NEURIPS 2026

ECHO-2

A Large-Scale Distributed Rollout Framework for Cost-Efficient Reinforcement Learning

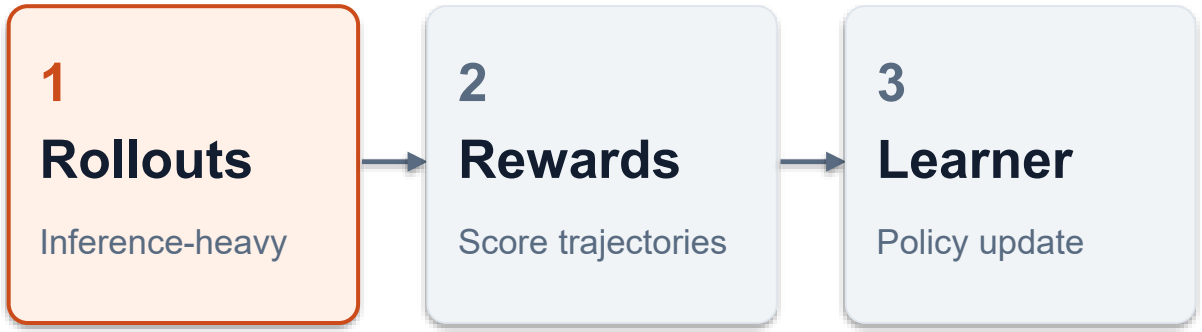
Jingwei Song*, Meng Chen*, Jie Xiao*, Qingnan Ren*, Jiaqi Huang, Yangshen Deng, Chris Tong, Wanyi Chen, Suli Wang, Zhisheng Chen, Ziqian Bi, Shuo Lu, Yiqun Duan, Xu Wang, Rymon Yu, Ween Yang, Lynn Ai, Eric Yang, Tianyu Shi†

* Equal contribution · † Corresponding author

Rollouts dominate RL post-training—on premium GPUs

The expensive part is often inference, not policy optimization.

CONVENTIONAL RL POST-TRAINING LOOP



The forward-pass workload is a candidate for cheaper GPUs.

REFERENCE RENTAL PRICES (USD / GPU-HOUR)

A100 80GB · Google Cloud

$3.06

RTX 5090 · vast.ai

$0.35

8.7× cheaper per GPU-hour

Question: RL does not care where a trajectory comes from. Can cheap, distributed GPUs generate the rollouts?

Cheap wide-area compute can still stall the learner

Wide-area policy dissemination is no longer negligible.

REPRESENTATIVE MEASURED SCALES (DIFFERENT EXPERIMENTS)

~24 min
Policy dissemination

~27 min
8B learner update

Slow snapshots
Checkpoint transfers across WAN links

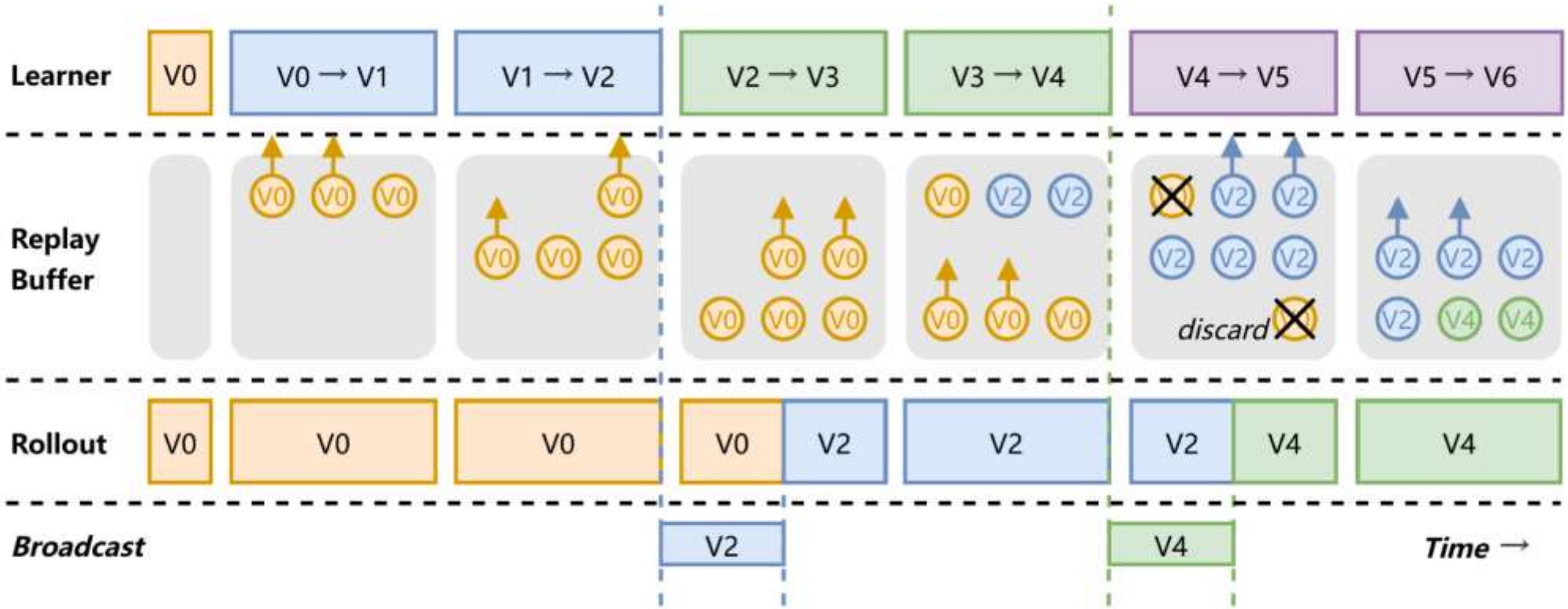
Uneven workers
Throughput and availability fluctuate

