从能力激发到能力进化：
大模型驱动具身智能的路径探索

张静
中国人民大学

2026年11月

提纲

Contents

01 技术趋势与研究路线

02 能力激发：RoboHarn-Evo

03 能力进化：Replan 与 ProcVLM

04 精准执行：VLA 工具的训练与推理

05 总结与展望

技术演进：泛化能力扩展，精准度与效率待突破

VLA 与 WAM 的泛化目标，在 Astra 等通用模型中展现出更大潜力

VLA
从视觉与语言输入
直接学习连续动作

→

WAM
联合建模环境变化与动作
学习操作的物理结果

→

Astra 等通用模型
通用推理与工具调用
扩大跨任务、跨场景泛化

当前局限：操作精准度不足，在线推理与调用效率仍低

效果：先激发，再内化

固定模型参数，用外部 Harness 积累经验、
编排工具，激发模型已有能力。

进一步让模型推理并调用工具，
与环境多轮交互，通过 RL 将能力内化。

效率：少量思考，长时间执行

构造感知、抓姿、轨迹预测等专有小模型，
提高执行精准度，降低高频调用开销。

大模型少量思考与调用。
小模型尽可能长时间执行，必要时交还控制。

提纲

Contents

01 技术趋势与研究路线

02 能力激发：RoboHarn-Evo

03 能力进化：Replan 与 ProcVLM

04 精准执行：VLA 工具的训练与推理

05 总结与展望

第一步：固定模型与执行器，让物理经验在框架中积累

RoboHarn-Evo：分层Skill

任务Skill约束目标关系与完成条件，动作Skill约束对象相对几何和物理效果

例：任务先按空间顺序盖住积木，再按红、绿、蓝顺序揭盖。



当前子任务：揭开绿块上方的盖子

层次	适用条件与策略	完成 / 效果证据
任务K^T	红已揭开、绿仍被盖 接下来揭开绿色积木	盖子移到一旁 绿块持续可见
动作K^A	**目标盖子把手可达** **夹住把手，先垂直抬升**	**盖子随夹爪抬升** **并与积木脱离**

$k = (\text{条件}, \text{策略}, \text{结果}, \text{证据}\ b, \text{状态}\ \sigma)$



闭合：未证实　　抬升：抓取成立

仍需移到一旁，才结束揭盖子任务。

整体框架：内环执行与外环知识演化

内环用知识完成当前任务，外环把源交互中的证据写成后续可复用知识

内环：当前回合内，检索任务知识与动作知识

观察与运行记忆 RGB-D / 进度 → 检索 K_i^T 选择子任务 u_t → 检索 K_i^A 确定几何 z_t → 生成工具调用 c_t 固定执行器 → 观察物理效果 更新进度 / 再规划

$$u_t \sim p_\theta(\cdot \mid s_t, l, K_i^T) \qquad z_t \sim p_\theta(\cdot \mid s_t, u_t, K_i^A) \qquad c_t \sim p_\theta(\cdot \mid s_t, u_t, z_t)$$

动作后重新观察；验证子任务完成后才推进。运行记忆只记录本回合状态。

外环：源交互（dev）回合结束后，评价证据并修订知识

源回合轨迹 τ_i 观察 / 调用 / 效果 → 分层反思与评价 提取任务、动作条目 → 维护知识与证据 合并 / 修订 / 暂停反证条目 → 写回 K_{i+1} 与技能索引 供后续回合检索

