \text{softmax}\left(\frac{Q_m K_{\text{all}}^\top}{\sqrt{d}}\right) V_{\text{all}}, \quad m \in \{v, a, u\}.$$

1.3 Motus 已经给出了整门具身智能的目录

现在回看三个 expert

1. understanding expert 怎样从像素获得空间和语义？这通向 ViT、DINO、VLM 与多模态融合；
2. action expert 如何做回归？这通向 BC、action chunk、Diffusion、DiT 与 flow matching；
3. video expert 为什么能帮助控制？这通向视频策略、world model、IDM 与 WAM；
4. 没有 action label 的视频怎样进入 action expert？这通向 latent action 与可辨识性；
5. 三个 expert 的数据从哪里来？这通向数据金字塔、仿真和跨本体对齐；
6. 模仿学习以后如何继续提升？这通向 RL、人工干预和在线数据飞轮。

2. 第一步 看懂蓝杯：understanding expert的多模态智能

2.1 Understanding expert 先把连续世界变成 token

相机输入通常先经过视觉编码器。Vision Transformer 将图像 $I \in \mathbb{R}^{H\times W\times C}$ 切成 $P \times P$ patch，得到 $N = HW/P^2$ 个视觉 token：

$$z_0 = [x_{\text{cls}}; x_p^1 E; \ldots; x_p^N E] + E_{\text{pos}}.$$

自注意力随后建立 token 间关系：

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}} + M\right)V.$$

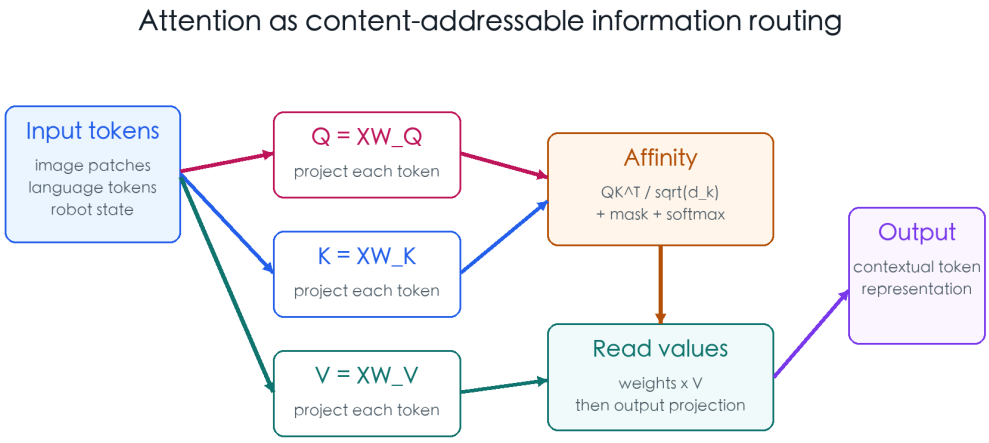


图 2–1 具身策略中的 attention：视觉、语言、状态与动作 token 通过 mask 和位置编码形成可执行的条件关系。教学绘制。

2.2 语义、几何和接触不是同一种视觉能力

“蓝杯”要求语义识别，“杯把朝右”要求局部几何，“夹爪是否接触杯壁”要求细粒度时序和力觉。一个视觉塔很难同时把这些能力做到最好，因此现代系统常组合不同先验：

表征来源	强项	对抓杯任务的价值	容易遗漏什么
DINO / DINOv2	对象区域与结构稳定	跨纹理、光照识别同一对象	接触力与绝对尺度
SigLIP / CLIP 类	图文语义对齐	将“蓝杯”“篮子”绑定到视觉区域	精细位姿与动力学
Video encoder	运动与时序	识别滑动、抓起、遮挡变化	精确机器人 action semantics
Depth / point cloud	显式 3D 几何	抓取位姿、碰撞和可达性	语言语义与材质属性
Proprioception / tactile	本体和接触	判断夹爪闭合、受力和执行状态	场景级语义

Motus 选择 Qwen3–VL–2B 作为 understanding prior，论文强调其空间理解、3D grounding 与定位能力。但即使 VLM 能准确指出杯子，它仍不知道目标机器人的关节限位、控制频率和夹爪闭合范围。这就是从 VLM 到 VLA 必须跨越的一步。