Uplink contention
With push-to-all, each of N workers gets only 1/N of the learner's upload bandwidth.

Strict synchronization creates idle learner "bubbles" that erase the savings.

Bounded staleness creates room for overlap

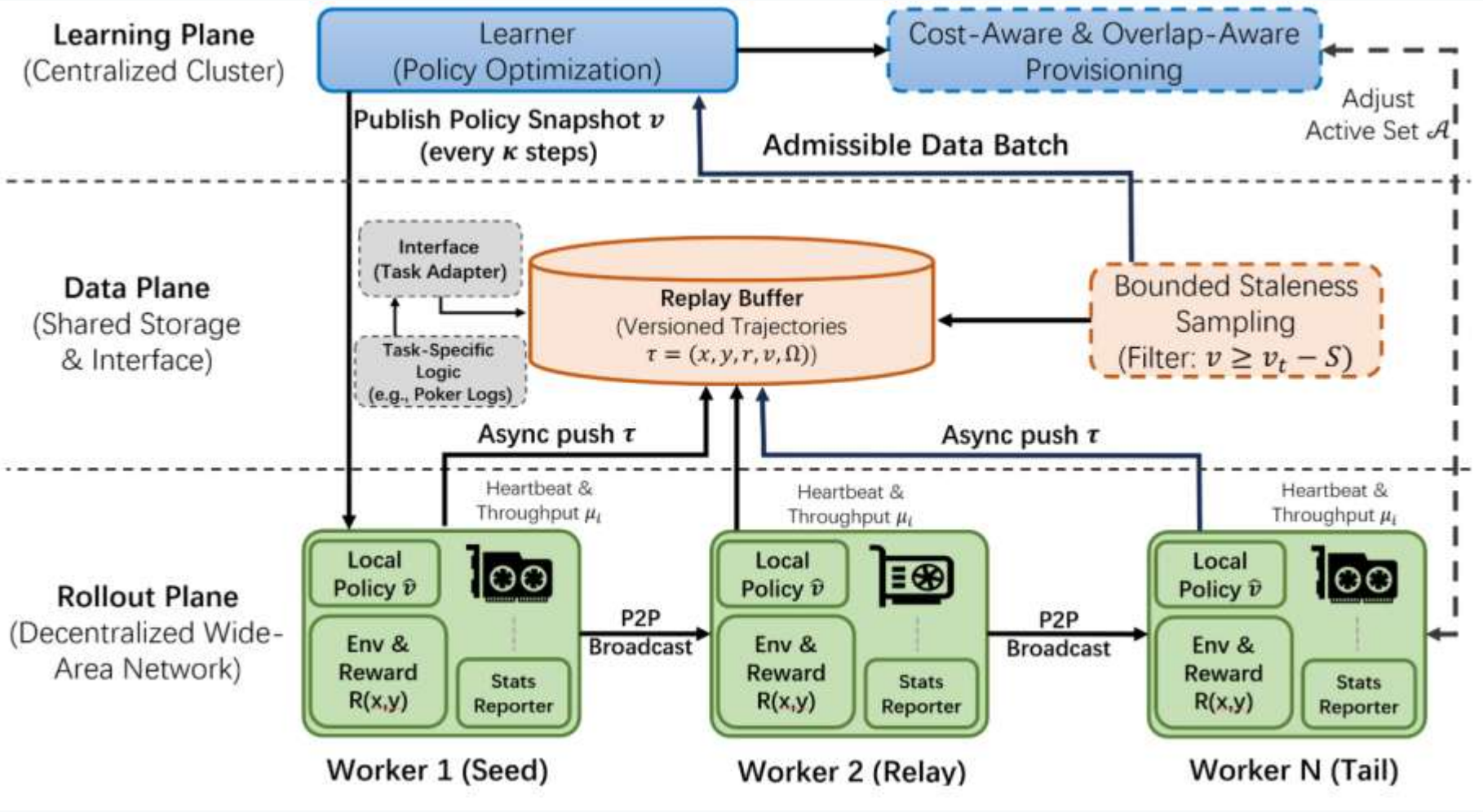
Freshness is a budget, not a global synchronization barrier.



