SoCC'25, November 19–21, 2025, Online, USA

STEVENS
INSTITUTE OF TECHNOLOGY
1870

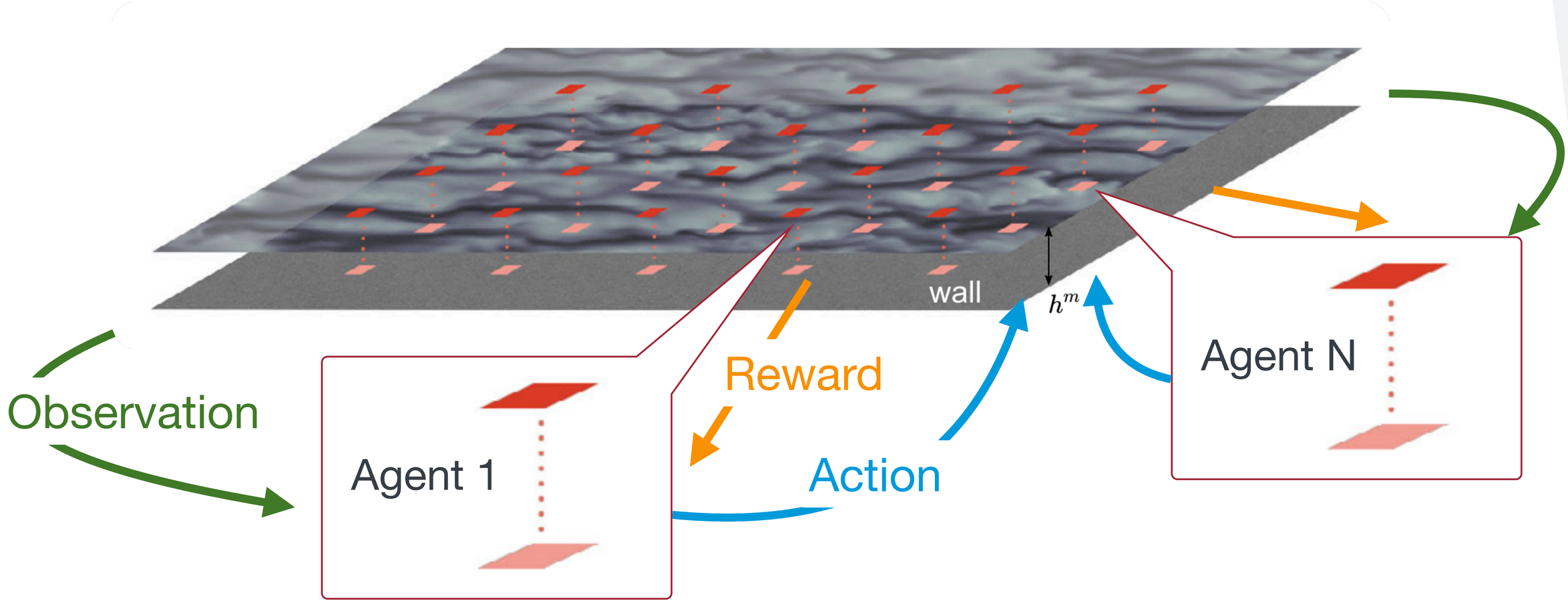
Multi-Agent Reinforcement Learning with Serverless Computing

Rui Wei[1], Hanfei Yu[1], Xikang Song[2], Jian Li[3], Devesh Tiwari[4], Ying Mao[5], Hao Wang[1]

Stevens Institute of Technology[1], University of Chicago[2], Stony Brook University[3],
Northeastern University[4], Fordham University[5]

IntelliSys Lab

Multi-Agent Reinforcement Learning



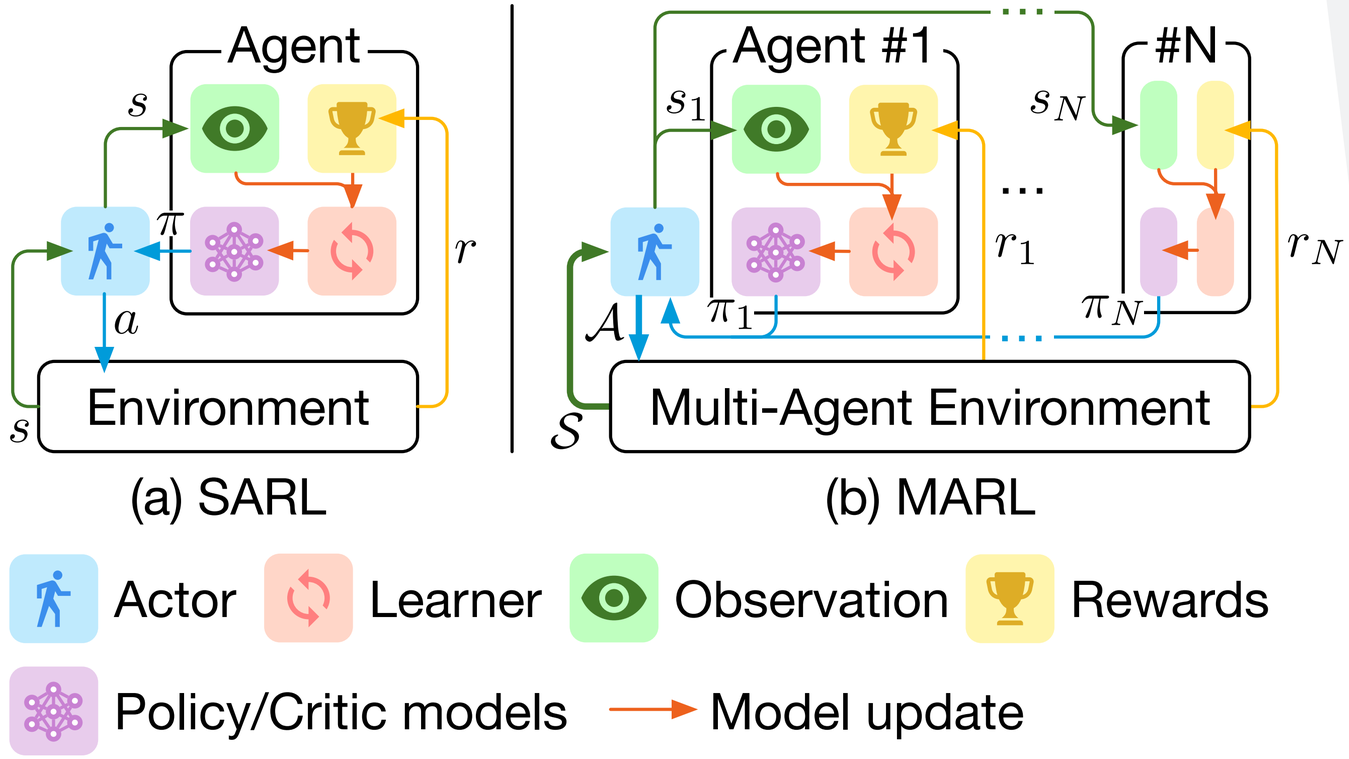
[1] H. Bae; et al. Scientific Multi-agent Reinforcement Learning for Wall-models of Turbulent Flows. Nature Communications 2022 Vol. 13