2.3 VLA：让语言模型开始输出机器人动作

VLA 将策略写成：

$$\pi_\theta(a_{t:t+H-1} \mid I_t^{1:V}, q_t, \ell, h_t, e),$$

其中 $I_t^{1:V}$ 是多视角图像，q_t 是本体状态，h_t 是历史或记忆，e 描述机器人本体。视觉 encoder、projector、多模态 backbone、state/embodiment adapter 和 action head 共同把通用语义先验落到一个具体 action space。

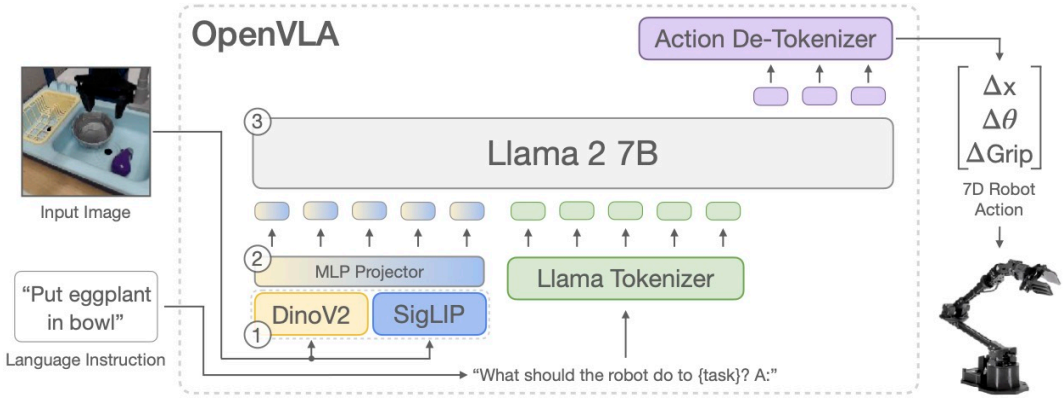


图 2–2 OpenVLA：DINOv2 与 SigLIP 特征经 projector 注入 Llama 2，再把 7D 动作离散为 token。来源：OpenVLA 项目。

模型	视觉语言先验	动作表达	它解决了什么	留给 Motus 的问题
RT–2	大规模 VLM	action token	把网页语义迁移到机器人控制	连续动作与物理未来主要隐式建模
OpenVLA	DINOv2 + SigLIP + Llama 2	256–bin 离散 action token	开源 7B VLA 与跨任务微调	逐维离散会切断维度相关性
RDT–1B	多模态条件	diffusion Transformer	双臂多峰连续动作与跨本体 action space	没有显式生成未来视觉
π_0	预训练 VLM	flow–matching action expert	语义 backbone 与连续动作生成结合	world dynamics 仍主要藏在 policy 内部
Motus	Qwen3–VL + Wan video prior	flow action expert	理解、视频与动作共同建模	统一训练成本与任务干扰更难

3. 第二次步：动作不是一个答案，而是一族可行未来

3.1 BC 是起点，它的失败解释了生成式策略的必要性

Behavior Cloning 在示范数据 $D = (o_t, \ell, a_t)$ 上最大化动作似然：

$$\theta^* = \arg\max_\theta \mathbb{E}_D \log \pi_\theta(a_t \mid o_t, \ell).$$

若策略用高斯均值和 MSE 训练，等价于把多个动作模式压到条件均值。对绕障抓取来说，数据中一半从左绕、一半从右绕，均值可能正好撞上障碍物。BC 还有第二个结构性问题：训练只见专家状态，部署中的微小误差会把机器人带到数据分布之外，后续误差继续累积。

3.2 Diffusion Policy：从噪声里逐步找回一条动作轨迹

前向扩散把真实动作序列 a^0 加噪：

$$q(a^k \mid a^0) = \mathcal{N}\left(\sqrt{\bar{\alpha}_k} a^0, (1-\bar{\alpha}_k) I\right),$$

训练网络在视觉、语言和状态条件 c 下预测噪声：

$$\mathcal{L}_{\text{DDPM}} = \mathbb{E}_{a^0,k,\epsilon} \left\| \epsilon - \epsilon_\theta(a^k, k, c) \right\|_2^2.$$

