

PYTORCH CONFERENCE 2026

Miles

RL post-training for the new era of scale and workloads

Yueming Yuan, RadixArk

BACKGROUND

The new challenges: larger scale, diverse workloads

Larger scale

- 3T-parameter models, like Kimi K3
- 1M-token agent trajectories
- 10,000+ GPU clusters

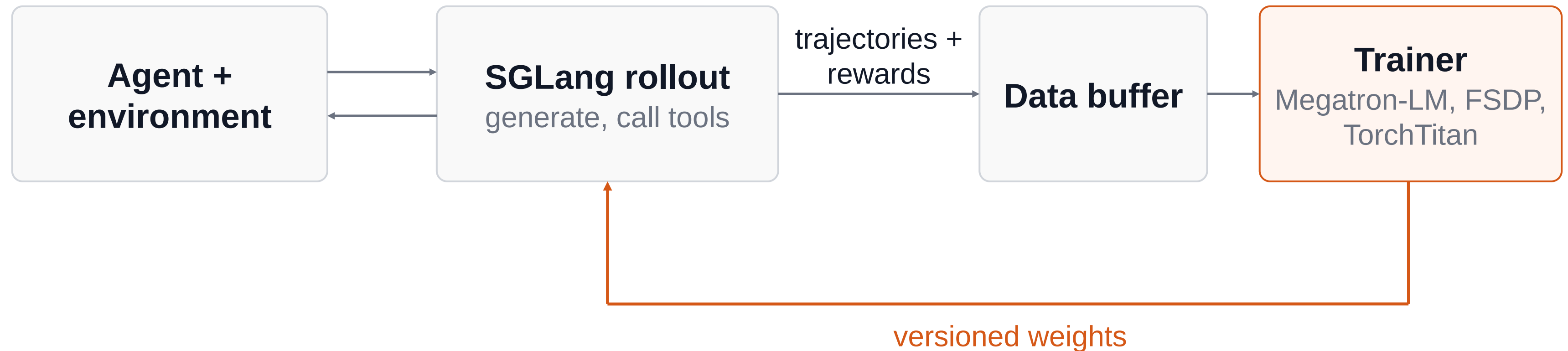
Diverse workloads

- Coding and tool-use agents
- Vision-language models
- Distillation, LoRA, diffusion
- NVIDIA, AMD and TPU

MILES

Miles: any post-training task, stably and efficiently

Open-source RL framework. SGLang for rollout; Megatron-LM, FSDP or TorchTitan for training.