S: the oldest policy the learner accepts, counted in training steps.
κ = S − 1: a snapshot is published every κ steps.
Here S = 3, κ = 2.Workers keep generating with the old policy while the new one is in flight.

Three planes

Decouple policy optimization, rollout inference, and versioned data.



LEARNING PLANE

A few data-center GPUs update the policy and publish a snapshot every κ steps.

DATA PLANE

A versioned replay buffer admits only rollouts within the budget S.

ROLLOUT PLANE

Wide-area workers generate and score trajectories, and relay snapshots to peers.

One overlap rule sizes the rollout pool

Time to train κ steps (one publication period) ≥ **Time to broadcast the new snapshot** + **Time to generate κ batches of rollouts**

OVERLAP CONDITION

$$\kappa T_{\text{train}} \geq T_{\text{bcast}} + \frac{\kappa R}{\sum_{i \in \mathcal{A}} \mu_i}$$

CAPACITY RULE

$$\sum_{i \in \mathcal{A}} \mu_i \geq \mu_{\min}(\kappa) = \frac{\kappa R}{\kappa T_{\text{train}} - T_{\text{bcast}}}$$

One throughput target for the whole heterogeneous pool.

Cost-aware activation

Rank workers by price ÷ throughput. Switch on the cheapest until the target is met.

