■ 从生成到创作 · AIGC RL 的 Agent 化跃迁

Hunyuan UniRL

面向统一多模态模型的分布式 RL 后训练框架

王淏楠 · **Principal Research Scientist**

多模态强化学习团队 / 腾讯 · 混元

通用观察 · THE SHIFT

AI 的交互方式，从「回答问题」到「完成任务」

变化的不只是模型会调用工具，而是交互形态本身

Chatbot	人给 prompt，模型给 response —— 单步问答
Collaboration	人在关键节点 review · comment · accept · reject
Agent Loop	Goal 模式：人给 done condition，系统 plan · act · observe · revise

模型解决单步生成，Agent 组织多步流程

通用观察 · WHY NOW

为什么 Agent，总在模型成熟之后出现？

CODING

代码模型能稳定补全、修 bug、读项目

↓

Cursor / Coding Agent 出现

↓

从「写一段代码」→「完成一次工程修改」

AIGC

图像视频模型能稳定生成高质量内容

↓

Agentic Image / Video 出现

↓

从「生成素材」→「完成一个创作」

Seedance 的意义不是又一个视频模型，而是让视频生成跨过了 Agent 化所需的可用性阈值

AIGC AGENT · ANALOGY

视频生成，更像 Cursor

不是 prompt → video 一步到底，而是 goal → 候选 → 选择 / 评论 → 修改 → 组装

CODING AGENT

− 旧实现

```
def get_user(id):
    user = DB.get(id)
    return user
  ...

def logout():
   session.clear()
 ....
```

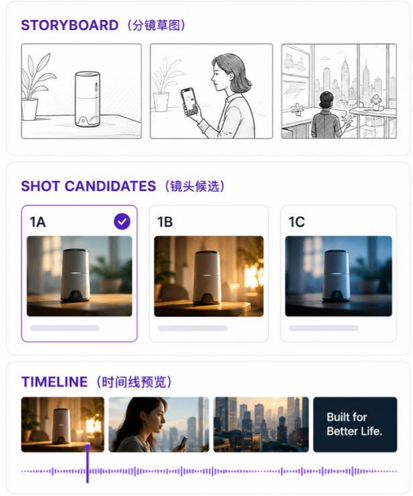
+ 新实现

```
def get_user(id):
    user = DB.find_user(id)
    if not user:
        raise NotFound()
    return user
  ...

def logout():
    session.invalidate()
  ...
```

plan · code diff · test → accept / reject / comment / edit

VIDEO AGENT



storyboard · 候选镜头 · timeline → 选 1/2/3 · comment · 改某镜头

好的 Creative Agent 不是一次性替你创作 —— 而是在关键节点，把创作权交还给你

反馈即信号 · FEEDBACK

每一次创作反馈，都是训练信号

生成是一次 sample，创作是一段 trajectory —— 中间反馈不是 UI 事件，而是训练数据

用户的协作动作		训练信号
选 1 / 2 / 3 候选	→	preference signal
写 comment	→	自然语言奖励 / 编辑指令
拒绝某个镜头	→	negative sample
重写分镜	→	planner correction
要求重生成	→	process feedback
采用最终版本	→	outcome reward