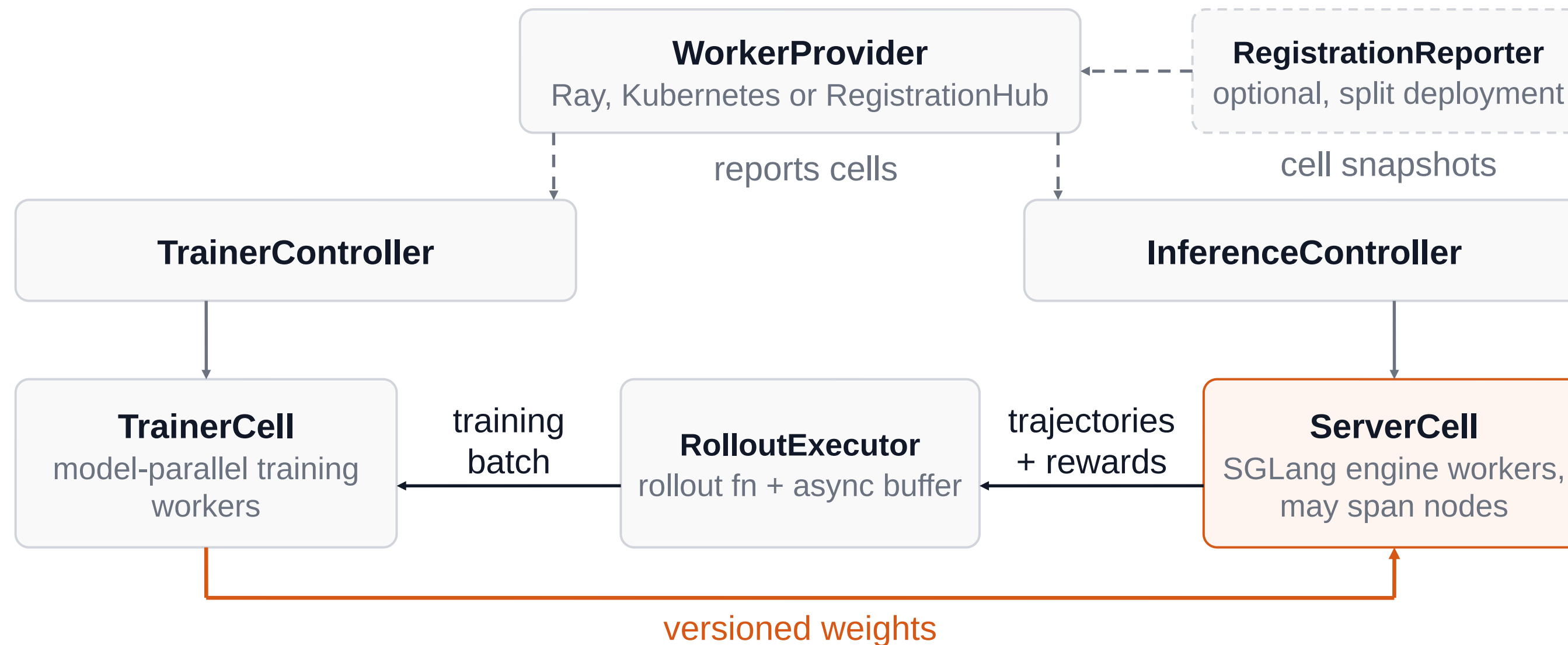


Larger scale

LARGER SCALE

Cells: the fleet changes at explicit boundaries

A **cell** = workers discovered, initialized, health-checked and retired **together**

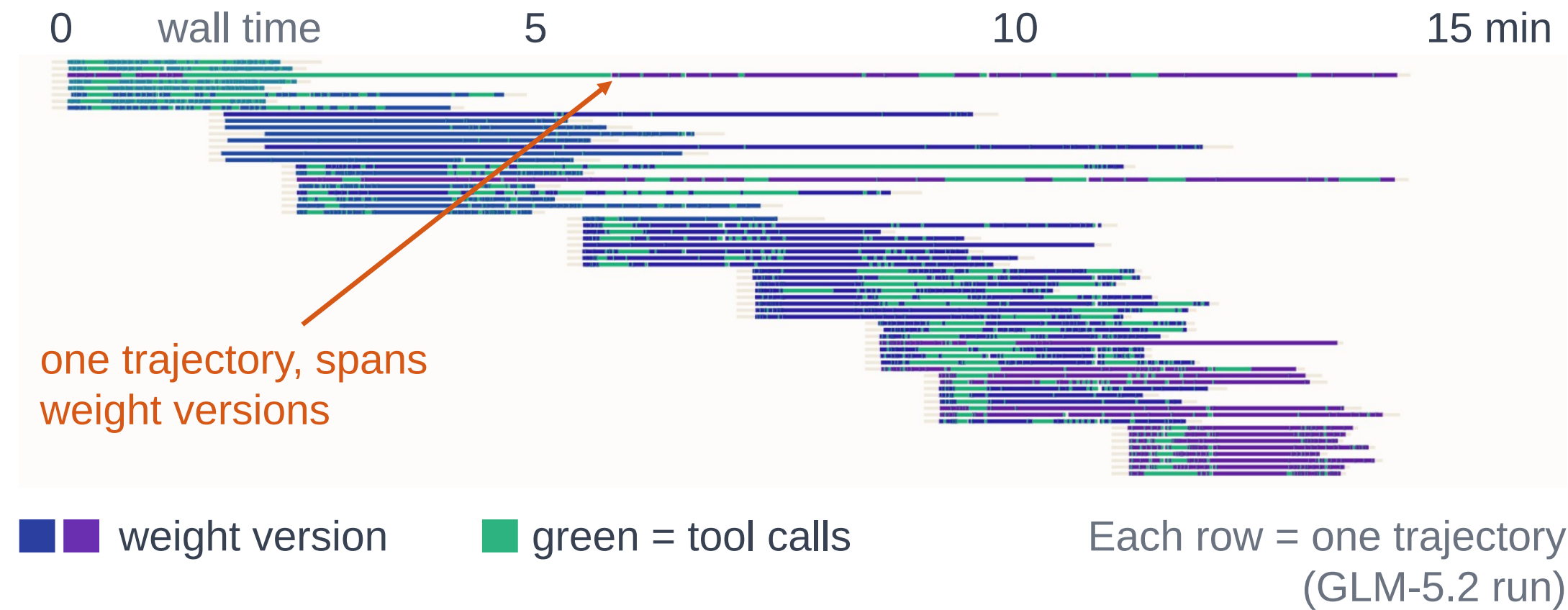


- Scale out with new engines
- Plug in external inference fleets
- Run on Ray or Kubernetes
- Tolerate failures cell by cell



