主流WAM的测试报告-v0.2

版本：v0.2

时间：2026-6-19

作者：Simple WAM Team，详见附录

一、报告目的

本报告面向当前主流 World Action Model（WAM）方案的阶段性复现评测与路线判断，目标是在统一任务与评测协议下回答三个核心问题：

1. **WAM 是否已经具备可靠的标准任务执行能力？**
   即在标准 LIBERO 测试集上，WAM 能否达到与强 VLA 基线相当的成功率。
2. **WAM 是否天然具备更强 OOD 泛化能力？**
   即在仅使用 LIBERO 训练、zero-shot 迁移到 LIBERO-Plus 的设置下，WAM 是否能稳定优于强 VLA 基线。
3. **WAM 的性能收益与主要瓶颈分别来自哪里？**
   重点分析 video generation backbone、机器人视频-动作预训练、test-time video generation 对 OOD 能力和推理效率的影响。

围绕上述问题，报告对 π0.5、DreamZero、Fast-WAM、Fast-WAM-Joint、Fast-WAM-IDM、LingBot-VA 等代表性方法进行复现、公开结果整理与对比分析。基于LIBERO / LIBERO-Plus上的测评结果，判断 WAM 的已验证能力、尚未成立的假设，以及后续最值得投入的方向。

1. 评测对象

1.1 已测评的模型

模型	定位	本稿证据
π0.5	当前公开可获得、在 LIBERO 上具备成熟微调链路的强 VLA 基线	内部 LIBERO；公开 LIBERO-Plus 作为背景参照
DreamZero	预训练型 / 自回归 WAM	内部适配、RLinf checkpoint / 续训
Fast-WAM	训练期video co-training，测试期直接动作	内部复现 + 官方结果
Fast-WAM-Joint	测试期 joint video-action denoise	内部复现 + 官方结果

Fast-WAM-IDM	先未来状态、后动作的 IDM 变体	内部复现 + 官方结果
LingBot-VA	具备大规模 embodied pretraining 的 WAM	官方 ckpt，当前仅 Long suite
其他VLA/WAM模型	作为Benchmark对比	

二、主要结论

结论1：WAM的基本能力成立（标准Libero测试集）

目标：在LIBERO数据集上验证WAM的基本能力

实验：使用不加载预训练权重的方式，从WAN2.2-5b冷启动训练Dreamzero和FastWAM

结果：内部结果显示，Fast-WAM 家族均达到 97% 以上平均成功率，说明 WAM 在标准 LIBERO 上具备完整任务执行能力。更重要的是，内部结果与公开结果整体接近，复现可信度较高。

表1 Libero数据集测评结果

模型	Spatial	Object	Goal	Long	Average
PI0	96.8	98.8	95.8	85.2	94.1
PI0.5	97.8	98.8	97.6	92.4	96.7
Dreamzero-RLinf release	95.3	-	-	-	-
Dreamzero-复现	96.2	88.8	90.0	79.8	88.7
FastWAM-release	96.4	**100**	96.4	93.6	96.6
FastWAM-复现	99.0	99.6	96.8	93.0	97.1
FastWAM-Joint	**99.2**	98.6	**98.4**	96.8	**98.3**
FastWAM-IDM	98.8	98.8	97.0	**97.6**	98.05

> 注：Dreamzero-复现训练方法：加载Dreamzero-RLinf release step18000，在libero四个suite上继续训练18000step。

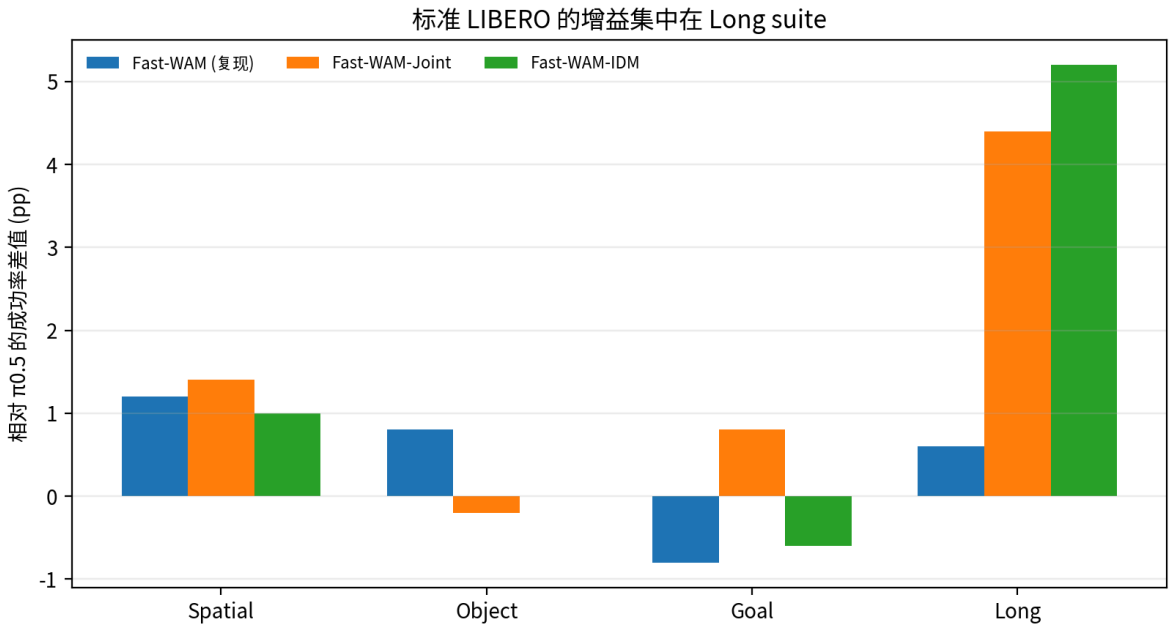


图 1. Fast-WAM 变体相对内部 π0.5 的 suite-level 差值。

深入分析：

1. 从 LIBERO 主评测结果看，WAM 路线已经具备基本任务能力：FastWAM 系列平均成功率达到 96.6%-98.3%，与强 VLA 基线 π0.5 基本持平甚至略优，说明 WAM 在标准任务上是可行的。
2. Fast-WAM-Joint 相比 π0.5 平均提升 1.6pp，Fast-WAM-IDM 提升 1.35pp；增益主要集中于 LIBERO-Long（通常为2-stage 长轨迹任务），分别提升 4.4pp 和 5.2pp。
3. DreamZero 复现效果效果不佳（包括NVIDIA开源版和RLinf开源版），可能由于该架构对 backbone 规模、机器人动作预训练、多视角对齐和闭环执行频率高度敏感。论文主结果依赖 14B 视频骨干、大规模多样化机器人数据和经过优化的异步推理系统；当前 NVIDIA-LIBERO 适配和 RLinf WAN2.2-5B cold-start 设置均偏离该最优条件。当前结果判断说明 DreamZero 开源链路的迁移成本较高，而尚不足以单独否定其完整预训练设定下的能力。

结论2：WAN基模本身不能直接使WAM具备强OOD泛化能力，现阶段观察到的WAM强 OOD能力更可能来自大规模机器人预训练+视频生成能力的进一步优化。

目标：测试 WAM 的 OOD 泛化能力，并分析 OOD 能力主要来自 视频Backbone、预训练数据还是推理时 video generation。

实验：仅在LIBERO上训练，Zero-shot迁移测评LIBERO_PLUS；对比冷启动WAM、加载预训练ckpt的WAM，以及强 VLA 基线 π0.5。

结果：冷启动 WAM 的 OOD 表现明显不足：DreamZero 内部适配、DreamZero-RLinf 续训、FastWAM 和 FastWAM-Joint 分别为 41.6%、54.0%、53.9% 和 59.0%，均显著低于 π0.5 的 85.7%。相比之下，加载预训练 ckpt 的 GE-Act 和 Cosmos-Policy 分别达到 80.3% 和 82.2%，明显更接近 π0.5。