把这些多模态反馈，转成 RL 后训练数据

发展阶段 · THREE STAGES

三个发展阶段

1.0

单次生成

prompt → 内容

画质 · 真实感 · 指令遵循

2.0

创作编排

LLM / VLM 搜索 · 规划 · 分镜 · 评审 · 修改，生成模型作为工具被调用

近期代表

Codex · Record & Replay

Gen-Searcher · CutClaw · Co-Director

3.0 正在走向

可训练创作闭环

把 plan · generate · critique · revise 形成 trajectory，让反馈进入 RL 后训练

UniRL 要回答的问题

Workflow 让 AIGC 开始像 Agent；RL 后训练让它有机会从反馈中改进

从编排到学习 · WORKFLOW → LEARNING

Workflow 让系统跑起来，RL 后训练让系统学起来

WORKFLOW 已经解决

把创作流程跑起来

planner　generator　critic　editor

✓ 已经能做出 agentic image / video 的 demo

但要从反馈中学习，还留下三个训练问题 ——

01 Feedback 停在 UI 层

选 1/2/3、写 comment、reject 某镜头，通常只影响下一次调用，不一定进入模型更新

02 Credit assignment 没被处理

视频变好，是 planner 改对、某镜头重采样对、还是 critic 选得好？能记录过程，不一定能学习归因

03 Scaling cost 变成系统问题

creative loop 是多轮生成 · 评审 · 修改；rollout · reward · weight sync · 分布式调度都成核心工程

UniRL 要回答的：当 workflow 产生大量多模态反馈，怎么把它变成可训练的 RL 后训练闭环

HUNYUAN · UNIRL

UniRL

为 Agentic AIGC，做一套通用的 RL 基础设施

出发点 · THE ASSUMPTION

现有 RL 框架，都建立在一个隐含假设上

隐含假设

一条 rollout 轨迹 = 一串离散 token ≈ KB

`Driver · single-controller`

所有 rollout 数据汇聚到 driver，毫无压力 —— `verl / slime` 正是这么设计

假设成立的范围

✓ **LLM**

Text → Text，本身就是 token 序列

✓ **VLM**

图像 → ViT → token，仍是 token 序列

✓ **通信量小**

KB 级，甚至可以走普通网络

在「轨迹 = token」的世界里，single-controller 不是缺陷，是最简洁、最正确的设计

转折 · THE BREAK

多模态生成，换掉了「轨迹」的定义

自回归 · AR

~ KB

TextSegment

`tokens [total_tokens]`

■ → ■ → ■ → ■

一串离散 token，逐个采样；logP 是离散分布的对数概率。

VS

扩散 · DIFFUSION

MB → GB

LatentSegment

`latents [N, K, C, T, H, W]`

x_T → ▒ → ▒ → ▒ → x_0

一条 K 步去噪 latent 轨迹；logP 来自每步的高斯 transition。

统一模型还要支持图文交替（interleaved）—— 一条轨迹里同时有 token 段和 latent 段

崩塌① 算法面 · ALGORITHM

ratio 怎么算？logP 与 group 切分都变了

LOGP 不再是 token 对数概率

AR 每个 token 一个离散对数概率

Diff 每个随机去噪步是一个高斯 transition，logP 来自连续高斯

只有 `eta > 0` 的步可训练 → `sde_logp [N, S]`

GROUP 切分

advantage 仍是组内 z-score（同 prompt 的 N 条样本）

`traj·1` `traj·2` ··· `traj·N`

但「一个样本」现在是一整条 K 步轨迹，advantage 要沿轨迹 broadcast 到每个可训练 SDE 步。

前提：rollout 与 train 必须逐位走同一条轨迹（σ schedule · x_T · fp32）

崩塌② 系统面 · COMMUNICATION

通信量从 KB 跳到 GB

token 走 driver 没事；GB 级 video latent 走 driver，driver 就是单点瓶颈 —— image / video 独有，LLM RL 永不会遇到

系统设计 · DATA PLANE

Driver with MetaData：只编排元数据，不搬数据

走 Driver	Ray object store · CPU + pickle	个位数 GB/s · 单点串行 → 瓶颈
NCCL · NVLink	机内 · 点对点	数百 GB/s · 并行
GPUDirect RDMA · IB	跨机 · NDR 400G	数十 GB/s · 卡 · 零拷贝绕 CPU