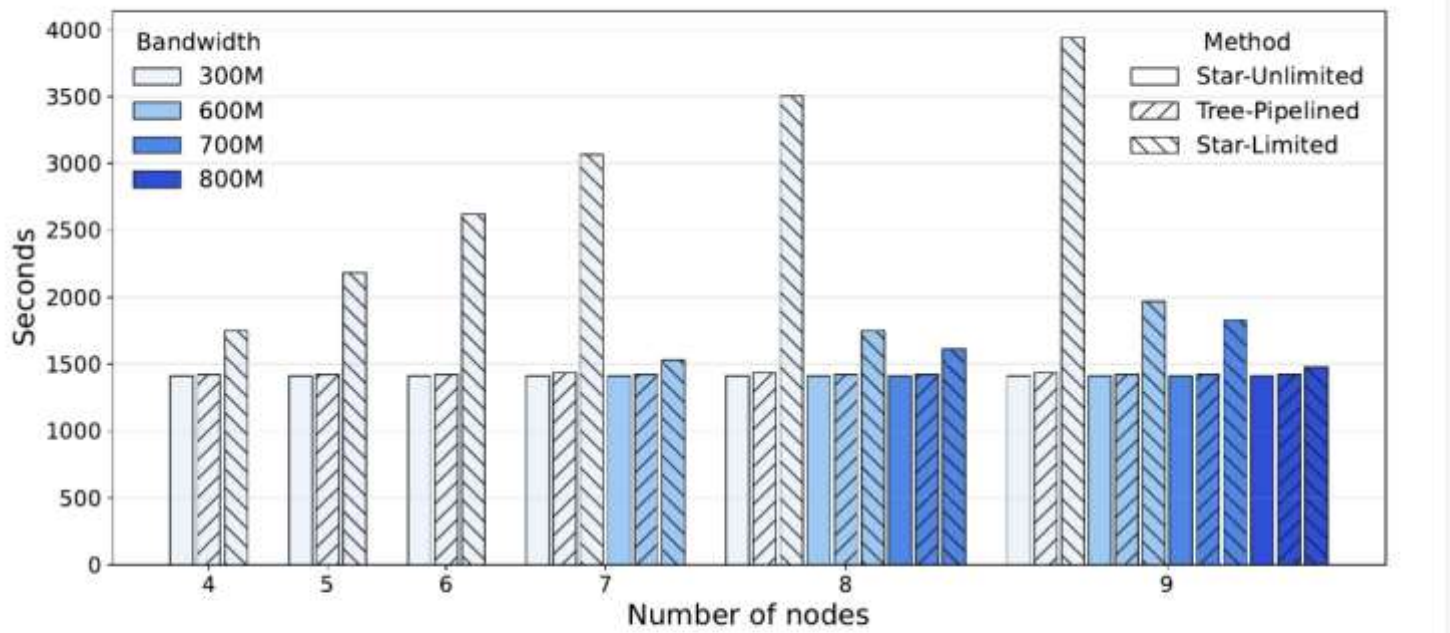
Closed-loop control

Aim 10% above the minimum (γ = 1.1). Add or release workers only after sustained drift.

Key insight: less broadcast delay means fewer workers needed to keep the learner busy.

Peer-forwarding chains keep broadcast time flat as the fleet grows

Decouple policy optimization, rollout inference, and versioned data.



Broadcast time vs. number of workers at four learner uplink caps (300–800 Mbps). Each worker is capped at 100 Mbps.

Push-to-all (star)
N workers share the learner's uplink, so broadcast time grows with N.



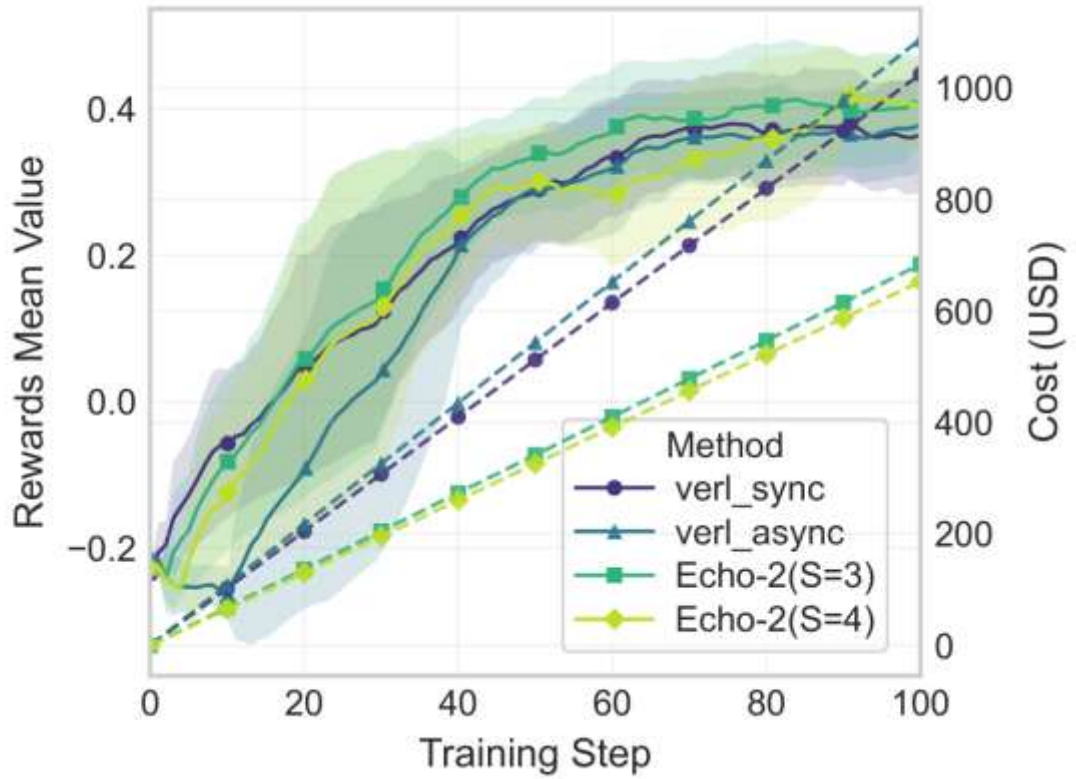
The learner feeds as many chains as its uplink supports. Each worker forwards chunks to one child and starts generating once its copy is installed.

