

基于 Newton 物理引擎与 Claude VLA 的具身智能交互演示

答辩报告

张培搏 学号 24121213320

第十六组

2026 年 6 月

目录

- ① 动机与目标
- ② 技术栈
- ③ 系统架构
- ④ 三种交互模式
- ⑤ 工业模式：双臂并行
- ⑥ v0.2.0：真实物理 · 双臂协作 · 物理课
- ⑦ 视觉反馈设计
- ⑧ 测试与评估
- ⑨ 工程教训
- ⑩ 总结与展望

为什么做这个 Demo?

大语言模型已经无处不在，但“身体”仍然是缺失的最后一公里。

- 学生很难直观感受“经典控制” vs “学习驱动”的分野
- 难以亲手验证大模型在闭环物理世界中的实际表现
- 大多数 VLA 教学需要 GPU 服务器或云端 API

设计目标

- 3 分钟 课堂级现场演示
- 零云依赖：MacBook CPU-only
- 60 fps 全屏渲染
- 毫秒级首响应：观众输入后机械臂瞬时启动
- 全自动回归：所有关键路径有测试

技术栈

物理与渲染

- NVIDIA **Newton** XPBD 物理引擎
- pygame 2D 草图风格渲染
- 闭式 3 自由度 IK
- Min-jerk + back-ease-out 缓动

人工智能与人机接口

- Claude CLI (`claude --print`) 作 VLA 大脑
- 关键词 + bigram LM 模糊匹配回退
- Google Web Speech 语音识别
- 中英双语关键词解析

Newton XPBD

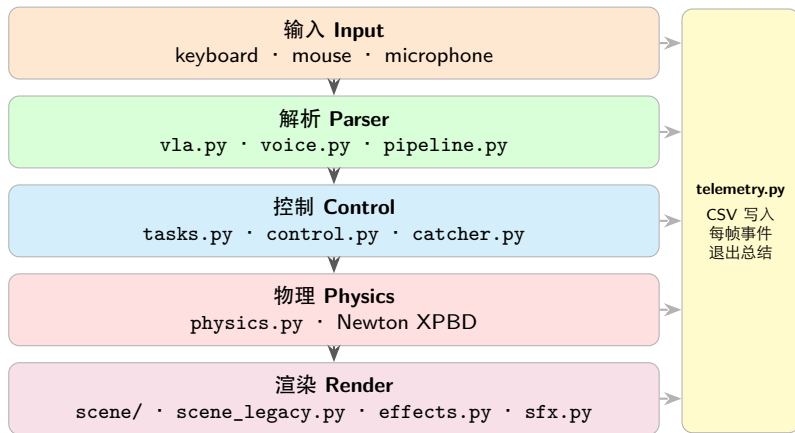
pygame UI

Claude CLI

Google STT

本机 CPU, 无 GPU, 无云推理

总体架构 (5 层)



5 层等宽圆角块自上而下严格对齐 → 数据流单向。Telemetry (黄) 横切监听每层关键事件。

模块清单

模块	行数	职责
__main__.py	1358	pygame 主循环、事件分发、模式切换
scene/arm.py	675	工业风机械臂
scene_legacy.py	671	课堂白板渲染
physics.py	659	Newton 世界、铰接臂、 真实刚体方块
voice.py	512	麦克风、Google STT、模糊匹配
vla.py	467	Claude CLI + 关键词回退
tasks.py	429	pick/place/stack/gesture
catcher.py	271	MPC + 弹道解算
control.py	214	PD slew + NaN 拒收
experiment.py	194	偏移塔稳定性物理课
collab.py	167	双臂协作搭塔接力
telemetry.py	163	CSV 写入 + 公式注入防护
ik.py	91	闭式 3-DOF IK
合计 (22 模块, 实代码)	7653	粗体为 v0.2.0 新增/扩展

交互模式 1/3: 接球 MPC

按 1 → 拖鼠标抛球

- 每帧实时拟合弹道: $z(t) = z_0 + v_z t - \frac{1}{2}gt^2$
- 求 $z(t) = z^*$ 的较大正根 → 截击时刻 t^*
- 闭式 IK 解算关节角, 提交到控制器
- “Commit” 瞬间末端执行器画琥珀光环 (audience 看到决策点)
- 实战命中率 62%–82%

判别式 $\Delta < 0$ 时退出当前预测并打一次性警告。

为什么是经典控制

- 闭式解, 零优化器
- 每帧 1 ms 内完成
- 弹道是高中物理, 可预测
- 对比 VLA 突出“学习驱动”的代价

交互模式 2/3: 自然语言驱动 (VLA)

按 2 → 输入命令 → Enter

Claude 把 “pizza tower” 解析成...

