

Scaling Multi-Model PyTorch Workloads for Closed-Loop Robotics Evaluation

Nan Zhu, Shenyuan Gao, Zi-ang Cao, Kaiyuan Zheng, Fangqi Zhu

NVIDIA



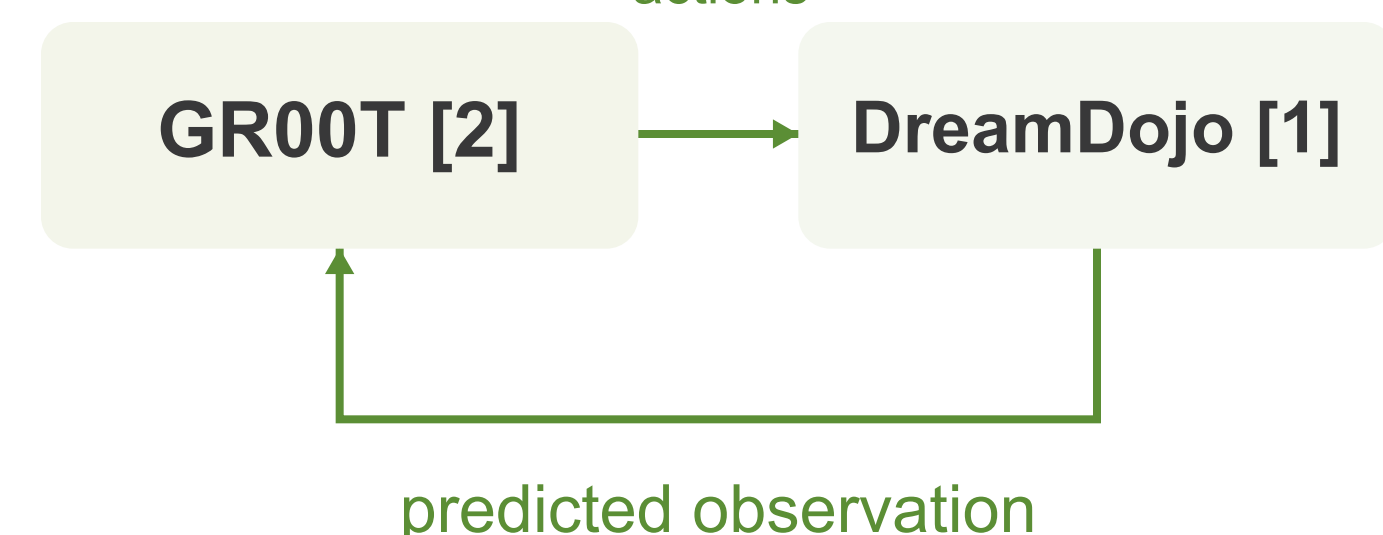
01 Closed-loop challenge

Closed-loop evaluation reveals compounding errors that open-loop testing misses.

Closed-loop evaluation repeatedly couples policy inference, world-model prediction, and online reward scoring. These models have different compute demands and execution times, making it difficult to keep GPUs busy as the number of concurrent rollouts grows.