LARGER SCALE

Fully async: rollout never waits for the training batch



- Rollout concurrency is set on its own, not by the training batch
- Every finished group enters the buffer at once

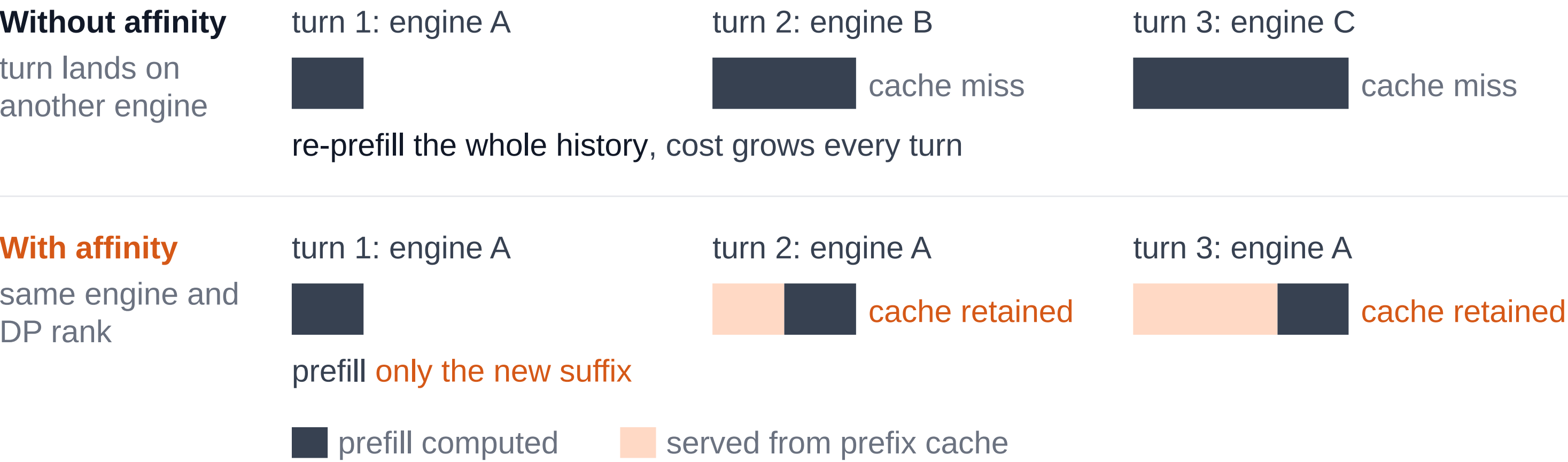
Disaggregated mode (default)
rollout and training overlap

Colocated mode (new)
they share GPUs and take turns

Efficiency

EFFICIENCY

Agentic rollout on SGLang: every session stays on its cache



- Each turn goes to the engine and DP rank that holds its KV cache
- New sessions go to the least-loaded engine
- Under fully async, prefix cache is kept separate per weight version

Observed: 96% prefix-cache hit (GLM-5.2 run)

EFFICIENCY

Training: three backends, faster kernels

Megatron-LM (default)

every parallelism axis, LoRA

FSDP2

Hugging Face models as they are

TorchTitan (new)

torchtitan's trainer, TP, PP, CP and EP

Recent optimizations

Fused log-prob and entropy kernel: loss head **13 to 16x** faster

Architecture-tuned sparse attention: up to **10x** faster forward and backward (GLM-5)

Load-balanced DeepSeek-V4 indexer across context-parallel ranks: **4 to 8%** more throughput

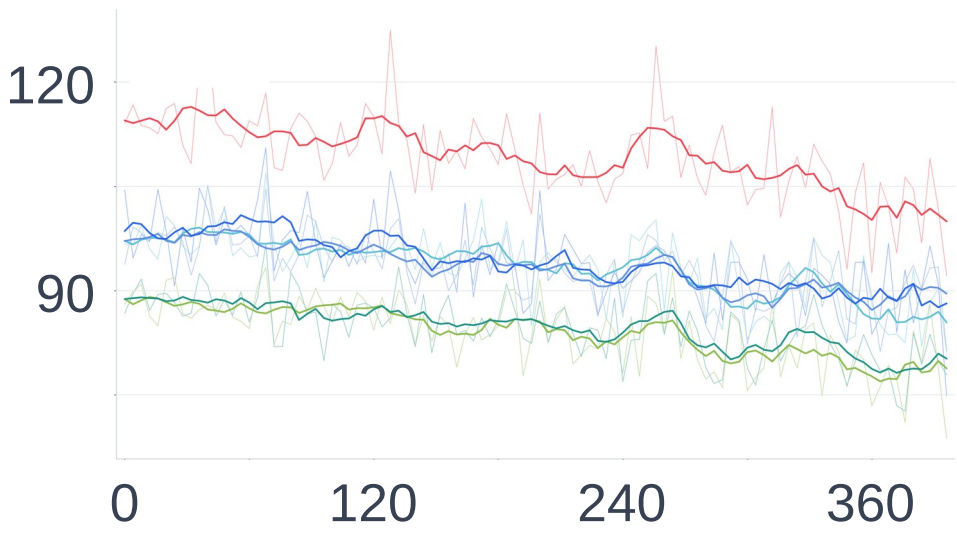
Optimizer state streamed from NVMe: 744B full-parameter RL on **half the nodes**