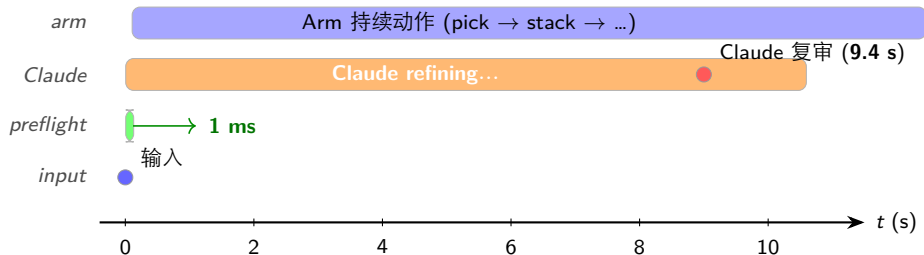
```
{"action": "stack", "color": null,  
  "colors": ["red", "green", "blue"],  
  "target": null,  
  "reason": "build a tower with default RGB"}
```

支持的命令

- pick up the red block / grab green / 拿起红色
- put it on the left / place at x=0.5
- stack red green and blue / build a tower / 搭个塔
- drive right / drive to red
- wave / say hi / 挥手 (装饰手势)
- go home / 回位

核心创新：混合 VLA 解析管线

问题：Claude 延迟 2–10 s，远超 60 fps 帧预算。



- 1 关键词 `_keyword_fallback` 同步运行 (~ 1 ms) → 机械臂立刻动
- 2 Claude 后台 daemon thread 跑 `claude --print` (~ 9 s)
- 3 Claude 返回后比较 preflight 结果：相同 → 静默确认，不同 → 打日志
- 4 generation counter 防止旧线程结果覆盖新命令

效果：观众感知的命令响应延迟从 9.4 s → 1 ms。

交互模式 3/3: 手势装饰

按 2 输入 “wave” / “dance” / “bow” / “point at the audience”

- make_wave: 手臂上举 + 左右扫
- make_point: 左/右/audience 三方向
- make_bow: 前倾下俯保持后归位
- make_dance: 4 拍方框轨迹

为什么不只是抓物?

教学叙事需要“表情”

—手势让机器人看起来有意图，而不只是一台精确但冷漠的设备。

修过的一个 **CRITICAL bug**: 手势的 preflight 白名单原本漏掉了 wave/point/bow/dance, 导致所有手势必须等 Claude 9.4 s 才能触发。修复后瞬时响应。

point 的方向参数原来被“颜色过滤”强制过滤为合法颜色，导致所有 point 都退化为“left”。也修了。

Arm B: 固定底座工件搬运

`--industrial`: 第二只机械臂 Arm B 固定在右侧支架上 (FK-only, 不进 Newton 物理)。

- 与 Arm A 共享 `world.blocks` (pick 互斥)
- 拥有专属 “workpiece” 灰色方块
- 空闲时循环搬运工件 (基础行为)
- 与 Arm A 完全并行: 你按 1 抛球时它仍在搬运