Distributed Reinforcement Learning

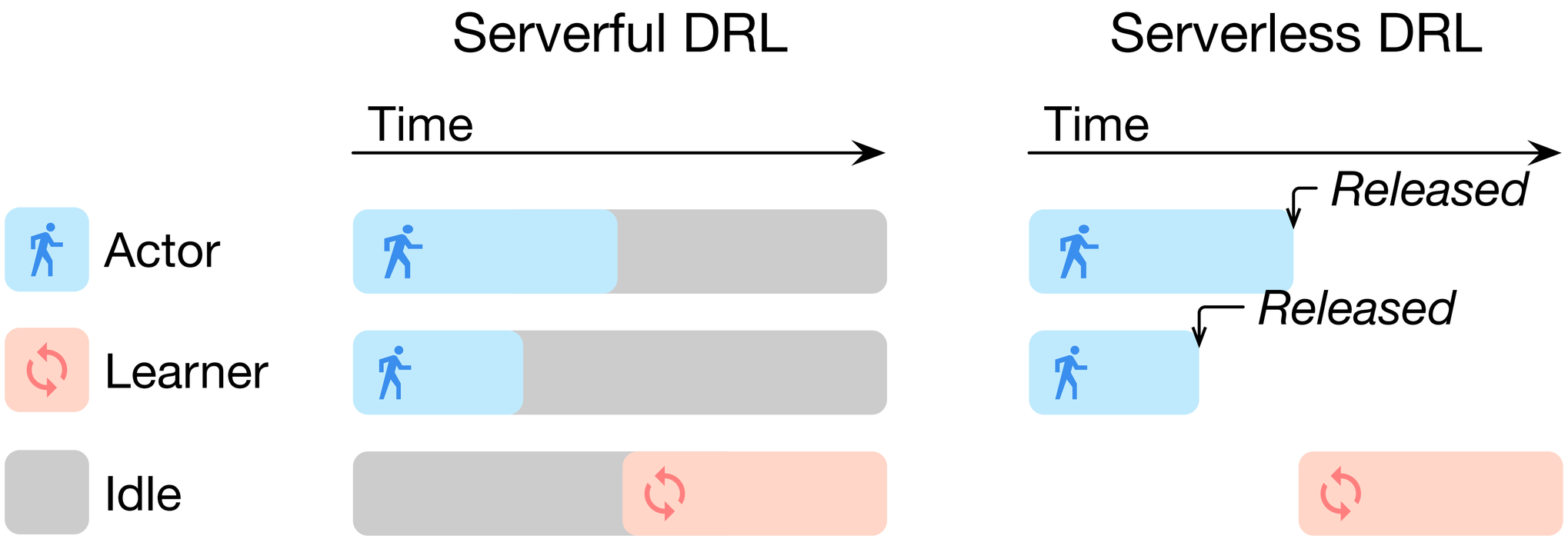
Actor Learner Architecture

Actor: run simulations in a separate environment to sample trajectories

Learner: use trajectories to calculate loss and update the policy



Serverless for DRL



Idleness between actors and learners leads to waste

Launch and release based on actual demands

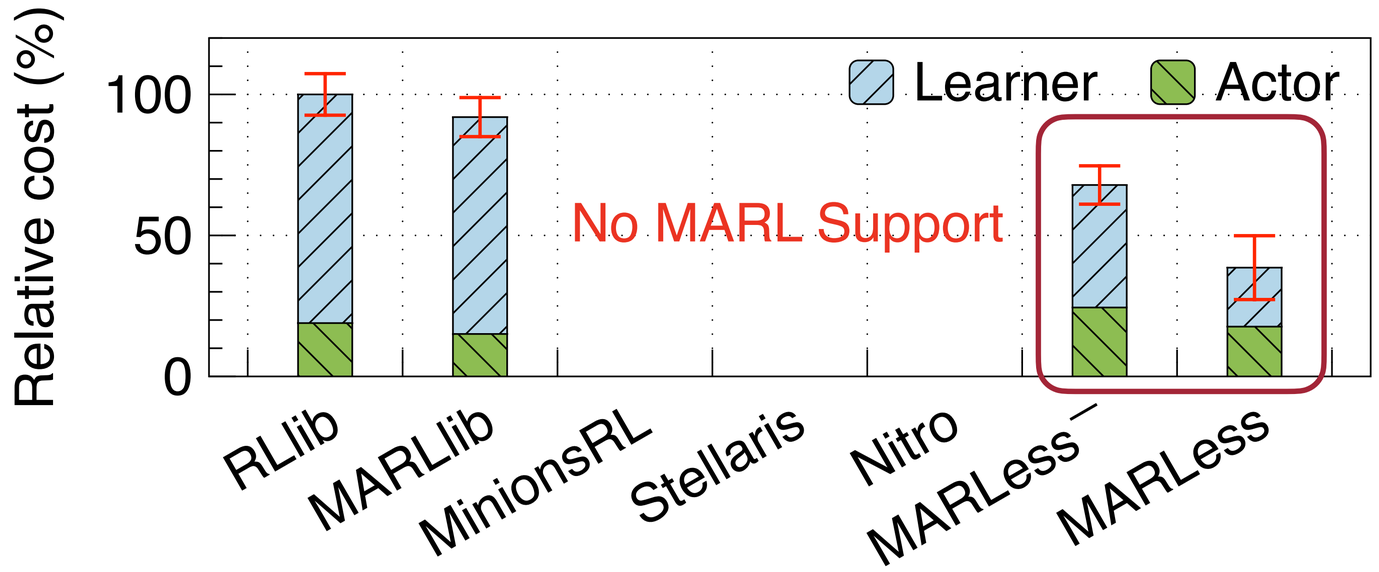
Existing MARL Frameworks

Serverful baselines

RLlib [1], MARLlib [2]

Serverless baselines

MinionsRL [3], Stellaris [4], Nitro [5]

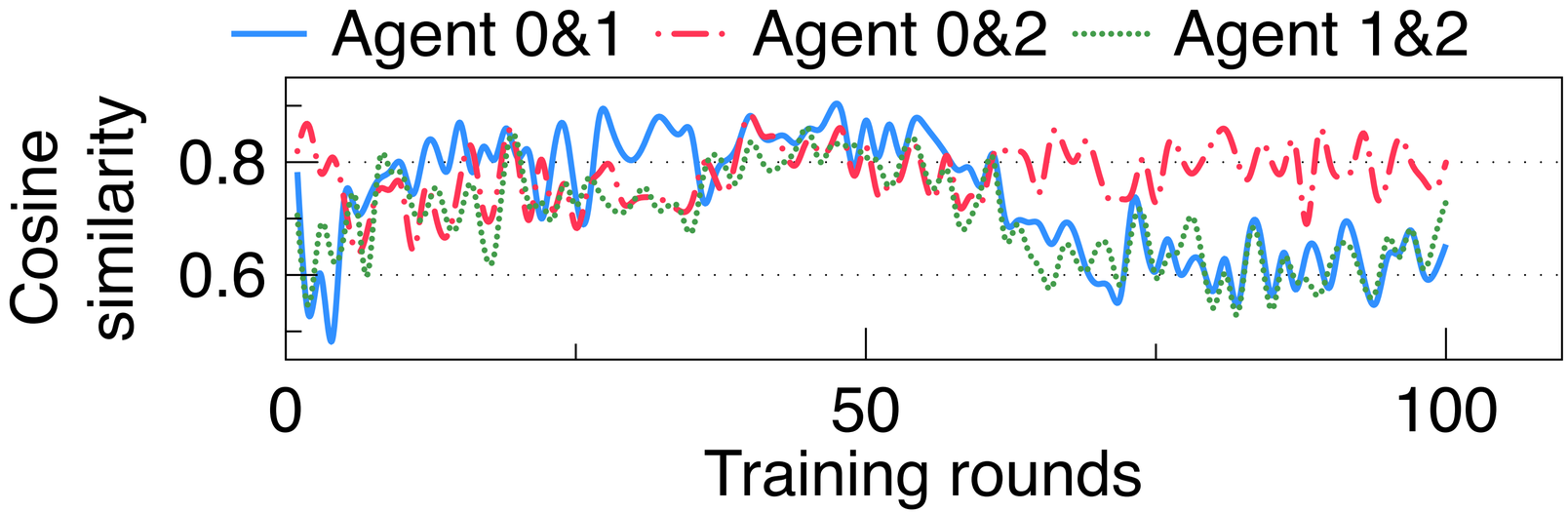


MARLess⁻
= MARL + Serverless

MARLess:
= MARLess⁻
+ Learner Sharing
+ Actor Scaling

[1] E. Liang; et al. RLlib: Abstractions for Distributed Reinforcement Learning. ICML 2018
[2] S. Hu; et al. MARLlib: A Scalable and Efficient Multi-agent Reinforcement Learning Library. PMLR 2023
[3] H. Yu; et al. Cheaper and Faster: Distributed Deep Reinforcement Learning with Serverless Computing. AAAI 2024
[4] H. Yu; et al. Stellaris: Staleness-Aware Distributed Reinforcement Learning with Serverless Computing. SC 2024
[5] H. Yu; et al. Nitro: Boosting Distributed Reinforcement Learning with Serverless Computing. VLDB 2024