推理从高斯噪声出发，反复去噪得到整段动作。不同随机初值可以落到不同可行模式，所以它能表达“从杯子左边绕”与“从右边绕”，而不是输出两者平均。

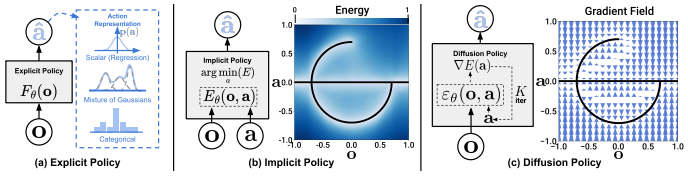


图 3–1 Diffusion Policy 的多峰轨迹与闭环执行。来源：Diffusion Policy。

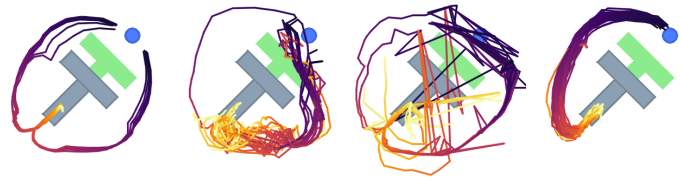


图 3–2 生成模型可保留多个动作模式，单点回归容易落在低概率平均区域。来源：Diffusion Policy 项目材料。

Diffusion Policy 还把生成模型放进 receding–horizon control：模型预测 H_p 步，只执行前 H_e 步，然后读取新观测再规划。这里必须区分三个 horizon：observation horizon H_o、prediction horizon H_p、execution horizon H_e。H_e 太长会降低反馈频率，太短则增加推理压力与动作不连续。

3.3 从 DDPM、DDIM 到 Flow Matching：为什么 Motus 预测 velocity

DDPM 采用随机反向采样，DDIM 可选择更确定的采样路径并减少步数；两者通常围绕 noise、score 或 clean sample 参数化。Flow Matching 直接学习从噪声分布流向数据分布的速度场：

$$a_\tau = (1-\tau)a_0 + \tau\epsilon, \qquad u_\tau = \epsilon - a_0,$$

$$\mathcal{L}_{\mathrm{FM}} = \mathbb{E}\|v_\theta(a_\tau, \tau, c) - u_\tau\|_2^2, \qquad \frac{da}{d\tau} = v_\theta(a, \tau, c).$$

推理时由 ODE solver 积分速度场。对机器人 action chunk 来说，Flow Matching 的吸引力在于连续轨迹建模、稳定训练和较少采样步的潜力；π_0 与 Motus 都沿这条路线构造 action expert。

DiT 则把去噪器从 U-Net 换成 Transformer。动作序列天然是 token 序列，Transformer 能在整个 chunk 内建模关节耦合、双臂同步和长程依赖。timestep、语言与视觉条件通常通过 AdaLN、cross-attention 或 joint attention 注入。Motus 的 action block包含 rectified-flow timestep AdaLN、FFN 和 Tri-model Joint Attention，因此“动作怎样去噪”与“动作读取哪些视频/语言信息”在同一层发生。