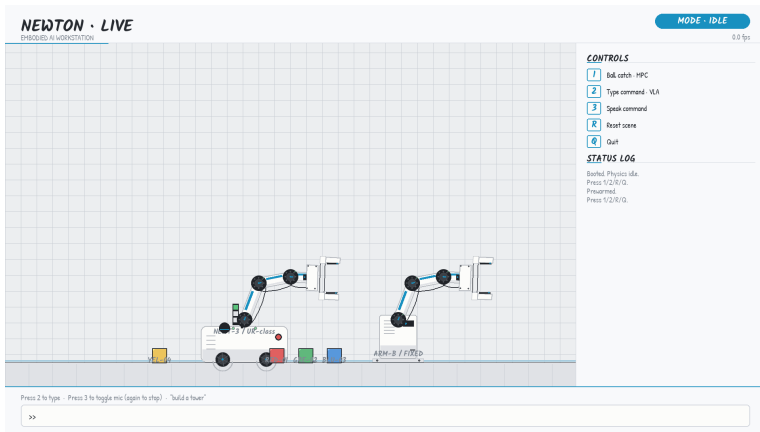
v0.2.0: 空闲搬运已升级为 `--collab` 双臂协作搭塔 (见后)。

```
def _ensure_reachable(self, world_xz):  
    if self.anchor_world_x is not None:  
        return [] # fixed-base arm: no drive  
    ...
```

修过的一个潜在 bug

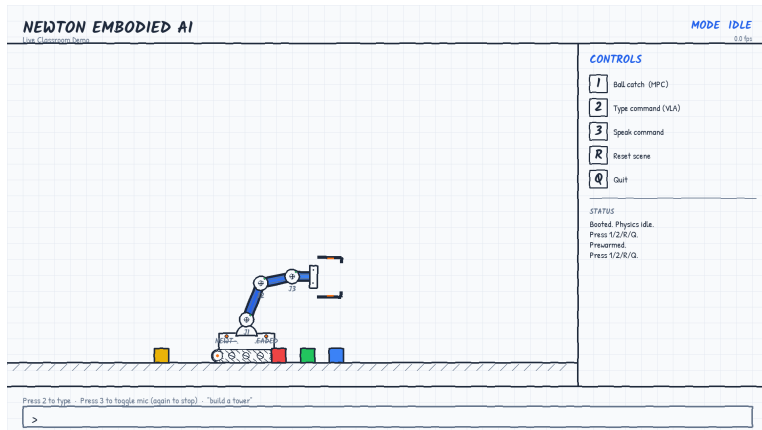
原 `_ensure_reachable` 给任何臂都派 `world.drive_to(...)`,
—这会把 Arm A 的底盘拖来拖去。
—修: 固定臂 (anchor 已设) 跳过 `drive`。

演示截图 1/5: IDLE 工业模式



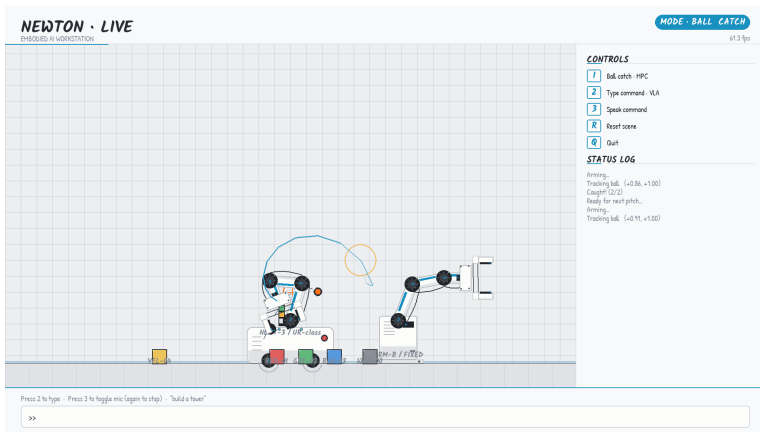
左：移动底盘 Arm A（学生命令驱动）。右：固定 Arm B 自主搬运 workpiece（基础工业模式；--collab 时升级为协作搭塔）。5 个方块、控制面板、状态日志一览。