Motivation 1: Similarity Among Agents



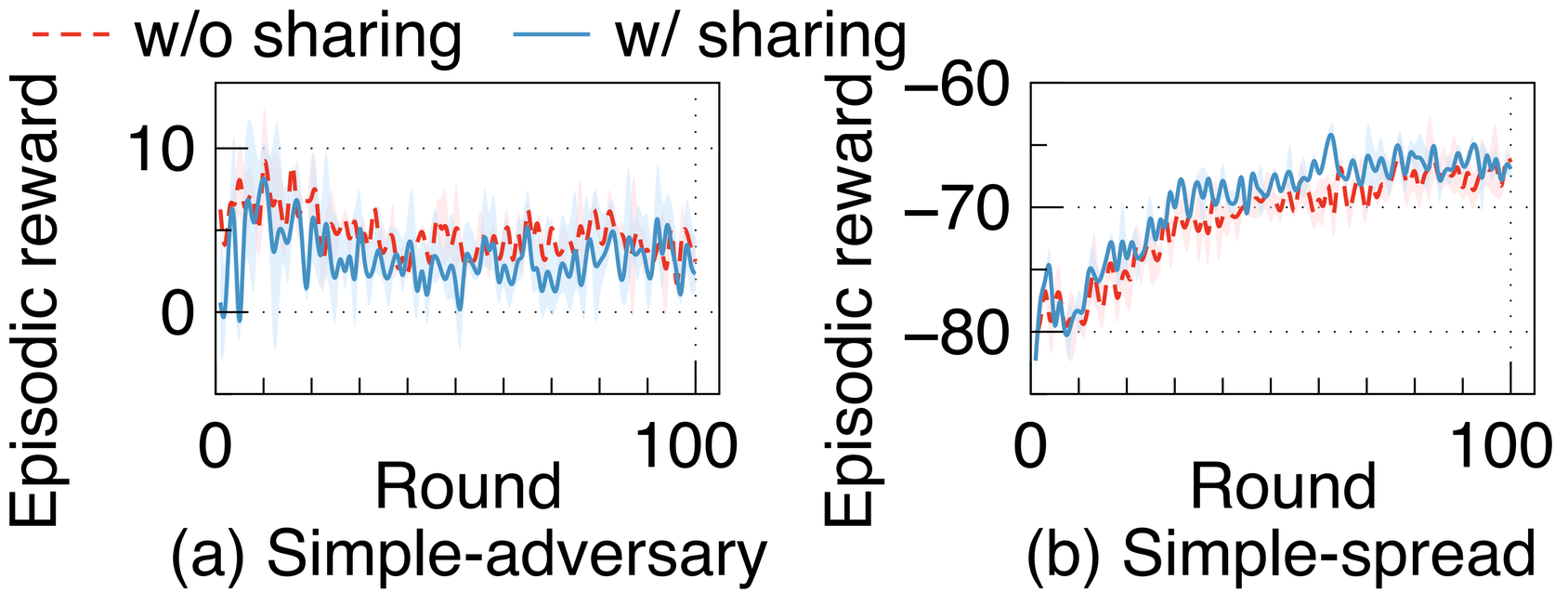
Agents behave similarly at some training rounds → Dynamically grouping them to use the same policy and learner → **Dynamic Learner Sharing**

💵 Lower Cost

Challenge 1: Balancing Cost and Quality

How to design dynamic learner sharing without affecting training quality?

Sharing learner/policy may lead to performance downgrade [1,2]



[1] T. Hu; et al. Measuring Policy Distance for Multi-Agent Reinforcement Learning. AAMAS 2024
[2] Y. Zang; et al. Automatic Grouping for Efficient Cooperative Multi-Agent Reinforcement Learning. NIPS 2023

Design 1: Dynamic Learner Sharing

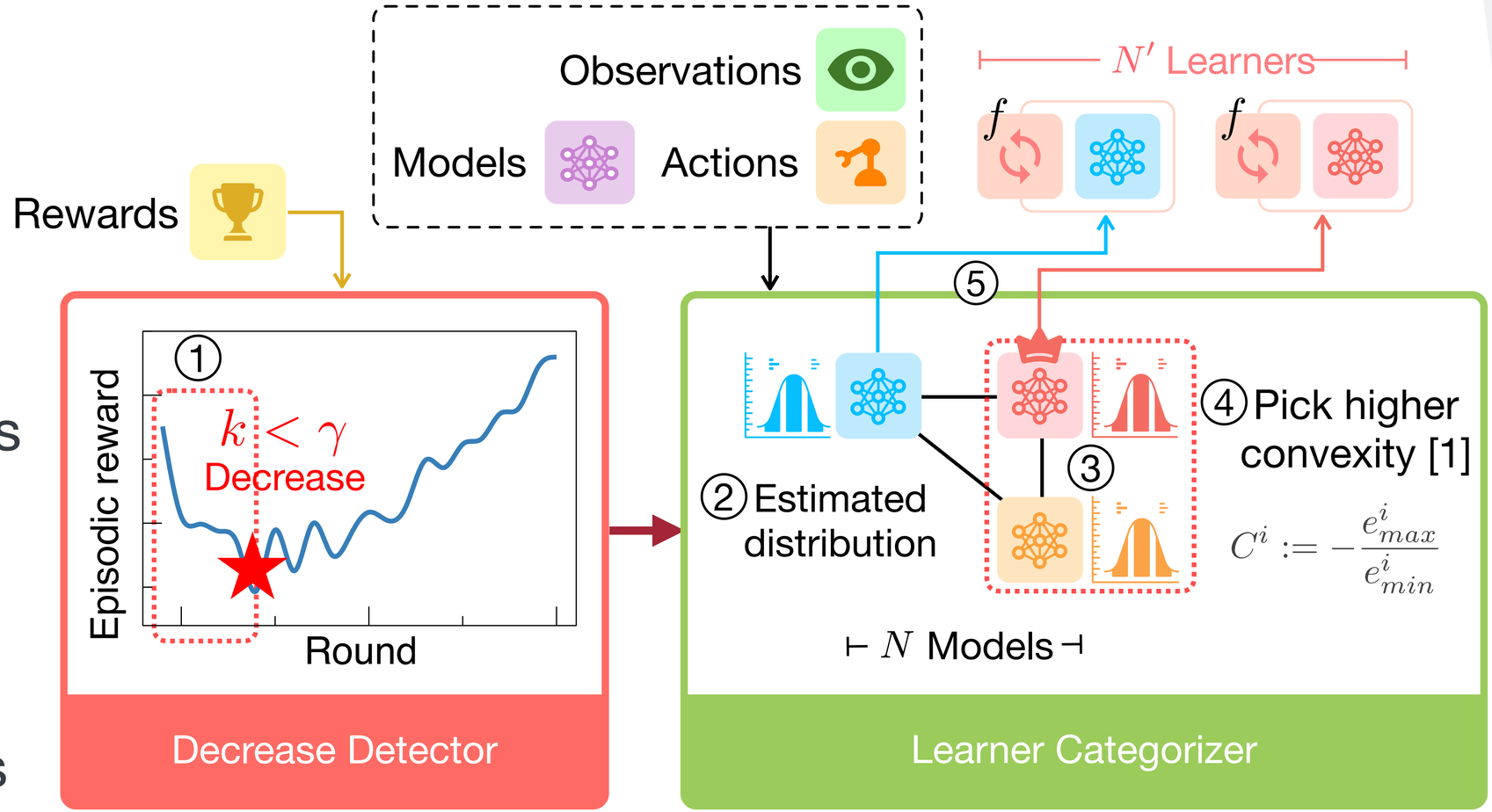
① Check reward trend in recent rounds

② Build distribution estimation

③ Group similar models

④ Select the shared model

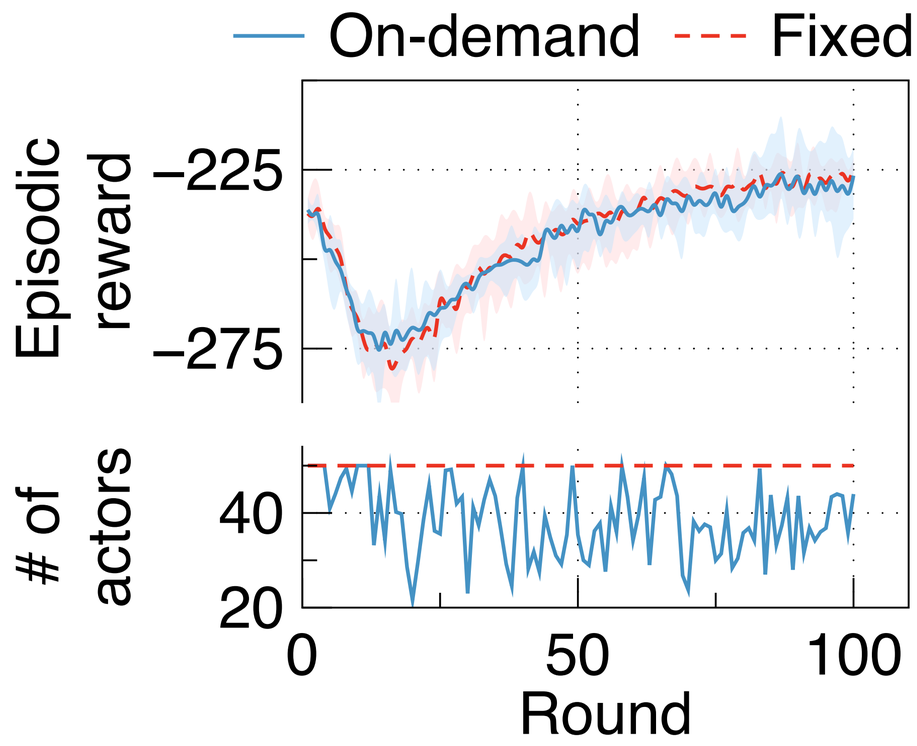
⑤ Launch new learners



[1] H. Yu; et al. Nitro: Boosting Distributed Reinforcement Learning with Serverless Computing. VLDB 2024