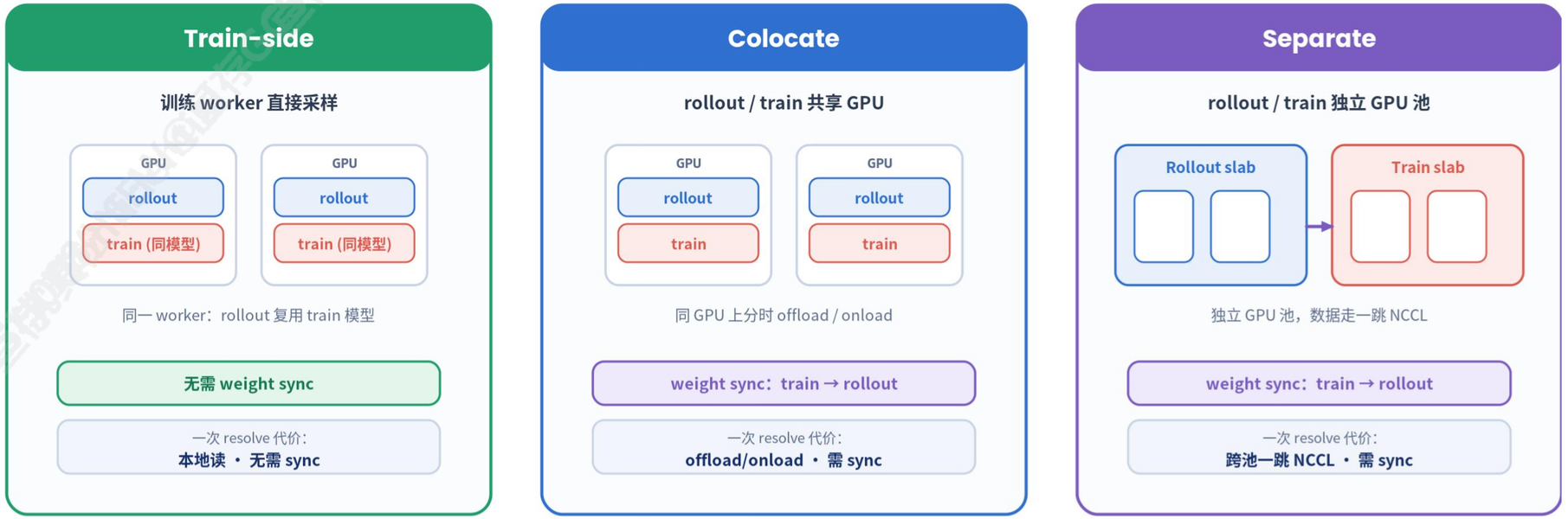
关键不是「快几倍」，而是差一两个数量级 + 消除单点瓶颈

部署拓扑 · TOPOLOGY

一套框架，三种部署拓扑

rollout 与 train 怎么摆 —— Train-side · Colocate · Separate，只有 separate 才有跨池数据传输

同一份代码，三种放置 —— 只有 separate 才有跨池数据传输



架构 · COMPOSABLE

一套 RL 流程，每个模块都可替换

固定的是这套流程的契约；⇄ 处都是可替换的槽 —— 换实现，不改主流程

Rollout ⇄

- Trainside
- SGLang
- vLLM-Omni
- Composed

推理引擎可换

→

Reward ⇄

- local scorers
- remote panel

HPS · CLIP · GenEval2 · VideoAlign

→

Advantage 共同

- group z-score

组内归一，沿轨迹分配（不变）

→

Train ⇄

- GRPO
- DanceGRPO
- DiffusionNFT
- ★ Flow-DPPO
- ★ DRPO

FSDP2 · VeOmni 后端

→

Sync ⇄

- LoRA / Full
- NCCL
- Tensor / IPC

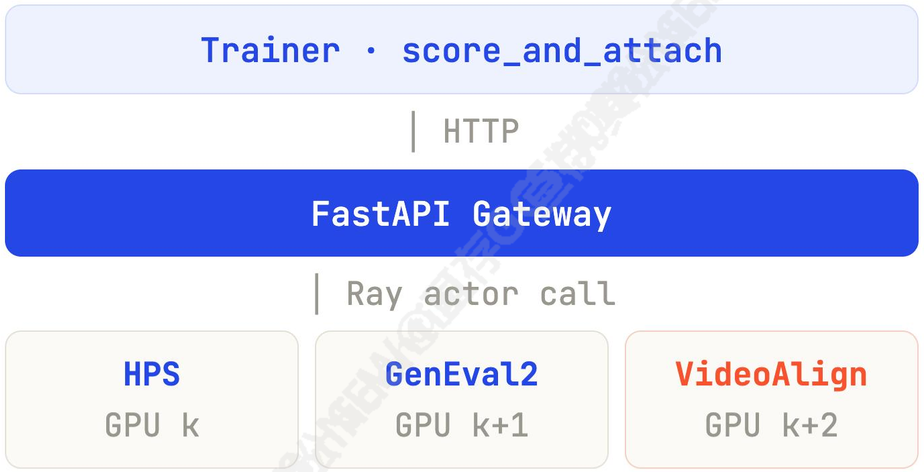
权重同步可换

模型 bundle 也是一个槽 ⇄ SD3 · WAN 2.1/2.2 · HunyuanVideo · Qwen-VL · HunyuanImage3 · Bagel