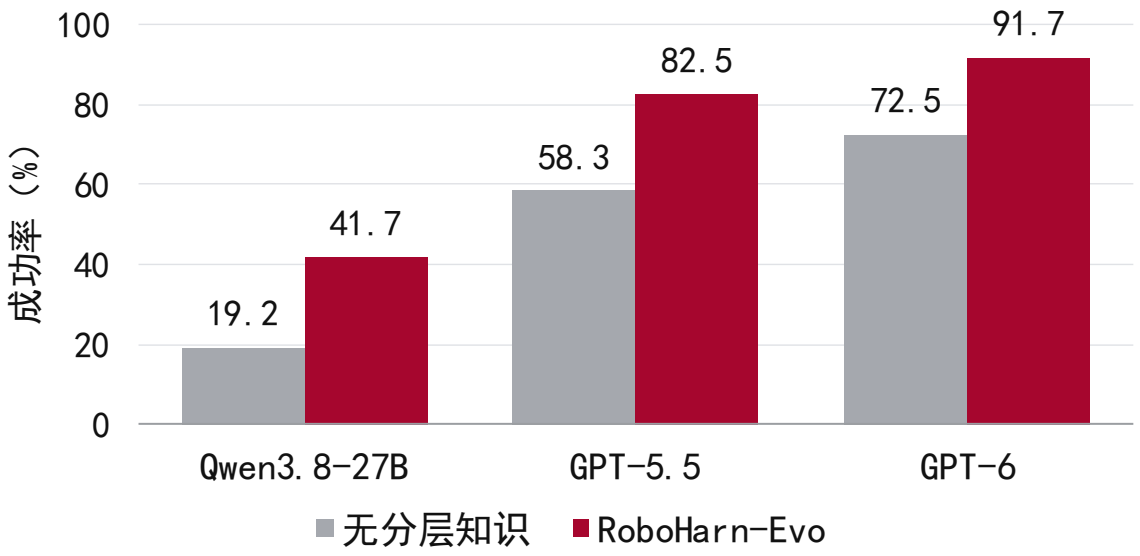
$$K_{i+1} = U(K_i, View(\tau_i)), \qquad K_i = (K_i^T, K_i^A), \qquad \theta \text{ 与执行工具始终固定}$$

技能是知识检索索引。留出评测时冻结知识库，测试反馈不进入外环更新。

实验：固定模型与工具， Skill积累能否改善执行？

过去交互形成的知识，能够提升同类任务中未见配置的操作成功率

六项仿真任务的平均成功率



相同源经验下，知识结构的贡献

方法	Qwen 成功率	提升 pp	GPT-5.5 成功率	提升 pp
无 HPK	19.2%	0.0	58.3%	0.0
平铺反思	24.2%	+5.0	64.2%	+5.8
仅任务知识	27.5%	+8.3	70.8%	+12.5
仅动作知识	28.3%	+9.2	66.7%	+8.3
RoboHarn-Evo	**41.7%**	**+22.5**	**82.5%**	**+24.2**

相同模型与执行器，仅改变知识条件。

每项任务 20 个留出回合。
平均提升：22.5 / 24.2 / 19.2 个百分点。

完整分层优于平铺反思与任一单层。

Qwen 指 Qwen3.8-27B。
各行相对“无 HPK”计算；HPK 为分层物理知识。

提纲

Contents

01 技术趋势与研究路线

02 能力激发：RoboHarn-Evo

03 能力进化：Replan 与 ProcVLM

04 精准执行：VLA 工具的训练与推理

05 总结与展望

第二步：让模型从可验证交互中学习恢复，并学习判断真实进展

Replan：模型要能真正修复错误

模型不能只会解释错误，要能真正修复错误，并继续完成任务

MR：操作关系恢复



电钻放错了容器。
重新抓起电钻，放入任务指定的塑料盒。

验证：错误容器关系消除，正确关系成立。

OR：遮挡恢复



机械臂挡住了工作区。
先调用 home 恢复视野，再继续分拣食物。

验证：回位后重新观察，目标重新可见。

MTR：目标缺失恢复



场景里没有饮料。
转而抓取菠萝，完成可行的水果放置。

验证：不反复找缺失目标，可行部分继续推进。

恢复是否有效，必须由执行后的实际结果验证。

因此需在可交互仿真环境中闭环训练 replan，不能仅凭语言解释或参考答案判断恢复。

Zhuoran Wang et al. Replan for Error Recovery: Training Embodied VLMs through Verifiable Interaction with Action-Grouped Turn-GSPO. https://arxiv.org/abs/2610.27003.

ERR-Env 与 ERR-Bench：从相同起点验证恢复

可恢复快照让模型从正常、中间或错误状态出发，用实际执行结果验证恢复

动作接口：模型选择高层动作与图像坐标

动作	环境执行的含义
pick(x, y)	抓取图像坐标对应的物体
place(x, y)	放入容器，或放回桌面
home()	机械臂回位，恢复视觉观测
done()	结束回合，检查完成与终止