演示截图 2/5: IDLE 课堂模式



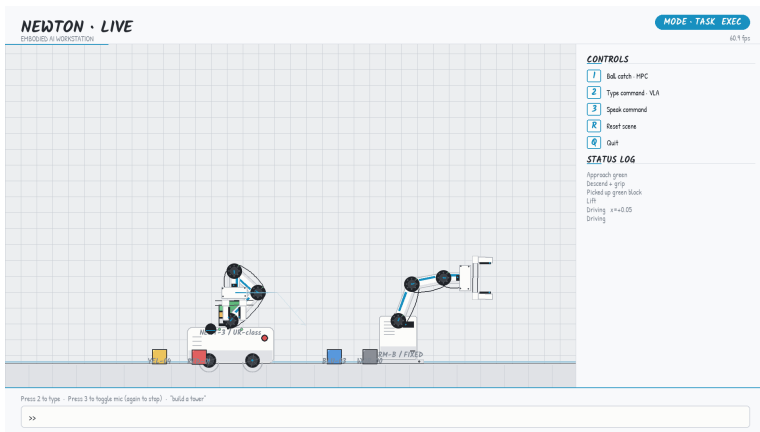
默认课堂白板风。手绘草图机械臂 + 4 教学色方块，模拟黑板演示。无 Arm B。

演示截图 3/5: BALL CATCH 实时追踪



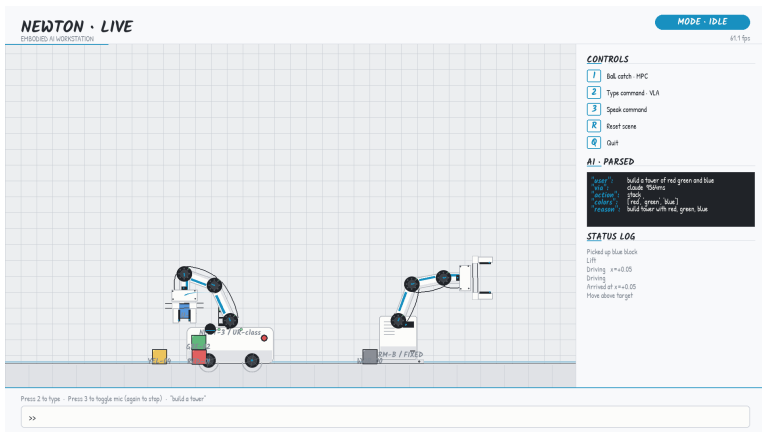
蓝色轨迹：每帧采样小球位置；橙色圆环：`catcher._predict()` 计算的截击点。状态日志显示连续 “Caught! (2/2)”。

演示截图 4/5: TASK EXEC 自主堆塔



Arm A 持绿块向左前行，红块已就位。status log 完整呈现 Approach → Descend + grip → Picked up → Lift → Driving 五步流水。

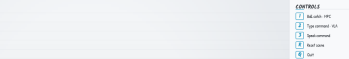
演示截图 5/5: VLA AI · PARSED 侧栏



用户输入 "build a tower of red green and blue" → Claude 后端 9564 ms 返回 → 解析为 stack [red,green,blue]。塔已堆好红 + 绿，正持蓝块行进。Claude 在想什么观众一目了然。

NEWTON - LIVE

https://www.newtonlive.com



CONTROLS

1. Ball catch - MPC
2. Type command - VCA
3. Speed command
4. Read status
5. Run

STATUS LOG

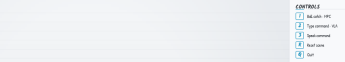
Assembly
Training log: 14:01:10:00
Gripper Pick!
Ready for next pick.
Assembly
Training log: 14:01:10:00

NEWTON - LIVE

NEWTON-1000

MODE - TALK TO MAN

3x1.1 Hz



VELOCITY

CONTROL

- 1 Ball catch - MPC
- 2 Type command - GA
- 3 Speed command
- 4 Read zone
- 5 Test

RECENT COMMANDS

uploading the methods

x=100 y=100

pick item

STATUS LOG

Grasp item at x=100 y=100

Grasping... item at 100

Arm moved at x=100 y=100

Approaching item

Grasped a grip

Press 2 to begin... Press 3 to begin and begin to stop! "Select item"

#HALYZING.