Motivation 2: Changing Data Demands

RL policy requires different amount of data across the training [1, 2]



More actors NOT necessarily lead to better training performance

Cost-Aware Actor Scaling

💵 Lower Cost

⏩ Faster Speed

[1] H. Yu; et al. Cheaper and Faster: Distributed Deep Reinforcement Learning with Serverless Computing. AAAI 2024
[2] H. Yu; et al. Nitro: Boosting Distributed Reinforcement Learning with Serverless Computing. VLDB 2024

Challenge 2: Balancing Data Needs

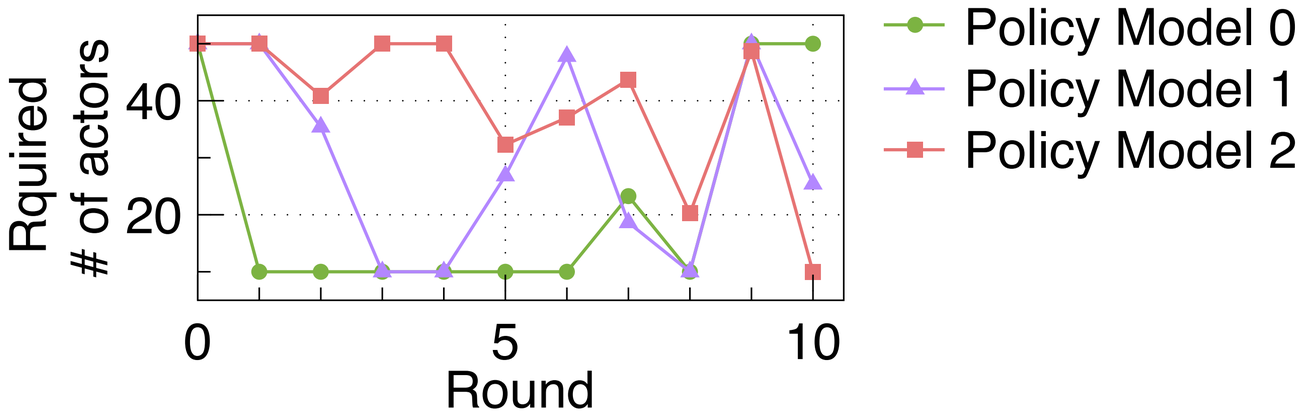
How to balance the varying data needs among agents?

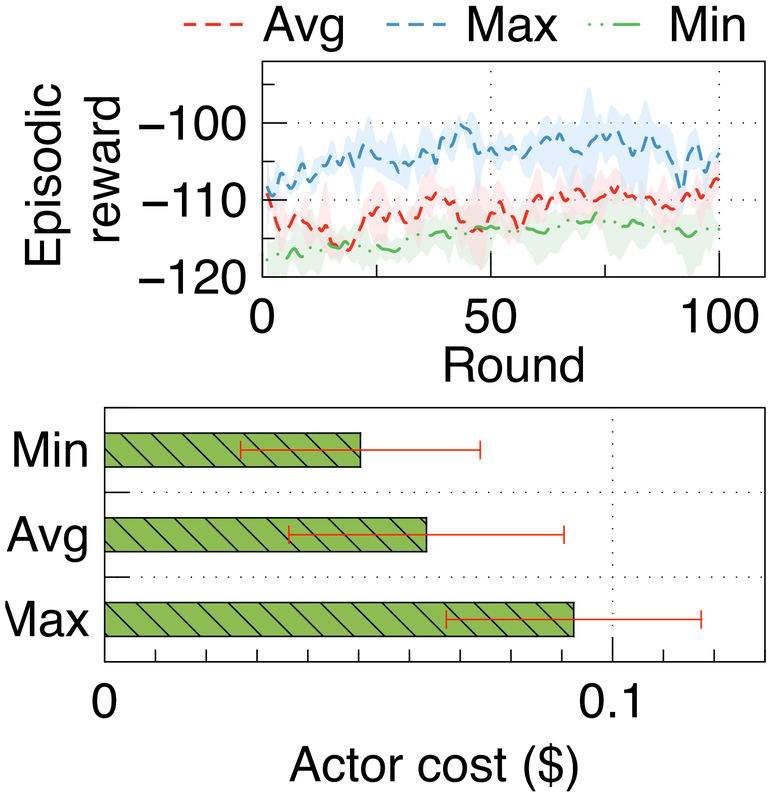
Existing actor scaling methods are for **SARL**

In **MARL**, each policy model can have varying data demands

Over-scaling ➡️ extra costs

Under-scaling ➡️ performance drop

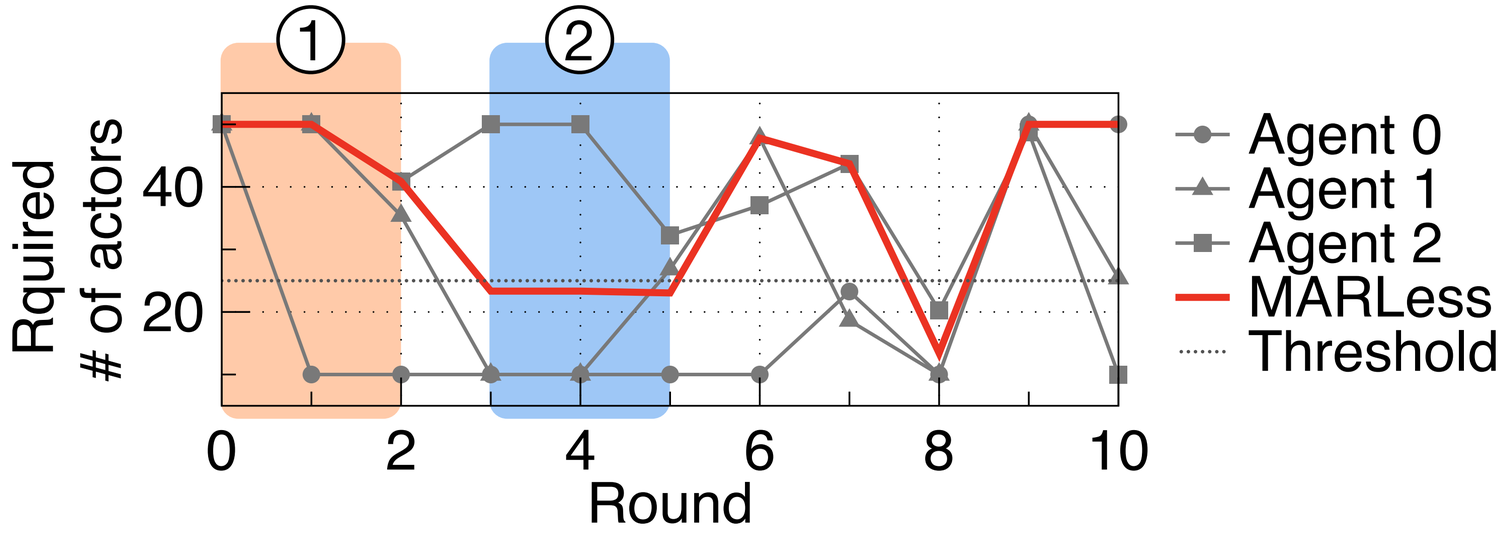




Design 2: Cost-Aware Actor Scaling

Single-Agent: $S_t^i := \frac{C_{max}^i - C_t^i}{C_{max}^i - C_{min}^i}$ $I_t^i := Clip(S_t^i \times I_{max}, I_{min}, I_{max})$

Multi-Agent: $I_t := \begin{cases} \max(I_t^j \mid j \in N'), & \text{if } I_{\text{average}} \geq I_{\text{threshold}}, \\ I_{\text{average}}, & \text{if } I_{\text{average}} < I_{\text{threshold}}, \end{cases}$