表2 Libero Plus数据集测评结果

	Model	机器人数据预训练	Original	Camera	Robot	Lang.	Light	BG	Noise
VLAs	π0	✅	94.2	13.8	6	58.8	85	81.4	79
VLAs	π0 (rerun)	✅	91.3	61	40.8	63.5	89.3	84.1	80.1
VLAs	π0-FAST	✅	85.5	65.1	21.6	61	73.2	73.2	74.4
VLAs	**π0.5**	✅	**96.9**	**75.4**	**77.5**	**85.6**	**96.9**	**94.6**	**89.7**
VLAs	OpenVLA-OFT_m	✅	97.6	55.6	21.7	81	92.7	91	78.6
VLAs	UniVLA	✅	95.2	1.8	46.2	69.6	69	81	21.2
VLAs	RIPT-VLA	✅	97.5	55.2	31.2	77.6	88.4	91.6	73.5
VLAs	X-VLA	✅	98.1	23.4	89.7	75.7	88.2	96	62.7
VLAs	HoloBrain0-GD	✅	96.7	65.5	58.2	78.7	88.1	90.3	66.9
VLAs	ABot-M0	✅	98.6	60.4	67.9	86.4	96.2	91.6	86.4
VLA+WM	VLA-JEPA	❌	97.2	64.2	67.7	88.1	91.8	93.4	65.8
WAM	GE-Act	✅	94.4	60.7	77	77.4	95.8	86	90.9
WAM	**Cosmos-Policy**	◐	**98.5**	**75.8**	**63.3**	**81.7**	**96.5**	**88.9**	**92.7**
WAM	DreamZero（内部适配）	❌	88.7	26.3	23.5	56.4	63.5	42	43.5
WAM	DreamZero（RL Inf版本）	❌		31.1	31	78.7	81.8	44.7	52.8
WAM	Fast-WAM	❌	97.1	25.6	42.9	73.9	61.7	54.9	43.6
WAM	Fast-WAM-Joint	❌	98.3	34.3	55.7	89.4	89.8	45.7	32.4
WAM	LingBot-VA	✅		44.2	84.5	93	80.7	73.4	65.5

深入分析：

1. FastWAM 在标准 LIBERO 上表现很强，但迁移到 LIBERO-Plus 后仅为 53.9%，Dreamzero的多个版本也有类似现象，说明仅基于WAN视频预测主干，标准任务能力并不会自动转化为 OOD 泛化能力。
2. 更关键的变量可能是机器人数据视频-动作预训练。GE-Act 和 Cosmos-Policy 同样属于 Video-Gen based WAM，但在加载预训练 ckpt 后达到 80% 以上，显著优于冷启动WAM。因此，各WAM论文中声称的强 OOD更可能来自**大规模机器人数据预训练**和**视频预测模型的视觉-动作 grounding**，而不是Video Generation Backbone结构本身。

3. **测试时视频生成对OOD能力有一定帮助，但不是根本来源**。FastWAM-Joint 相比 FastWAM 提升 5.1pp，说明推理时引入 video generation 能改善部分扰动鲁棒性，但绝对性能仍明显低于 π0.5。

表3 Fast-WAM及变体在Libero Plus数据集测评结果对比

扰动类别	Fast-WAM	FastWAM-Joint	差值 (pp)
Camera	25.4	34.0	+8.6
Robot	42.8	55.1	+12.3
Language	73.2	88.9	+15.7
Background	53.5	44.9	-8.6
Noise	44.5	33.3	-11.2
Layout	66.6	73.4	+6.8
Light	79.0	90.0	+11.0

结论3：WAM未优化时的推理速度显著低于VLA，优化后可接近

目标：在 LIBERO 数据集上验证 WAM 的推理速度，并分析不同推理范式之间的速度差异。

实验：我们从 WAN2.2-5B 初始化 DreamZero 与 Fast-WAM 系列模型，并在统一的 LIBERO 推理流程中统计单次 policy update 的平均 wall-time。需要说明的是，本实验统计的是当前内部代码链路下的端到端推理耗时，包含模型前向、denoising、action chunk 生成等实现开销；

结果：

内部实测显示，在当前开源实现和硬件配置下，WAM 的单次 policy update 耗时整体高于 VLA。π0.5 在本地链路中约为 100ms/update，而 Fast-WAM 为 568ms，Fast-WAM-Joint 和 Fast-WAM-IDM 分别上升至 974ms 和 1189ms；DreamZero 未优化顺序推理链路进一步达到 5616ms。

上述内部实测并不代表各方法的系统上限。例如 Fast-WAM 论文报告其 action-only 推理可达到约 190ms；DreamZero 通过 CFG parallelism、DiT caching、CUDA Graph、量化和 Flash-style denoising 等优化后可降至约 150ms。

表4 主流VLA和WAM模型推理速度结果对比

	模型	推理耗时	GPU / 硬件	数据来源
VLAs	π0.5，内部实测	100 ms	A100	内部 LIBERO profiling
	π0 / OpenPI JAX，2 views	53.7 ms	RTX 4090	Realtime-VLA 在单卡 RTX 4
	π0 / optimized Triton，2 views	27.3 ms	RTX 4090	Realtime-VLA 优化后 1/2/3
	OpenVLA，LIBERO 原版	240 ms	A100	OpenVLA-OFT 论文在 100 c
	OpenVLA-OFT / L1 action head	73 ms	A100	OpenVLA-OFT 通过 continu
	OpenVLA-OFT+，ALOHA 设置	321 ms	A100	ALOHA 设置额外处理多视角
	RDT-1B，ALOHA 设置	297 ms	A100	OpenVLA-OFT 论文作为 AL
	ACT，ALOHA 设置	58 ms	A100	OpenVLA-OFT 论文中 ACT
	Diffusion Policy，ALOHA 设置	90 ms	A100	OpenVLA-OFT 论文中作为从
WAMs	Fast-WAM，论文报告	190 ms	RTX 5090D V2 32GB	Fast-WAM 论文明确所有 lat
	Fast-WAM，内部实测	568 ms	A800	内部 LIBERO profiling
	Fast-WAM-Joint，内部实测	974 ms	A800	内部 LIBERO profiling
	Fast-WAM-IDM，论文报告	810 ms	RTX 5090D V2 32GB	Fast-WAM 论文报告 IDM 作
	Fast-WAM-IDM，内部实测	1189 ms	A800	内部 LIBERO profiling
	DreamZero，未优化 baseline / 内	5.7 s / 5.616 s	GB200 / A800	DreamZero 论文报告 basel
	DreamZero-Flash，优化后	150 ms	GB200	DreamZero 论文称通过 CF
	GE-Act	<200 ms	ommodity GPU，未明确型	GE-Act 公开资料称可在 200
	Cosmos Policy	公开 per-update ms		Cosmos Policy 官方仓库主
	VLA-JEPA	公开 per-update ms		公开仓库/论文主要给评测并
	GR00T-N1 / N1.5	公开 per-update ms		NVIDIA 资料给部署硬件和模