Press 2 to stop - Press 3 to toggle into logic to stop - Toggle back

##ALIVE##

[illegible]

17 / 30

真实刚体方块 `--real-blocks`

默认 **teleport** (Python 解析驱动) $\rightarrow$ `--real-blocks` 让方块成为真正的 **Newton XPBD** 刚体：会堆叠、会碰撞、会倒塌。

难点：XPBD 没有 weld / 焊接约束
(equality 约束是 MuJoCo 求解器专属)。怎么让夹爪“抓住”方块？

KINEMATIC $\leftrightarrow$ DYNAMIC 切换

- 抓住：grab_block 把 body 设 `BodyFlags.KINEMATIC`，每帧把它的位姿写成夹爪 TCP（求解器原样透传）
- 放下：设回 `DYNAMIC`，在重力下落到下方物体上
- 切换后必须 `notify_model_changed` 让 XPBD 重算 kinematic 集

工程细节

- 碰撞过滤：块 $\leftrightarrow$ 块、块 $\leftrightarrow$ 地 开；臂 $\leftrightarrow$ 块、球 $\leftrightarrow$ 块 关（臂用 kinematic 抓取，物理接触只会和 PD 控制器打架）
- `_held_body_ids` 防双抓 / 防空放
- OOB 恢复跳过手持块（避免抖动）
- 20 cm 立方体，`iterations=8`，`substeps=2`，`relaxation=0.8`

双臂协作搭塔 --collab

舞台空闲 3 s 后自动开演：两臂接力搭塔，是”恰好是机器人协作的屏保”。

接力流水线 (CollaborativeBuild)

- 1 Arm A（移动）取色块 → 放到 交接槽 $x = 1.50$
- 2 Arm B（固定）从交接槽取走 → 堆到塔列 $x = 1.70$
- 3 红 → 绿 → 蓝自底向上；admire 2.5 s
- 4 角色互换拆塔（自顶向下，绝不抽承重块）
- 5 归位 → 循环

安全联锁

单交接槽就是全部互锁：A 只在槽空时投放，B 取走的瞬间释放槽。无显式碰撞几何，靠 时序不对称（B 抬升 ~ 2 s vs A 补给 ~ 10 s）保证两臂永不同时伸向槽。

观众一动（抛球/输入/语音）→ 立即让出双臂。

修过的 bug：collab 曾在任何 --bench 下静默失效（误挂在 Arm B 的 idle 门控上）→ 已给独立门控。

Arm B 物理课：偏移塔稳定性 --experiment

让真实 **XPBD** 当裁判：Arm B 用 3 块灰色工件按逐层偏移 $0 \rightarrow 4 \rightarrow 9$ cm 堆塔，物理判定倒不倒——判据是算出来的，不是脚本写死的。

倒塌判据（高中物理）

半宽 $r = 10$ cm 的等质量立方体，上两层组合重心偏离底块中心 $1.5d$ 。重心铅垂线超出底块支撑面即倒：

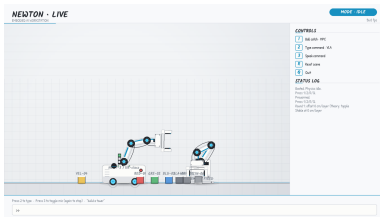
$$1.5d > r \Rightarrow d > \frac{r}{1.5} \approx 6.67 \text{ cm}$$

6.67 是刚体理想上界；真实 XPBD 接触柔度让边界更靠前——实测 $(4, 5]$ cm。选 0/4/9 三档，稳/倒与理论同号。

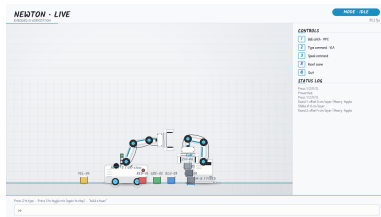
实现

- `toppled()`：任一块 $|\text{angle}| > 0.30$ rad ($\sim 17^\circ$) 或顶块高度跌落 $> r$
- 重心铅垂线 + 支撑括号实时叠加，越界翻琥珀色 + “CoM $\rightarrow$ OUT”
- 向 $-x$ 倒（散块落进空带，自动收拾）
- 隐含 `--real-blocks + --industrial`；与 `--collab` 互斥

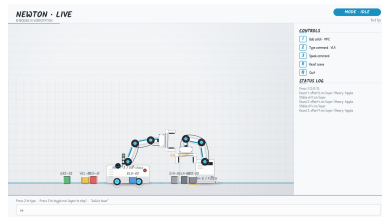
偏移塔实测：Stable $\rightarrow$ Stable $\rightarrow$ TOPPLED