EFFICIENCY

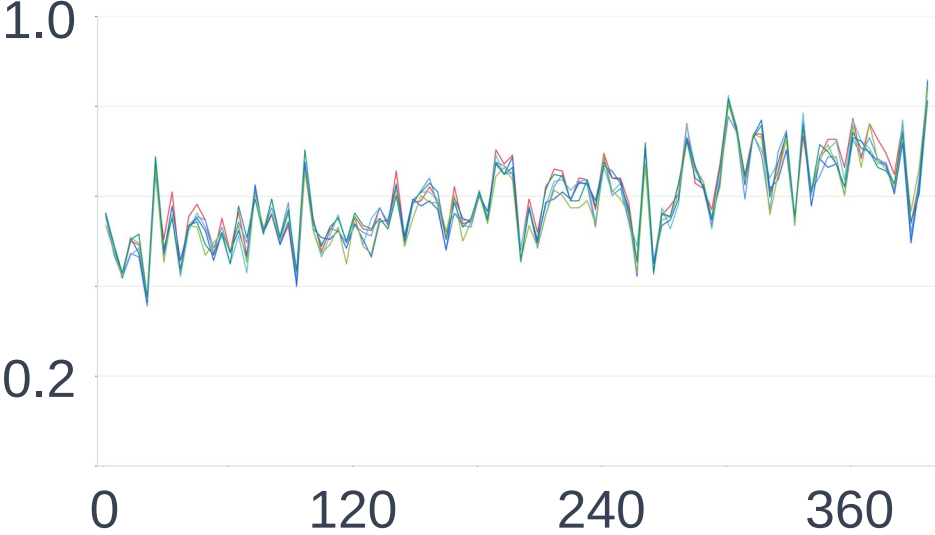
Low-precision RL recipes

Recipe	Rollout	Training	Hardware
FP8	FP8	FP8 or BF16	Hopper, Blackwell, AMD MI350X and MI355X
MXFP8	MXFP8	MXFP8 or BF16	Blackwell
NVFP4	NVFP4	NVFP4 experts	Blackwell
INT4 QAT	INT4 weights	BF16 with fake quantization	Hopper
BF16 train, FP8 serve	FP8	BF16	FP8 GPUs

Rollout time per step (s)