深入分析：

1. **当前未优化 WAM 链路相比 VLA 存在明显推理延迟劣势**。内部实测中，π0.5 约 100ms，而 Fast-WAM 为 568ms，Fast-WAM-Joint/IDM 接近或超过 1s，DreamZero 达到 5.6s。这说明在实际工程落地前，WAM 必须优先解决推理效率问题。
2. **WAM 的速度瓶颈主要来自测试时显式 imagination，而不是 world supervision 本身**。Fast-WAM action-only 明显快于 Joint/IDM，说明训练期使用 video/world supervision 可以保留，但测试期如果继续生成未来视频或进行联合 video-action denoising，就会显著放大延迟。这支持“训练期 world-aware，推理期 action-only”的轻推理路线。
3. **公开优化结果表明 WAM 仍有工程优化空间**。Fast-WAM 论文值 190ms 明显快于内部评测 568ms；DreamZero 从未优化 5.7s 降至 Flash 版本 150ms，说明缓存、并行、少步 denoising、量化和 CUDA Graph 对 WAM 至关重要。

结论4：目前WAM缺乏有效的3D能力，是影响OOD泛化能力因素之一

目标：本部分实验旨在验证当前动作世界模型（World Action Model, WAM）在三维空间理解与精细化操作能力上的不足，并进一步分析 3D 能力缺失是否会影响模型在空间扰动和长尾场景下的 OOD 泛化能力。

实验设计

本部分包括两组实验。

1. **实验 I：现有 WAM 的 3D 操作能力评测。**

基于 RoboTwin 和 LIBERO-Plus 等仿真任务，对 Fast-WAM 等现有 WAM 模型进行评测。任务类型覆盖两类代表性操作：一类是对 3D 空间定位不敏感的粗粒度操作，例如较大目标物体的抓取和移动；另一类是对 3D 位置、物体部件和接触点要求更高的精细化操作，例如打开微波炉门、拨动开关、抓取特定容器等。通过对比不同任务上的成功率，分析当前 WAM 是否具备稳定的三维空间理解能力。

2. **实验 II：3D 辅助监督对 WAM 的能力提升评测。**

在保持原有 WAM 模型主干和推理流程基本不变的前提下，引入 VGGT-Omega 等 3D Foundation Model 生成的几何信息，为模型提供 Gaussian、深度、几何特征等辅助监督。该方案仅在训练阶段引入 3D 监督头，用于约束视频—动作潜变量的空间一致性；推理阶段移除 3D 头，模型仍沿用原有 WAM 主干，因此不会引入额外推理成本。实验重点观察该方案是否能够提升 WAM 在精细化操作和空间 OOD 场景下的任务成功率。

实验结果与分析

1. ***现有 WAM 在精细化 3D 操作任务上表现明显下降***

表 1 展示了 Fast-WAM 与 PI-0.5 在 RoboTwin 部分任务上的评测结果。可以看到，在 Adjust Bottle 和 Click Alarmclock 等任务中，Fast-WAM 基本可以达到接近满分的性能，在 Easy 和 Hard 设置下均表现稳定。这类任务通常具有较明确的目标区域，且对精确的三维部件定位和接触点控制要求相对较低，因此更容易被当前 WAM 模型完成。相比之下，在 Move Microwave 和 Turn Switch 等任务中，模型性能明显下降。

以 Fast-WAM 为例，Move Microwave 在 Easy 和 Hard 设置下分别只有 62 和 45，Turn Switch 也仅为 61 和 59。这类任务通常需要模型理解物体的局部结构、三维空间位置、可操作部件以及动作接触关系。例如，微波炉任务不仅需要识别目标物体，还需要准确定位门把手或可交互区域；开关任务则要求模型对小尺度部件的位置和方向变化具有较强敏感性。因此，这些任务更能暴露 WAM 在三维空间理解和精细化动作生成方面的不足。

表5 Robotwin上部分任务测试结果

仿真任务	FastWAM		PI 0.5	
	Easy	Hard	Easy	Hard
Adjust Bottle	100	100	100	99
Click Alarmclock	100	100	98	89

Move Microware	62	45	34	77
Turn Switch	61	59	62	54

从结果可以看出，当前 WAM 并非在所有任务上都表现较弱，而是主要在需要精确空间定位、物体部件识别和接触点控制的任务上出现明显退化。这说明模型已经具备一定的视觉语义理解和粗粒度操作能力，但其空间表征能力仍不足以稳定支撑复杂三维交互。

这一结果也从侧面说明，WAM 的 OOD 泛化能力下降并不完全来自训练数据规模不足或动作分布偏移，3D 空间能力缺失可能是其中一个重要因素。当测试场景出现视角变化、物体姿态变化、部件位置变化或机器人本体差异时，如果模型缺乏稳定的几何表征，就容易出现目标定位错误、接触点偏移或动作执行失败。

2. ***注入 3D 能力后，WAM 在精细化操作上取得明显提升***

为了进一步验证 3D 能力对 WAM 操作性能的影响，我们在 Fast-WAM 基础上引入 Gaussian / 深度 / 几何辅助监督，形成 GaussianWAM 方案。该方法的核心特点是：训练阶段利用 3D Foundation Model 提供几何伪标签，对模型中间表征进行空间约束；推理阶段仍使用原有 WAM 主干，不额外引入 3D 模块，因此能够在不增加推理成本的情况下增强模型的三维空间表征能力。

表 2 展示了 GaussianWAM 在 RoboTwin 精细化任务上的结果。可以看到，在 Open Microwave 任务上，GaussianWAM 从 Fast-WAM 的 62 提升到 74，绝对提升 12 个百分点；在 Turn Switch 任务上，从 59 提升到 62，绝对提升 3 个百分点。两个任务平均提升 7.5 个百分点，说明 3D 辅助监督能够有效改善模型对空间结构和可交互部件的理解能力。

其中，Open Microwave 的提升更加明显，说明当任务失败主要来自目标部件定位、空间关系理解和接触区域判断时，3D 几何监督能够带来更直接的收益。Turn Switch 的提升相对有限，可能与该任务对动作精度、旋转控制和接触稳定性的要求更高有关。换言之，3D 能力可以改善模型对空间结构的理解，但对于强接触、强控制精度依赖的任务，仍需要进一步结合更细粒度的动作建模、接触动力学建模或闭环反馈机制。

表6 Robotwin上部分任务测试结果

	Open Microware (Clean)	**Turn Switch (Random)**
Fast-WAM	62	59
GaussianWAM (我们的方案)	74 (+12)	62 (+3)

进一步地，我们在 LIBERO-Plus 上评估 GaussianWAM 在不同 OOD 设置下的泛化能力。实验结果如表 3 所示。相比 Fast-WAM，GaussianWAM 在 Camera、Robot 和 Light 三类扰动设置下均取得明显提升：Camera 设置从 25.63 提升到 39.40，提升 13.77；Robot 设置从 42.88 提升到 64.84，提升

21.96；Light 设置从 61.70 提升到 89.23，提升 27.53。三类设置的平均成绩由 43.40 提升到 64.49，平均绝对提升 21.09。

表7 Libero-Plus上部分任务测试结果

	Camera	Robot	Light
Fast-WAM	25.63	42.88	61.70
GaussianWAM (我们的方案)	39.40 (+13.77)	64.84 (+21.96)	89.23 (+27.53)