底层执行器固定；快照保存完整机器人和场景状态。

ERR-Bench：1,503 个留出任务快照

起点	数量	评测目标
正常初始	501	从头完成整个任务
执行中间	501	从当前状态继续完成
指定错误	501	修复错误并继续完成

错误快照：MR / OR / MTR 各 167 个

保存并直接恢复快照 → 各模型面对相同起点 → 执行后检查实际状态

训练可直接采样错误，避免长前缀重放；评测可控制起点，公平比较恢复行为。

ERR-Env：随任务同步合成 checker

合成仿真任务时，可同步生成场景、执行程序与 checker

示例任务：魔方放入木盒，刀放入蒸笼，蜡烛台和干扰物不参与任务

```
def checker(self, success_mode=False,
            progress_mode=False, penalty_mode=False,
            visibility_mode=False):
    r = self.check_on(self.rubikscube, self.wooden_box)
    k = self.check_on(self.knife, self.steamer)
    if success_mode:
        return r and k

    if progress_mode:
        held = (self.check_grasp(self.rubikscube)
                or self.check_grasp(self.knife))
        grasp_bonus = 0.5 if held else 0.0
        return r + k + grasp_bonus, 2.0

    # visibility_mode / penalty_mode: ...
```

完成 success

check_on 检查两个目标关系，
二者都成立才完成。

进度 progress

每个正确放置计 1，抓住目标加 0.5，
并返回参考总进度 2.0。

错误 penalty

目标放错容器 +1。无关物体被抓 +0.5，
进入目标容器各 +1。

可见性 visibility

count_visible_objects(scene)
返回当前可见物体数。

任务给出对象、目标关系与错误类型，checker 即可同步合成

checker 读取仿真状态，仅用于奖励与评测，检查结果不作为策略输入

强化学习核心：Action-Grouped Turn-GSPO

按实际效果逐轮给分，同类动作比较优势，再对整轮推理与动作共同更新

① 状态评分：消除错误也应得到正反馈

$$\mathrm{q(s)} = \mathrm{progress(s)} - \mathrm{penalty(s)}$$

② 动作奖励：按动作的实际目的给分

$$r_t^{act} = \begin{cases} r_t^{exec} + \Delta q_t & \text{pick, place} \\ r_t^{exec} + \Delta v_t & \text{home} \\ r_t^{exec} + \mathrm{success}(s_{t+1}) & \text{done} \end{cases}$$

执行奖励检查合法性，Δq / Δv 为评分 / 可见数变化。

③ 总奖励：定位能力、局部纠错与整体完成

$$r_t = 0.25\, r_t^{grd} + 0.50\, r_t^{act} + 0.25\, r_t^{out}$$

grd 保留定位，out 兼顾净进展、效率与终止。

④ 动作分组：比较同类动作的相对好坏

$$A_t^{raw} = \frac{r_t - \mu_{G_t}}{\sigma_{G_t} + \delta}$$

pick / place：同一起点的 6 条轨迹内，同类比较。
home / done：整批次同类比较，缓解样本稀少。
理由：避免跨动作的奖励尺度与频次干扰。

⑤ 语义修正：避免坏动作因整局成功获奖

$$\tilde{A}_t = \begin{cases} \min(A_t^{raw}, -0.02) & r_t^{exec} < 0 \\ \min(A_t^{raw}, -0.01) & r_t^{exec} \geq 0, \Delta q_t \leq 0 \\ \max(A_t^{raw}, +0.02) & r_t^{exec} \geq 0, \Delta q_t > 0 \end{cases}$$

无效或未改善为负，确有改善为正。

⑥ Turn-GSPO：逐轮信用、整轮更新

$$\rho_t = \left(\frac{\pi_\theta(y_t)}{\pi_{old}(y_t)}\right)^{1/|y_t|}$$

$$J = \mathrm{E}[\min(\rho_t \tilde{A}_t, \mathrm{clip}(\rho_t, 1-\varepsilon, 1+\varepsilon)\tilde{A}_t)] - \beta\, L_{KL}$$