Reward



BF16

MXFP8

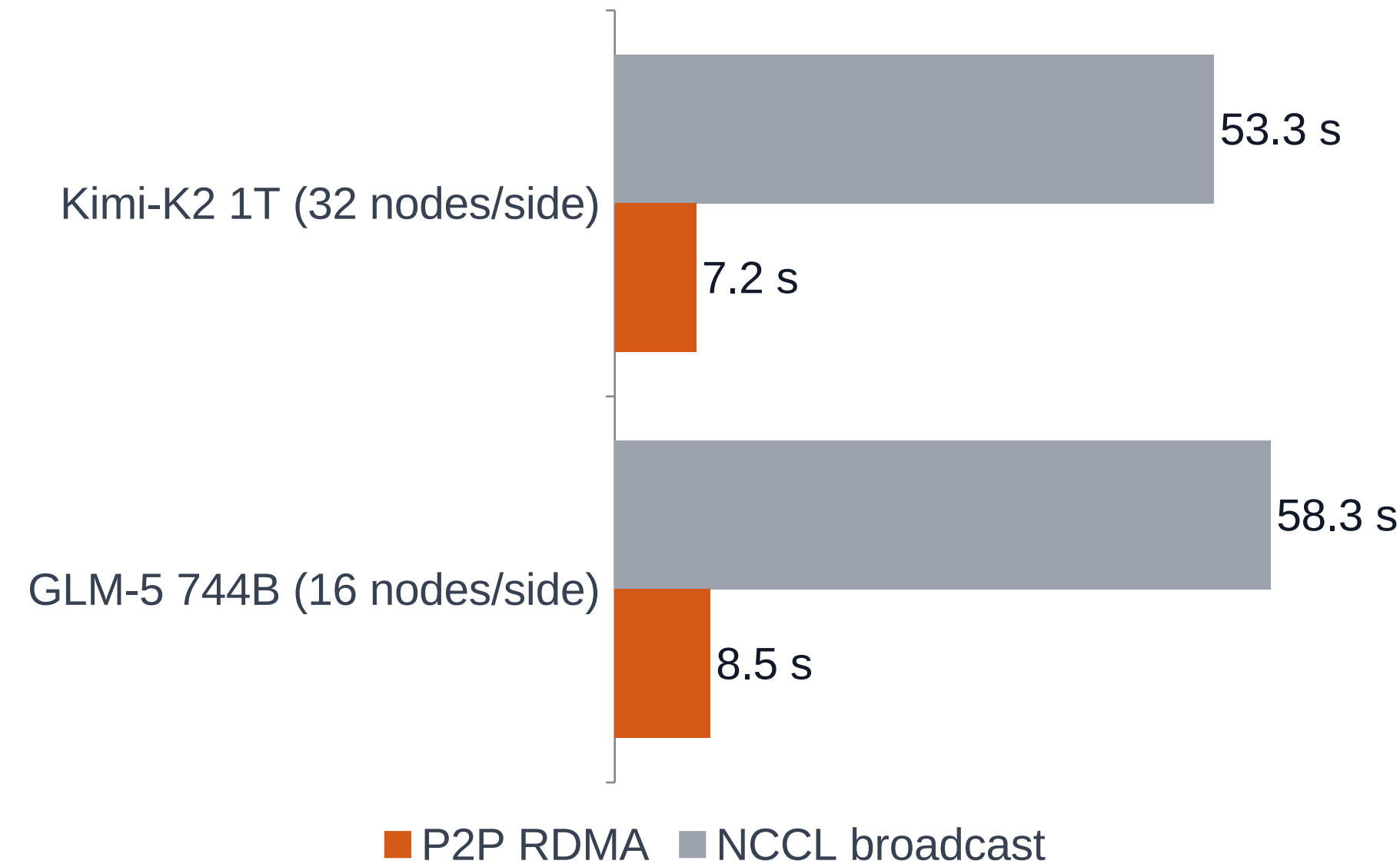
NVFP4

Qwen3-30B-A3B on 8 B200 GPUs,
rollout step on the x-axis

EFFICIENCY

Weight update: one interface, many transports

P2P vs broadcast, 1T and 744B



H100. Weight-update pause only, not the whole RL step.

Collocated
CUDA IPC

Shared NCCL fabric
broadcast

Large fleets
P2P over RDMA, with Mooncake, NIXL or
Ray RDT

No shared fabric
delta updates, **62.4 GB to 0.7 GB** per sync
(GLM-4.7-Flash)

Stability

STABILITY

Infra-level fixes for train-inference mismatch

Where the mismatch comes from	How Miles removes it
Tokens: harness re-tokenizes the text	session server keeps the sampled token IDs
MoE routing: tiny diffs flip experts	R3 replays rollout's expert choices
Sparse attention: indexer picks differ	replay rollout's indexer top-k
Kernels: batch-dependent reductions	batch-invariant kernels: diff exactly 0 (Qwen3)