实验结论

综合上述实验，可以得到以下结论。

1. 当前 WAM 已经具备较强的视觉语义理解和粗粒度操作能力，但其 3D 空间理解能力仍然不足。对于大目标、低精度、语义明确的操作任务，现有 WAM 可以取得较高成功率；但在需要精确定位物体部件、判断空间关系和执行细粒度接触动作的任务中，模型性能明显下降。这说明 3D 能力缺失已经成为限制 WAM 精细化操作能力的重要因素之一。
2. 3D 能力不足会进一步影响模型的 OOD 泛化能力。当前 WAM 容易依赖二维图像外观、视角分布和训练数据中的统计相关性。当测试场景出现相机视角变化、机器人本体变化、光照变化或物体局部姿态变化时，缺乏稳定几何表征的模型更容易出现目标定位偏差、动作接触点错误和操作失败。因此，WAM 的 OOD 泛化问题不仅是数据覆盖问题，也与模型内部空间表征能力不足密切相关。
3. 在当前基座模型和数据规模仍然有限的情况下，通过 3D Foundation Model 为 WAM 注入几何归纳偏置，是一种有效且成本较低的改进路径。实验结果表明，在不改变原有 WAM 推理主干、不增加推理阶段额外成本的前提下，引入 Gaussian、深度和几何辅助监督，可以明显提升模型在精细化操作任务和空间 OOD 场景中的表现。
4. 3D 注入并不能完全解决 WAM 的所有操作失败问题。对于 Turn Switch 等强接触、强动作精度依赖的任务，仅增强空间表征仍然不够，后续还需要结合更精细的动作建模、接触动力学建模、闭环反馈和更高质量的具身交互数据。因此，本部分实验更准确的结论是：3D 能力缺失是影响当前 WAM 精细化操作和 OOD 泛化的重要因素之一，而不是唯一因素；但从实验结果看，增强 3D 空间表征已经能够带来稳定且显著的性能收益，值得作为后续 WAM 改进的重要技术方向。

结论5：推理阶段视频生成可减少操作失败，但未改变主要失败机制。

目标：分析Fast-WAM变体在标准LIBERO上的失败模式，推理时做video generation的具体收益。

实验：对Fast-WAM、Fast-WAM-Joint 和Fast-WAM-IDM的失败 rollout进行人工归类，类别包括 perception failure、action error、recovery failure、timeout / stagnation 和 unclassified。由于

π0.5 与 DreamZero 的 failure taxonomy 尚不完整，本节主要比较 Fast-WAM 家族内部变体。

失效分类定义：

Perception Failure	Action Error	Recovery Failure	Timeout	抖动
目标物体感知失败，如：未出现抓取目标物体趋势	目标物体感知成功（趋势正确），但无法抓起/摆放等	扰动后难以自主恢复到原有任务轨迹上	固定时间过去后尚未完成任务	Policy执行过程抖动

结果：Fast-WAM、Fast-WAM-Joint 和 Fast-WAM-IDM 的失败数量分别为 68、35 和 39。相比 Fast-WAM，Joint 的 failure 数量下降 48.5%，IDM 下降 42.6%。其中，Joint 的 action error 减少 55.6%，timeout 减少 32.1%；IDM 的 action error 减少 63.9%，但 timeout 仅减少 14.3%。这说明 Joint / IDM 都能显著减少动作执行错误，但剩余失败逐渐集中到 timeout / stagnation。

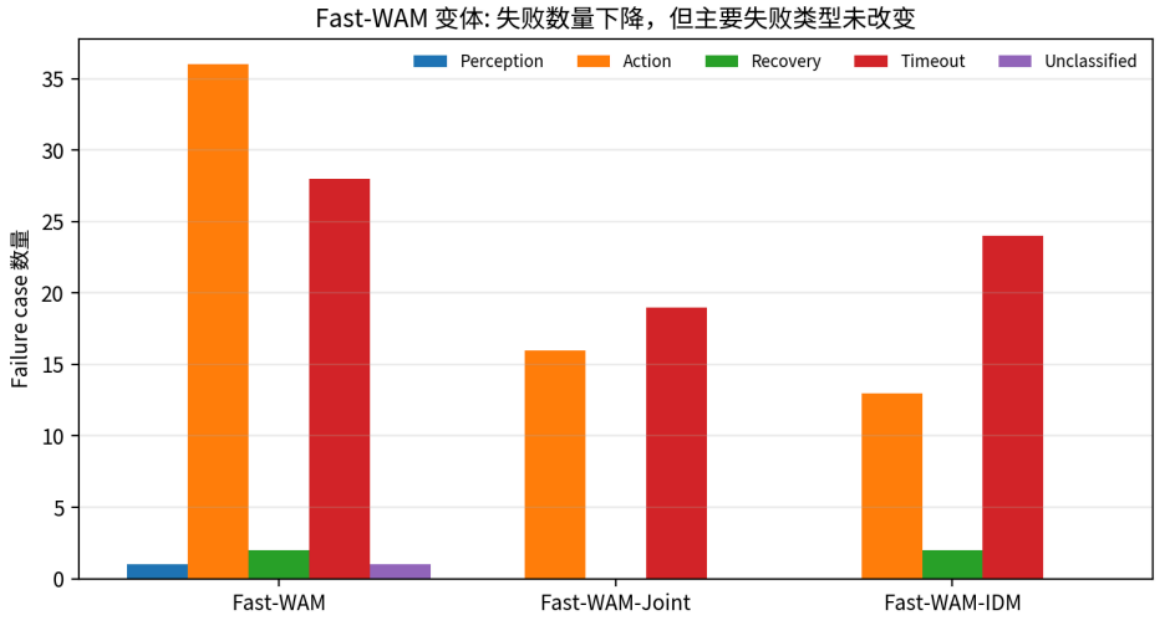


图 2 Fast-WAM 及变体的失败分类分析

表8 不同FastWAM变体的Failure Case结果分析

模型	Failure Case 数量	Perception Failure	Action Error	Recovery Failure	Timeout	抖动
FastWAM	67	1	36	2	28	0
FastWAM-Joint	35	0	16	0	19	0

FastWAM-IDM	39	0	13	2	24	0

深入分析：

1. 从 failure taxonomy 看，Fast-WAM-Joint 和 Fast-WAM-IDM 相比 action-only Fast-WAM 均显著减少失败数量。Fast-WAM、Joint 和 IDM 的 failure case 分别为 68、35 和 39，Joint 下降 48.5%，IDM 下降 42.6%。这说明推理阶段引入 world/video modeling 或 IDM-style imagination 能够提升标准 LIBERO 上的 rollout 稳定性。
2. 进一步看失败类型，三者的 perception failure 都接近于 0，说明 Fast-WAM 家族在标准 LIBERO 上的主要瓶颈不是目标识别或语义理解，而是动作执行、接触控制和任务推进。Joint 和 IDM 分别将 action error 从 36 降至 16 和 13，说明 test-time world/video modeling 主要改善了动作序列的一致性，减少了抓取失败、放置不准、碰倒物体或只完成部分操作等执行错误。
3. 但剩余失败并没有被根本改变，而是逐渐集中到 timeout / stagnation。Joint 和 IDM 的 timeout 占比分别达到 54.3% 和 61.5%，高于 Fast-WAM 的 41.2%。这说明当明显 action error 被减少后，模型的主要瓶颈转向目标附近卡住、无法完成最后接触、失败后缺少重试，以及任务阶段切换不稳定等问题。

主要结论

1. 推理阶段 world/video modeling 能显著减少 Fast-WAM 家族的失败数量，尤其降低动作执行错误；但它主要是减少已有失败类型的发生频率，并没有改变主要失败机制。当前标准 LIBERO 上的核心瓶颈已经不是 perception，而是 execution、local correction、recovery 和 completion judgment。因此，后续仅增加 test-time video generation 并不足够。

参考文献

[1] Yuan et al. Fast-WAM: Do World Action Models Need Test-time Future Imagination? arXiv:2603.16666. https://arxiv.org/abs/2603.16666

[2] Ye et al. World Action Models are Zero-shot Policies (DreamZero). arXiv:2602.15922. https://arxiv.org/abs/2602.15922