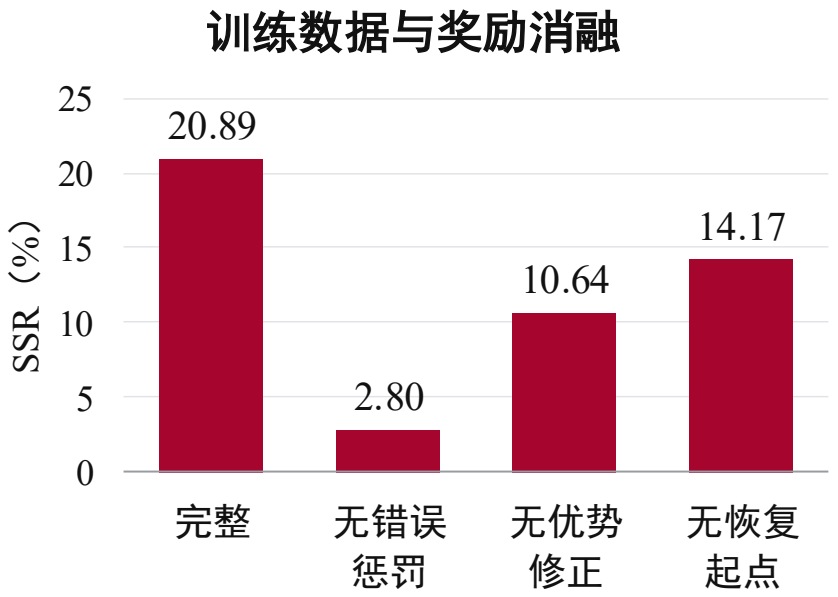
实验：恢复当前错误，还要可靠完成剩余任务

2B 模型的指定错误恢复率最高，但严格完成仍落后于最强通用模型

统一提示、工具接口与 500-token 思考预算，评测 1,503 个留出任务快照；下表全部为百分数。

模型	PG	SSR	RR	R-SSR
GPT-5.6 Sol	76.81	15.44	49.10	11.78
Qwen3.8-Max	82.66	27.21	57.69	26.55
RoboBrain2.5-8B	49.21	4.06	31.54	2.40
Embodied-R1.5	54.86	3.39	24.35	1.40
ERR-VLM (SFT, 2B)	62.36	10.64	42.52	8.58
ERR-VLM (Full, 2B)	**71.08**	**20.89**	**65.47**	**22.16**

闭源：前两行。具身模型：中间两行。
去掉 RL 恢复起点，RR 降至 39.12%。



去掉错误惩罚，PG 仍有 72.45%，SSR 却降至 2.80%。

PG：完成子任务的比例。SR 只要求全部完成，SSR 还要求无新增执行错误并以 done 结束。
RR：修复起始错误。R-SSR：从错误状态出发，严格完成剩余任务。修复一个错误仍不等于可靠做完。

过程奖励模型：把可验证反馈扩展到真机

Replan 依赖环境状态检查，真机学习需要从视觉判断任务是否真正推进

反馈条件发生变化

场景	可用反馈	反馈特点
仿真	状态与规则检查	可验证动作效果
真机	图像与任务指令	需学习视觉状态估计

时间与运动不直接代表进度

抓起杯子
抓取阶段完成

→

移动 / 重试
耗时与图像变化

→

放置完成
到达目标位

同一阶段可能反复尝试，失败还可能使已完成状态丢失。

需要怎样的过程反馈

任务指令 + 近期观察
明确目标与当前物理状态

→

当前阶段 + 剩余步骤
区分有效推进、停滞与恢复

→

连续进度估计
为策略学习提供稠密反馈

把子任务语义、剩余动作推理和连续数值预测纳入同一奖励模型

ProcVLM：先推理剩余步骤，再估计当前进度

训练好的模型同时输出过程解释与连续进度，失败和重试由实际状态决定



过程标签：子任务预算与阶段内连续进展

先自动划分语义步骤，再在各步骤的预算内分配连续进度

完整轨迹
VLM 生成计划与子任务区间

→

去重关键帧
完成状态、剩余动作与推理

→

扩展到逐帧标签
每帧绑定当前子任务与进度

T 为轨迹时长，K 为有效子任务数；第 k 步区间为 $[s_k, e_k]$，φ 表示视觉表征。

$$p(t) = \frac{\int_0^t w(\tau) r(\tau) d\tau}{\int_0^T w(\tau) r(\tau) d\tau}$$

$$w(\tau) = clip\left(K \cdot \frac{e_k - s_k}{T}, 0.75, 1.25\right), r(\tau) = \frac{||\dot{\varphi}(\tau)||}{\int_{s_k}^{e_k} ||\dot{\varphi}(u)|| du}, k = k(\tau)$$

子任务预算 w

按相对时长修正，再截断到 0.75–1.25，
避免极长或极短步骤主导整体进度。

阶段内分配 r

在同一子任务内归一化感知变化，
据此分配该步骤的连续进展。

重点是语义过程监督，数值插值是标签构造；质量依赖分段与有效状态变化

联合训练：用过程标签构造三个互补任务

进度估计是主任务，动作分段与未来规划提供过程理解监督

ProcCorpus-60M：30 个数据集、约 40 万轨迹，生成 6000 万逐帧过程标签

训练任务	输入与预测目标	对进度估计的作用
动作分段	整段视频 + 指令 输出动作语义及起止帧	识别执行阶段与阶段切换
未来规划	近期观察 + 指令 输出下一动作或剩余动作序列	理解还需完成哪些步骤
进度预测	近期观察 + 指令 先输出剩余动作，再回归进度	将步骤完成证据映射到连续数值