Single outlier won't trigger over-provisioning as long as

$$I_{\text{threshold}} \geq \frac{I_{\max} - I_{\min}}{\#\text{ of learners}}$$

Motivation 3: Serverless MARL

Pay-as-you-go

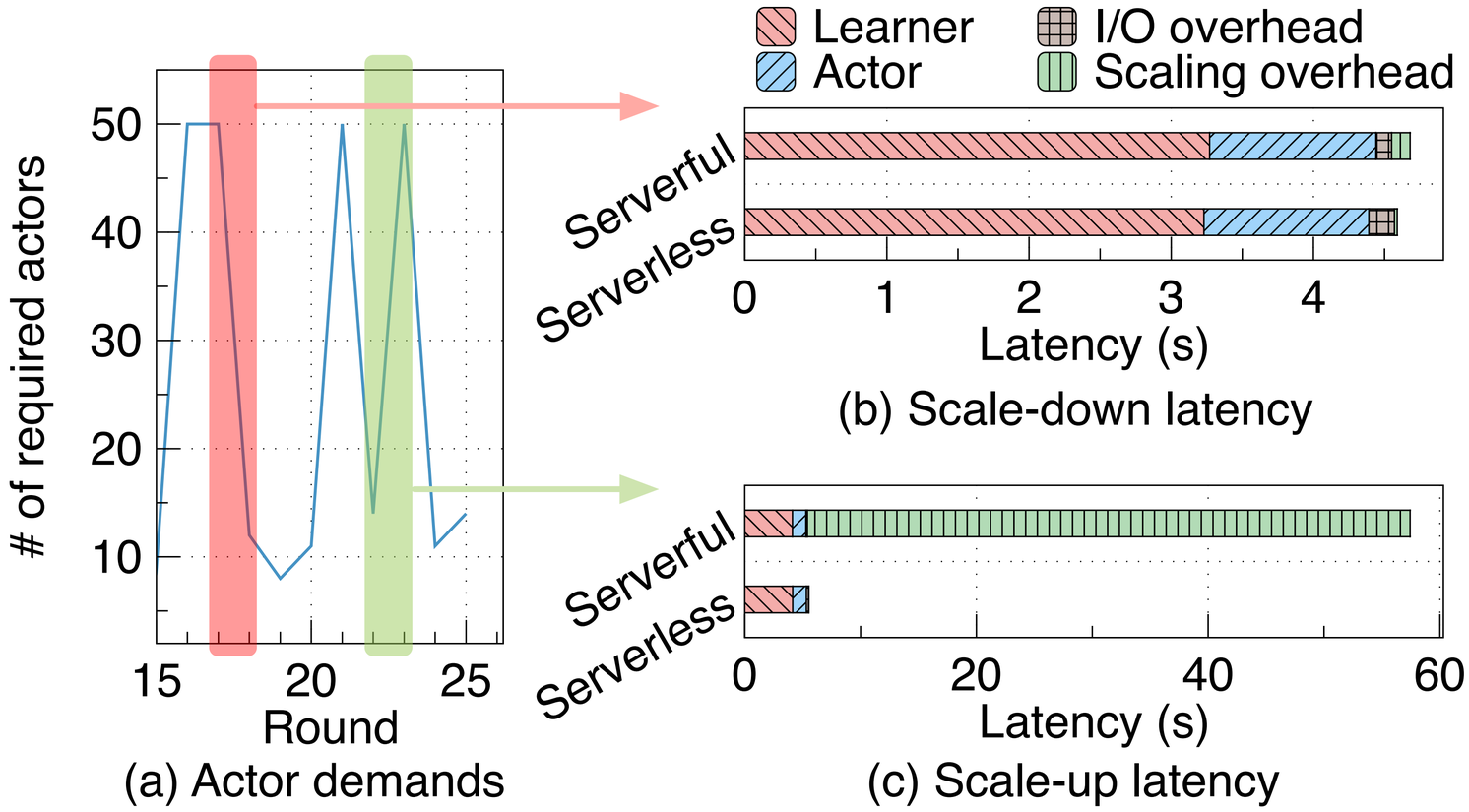
Lower costs

Quick response

Low scale-up overhead

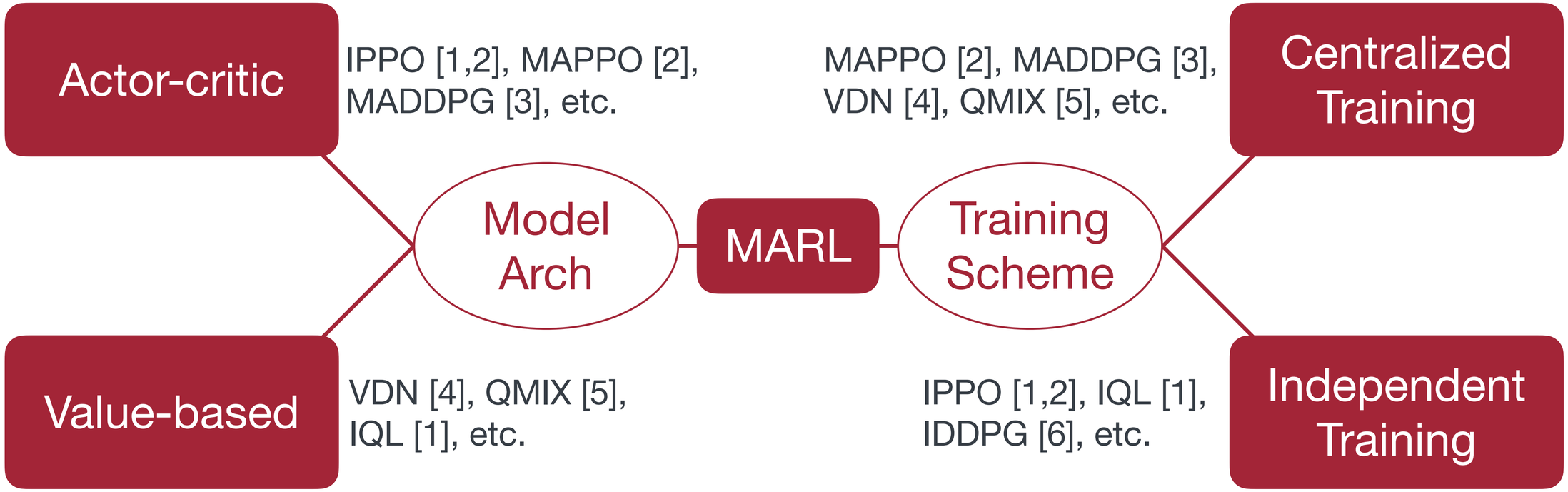
Fine-grained resource management

Scale up/down more flexibly



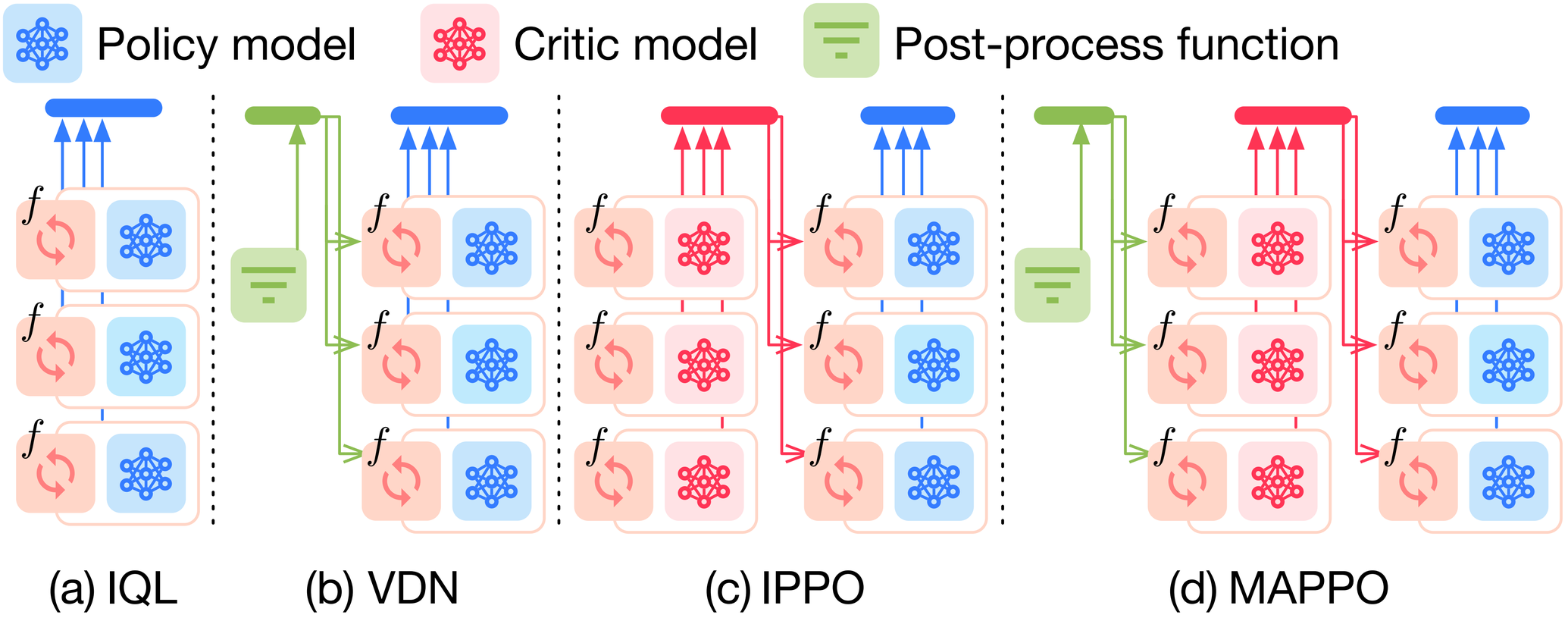
Challenge 3: Compatibility for MARL

How to implement MARL training in a serverless manner while supporting varying algorithms?