差异压进可替换组件，正确性留在共同契约 —— 新增第 N+1 个组合，不改全局流程

功能 · REWARD SERVER

多模态 reward 是重模型



- **GPU 资源隔离** — reward model 独占 GPU，互不挤占
- **依赖隔离** — 每个 scorer 独立 venv，不同环境隔离
- **图像 + 视频 judge** — PickScore · Qwen-VL · GenEval2 · HPv3++
- **远程 panel**— 多 reward 加权聚合（weighted / mean / min / max）
- **错误隔离** — 高并发下单 reward 失败不污染同批其它reward

对 trainer 只是 `score_and_attach` — 本地或远程Reward

功能 · PERFORMANCE

效率优化

为多模态，把 rollout · Driver · train 三处重做

01 Rollout

采样又快又对

- **连续批处理** — 扩散 rollout 骑 SGLang，GRPO 整组批进去噪流水线，`forward_batch_size` 控显存
- **rollout 即「校验边界」** — `req.sigmas` 单一真源 + 逐步确定性 SDE 噪声；`bypass-mode ratio` 扛住 train↔infer 数值差
- **SDE 采样核可插拔** — `Flow / Dance / CPS` 等 StepStrategy 自由换

→

02 Driver

只编排 metadata，不搬数据

- **GB latent 不过 driver** — train 直接从 rollout 拿（dehydrate→TensorRef→localize）；跨机 `GPUDirect RDMA / Mooncake` 零拷贝
- **异步重叠 + 连续缓冲** — 生成与训练重叠（常驻引擎 + `max_inflight`）；`group-keyed buffer` + 有界 staleness
- **reward 不抢卡** — 独立 reward slab / 远程服务，独占 GPU，不和 policy 争显存

→

03 Train

更新到 scale

- **双后端 FSDP2** — torch-native `fully_shard` + `VeOmni`；ZeRO-2/3、fp32-master / bf16-compute
- **长序列 / 大模型** — `Ulysses` 序列并行、HSDP、DCP 分片 checkpoint（80B meta-init）
- **省通信** — no-sync 累积 + forward-prefetch 重叠；AR replay 用 `FA2 / flex` 跳过跨样本 attention

`token → latent`，把 rollout / Driver / train 的三套假设全部打破

覆盖范围 · MODEL × ALGORITHM

把 model × algorithm 变成可组合关系

不为每个模型重写一套 RL，而是让模型与算法自由组合

模型覆盖

图像扩散

SD3 / SD3.5 · Qwen-Image · FLUX · …

视频扩散

WAN · HunyuanVideo · LTX-Video · …

AR / 统一模型

Qwen-VL · Qwen3 · HunyuanImage3 · Bagel · …

组合系统 (for agents)