共享 Qwen3-VL-2B 骨干

语言分支学习三个任务的文本输出
Value Head 仅在进度样本上学习连续回归

$$L = L_{LM} + \lambda \cdot I_{prog} L_{value}$$

三个任务均优化文本损失，
进度样本额外优化数值误差。

实验：少量示范后的奖励迁移与 baseline 对比

一条成功示范改善进度排序和完成判断，失败示范进一步补足异常状态

VOC：成功轨迹进度排序，越高越好；MAE：失败定位误差，越低越好；MCC：完成判断，越高越好

适配设置	奖励模型 / baseline	VOC ↑	MAE ↓	MCC ↑
Zero-shot	Qwen3.5-27B	0.6191	0.1267	0.5678
Zero-shot	RoboDopamine-8B	0.9095	0.2268	0.5506
Zero-shot	Robometer-4B	0.4129	0.1218	0.4516
Zero-shot	**ProcVLM-2B**	**0.4920**	**0.2001**	**0.6665**
1-shot 成功	Robometer-4B	0.4094	0.1222	0.4435
1-shot 成功	**ProcVLM-2B**	**0.9137**	**0.1241**	**0.7918**
成功 + 失败	Robometer-4B	0.4040	0.1221	0.4488
成功 + 失败	**ProcVLM-2B**	**0.9301**	**0.1187**	**0.8053**

同为 1-shot 成功：ProcVLM 的排序 MCC 更高，MAE 与 Robometer 接近。

“成功 + 失败”额外为每种失败类型各提供一条示范。零样本并非所有指标领先，适配后再比较迁移效果。

实验：过程奖励改善策略学习，收益依任务而异

仿真比较不同奖励模型，真机比较同一示范集上的 SFT 与奖励微调

仿真在线 RL：不同奖励模型

OpenVLA-OFT + GRPO，末端成功率（%）

奖励模型	Plate 约 30h	Can 约 120h
Robometer-4B	84.3	51.6
RoboDopamine 多视角	87.5	2.1
ProcVLM-2B	**84.2**	**75.6**

Plate 上各方法接近，RoboDopamine 略高。
Can 上 ProcVLM 比 Robometer 高 24.0 个百分点。

真机离线 RFT：对照 SFT

JAKA 叠碗，π0.5，50 条相同示范

训练步数	SFT	RFT	增益 pp
5k	**37.5**	**62.5**	**+25.0**
10k	**70.8**	**83.3**	**+12.5**