4. 第三次步：在 WAM 之前，人们怎样把“未来”接到策略上

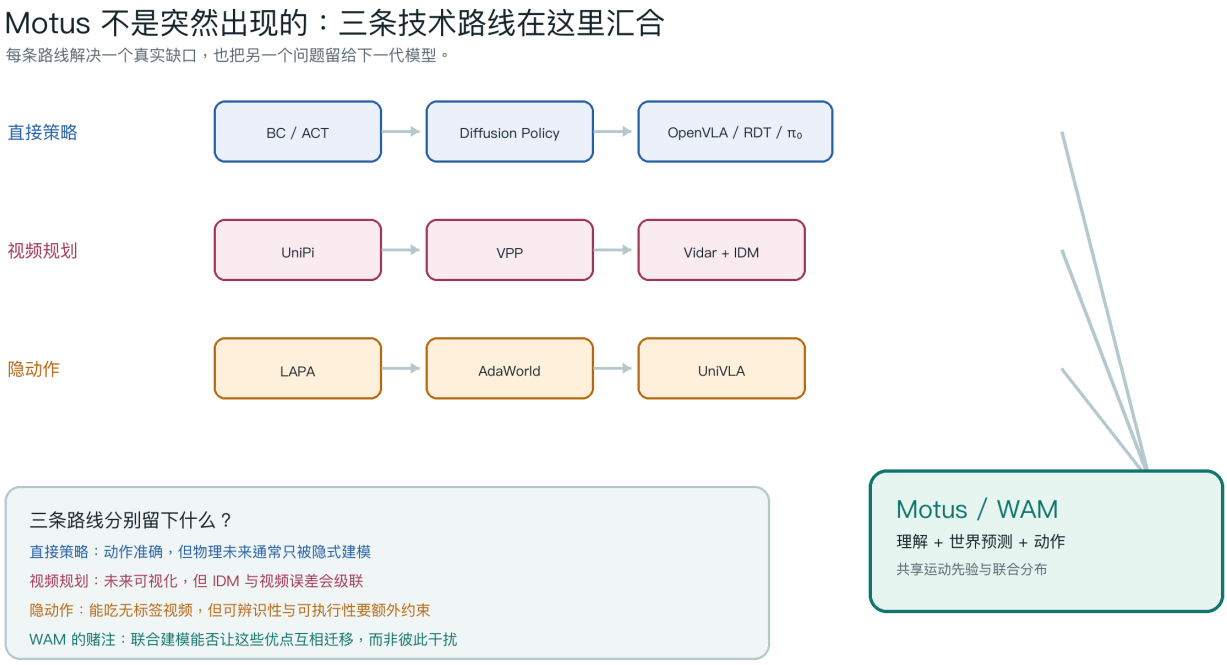


图 4-1 三条技术路线汇入 World Action Model：直接策略、视频规划加 IDM、latent action。教学绘制。

4.1 路线一：直接策略，把未来藏在参数里

BC、ACT、Diffusion Policy 和 VLA 都直接学习 $o, \ell \mapsto a$。只要示范数据覆盖充分，这条路线通常最短、推理也最直接。OpenVLA 继承图文语义，RDT 用 diffusion 表达双臂动作，π_0 用 flow matching 生成连续 chunk。

4.2 路线二：先生成视频计划，再用 IDM 取出动作

UniPi 把顺序决策写成文本条件视频生成：先画出任务成功的未来帧，再从帧间变化提取控制动作。这个思路把跨环境、跨 action space 的计划统一到像素空间。

随后，VPP 不一定把完整视频作为最终输出，而是利用视频扩散模型内部的 predictive representation，通过隐式 IDM 学动作；Vidar 使用互联网预训练视频 diffusion 作为 generalizable prior，再以 masked IDM 适配目标本体，并用 action-relevant mask 抑制背景干扰。

这条路线的因果链是：

$$(o_t, \ell) \xrightarrow{\text{VGM}} \hat{o}_{t+1:t+H} \xrightarrow{\text{IDM}} \hat{a}_{t:t+H-1}.$$

它的优势是未来可视化、能吸收无动作标签视频；它的风险也来自这条链：视频 hallucination 会传给 IDM，视觉上相似的变化可能对应不同控制，长视频生成还会占用闭环延迟预算。

4.3 路线三：不要求视频直接变成动作，先学习 latent action

LAPA 先用 VQ-VAE 风格 quantizer 从相邻帧学习离散 latent action，再训练 latent VLA 根据当前观测和语言预测 latent token，最后用少量带真实 action 的机器人数据完成 action finetuning。

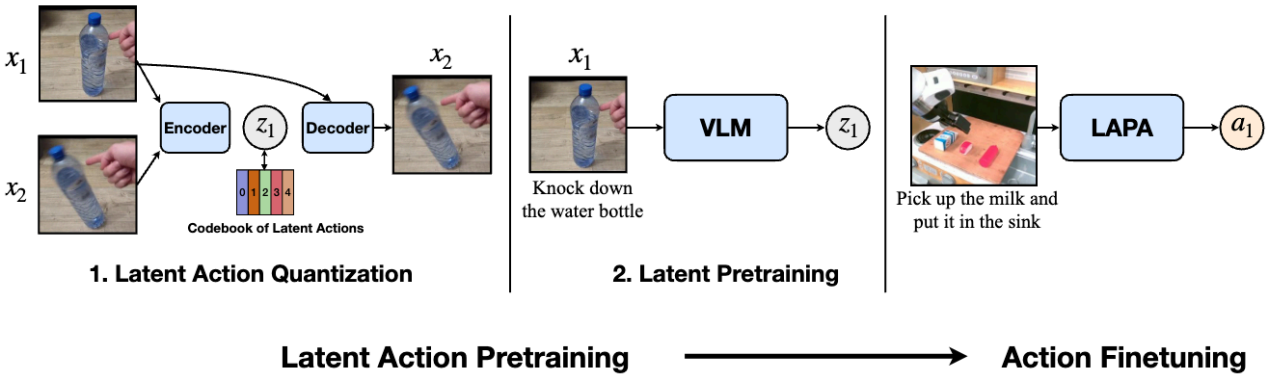


图 4-2 LAPA：action quantization、latent-action pretraining、真实 action finetuning。来源：LAPA 论文。