Qwen-VL + SD3.5 / LTX · …

算法覆盖

GRPO　DiffusionNFT　DanceGRPO　MixGRPO　CPPO

★ Flow-DPPO　★ DRPO

Flow-DPPO · 面向视觉生成（扩散 / 流匹配）的自研策略优化

DRPO · 面向长轨迹决策的自研强化优化

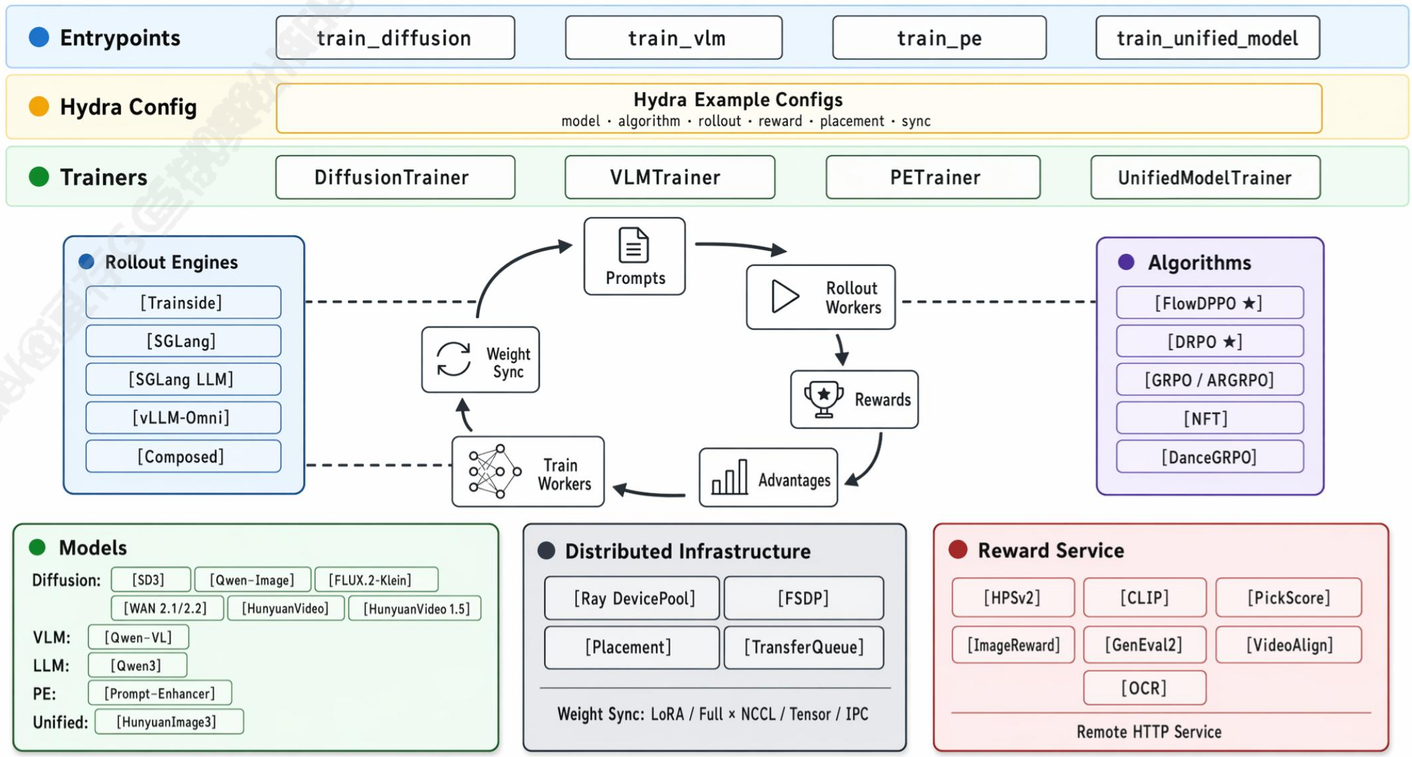
系统能力 · CAPABILITIES

三类核心系统能力

核心能力	技术表述	对 AGENT 化 AIGC 的意义
统一多模态 RL loop	generate → score → advantage → update → sync，覆盖 diffusion · AR · PE · unified 等入口	让图像 · 视频 · 文本 · VLM · AR-Diffusion 在相近流程下迭代
可插拔 rollout / reward / algorithm	rollout engines · reward service · algorithms · domain trainers 解耦	频繁替换 planner · generator · critic · reward，降低实验成本
分布式训练 · 推理协同	Ray runtime · DevicePool · FSDP · Transfer Queue · weight sync 支撑大规模采样与训练	creative loop 成本主要在多轮 rollout 与 reward，系统层必须先稳

系统总览 · PIPELINE

把多模态的 RL 后训练，放进一套流程



Agentic AIGC RL：先统一生成模型的 RL 后训练，再把多步创作过程接入这个闭环

落地路线 · ROADMAP

Creative AIGC Agent · 三阶段路线

1 **生成模型 RL 后训练**

让图像 / 视频生成模型在指令跟随 · 审美 · 一致性 · 可控性上更好

2 **创作模块 RL 后训练**

训练 prompt enhancer · planner · captioner · VLM judge · editor，更懂创作与生成边界

3 **多步创作闭环 RL**

把 plan → generate → critique → revise 组织成 trajectory，研究 process reward 与 credit assignment

---

很多工作在推理侧编排 workflow或者单模态模型的RL；我们更关心：它如何通过 RL 后训练，变成可学习的创作Agent

总结 · SUMMARY

核心总结

01 AIGC 正从单次生成走向多步创作，Agent 化是图像与视频生成自然的下一步

02 推理侧 workflow 解决「能不能串起来」；RL 后训练解决「能不能从反馈中学会」

03 UniRL 当前统一了多模态模型的 RL 后训练；Creative Agent 是正在探索的下一层

致谢 · THANK YOU

先把多模态生成模型的 RL 后训练统一 起来，再把这条路，推向可训练的 creative loop

王溪楠 · 腾讯 · 混元 多模态强化学习团队

从生成到创作 —— single-shot sample → long-horizon trajectory

扫码了解更多

UniRL · 联系我们