指标：soft success（%，允许部分得分）

过程奖励生成优势标签，
改善同一示范集上的早期微调效果。

提纲

Contents

01 技术趋势与研究路线

02 能力激发：RoboHarn-Evo

03 能力进化：Replan 与 ProcVLM

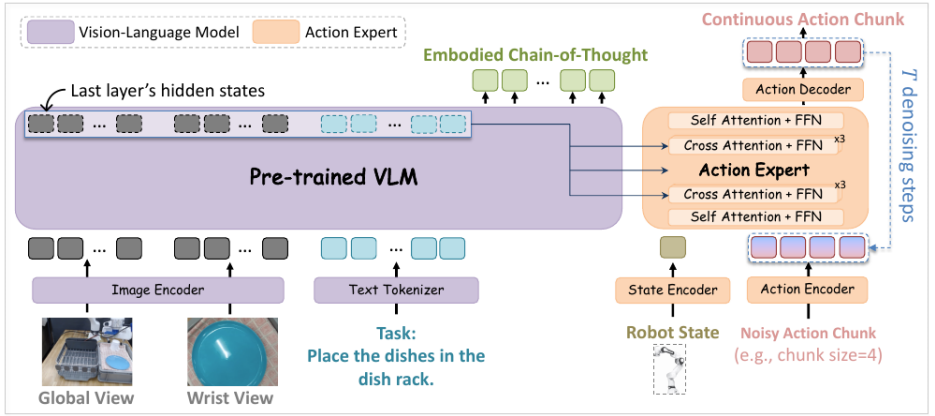
04 精准执行：VLA 工具的训练与推理

05 总结与展望

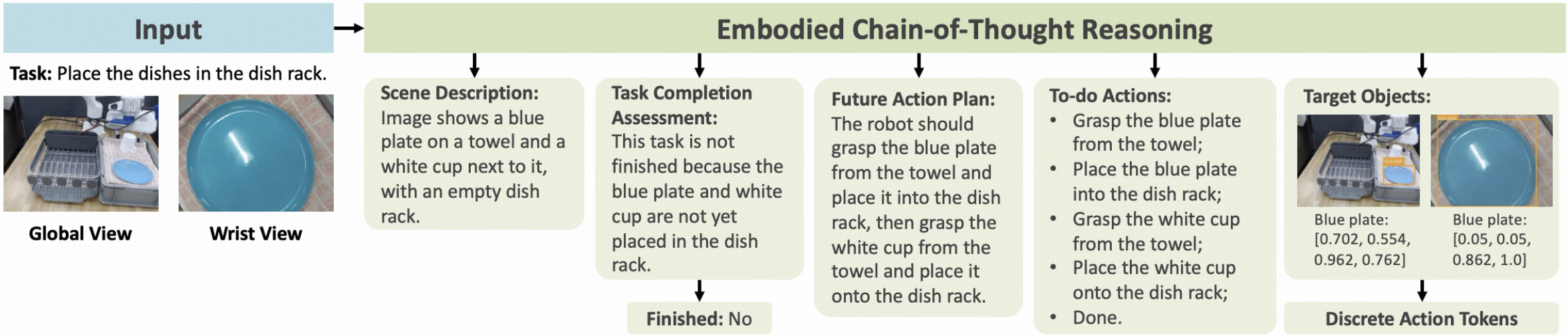
第三步：预训练改善执行，推理优化筛选动作，再探索无 LLM 的视觉轨迹模型

ZR-0：预训练 VLA 的双流架构

训练时用逐帧语义监督塑造表征，
部署时直接生成连续动作



VLM 语义表征　　图像与任务指令共同编码。训练时预测逐帧 ECoT。



ECoT 覆盖场景、进度、计划与原子子任务，并提供目标定位和离散动作监督。

ZR-0：密集语义监督改善长程执行

LIBERO 长程任务领先，跨具身迁移的收益取决于预训练技能覆盖

LIBERO 成功率（%）

方法	Long	平均
OpenVLA	53.7	76.5
CoT-VLA	69.0	83.9
π0.5	92.4	96.9
GR00T-N1.7	94.4	97.0
DeepThinkVLA	96.2	97.0
MolmoAct2	93.2	97.2
ZR-0	**96.4**	**97.8**