AdaWorld 关注可适配 world model 和 latent factor 解耦；UniVLA 在 DINO feature space 中学习 language-conditioned、task-centric latent action，试图抑制与任务无关的背景和视角变化。

4.4 WAM：把视频和动作从流水线关系变成联合分布

World Action Model 进一步学习：

$$p_\theta(o_{t+1:t+H}, a_{t:t+H-1} \mid o_t, \ell).$$

视频提供稠密的运动与结果监督，动作提供可执行性和因果 grounding。Motus 用三 expert、双时间 scheduler 和 optical-flow latent action 统一五种条件分布；LingBot-VA 强调因果视觉动作建模与闭环执行；DreamZero 从大型预训练视频 diffusion 出发联合生成 video/action，并通过系统优化推进实时闭环。

路线	主要监督	最强归纳偏置	典型失败
Direct VLA	robot action	直接优化控制	对新物理运动和无标签视频利用有限
VGM + IDM	future video + 少量 action	先计划视觉未来	视频误差与 IDM 误差级联
Latent Action	frame transition + 少量 action	共享运动表示	latent 不可辨识或不可执行
Joint WAM	video + action + language	视觉未来和动作互相约束	多目标干扰、计算量和闭环延迟

到这里，Motus 的 video expert 与 action expert 为什么相连已经清楚了。但连接两个 expert 还不够，互联网视频中没有关节角或末端位姿。故事的下一步，是构造一座从像素运动通向控制空间的桥。

5. 第四次步：没有动作标签的视频，怎样教会 action expert

Motus 的回答：把 optical flow 当作 pixel-level delta action

相邻两帧之间的 optical flow 描述像素位移。它不等于机器人动作，但比原始 RGB 更集中地表达“哪里发生了运动”。Motus 的 latent-action 路径是：

```
(frame_t, frame_t+1)
  -> DPFlow optical flow
  -> RGB-like flow representation
  -> DC-AE: four 512-D tokens
  -> lightweight encoder
  -> 14-D latent action z_t
```

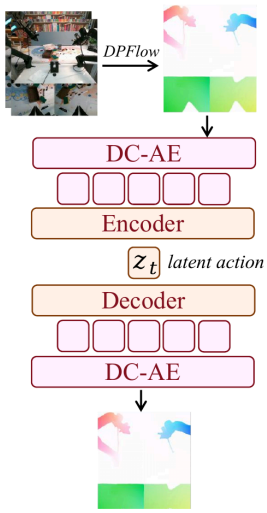


图 5–1 Motus Latent Action VAE：DPFlow 提取光流，DC–AE 压缩后映射到 14 维 latent action。来源：Motus 论文。

14 维大致贴近常见双臂 action 的量级，但“维度相似”不代表每一维天然对应某个关节。论文用 90% 无标签视频做 flow reconstruction，再混入 10% 带 action 的轨迹进行弱监督，其中包括标准机器人示范和 AnyPos 风格的 task–agnostic 交互数据。目标为：

$$\mathcal{L} = \mathcal{L}_{\text{recon}} + \lambda_a \|a_{\text{real}} - a_{\text{pred}}\|_2^2 + \beta \mathcal{L}_{\text{KL}}.$$

重建项让 latent 保留运动，action alignment 把它拉向可执行控制分布，KL 项约束潜空间。这样，action expert 在真正看到目标机器人数据以前，也能从人类视频、互联网视频和多机器人轨迹中预训练“运动如何发生”。