0 cm · 重心正对底座 · 稳



4 cm · CoM 偏 6 cm $< r$ · 稳



9 cm · CoM 偏 13.5 cm · 倒

真求解器回归测试钉死“4 cm 稳 / 5 cm 倒 / 9 cm 倒”的实测包夹。6.67 cm 是刚体理想上界，真实边界更靠前（实测 (4, 5] cm）；这三档稳/倒与理论同号。

戏剧化视觉反馈

接球瞬间

- 琥珀光环 标记决策时刻
- 接到时彩色粒子爆裂
- 1.5 s 慢动作 (0.33×)
- 横幅 “CAUGHT · N”

命令未识别

- 琥珀横幅: “DIDN'T CATCH THAT”
- 输入态 footer 自动切到命令示例
- “miss” 音效

AI 推理迹侧栏

7 行实时展开:

- user: 学生原话
- via: preflight / claude / fallback
- latency: 毫秒数
- action: pick/stack/...
- color / colors
- target: [x, z]
- reason: 自然语言解释

慢动作 gate

只在 !executor.busy 时触发, 避免 pick/stack 中接球导致动作脱钩。

测试矩阵

测试文件	测试数	重点
test_tasks	27	程序构造器 + 手势 + 抓取回调
test_vla_parser	24	关键词 (en + zh)
test_voice_fuzzy	21	噪声转录用例 (en + zh)
test_render_smoke	21	双渲染路径全部 draw_*
test_pipeline	20	所有 action 分支
test_effects	19	粒子 / 光环 / 横幅生命周期
test_catcher	18	弹道反解 + 状态机
test_vla_subprocess	17	Claude CLI mock
test_control	13	PD slew + NaN 拒收
test_telemetry / ik / docs_site	10×3	CSV / IK / 落地页校验
test_real_blocks	9	KINEMATIC 抓取 + 堆叠 + OOB
test_experiment	11	倒塌判据 4 稳/5 倒实测包夹
test_scripted_constants	7	彩排数据
test_scripted_flows	6	端到端流程
test_collab	5	双臂接力 build/teardown/loop
test_display_mode	2	CLI 解析
合计 (18 文件)	250	100% 通过 / ~105 s

FPS 实测 (Apple Silicon, CPU-only, capped@60)

场景 (18 s headless bench)	min	avg	max	样本
IDLE 工业模式	39.1	54.9	277.6	973
Scripted stack	40.4	54.9	478.3	974
Scripted VLA	42.0	53.9	319.4	963
--collab 双臂接力	40.6	54.6	308.2	970
--real-blocks 真刚体	47.7	55.9	134.3	1005
--experiment 偏移塔	40.9	56.5	129.5	1014

读法: avg 被 `clock.tick(60)` 钉在 ~ 55 (+macOS 调度抖动/热), 不是算力瓶颈——**max 277–478** 暴露了真正的余量。真刚体 / 偏移塔的 max 较低, 因为它们要跑真实碰撞管线 (见下页)。

性能优化：把物理从 71% 帧预算压到 ~2%

反直觉发现：原 `world.step()` 吃掉 **11.9 ms** (71% 的 16.7 ms 帧预算)，但 Newton 只模拟 4 个 body。那 12 ms 几乎全是 **Warp kernel** 启动开销 (`substeps × collide` 管线)，不是算力。

三个杠杆

- 1 iterations 20→8、substeps 4→2、sim_dt 1/240→1/120
约束：substeps×sim_dt = 1/60
- 2 teleport 模式复用空 contacts，跳过 per-substep collide（无真实接触时）
- 3 teleport solver iterations 8→2（无人读它的物理臂位姿）

结果

- `step()`: 11.9 ms → **0.30 ms** ($\approx 40\times$)
- uncapped 吞吐: 64 → **279 fps**
- 60 fps 预算下 **4× 余量**
- 250 测试全绿、渲染臂 **0 px** 变化