1,437 s with peer forwarding

1,830 s without, same 9 workers

Comparable quality at 33–36% lower estimated cost

Qwen3-8B GRPO on DAPO-Math WAN rollout fleet - AIME24 evaluation



33.3–36.3%

lower cumulative GPU rental cost at matched AIME24 accuracy

Solid: reward curves

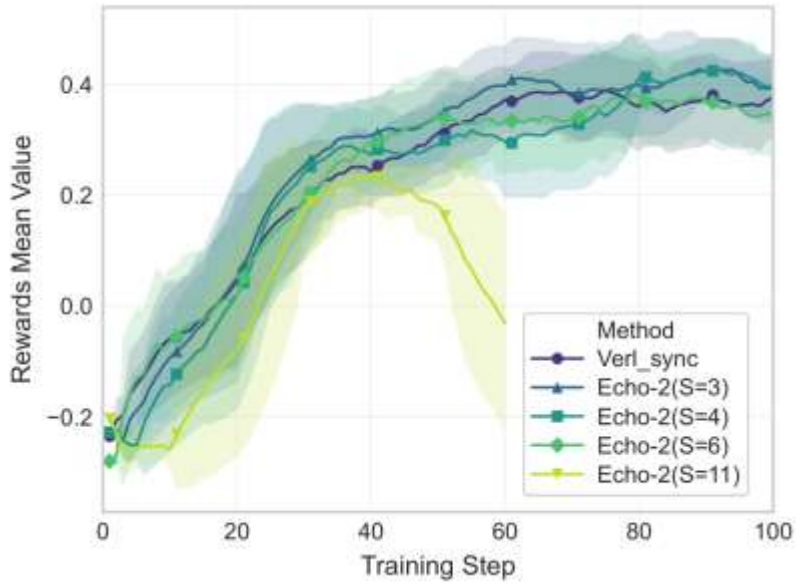
Dashed: estimated cumulative cost

Baselines: co-located veRL sync / async

Prices: $3.06/A100-hour; $0.35/5090-hour (2026-05-01).

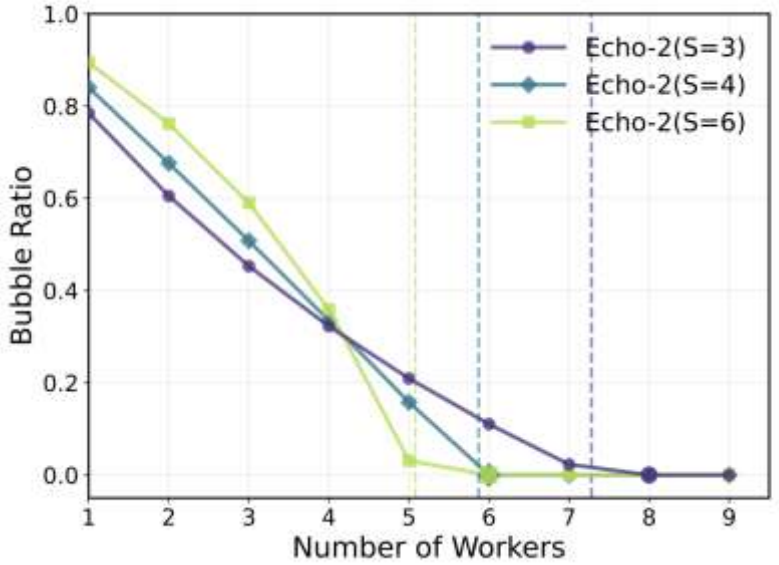
Staleness is useful—but not free

Learning stability and the predicted capacity threshold both matter.



Quality boundary

S ≤ 6: comparable here; S = 11: divergence.



Utilization threshold

Near-zero learner bubbles at the predicted threshold.

TRY IT OUT

Explore Logits

LLM training & sampling



logits.dev

Try ECHO-2!

Centralized learning + distributed rollouts can be cost-efficient.