6. 第五次步：SCALING LAW！数据在哪

6.1 Motus 的数据金字塔：越往上越少，也越接近最终控制

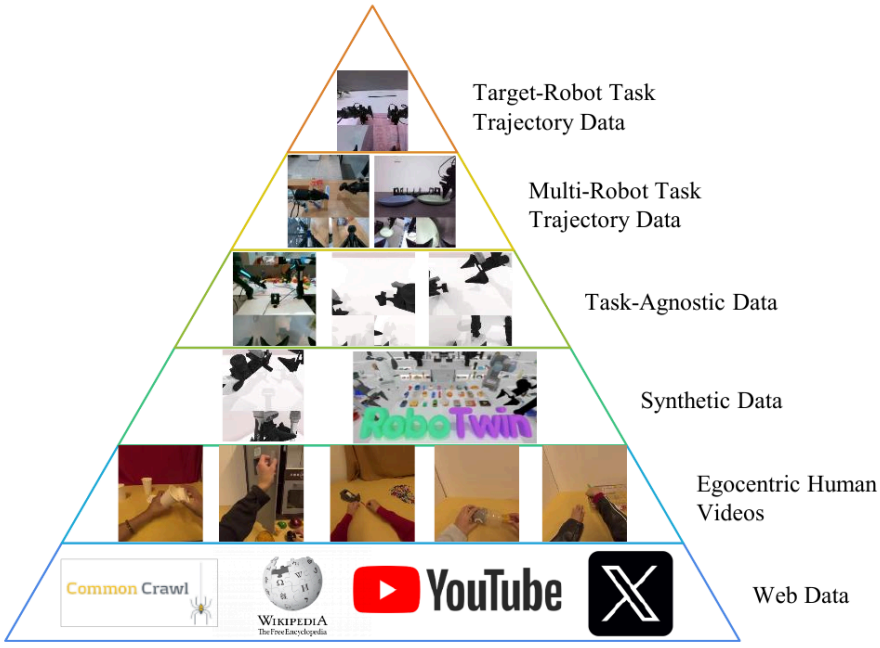


图 6–1 Motus 数据金字塔：从 web data 到 target–robot task trajectory，数量递减、控制相关性递增。来源：Motus 论文。

层级	典型内容	主要提供的先验	无法单独提供什么
L1 Web data	文本、图文、互联网视频	语义、常识、对象与广泛视觉模式	目标机器人控制
L2 Egocentric human video	第一视角人类操作	手物交互、任务过程、运动模式	机器人 action label

层级	典型内容	主要提供的先验	无法单独提供什么
L3 Synthetic data	仿真视频与轨迹	可控随机化、长尾场景、状态真值	完整真实接触与传感噪声
L4 Task-agnostic data	随机可行动作、play data	action space 覆盖、可达性、真实动力学	明确任务成功语义
L5 Multi-robot task trajectory	多本体示范	跨机器人技能与动作先验	目标本体精确标定
L6 Target-robot trajectory	目标机器人任务数据	最终 kinematics、频率、相机与动作接口	大规模开放世界覆盖

6.2 三阶段训练：每一阶段只解决下一座桥

阶段	使用数据	更新对象	这一阶段真正学什么
Off-the-shelf priors	L1 Web data	已有 VLM、VGM	通用语义与视频生成先验
Stage 1: Visual Dynamics	L2、L3、L5	只适配 VGM	让通用视频模型进入操作域，学 plausible future
Stage 2: Unified Latent Action	L2–L5	三个 expert，VLM 冻结	用 latent action 把视频运动灌入 action expert
Stage 3: Target SFT	L6	三个 expert + true action	绑定目标机器人的 kinematics 与 action semantics

6.3 跨本体不是 padding，而是物理语义对齐

不同数据集的 action 可能是关节位置、关节速度、末端绝对位姿、末端增量或力矩。即使张量都是 14 维，也可能具有完全不同的单位、坐标系和控制频率。一个可训练的数据 schema 至少需要：

```
episode_id, task_id, embodiment_id
timestamp, camera_intrinsics, camera_extrinsics
rgb[view], depth[view], proprioception
action_raw, action_type, action_frame, action_units
control_hz, gripper_semantics
terminated, truncated, success, intervention
calibration_version, preprocessing_version
```

跨本体 action adapter 需要显式处理 mask、归一化、旋转表示、左右臂顺序和 embodiment token。RDT 的 Physically Interpretable Unified Action Space、Motus 的 optical-flow latent action，以及 embodiment-specific decoder，代表三种不同选择：统一真实物理维度、转向共享视觉运动空间、或保留本体专用映射。

7. 第六步：强化学习

7.1 为什么 SFT 后仍需要 RL

BC 优化的是训练分布 $d_{\text{data}}(s)$ 下的动作误差，部署时状态来自策略自己的分布 $d_\pi(s)$：

$$\mathbb{E}_{s\sim d_{\text{data}}}\ell(\pi(s),a^*) \quad \not\Rightarrow \quad \mathbb{E}_{s\sim d_\pi}\ell(\pi(s),a^*)\ \text{small}.$$