[3] Fei et al. LIBERO-Plus: In-depth Robustness Analysis of Vision-Language-Action Models. arXiv:2510.13626. https://arxiv.org/abs/2510.13626

[4] Physical Intelligence. openpi LIBERO README and π0.5 checkpoint results. https://github.com/Physical-Intelligence/openpi/blob/main/examples/libero/README.md

[5] Li et al. Causal World Modeling for Robot Control (LingBot-VA). arXiv:2601.21998. https://arxiv.org/abs/2601.21998

[6] Kim et al. Cosmos Policy: Fine-Tuning Video Models for Visuomotor Control and Planning. arXiv:2601.16163. https://arxiv.org/abs/2601.16163

[7] Zhang et al. Do World Action Models Generalize Better than VLAs? A Robustness Study. arXiv:2603.22078. https://arxiv.org/abs/2603.22078

[8] Ye et al. GigaWorld-Policy: An Efficient Action-Centered World-Action Model. arXiv:2603.17240. https://arxiv.org/abs/2603.17240

[9] Zhang et al. DO WORLD ACTION MODELS GENERALIZE BETTER THAN VLAS? A ROBUSTNESS STUDY. arXiv:2603.22078. https://arxiv.org/abs/2603.22078

附录

A. 作者及分工

Dreamzero测评分析：Bowen Jing，Junjie He，Jiajun Lu

FastWAM测评分析：Boyuan Zhang，Jinhao Zhang

Lingbo VA测评分析：Junjie He，Jiajun Lu

VLA模型测评分析：Zijian Zhang

报告整理及分析调研：Jiajun Lu，Yuntian Bo

Project Leader：Weitao Zhou，Haibao Yu

B.libero_plus完整实验结果

Model	Setting	Suite	Total	Camera	Robot	Language	Light	Background	Noise	Layout
DreamZero	RLinf release ckpt	Spatial	68.59	40.37	44.66	90.33	95.55	61.80	63.84	84.65
DreamZero	内部适配	Spatial	62.62	36.41	34.26	69.46	88.35	68.54	70.62	76.47
DreamZero	内部适配	Object	46.95	30.17	22.30	77.16	73.73	53.33	49.88	64.64

DreamZero	内部适配	Goal	28.85	22.57	20.38	29.47	52.69	24.82	25.06	33.02
DreamZero	内部适配	Long	30.15	16.03	17.12	49.48	39.05	21.28	28.50	44.93
DreamZero	RLinf续训	Spatial	69.04	38.78	42.13	89.31	95.20	69.66	67.23	84.14
DreamZero	RLinf续训	Object	56.44	36.15	17.64	90.81	86.53	45.10	56.67	69.73
DreamZero	RLinf续训	Goal	50.38	40.29	34.77	58.69	81.36	38.46	53.26	52.42
DreamZero	RLinf续训	Long	41.72	8.96	29.28	76.03	64.23	25.67	33.98	66.14
Fast-WAM	Action-only inference	Spatial	59.60	27.10	41.70	79.20	96.20	74.80	35.90	71.40
Fast-WAM	Action-only inference	Object	73.70	45.50	59.00	90.70	86.20	80.20	82.50	78.90
Fast-WAM	Action-only inference	Goal	36.90	9.60	24.00	57.80	30.20	30.20	16.60	51.80
Fast-WAM	Action-only inference	Long	45.90	20.30	46.80	67.80	34.20	34.30	39.20	65.10

Fast-WAM-Joint	Joint video-action inference	Spatial	70.00	46.00	68.60	98.50	95.20	67.10	24.50	90.40
Fast-WAM-Joint	Joint video-action inference	Object	63.70	36.60	47.20	98.90	97.00	52.80	48.10	74.40
Fast-WAM-Joint	Joint video-action inference	Goal	44.60	26.50	35.00	71.00	87.80	31.70	15.80	51.50
Fast-WAM-Joint	Joint video-action inference	Long	58.90	27.90	71.80	89.00	79.20	31.10	41.00	81.10
LingBot-VA	官方 ckpt	Long	74.32	44.15	84.48	92.95	80.66	73.36	65.48	87.18