STABILITY

Algorithm-level corrections, and score centering

Supported corrections

TIS

MIS

IcePop

OPSM

GSPO

Decoupled PPO

Score centering

Score centering

Removes the drift that stale samples add to the gradient

What Miles built for it

Rollout records the top 128 candidates per token

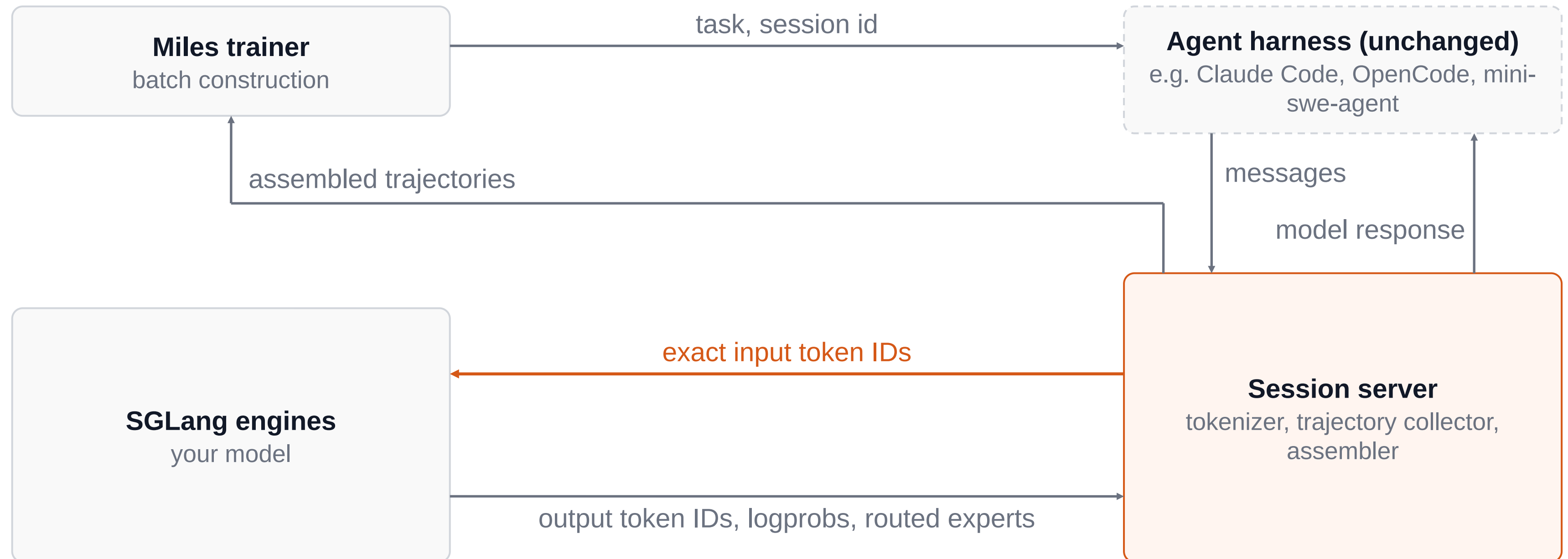
FP32 LM head in rollout for exact probabilities