抓杯时一次两毫米偏差，可能让下一帧出现训练集中从未见过的半抓取状态。DAgger 通过在线访问专家、收集纠正动作来缩小这个分布差；RL 则允许系统根据 reward、成功信号、偏好或人工干预，优化整段行为的长期结果。

7.2 Q-learning：先理解“动作之后的未来价值”

策略 π 的 return、state value 和 action value 定义为：

$$G_t=\sum_{k=0}^{\infty}\gamma^k r_{t+k}, \qquad V^\pi(s)=\mathbb{E}_\pi[G_t\mid s_t=s],$$

$$Q^\pi(s,a)=\mathbb{E}_\pi[G_t\mid s_t=s,a_t=a].$$

Q-learning 是一种 off-policy temporal-difference control 方法：它直接学习最优动作价值 $Q^*(s,a)$，并用 Bellman optimality backup 逼近“执行动作 a 后，未来继续采取最优动作时的总回报”。单步 target 为：

$$y_t=r_t+\gamma(1-d_t^{\text{term}})\max_{a'}\bar{Q}(s_{t+1},a').$$

公式中每一项都有明确含义：

- r_t 是当前动作已经得到的真实反馈；
- $\bar{Q}(s_{t+1}, a')$ 是模型对下一状态未来回报的估计；
- $\max_{a'}$ 表示下一步按当前价值函数选择最有价值的动作；
- γ 决定未来回报折扣；
- d_t^{term} 只在环境真正终止时将未来价值置零。

这里的 **bootstrap** 指 target 没有等到整条 episode 的真实 return，而是用当前价值估计 $\bar{Q}(s_{t+1}, a')$ 代替尚未发生的未来。它提高样本效率，却也带来偏差、过估计和 moving-target instability。target network、Double Q、replay buffer 等技术都在缓解这些问题。

时间上限 `truncation` 与环境终止 `termination` 不能混为一谈。杯子掉落、机器人进入不可恢复状态属于 termination，未来价值应为零；rollout 因采集窗口到时被截断通常只是 truncation，环境本可继续，target 一般仍应 bootstrap。只有当时间限制本身就是任务状态和失败定义的一部分时，才应把它当真正终止。

7.3 连续机器人动作为什么转向 Actor-Critic、PPO 与 SAC

机械臂动作连续且高维，直接计算 $\max_{a'} Q(s', a')$ 很困难。Actor-Critic 用 actor π_θ 生成动作，用 critic V_ϕ 或 Q_ϕ 评估动作。advantage 表示“这个动作比当前状态的平均选择好多少”：

$$A^\pi(s, a) = Q^\pi(s, a) - V^\pi(s).$$

PPO 用 clipped ratio 限制一次策略更新幅度：

$$L_{\text{PPO}} = \mathbb{E}_t \left[\min \left(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t \right) \right].$$

SAC 是 off-policy maximum-entropy 方法，其 critic target 可写为：

$$y_t = r_t + \gamma(1 - d_t^{\text{term}}) \left[\min_i \bar{Q}_i(s_{t+1}, a') - \alpha \log \pi(a' \mid s_{t+1}) \right].$$

PPO 适合大规模并行仿真和 on-policy 更新；SAC 能复用 replay，适合连续控制但要处理 Q overestimation、离线数据分布外动作和真实机器人采样成本。对 VLA 或 diffusion policy，actor 不一定是简单高斯头，它可以是 action-token 模型、flow action expert 或整段 chunk 生成器；critic 则需要明确评价单步 action 还是整个 chunk，并处理 reward 延迟归因。

7.4 真实机器人更常见的路线：干预、偏好与离线数据飞轮

真机从随机策略开始探索往往不安全也不经济。更现实的顺序是：

$$\textbf{pretraining} \rightarrow \textbf{BC/SFT} \rightarrow \textbf{deployment failures} \rightarrow \textbf{intervention/preference/reward} \rightarrow \textbf{policy improvement}.$$

HG-DAgger 与 Fleet-DAgger 用人工接管或多机器人 fleet 收集分布外纠正；RLIF 把 intervention signal 本身作为 reward，以 off-policy RL 学习，允许策略超过次优干预者；SOP 把 fleet、云端 learner 和在线后训练连成系统；From Prior to Pro / DICE-RL 则代表用 RL 将宽覆盖生成式 prior 收缩成高成功率 expert policy 的方向。