C.训练进展曲线

图 B1. π0.5 训练进展曲线。

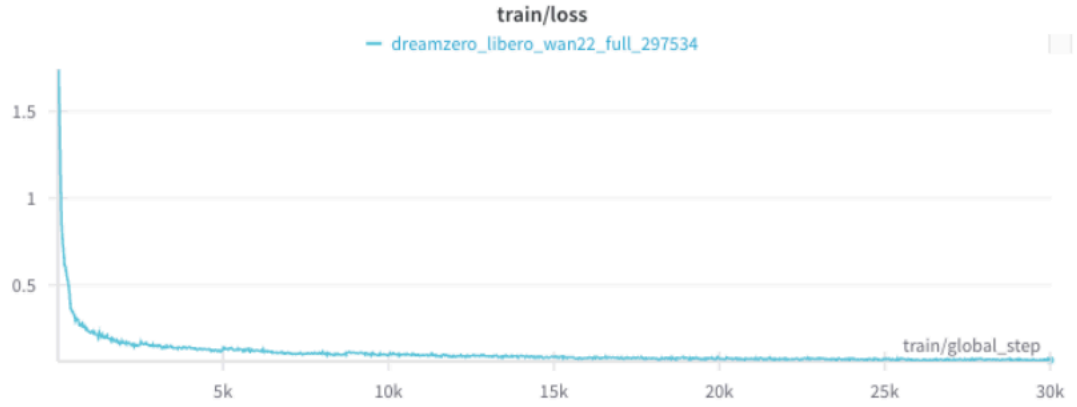


图 B2. DreamZero 训练进展曲线。

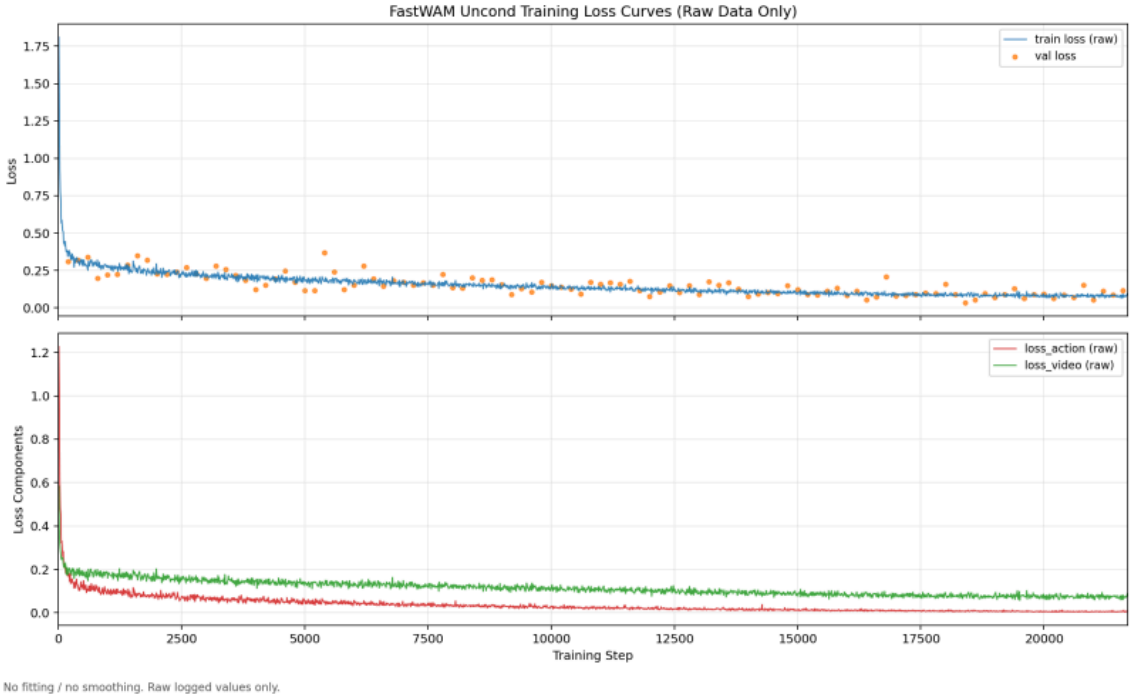


图 B3. Fast-WAM 训练进展曲线。

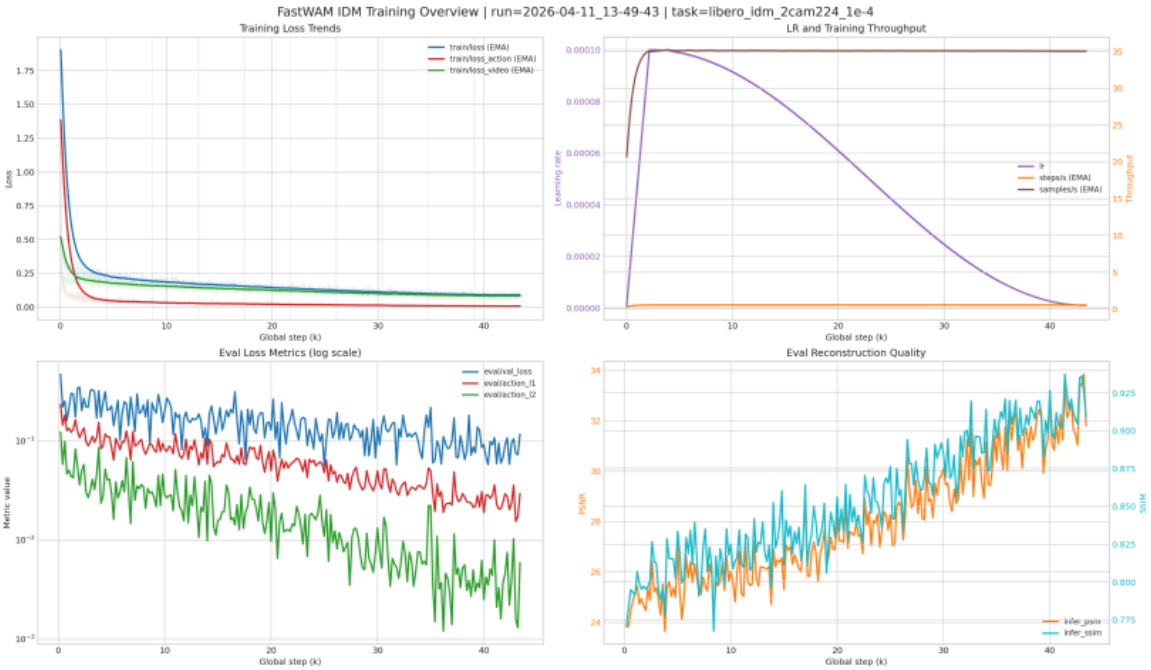
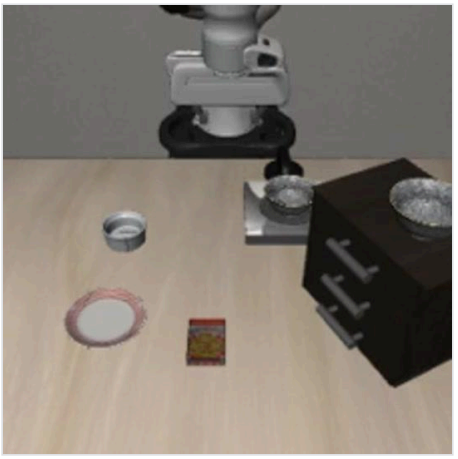


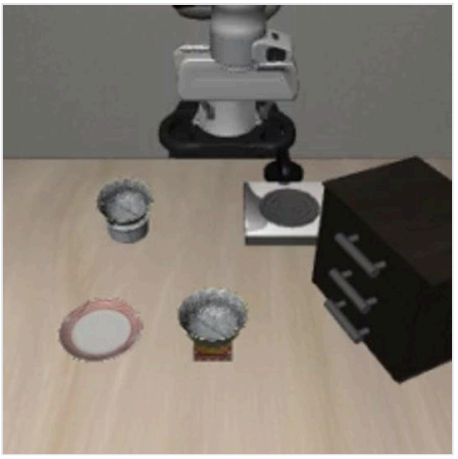
图 B4. Fast-WAM-IDM 训练进展曲线。

D. 典型Faliure Case

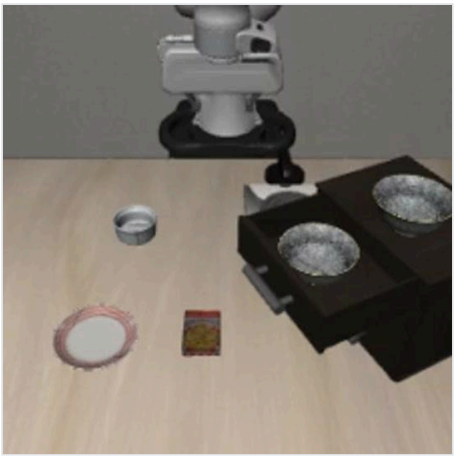
- PI0.5：
  - 最常见：似乎放到了正确的位置，但是在这个基础上又在反复触碰 （不是很确定是哪一类，暂时归为了action）



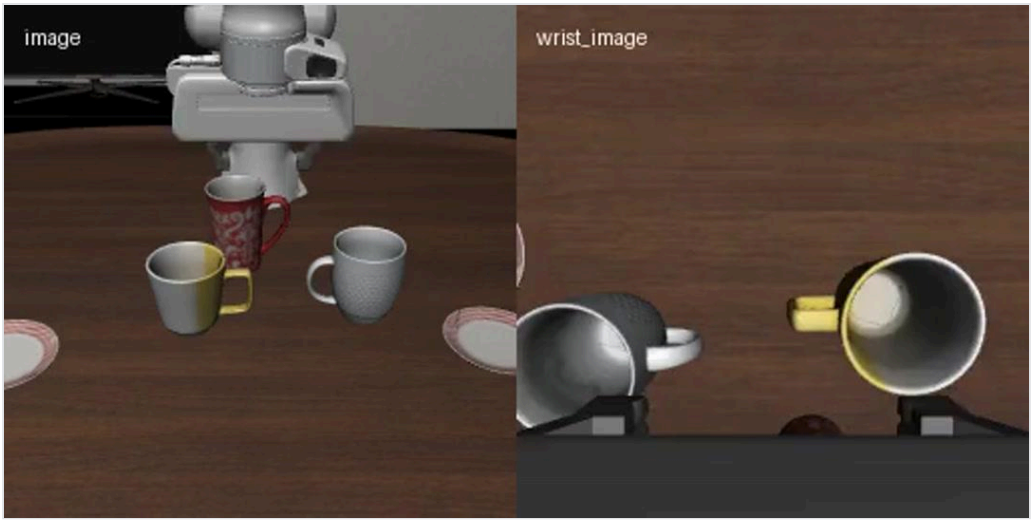
  - 另一类常见：抓取两个物体 or 仿真环境问题？ （不是很确定是哪一类，暂时归为了action）