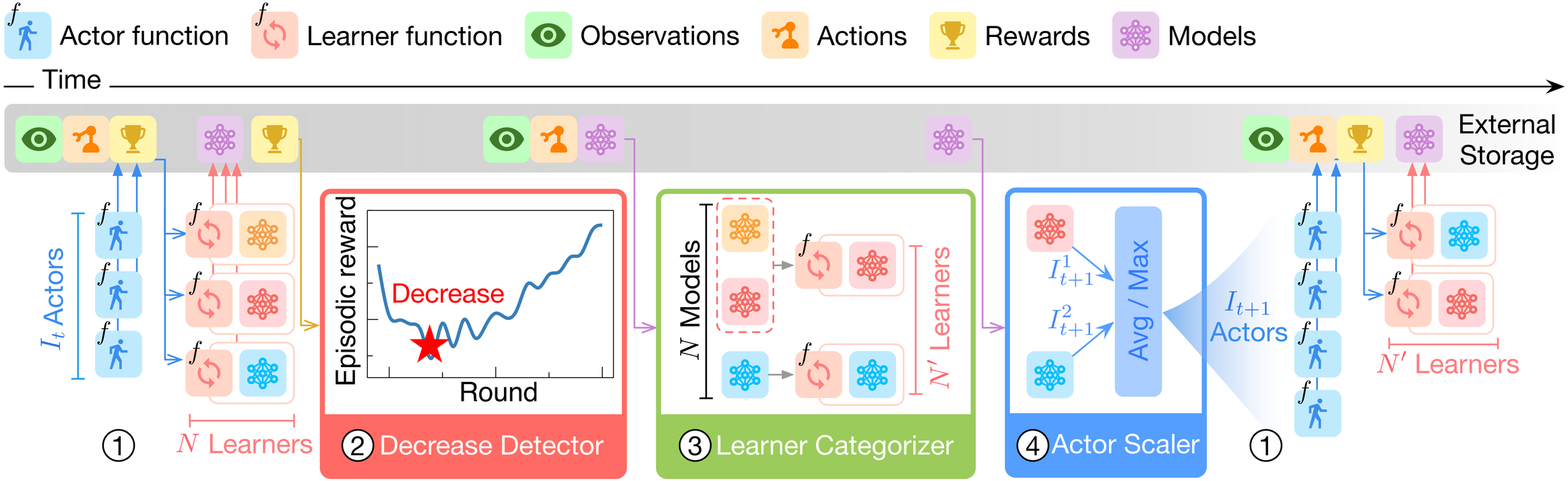


[1] C. Witt; et al. Is Independent Learning All You Need in the StarCraft Multi-Agent Challenge? Preprint
[2] C. Yu; et al. The Surprising Effectiveness of PPO in Cooperative, Multi-Agent Games. NIPS 2022
[3] R. Lowe; et al. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. NIPS 2017
[4] P. Sunehag; et al. Value-Decomposition Networks For Cooperative Multi-Agent Learning. AAMAS 2018
[5] T. Rashid; et al. QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning. PMLR 2018
[6] T. Lillicrap; et al. Continuous control with deep reinforcement learning. ICLR 2016

Design 3: MARL-Algorithm Compatible Design



System Overview



① Sampling & training
② Decrease detection
③ Agent categorization for sharing
④ Scaling actors

Evaluation

Implementation

Ray RLlib
Docker Containers
AWS EC2

Baselines

Ray RLlib [1]
MARLlib [2]

Metrics

Episodic reward
Training cost ($)

Benchmarks

MPE [3]

Simple-spread
Simple-adversary

SMAC [4]

8m
3s5z

[1] E. Liang; et al. RLlib: Abstractions for Distributed Reinforcement Learning. ICML 2018
[2] S. Hu; et al. MARLlib: A Scalable and Efficient Multi-agent Reinforcement Learning Library. PMLR 2023
[3] R. Lowe; et al. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. NIPS 2017
[4] M. Samvelyan; et al. The StarCraft Multi-Agent Challenge. NIPS 2019