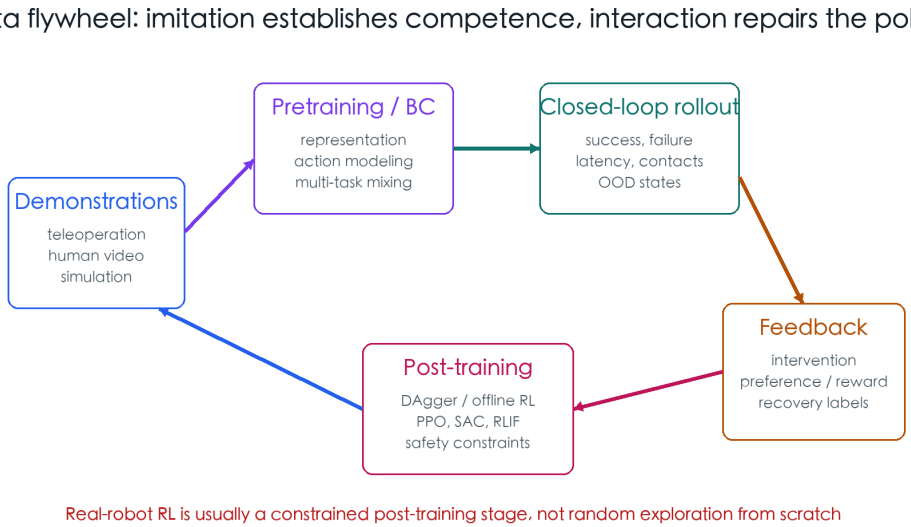


图 7-1 数据飞轮：部署、失败分类、人工接管、回放与策略更新共同构成 post-training。教学绘制。

7.5 Motus 与 RL 的关系必须说准确

Motus 论文的主要训练路线是 foundation-model adaptation、latent-action unified training 和 target-robot SFT，**它本身不等于一个 RL 算法**。从系统角度可以推导出若干结合方式，但应明确标记为后续设计，而不是论文已经证明的结论：

1. action expert 可作为初始化 actor，避免真机从零探索；
2. understanding/video features 可为 critic 或 reward model 提供状态表示；
3. video expert 可生成 imagined rollout，降低真实交互量；
4. intervention、success detector 或 VLM preference 可产生 post-training 信号。

最大的风险是 model bias exploitation：critic 可能偏爱 video expert 容易生成、却在真实物理中不可行的轨迹。想象数据必须与真实 rollout 校准，并持续监控接触、约束违规、视频动作一致性和 OOD 状态。

RL 让系统能够利用失败，但策略最终仍要在毫秒级控制循环中运行。一个联合模型的能力越多，部署时越需要精确决定哪些分支必须计算、多久重规划一次、哪一层负责安全。

附录 A：沿故事主线阅读论文

A.1 主角与统一架构

1. Motus: A Unified Latent Action World Model
2. UniDiffuser: One Transformer Fits All Distributions
3. Understanding World or Predicting Future? A Comprehensive Survey of World Models
4. General World Models Survey

A.2 视觉、多模态与 VLA

1. An Image is Worth 16x16 Words / ViT
2. DINO
3. RT-2
4. OpenVLA
5. π_0
6. RDT-1B
7. ACT
8. H-RDT

A.3 生成式动作

1. Denoising Diffusion Probabilistic Models
2. DDIM
3. Classifier-Free Diffusion Guidance
4. Latent Diffusion Models
5. Diffusion Transformer
6. Flow Matching for Generative Modeling
7. Diffusion Policy

A.4 视频策略、IDM 与 latent action

1. UniPi
2. VPP
3. Vidar
4. Vidarc
5. LAPA
6. AdaWorld
7. UniVLA
8. AnyPos

A.5 World Action Model 与后续路线

1. Motus
2. LingBot-VA / Causal World Modeling for Robot Control
3. DreamZero

4. RISE
5. Act2Goal

A.6 IL、RL 与机器人后训练

1. DAgger
2. HG-DAgger
3. Fleet-DAgger
4. RLIF
5. Precise and Dexterous Robotic Manipulation via Human-in-the-Loop RL
6. SOP
7. From Prior to Pro / DICE-RL

A.7 数据与模拟器

1. Open X-Embodiment
2. RoboTwin
3. RoboTwin 2.0
4. AgiBot World Colosseo