Candidates sent as flat arrays: **130 s to 10 s** for 32 sessions

Diverse workloads

DIVERSE WORKLOADS

Session server: train inside any agent harness



DIVERSE WORKLOADS

Environments: plug in at the layer you need

Rollout function also batching and group rewards

Prime Intellect Verifiers

Generate function also token recording

HUD, Strands Agents, τ -bench

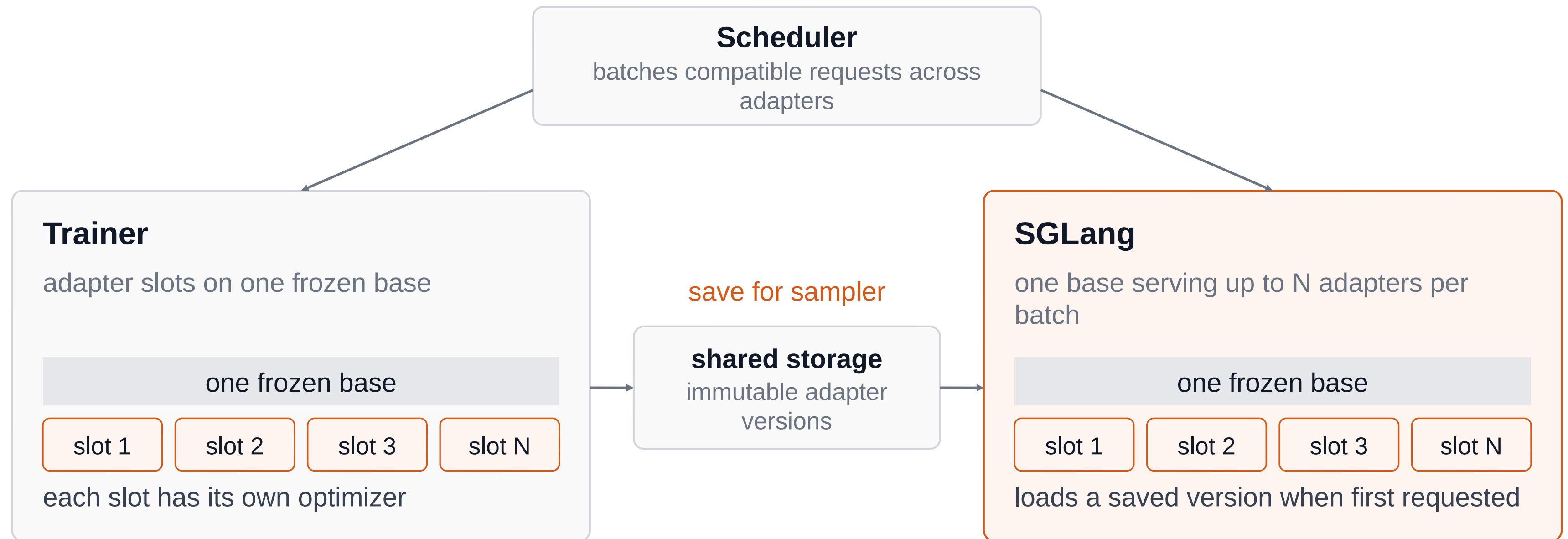
Agent function just the episode loop

Harbor, OpenEnv, NeMo Gym

Sandboxes: Daytona, E2B, Modal, AgentENV, Docker

DIVERSE WORKLOADS

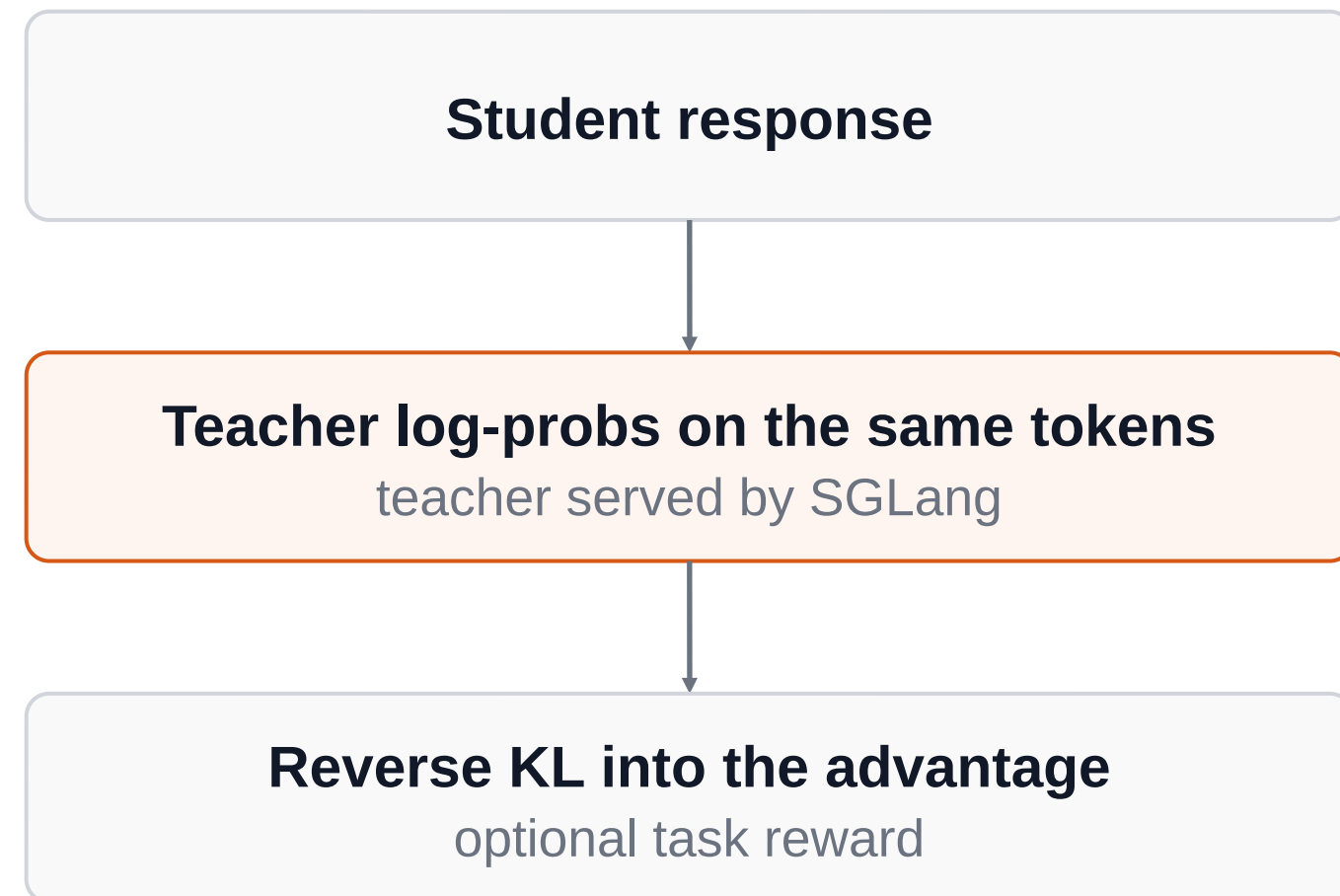
Multi-LoRA and Tinker: many adapters, one base