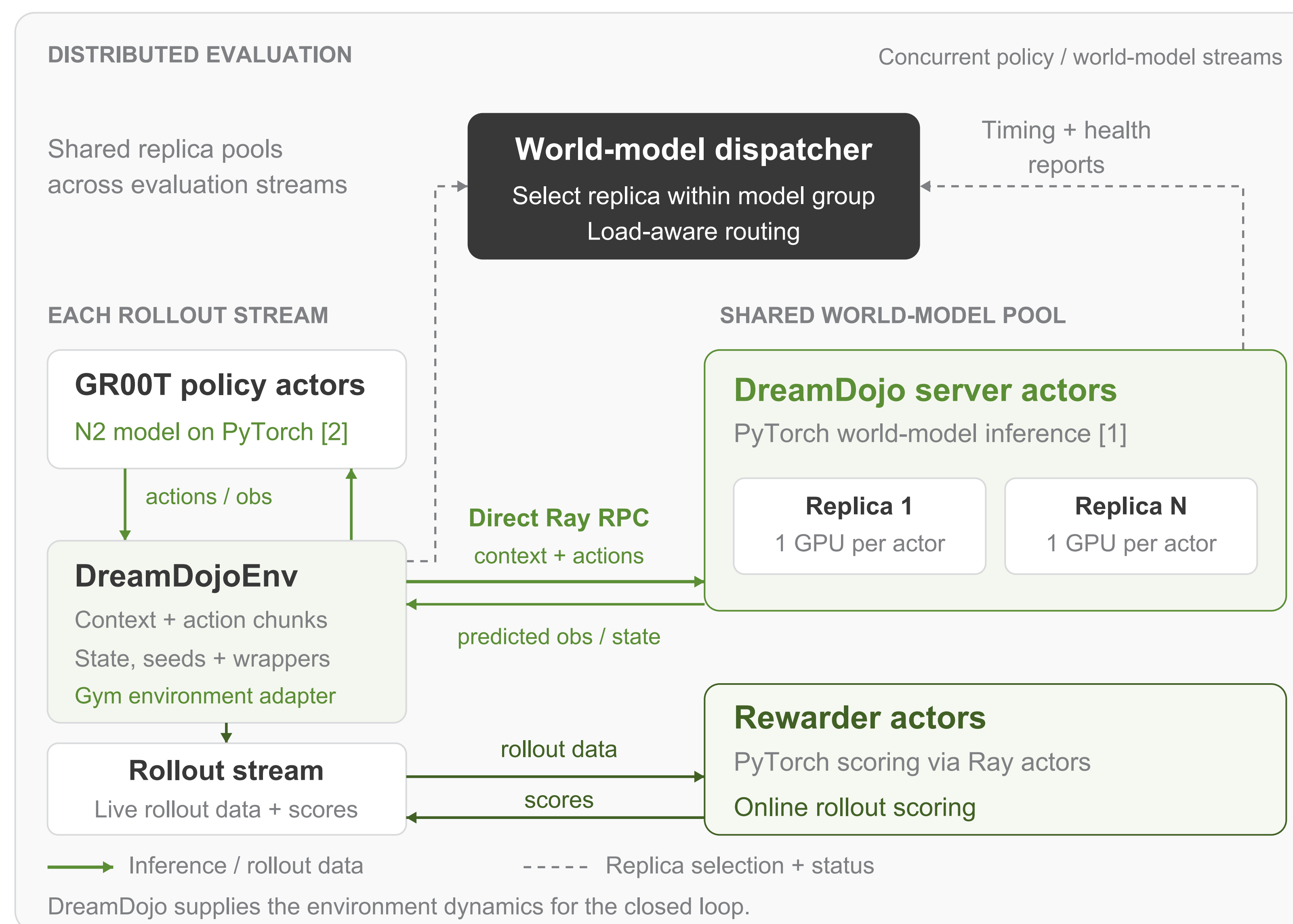
THE CLOSED LOOP

Start from real episode context
actions



Each policy replans from predicted observations as errors accumulate.

02 PyTorch execution, Ray coordination



03 Scaling results

MULTI-MODEL THROUGHPUT

>10×
higher throughput

Tens to hundreds
of GPUs



Act, predict, score, and repeat.

Heterogeneous execution

Independent policy streams share world-model replicas, accommodating differences in model execution time.

Reproducible evaluation

Consistent multi-model rollouts support policy comparison at scale.

Concurrent evaluation streams

Coordinate policy, world-model and rewarder actors. Score live rollouts while evaluation streams run across the cluster.

Load-aware replica routing

Select replicas using queued work and observed latency. Send inference arrays directly to the chosen actor via Ray RPC.

Reproducible rollouts

Preserve rollout context and control model randomness across distributed execution for repeatable policy evaluation.

[1] S. Gao et al. DreamDojo: A Generalist Robot World Model from Large-Scale Human Videos. ICML 2026.

[2] NVIDIA Isaac GR00T. Official model repository.