性能数据为优化迭代中 cProfile 实测（uncapped headless）；当前代码即优化后状态（substeps=2, sim_dt=1/120），其余量由上页 FPS 表 max 列（277–478）独立佐证。关键洞察：渲染臂由控制器 FK（`arm_state_from_target`）驱动，不读 solver——所以降迭代对画面零影响。--real-blocks 保留完整 collide（堆叠需要接触）。

实战演示数据

日期	catch	VLA	备注
05-21 12:05	7/10	0	首次现场
05-21 12:30	8/13	1	Claude 9.4 s 解析 “pizza tower”
05-21 17:42	14/17	0	30 s 密集投球

典型 telemetry CSV 节选

elapsed	event	detail
1.96	mode	ball_catch
4.48	catch	1/2
6.45	catch	2/3
...	...	...
30.10	catch	14/17

工程教训：发现与修复的 bug

第三方专业评审找出 13 个问题，全数闭环

- **CRITICAL** preflight 白名单漏手势 → 手势必须等 Claude 9.4 s
- **CRITICAL** point 方向被颜色过滤吃掉 → 永远退化为 left
- **HIGH** slow-mo 在 pick/stack 中触发 → 动作脱钩
- **HIGH** parse_thread 旧线程结果覆盖新命令
- **HIGH** test_scripted_catch 抗 flake 不足
- **MEDIUM** voice socket.setdefaulttimeout 进程级污染
- **MEDIUM** Arm B 在 rehearsal 里重复 pipeline 逻辑
- **MEDIUM** CSV =cmd Excel 公式注入
- **MEDIUM** pipeline.build_plan_from 无单测
- **MEDIUM** telemetry 类型注解不准
- **LOW** $O(n^2)$ list.index() 块越界恢复
- **LOW** launch_ball 速度无上限
- **LOW** 死代码桩 _request_exit/_finish

发现的隐藏 bug：固定臂 _ensure_reachable 误驱动主臂底盘。

代码进化: v0.1.0 → v0.2.0

轮次	重点
A-D	测试安全网、scene 拆分、表现力、4 新功能
R-S	13 个评审修复 (2 CRITICAL) + 抛光
T-V	effects/control/catcher/VLA mock 测试
W-Y	Arm B 构造收敛、空闲循环、修固定臂 drive bug
v0.2.0 (提升技术上限)	
性能	三杠杆: 物理 11.9 ms → 0.30 ms ($\approx 40\times$)
真物理	--real-blocks KINEMATIC 抓取 + 真实堆叠/倒塌
协作	--collab 双臂接力搭塔 (多智能体编排)
物理课	--experiment 偏移塔倒塌 (理论 vs 仿真)
落地页	GitHub Pages + 校验测试 + 设计报告/slides
合计	测试 41 → 250 、零回归、上线演示站

已完成 vs 局限

已完成 ✓

- 三模式交互 (catch / VLA / gesture)
- 混合 VLA 解析 (preflight + Claude)
- 真实刚体物理 (--real-blocks)
- 双臂协作搭塔 (--collab)
- 偏移塔物理课 (--experiment)
- 物理 $\approx 40\times$ 优化 (11.9 $\rightarrow$ 0.30 ms)
- 250 测试, 稳定 60 fps, 上线演示站

未来工作

- 1 AppState dataclass 重构主循环 $\rightarrow$ `__main__.py` ≤ 200 行
- 2 --collab --real-blocks 跨周期防漂移 (释放阻尼 / 投放微调)
- 3 完整逐帧 snapshot replay (D.3 原版)
- 4 多语言扩展 (日 / 韩 / 西)

局限 ●

- `__main__.py` 1358 行 > 800 上限, 需 AppState 重构
- --collab --real-blocks 跨周期会漂移, 演出用 teleport 接力
- 慢动作是简化版, 非逐帧 replay
- 语音仅 zh-CN / en-US
- 接球 MPC 单步式无前瞻

Q & A

Demo 入口（规范仓库根 Makefile）：

```
make demo / industrial / collab / experiment
```

测试入口：

```
make test    （完整 250 测试，需兄弟 ../newton）
```

```
make test-ci （101 个 no-newton 子集）
```

在线：hollis36.github.io/newton-vla-demo

release: v0.2.0 （GitHub tag + release）