相对 π0.5，Long 提升 4.0 个百分点
相对 JoyAI-RA，RoboCasa 提升 6.1 个百分点

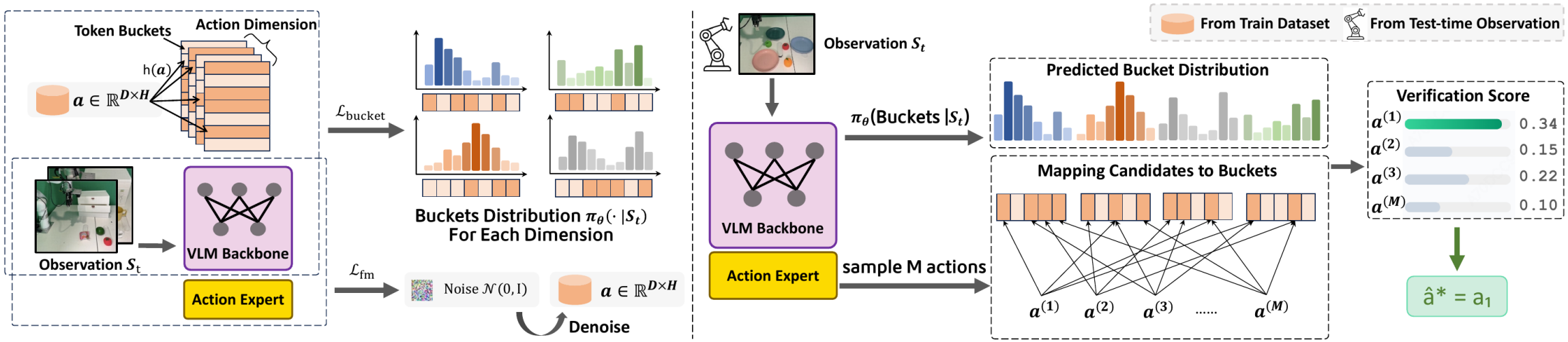
RoboCasa GR-1 成功率（%）

方法	24 任务平均
GR00T-N1.6	47.6
ABot-M0	58.3
JoyAI-RA	63.2
ZR-0	**69.3**

跳过预训练：Long 从 96.4 降至 92.6。
xArm 进度评分从 67.8 提升至 76.0。
抓放迁移较强，关门与挂杯仍有不足。

TokenPB：动作候选的同模型自验证

连续动作专家负责生成，VLM 的动作 token 概率负责排序



(a) Training Stage

(b) Inference Stage

训练：同一检查点既生成也评分

连续动作学习 flow matching。
示范动作转成 token，学习条件概率。

推理：多候选共享观察前缀

采样 M 个动作，批量计算 token 概率。
选择得分最高的连续候选，直接执行。

核心：把生成与验证放进同一模型，使用共享 KV 缓存控制候选筛选开销。

Chen Zhao et al. TokenPB: Enabling VLA Action Self-Verification via VLM Token Probability Bucketing. Neurips 26.

TokenPB：强验证基线与推理扩展收益

对比外部验证器和内部评分，16 候选筛选在多个主干上稳定受益

验证方法对比（%）

方法	Long	LIBERO	Simpler
MG-Select 单次	79.6	92.0	47
MG-Select	82.7	93.1	50
AsyncVLA	96.0	97.4	71
π0.5 单次	93.0	96.5	66
RoboMonkey 复现	95.0	97.3	71
TokenPB	**97.6**	**98.4**	**76**

同一检查点的筛选收益

主干	M=1	M=16
Qwen2.5-VL	93.9	96.8
InternVL3.5	94.7	97.8
π0.5	**96.5**	**98.4**

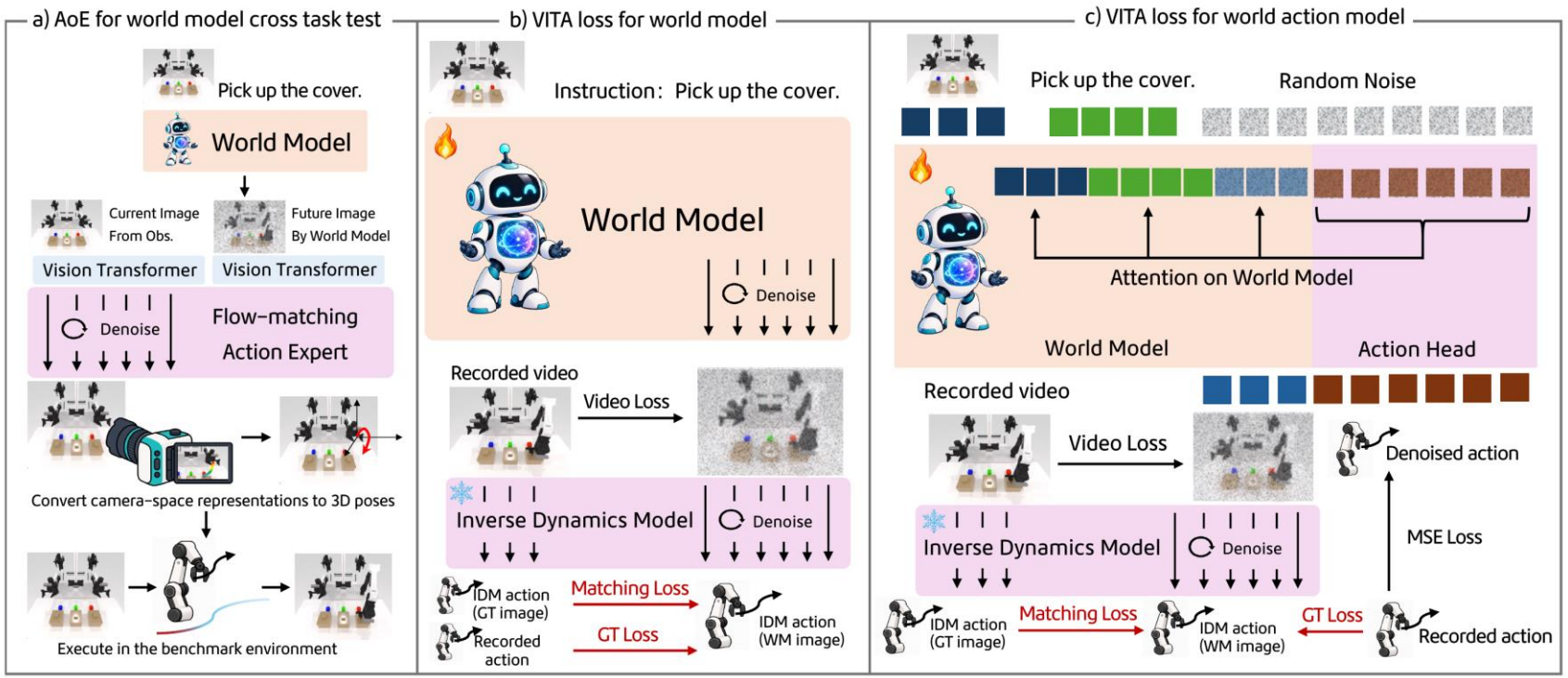
真机 Unseen 完成评分
π0.5：30.3 提升至 44.0。
CALVIN 平均连续任务长度
π0.5：4.07 提升至 4.27。