Testbeds & Clusters

Default Testbed

AWS *c6a.16xlarge* instance,

64 vCPU cores

128 GB memory

AWS *g5.16xlarge* instance,

8 NVIDIA A10G tensor cores

24 GB VRAM

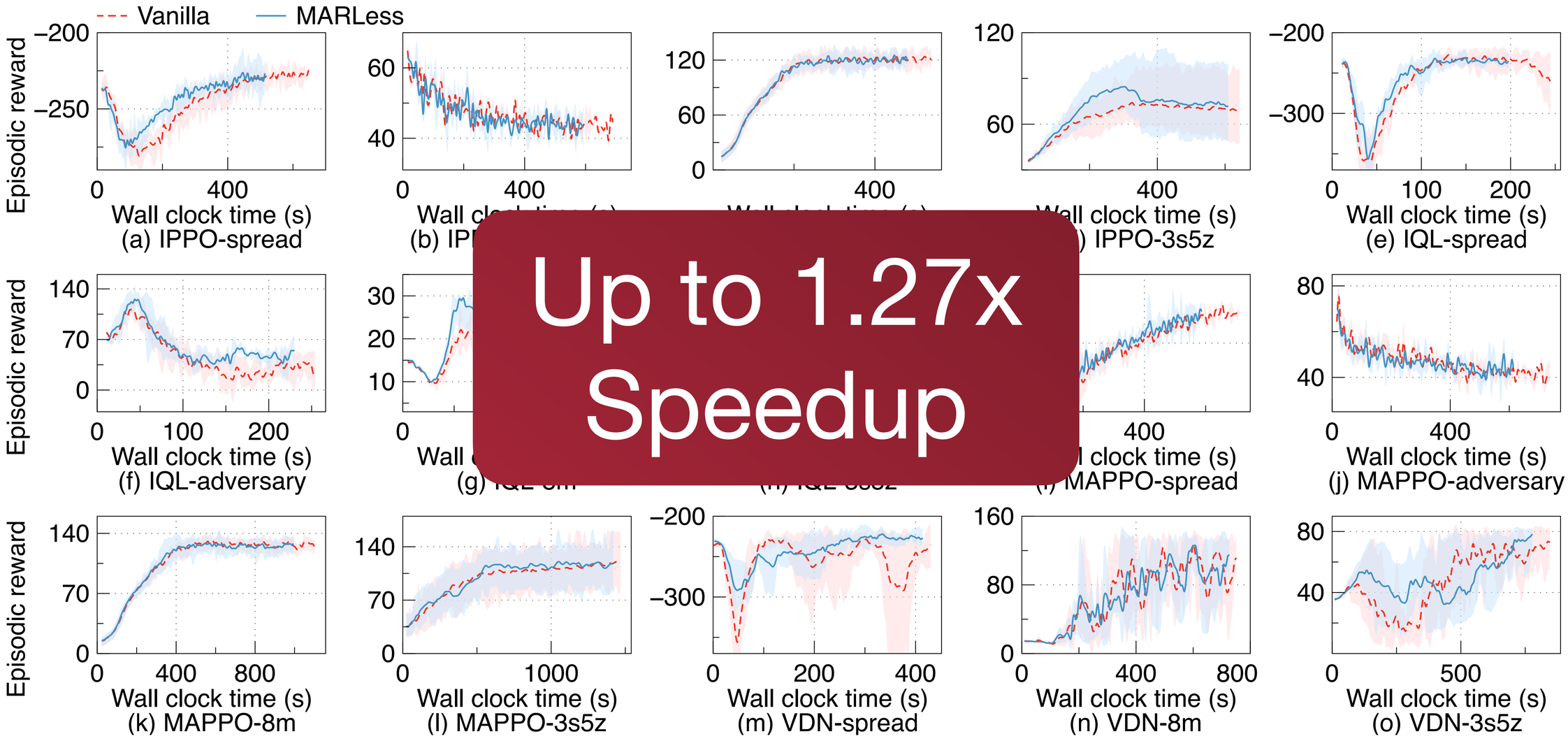
Large-Scale Testbed

15 x AWS *c6a.32xlarge* instances

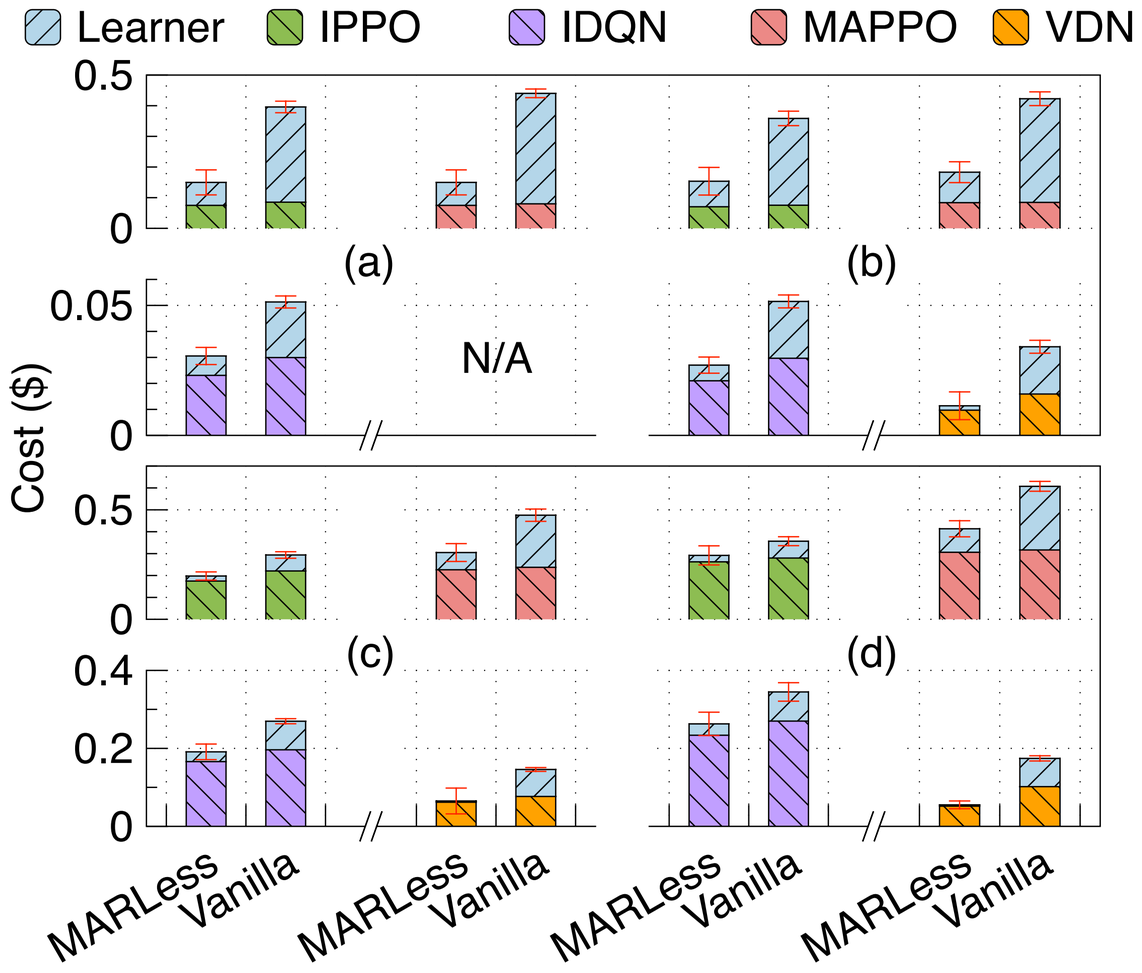
1,920 vCPU cores

3,840 GB memory

Overall Performance - Algorithm Baselines



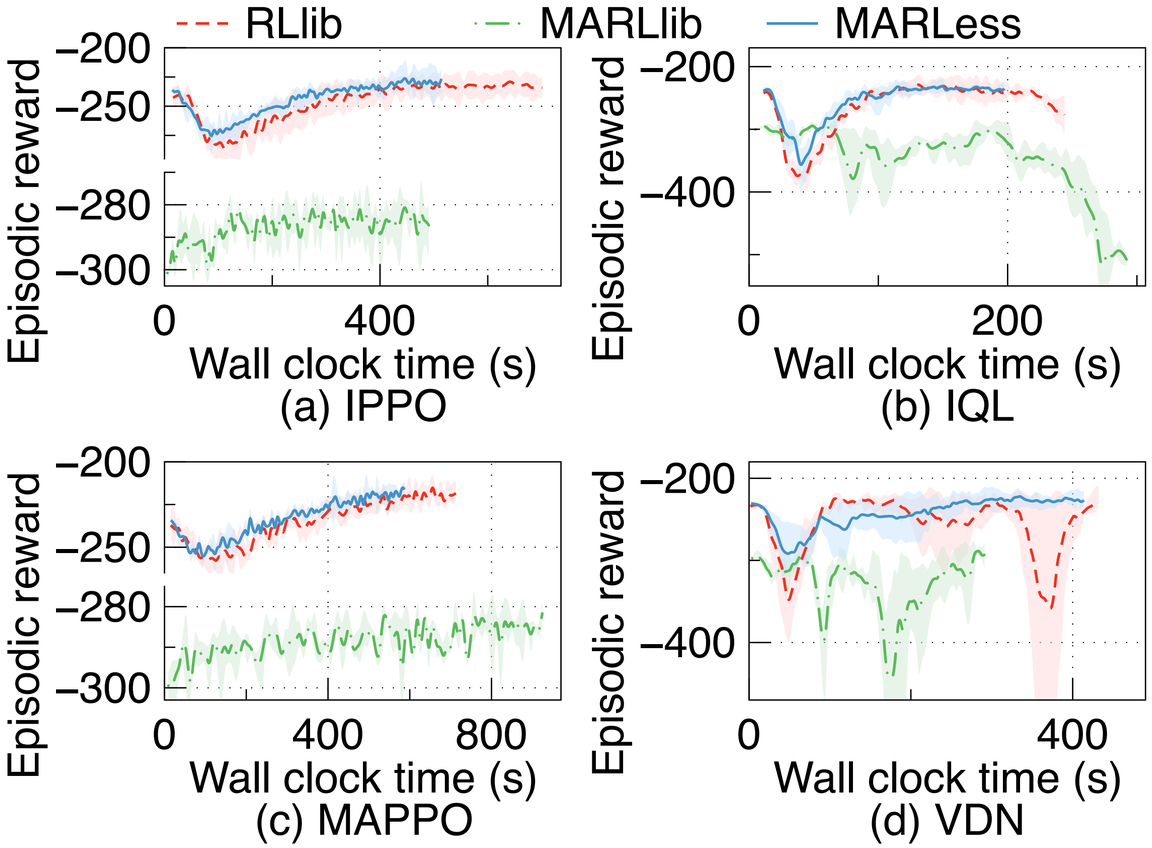
Overall Performance - Algorithm Baselines