- 其余错误都是很少的几个个例，
- 在目标附近反复尝试扎抓取

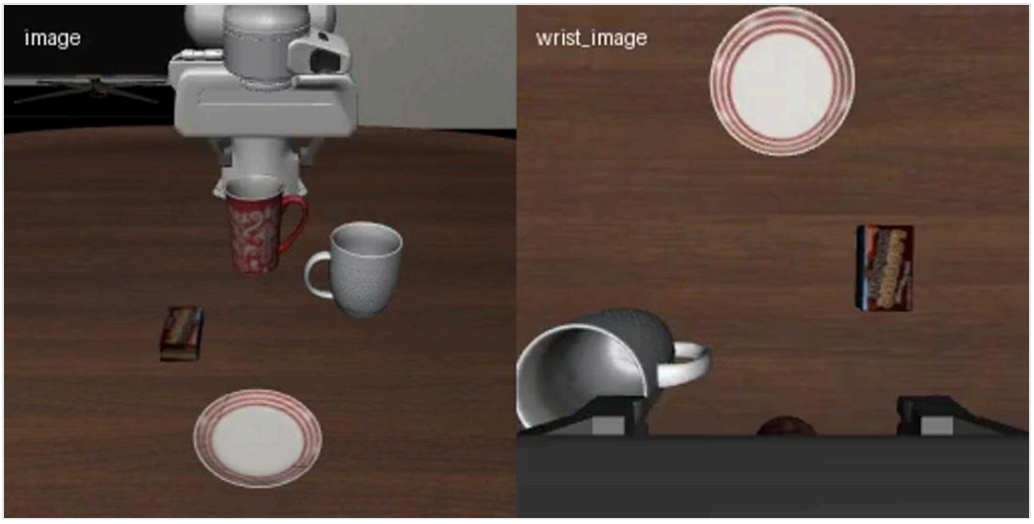


- FastWAM： 几乎只有两类情况
  - 机械臂能准确完成动作，但是在结束回收的时候将物体碰倒 （类似的情况分类为action）

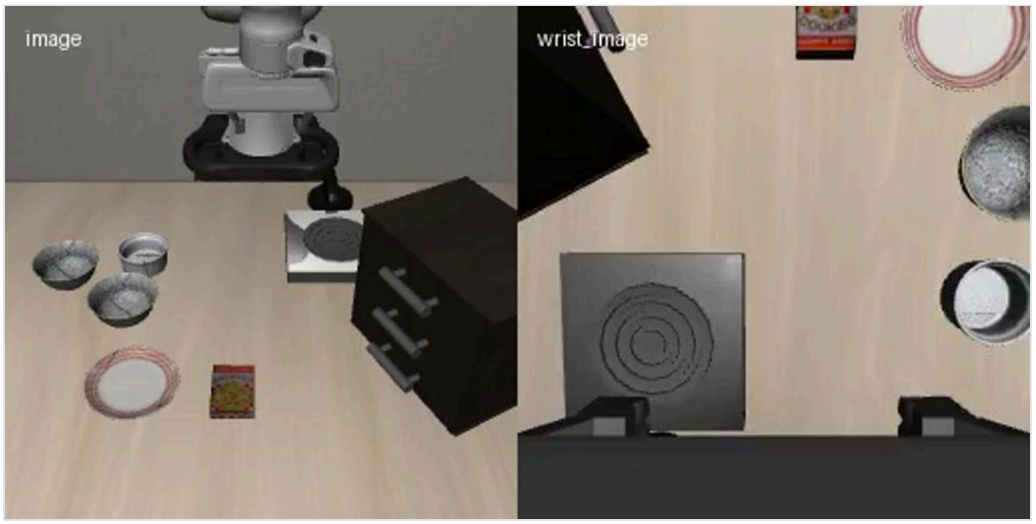


- 两个动作中只完成一个

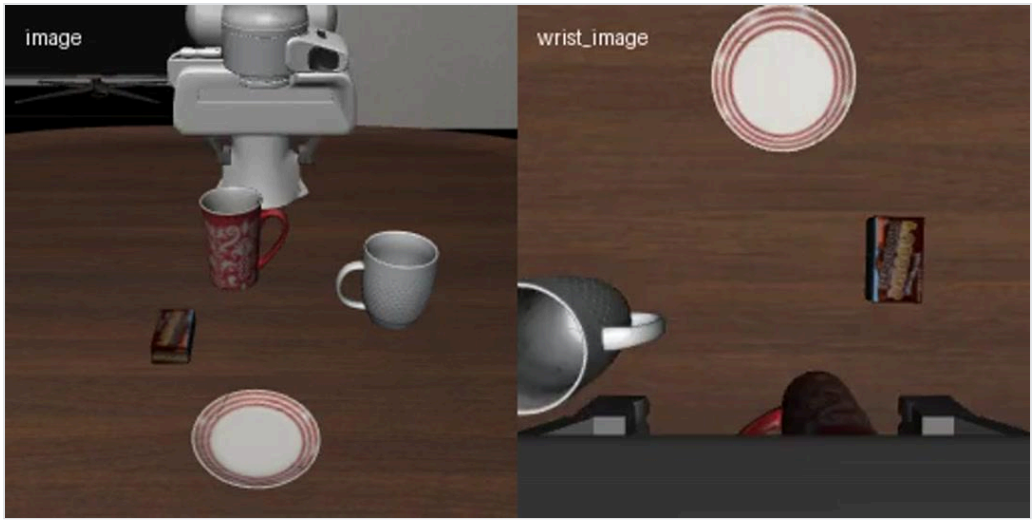
- 目标周围晃动但是不能接触到目标 （分类为timeout）

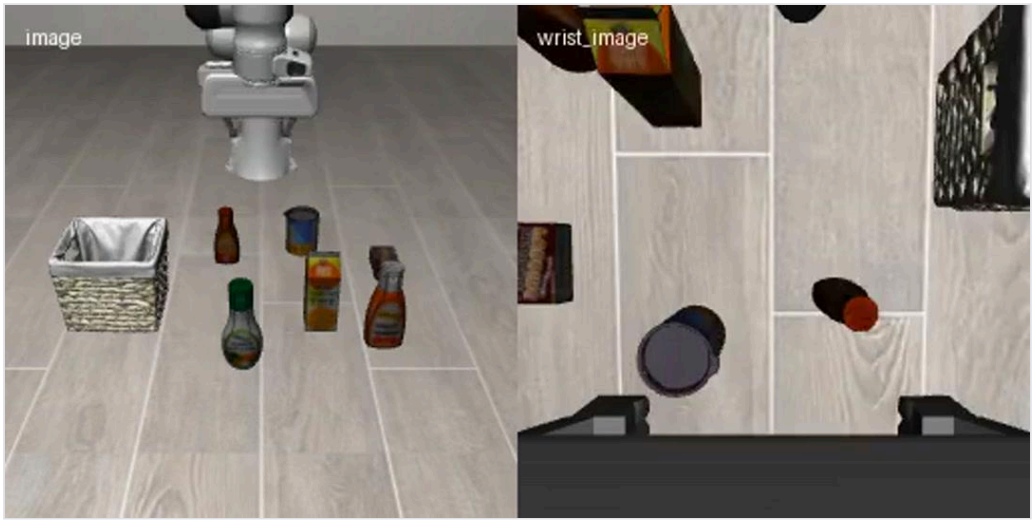


- FastWAM-Joint：
  - 基本上所有的问题都是FastWAM问题的子集，以下是常见的两类
  - 操作没做到（比如pick up.... And put..... 但是没pick up到就进入下一步了）