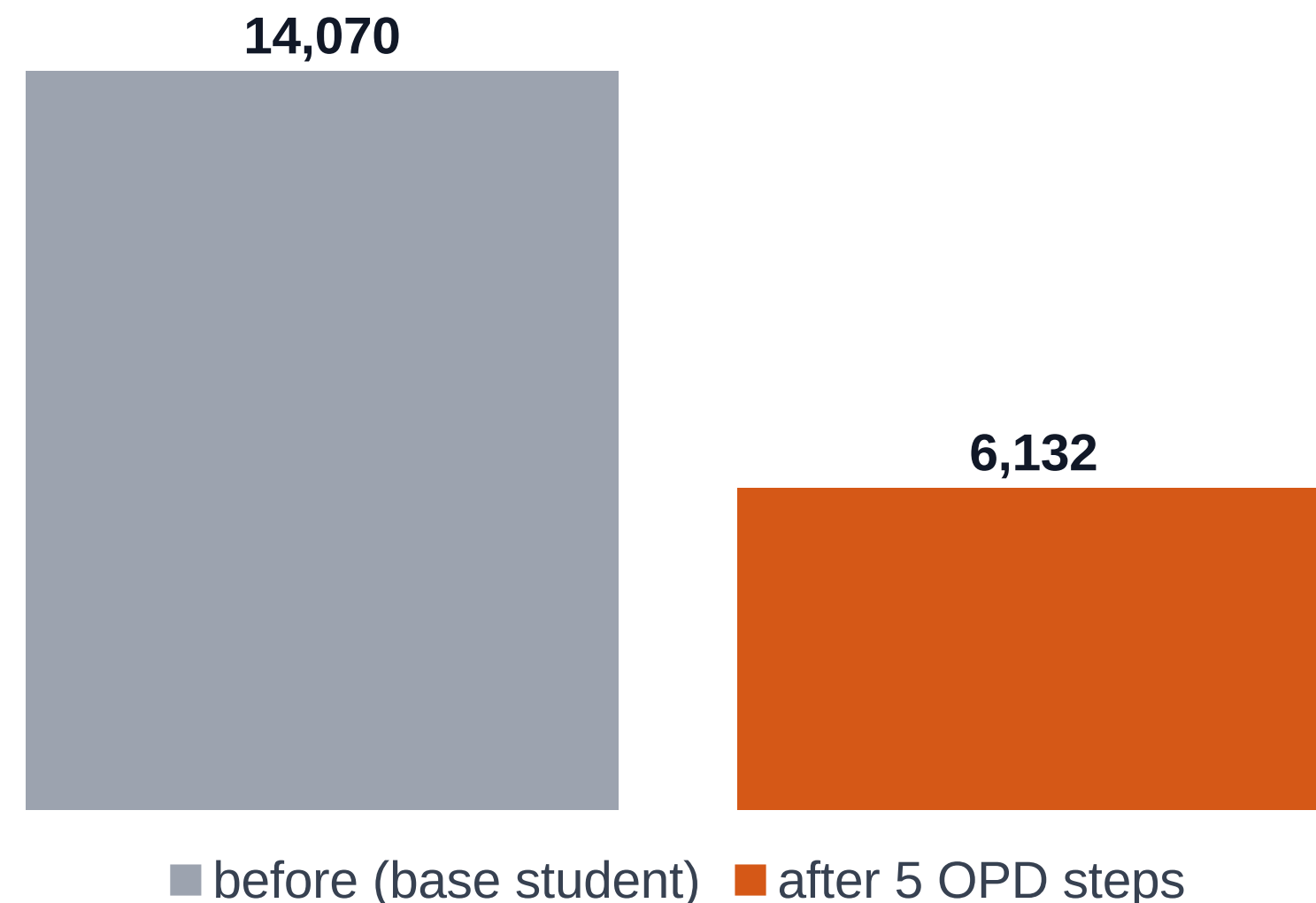
Built beside the RL loop: a separate Tinker package and one trainer subclass

DIVERSE WORKLOADS

On-policy distillation (OPD), without a task reward



Held-out mean response length (tokens)



Qwen3.5-35B-A3B, task reward 0. Accuracy: no detectable change (84.0% vs 85.2%).

DIVERSE WORKLOADS

Diffusion: same loop, different trajectory

	Engine	Trajectory	Loss	Scheduling	Weight sync	Evaluation
Text RL	SGLang	tokens	GRPO / PPO	same architecture and contracts		
Diffusion RL	sglang-diffusion	denoising steps	Flow-GRPO			

[PLACEHOLDER: diffusion results, to be filled by the diffusion team]

DIVERSE WORKLOADS

Same loop on NVIDIA and AMD

NVIDIA

AMD

TPU, next

ANNOUNCEMENT

[PLACEHOLDER: TPU training announcement — user fills]

[PLACEHOLDER: announcement headline + 2–3 facts, visual — user-supplied]

Thank you

github.com/radixark/miles

miles.radixark.com

arXiv 2609.08368