相对 RoboMonkey 复现：LIBERO +1.1，Simpler +5 个百分点

M=64 时额外验证耗时：TokenPB <0.2 秒，RoboMonkey >4 秒，MG-Select >1.5 秒。

AoE：预训练动作专家模型

AoE 将视觉变化恢复为动作，VITA 用动作误差约束生成的未来



AoE：恢复动作

输入当前帧、未来帧与机器人状态。
预测相机空间动作，通过标定转换并执行。

VITA：约束未来

固定 AoE，反传误差，让世界模型保留夹爪、物体位置等执行细节。

AoE 不依赖 LLM，用 9,637 小时图像—动作对预训练，冻结后承担执行接口与监督器。

AoE：生成得像，仍可能执行失败

固定 AoE 与执行协议，更换视频来源即可暴露未来预测的控制缺陷

未来视频的跨任务执行

视频 / 动作来源	采样	LIBERO	RT Hard
记录动作回放	参考	72.3	70.0
记录图像 + AoE	参考	57.0	60.7
Cosmos + AoE	pass@1	0.4	1.1
Cosmos + AoE	pass@50	5.9	4.7
WAN2.2 + AoE	pass@1	0.3	1.3
WAN2.2 + AoE	pass@50	4.6	5.8
Cosmos + VITA + AoE	**pass@1**	**51.7**	**36.4**
WAN2.2 + VITA + AoE	**pass@1**	**47.0**	**40.4**

动作表示决定迁移能力

动作表示	Zero-shot	10-shot
3D pose	0.0	2.3
相机空间	**29.2**	**37.8**

相机空间避免记忆环境坐标，
标定过程负责恢复三维位姿。

Cosmos 的 LIBERO 成功率
从 0.4 提升至 51.7。
关键是保留动作所需的几何细节。

增加采样次数仍难弥补几何错误，动作监督带来的改善远大于单纯增加视频候选。

提纲

Contents

01 技术趋势与研究路线

02 能力激发：RoboHarn-Evo

03 能力进化：Replan 与 ProcVLM

04 精准执行：VLA 工具的训练与推理

05 总结与展望

报告总结：推理、执行与反馈需要形成闭环

具身能力要体现为可验证的物理结果，并通过交互经验持续改进

报告内容	最值得带走的启示
能力激发 **RoboHarn-Evo**	外部经验与工具编排可以激发模型已有能力。 通过真实执行找到能力边界，再决定怎样增强系统。
能力进化 **Replan / ProcVLM**	交互 RL 训练错误恢复，过程奖励提供学习信号。 可靠的成败与进度反馈，是能力内化的基础。
精准执行 **VLA / WAM / AoE**	表征学习、动作监督与推理筛选共同改善执行。 专用模型让高层决策拥有可调用的操作能力。

落地选择：多少决策留给在线推理，多少经验提前固化为执行逻辑

泛化需求、任务变化频率与运行成本，决定两条路线各自的适用范围。

未来展望：在线通用模型与离线工作流

核心区别：任务级推理发生在运行时，还是部署前

在线通用具身模型

家庭等开放环境，追求强泛化

训练重点

规模化训练环境，开展稳定高效 RL，
学习具身推理与工具调用。

在线部署

按任务实时推理并调用工具，
根据环境反馈调整操作与恢复。

大模型公司 / 头部具身公司的重点探索

离线 VLM Coding

工业固定任务，支持换产重配

部署前

大模型生成并验证 workflow，
换产时重新生成或调整代码。

运行时

直接执行代码，分支处理有限异常。
调用定位、分割、位姿、轨迹等小模型。

适应物品、位置、背景等有限变化

谢谢！ 敬请批评指正！