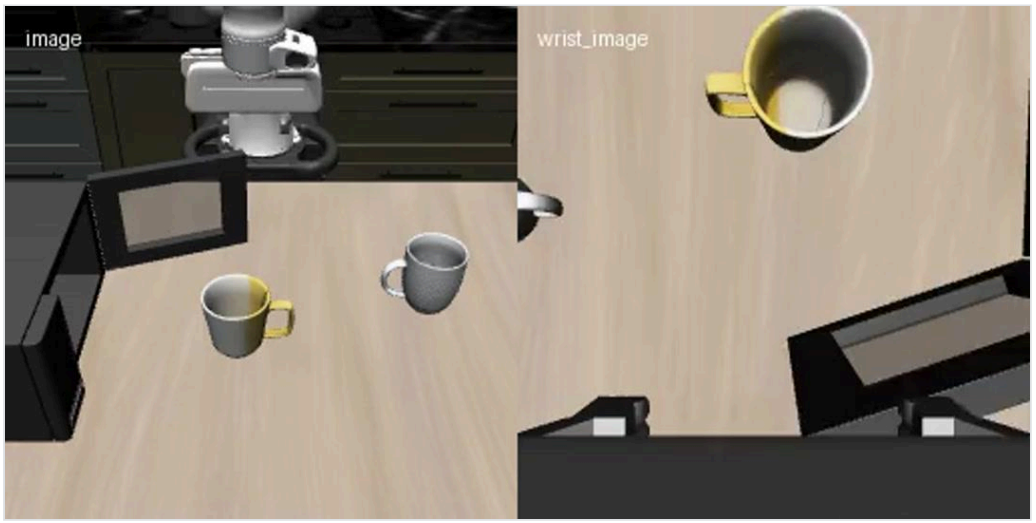


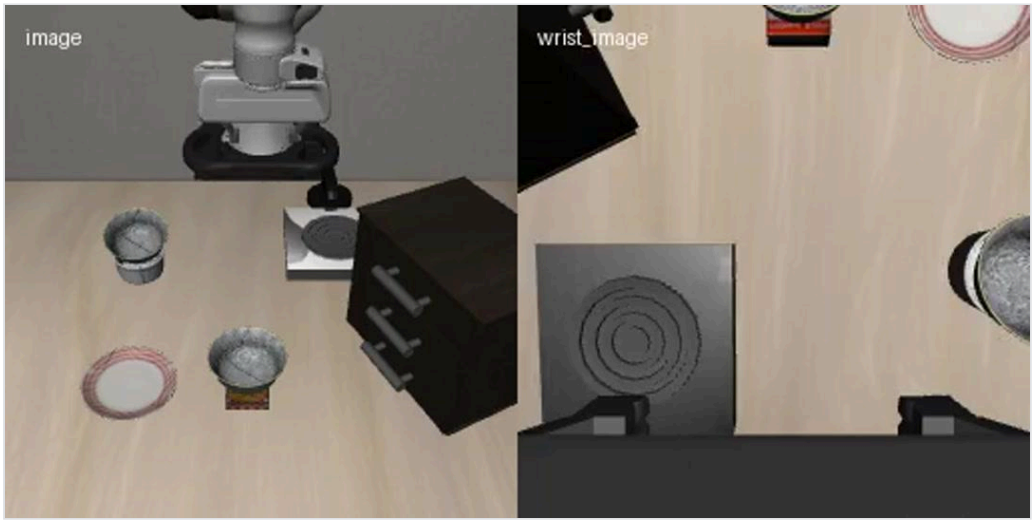
  - 目标物附近抖动或者在一个动作卡住





- FastWAM-IDM：



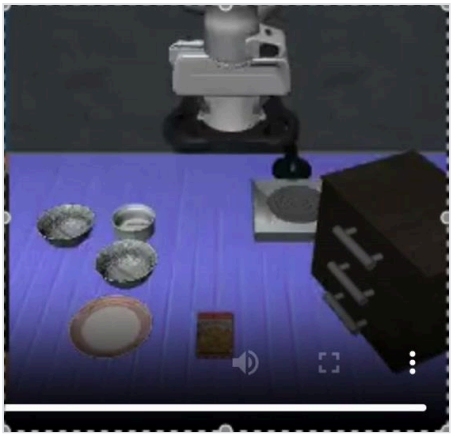


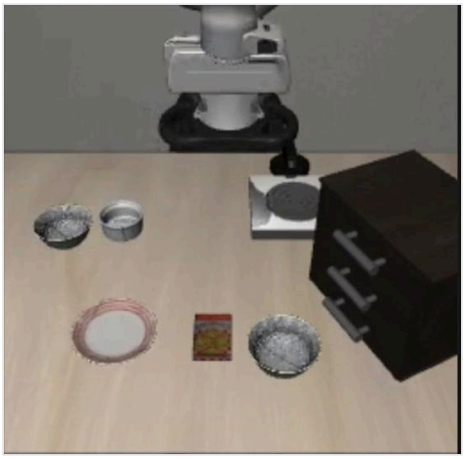
FastWAM 三个变体的failure case对应的任务基本上一致，出现的问题也基本一致，joint和IDM 出错少但类型没有什么变化。主要问题：

在目标物周围抖动，无法成功执行动作

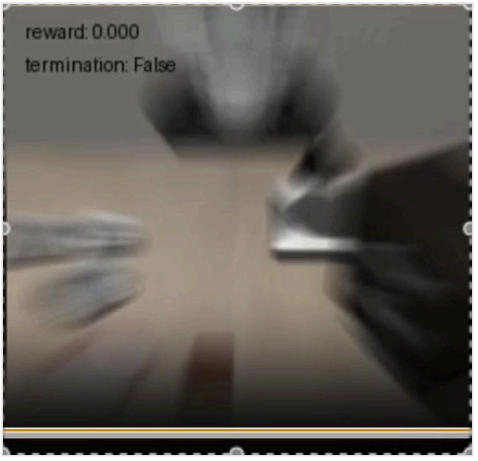
执行一次抓取但是没抓到，进入下一个动作

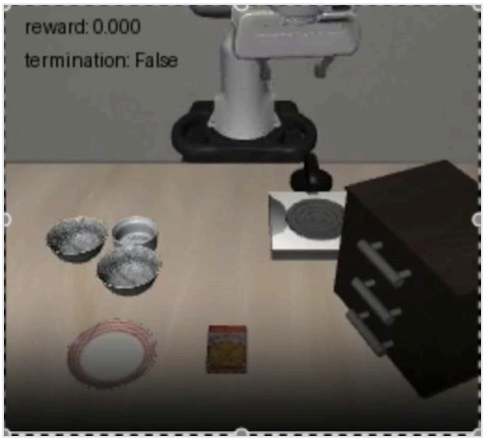
- Dreamzero （libero_plus)：






reward: 0.000
termination: False





E. Failure Case视频分析、原因溯源

	Success Case数量	Failure Case数量	**Perception failure**	**Action error**	**Recovery failure**	**Timeout**	**抖动**
PI0.5	2102	61	4	43	2	10	

	Failure Case数量	**Perception Failure**	**Action Error**	**Recovery Failure**	**Timeout**	**抖动**

FastWAM	68	1	36	2	28	0
FastWAM-Joint	35	0	16	0	19	0
FastWAM-IDM	39	0	13	2	24	0

F.Libero-plus测试备注

Fastwam 原版 评测时间8卡4090 6-7个小时 总体54%

Fastwam-Joint 评测时间8卡4090 差不多12个小时 总体59%

fwalign是原版，fwjoint是joint denoise

Cosmos-policy预训练权重：nvidia/Cosmos-Predict2-2B-Video2World