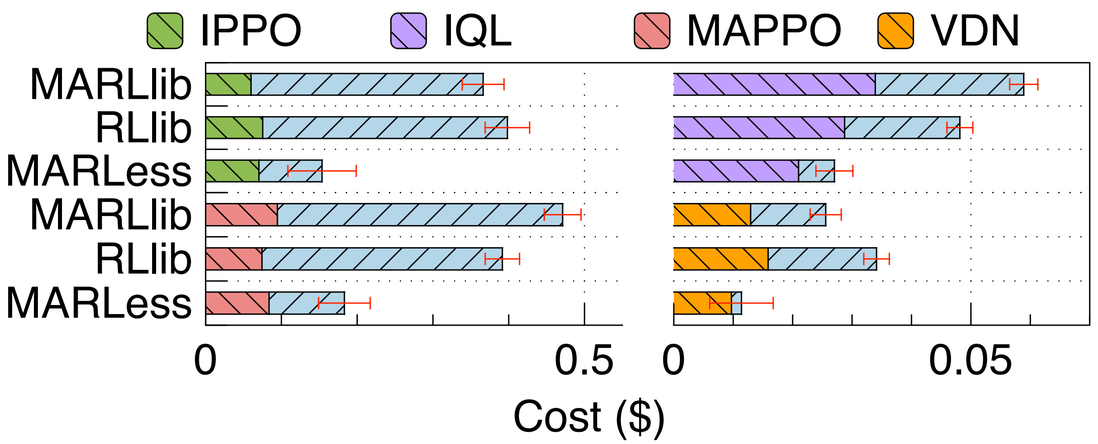


(a)simple-adversary
(b)simple-spread
(c) 8m
(d) 3s5z

Up to 68% Cost Reduction

Overall Performance - System Baselines

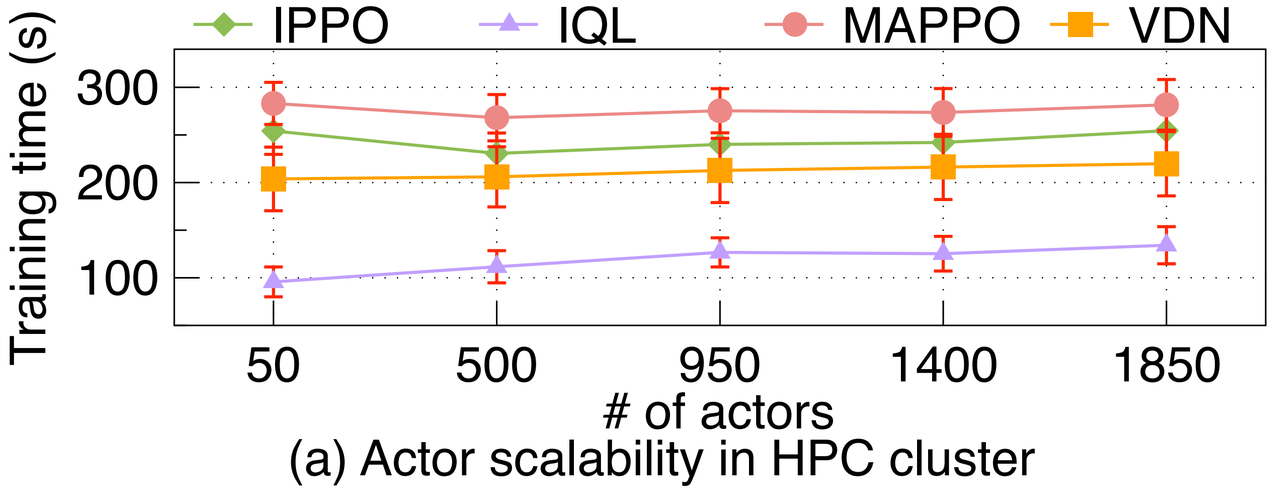




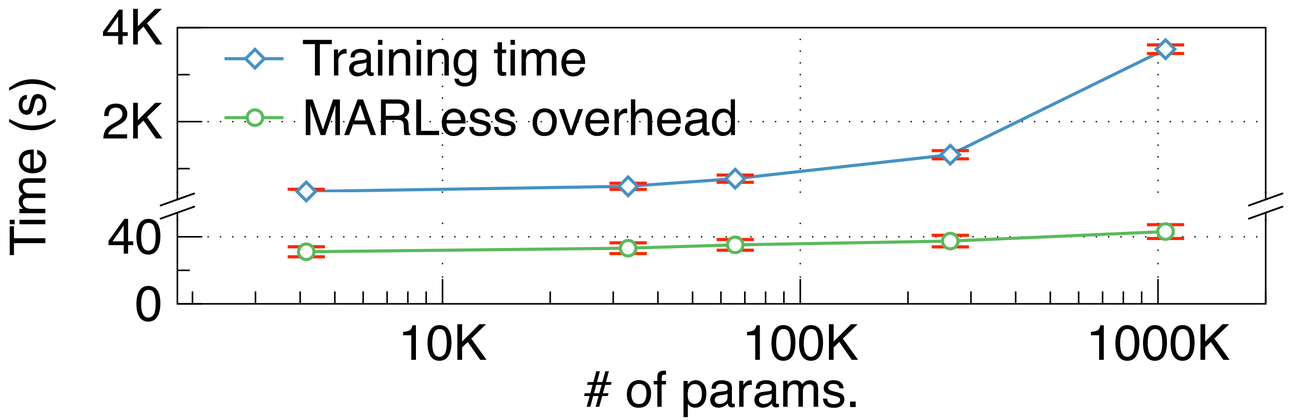
Up to 1.36x Speedup

Up to 67% Cost Reduction

Scalability



(a) Actor scalability in HPC cluster



(b) Model size scalability

Large-Scale Testbeds:

15 x AWS *c6a.32xlarge* instances

Each with

128 vCPU cores

256 GB memory

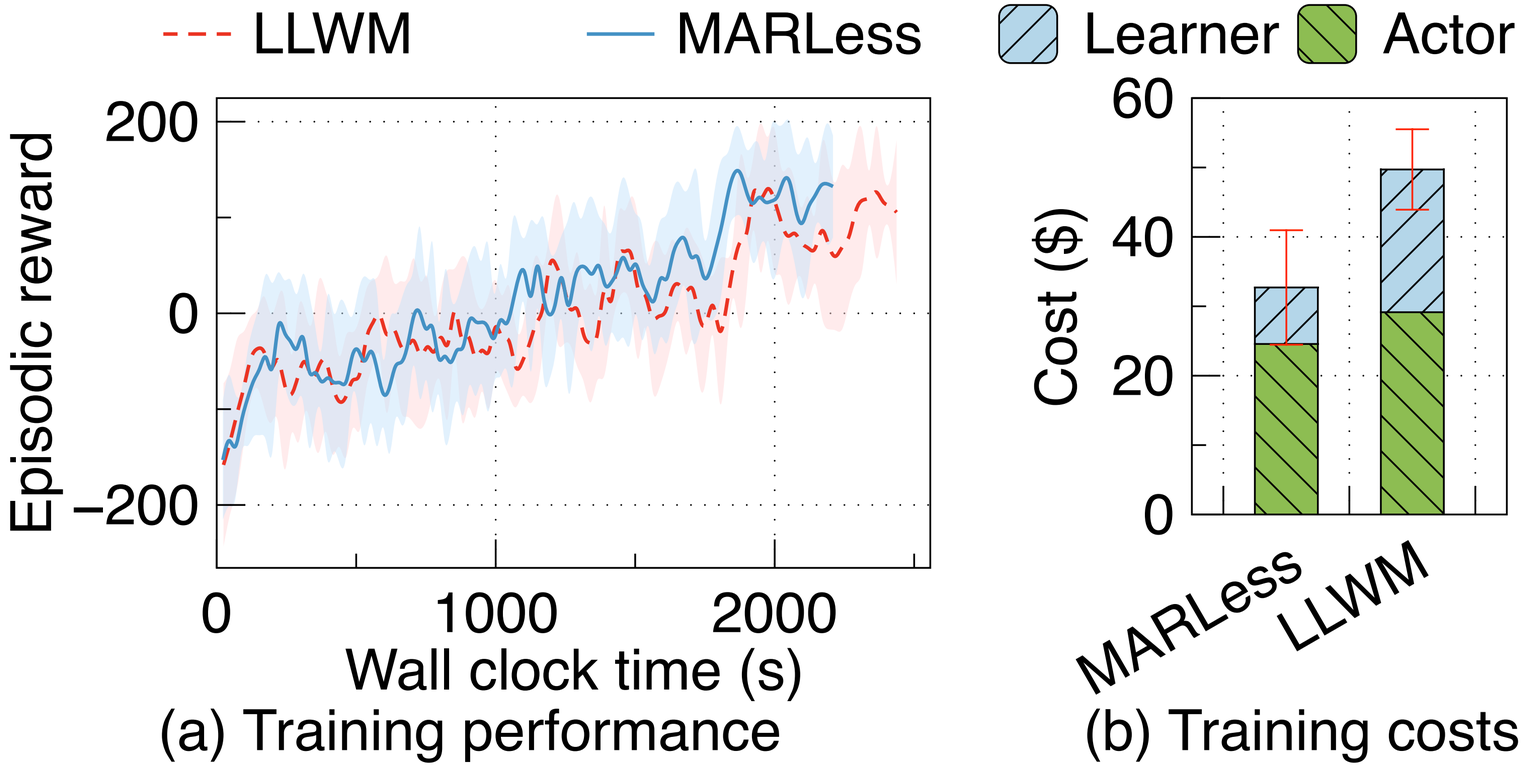
Algorithms:

IPPO, **IQL**, **MAPPO**, **VDN**

Environment:

MPE simple-spread

Improving Large-Scale Scientific MARL [1]



[1] H. Bae; et al. Scientific Multi-agent Reinforcement Learning for Wall-models of Turbulent Flows. Nature Communications 2022 Vol. 13

Dynamic Learner Sharing

Actor Scaling

Multi-Algorithm Compatible

MARLess

Compared to algorithm baselines,

Up to 1.27x

Training speed improvement

Up to 68%

Training cost reduction

STEVENS
INSTITUTE OF TECHNOLOGY
1870

IntelliSys Lab

NAIRR Pilot

THANK YOU

NSF

aws

Jetstream2

Stevens Institute of Technology
1 Castle Point Terrace, Hoboken, NJ 07030