SC '24, Nov 17–22, 2024, Atlanta, USA

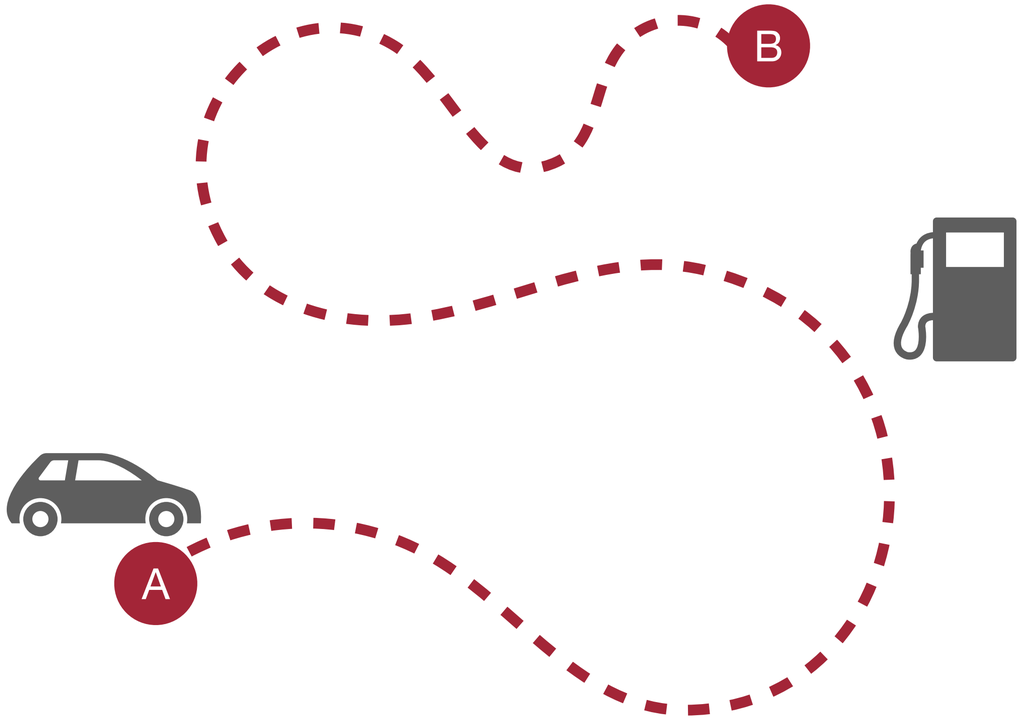
Stellaris: Staleness-Aware Distributed Reinforcement Learning with Serverless Computing

Hanfei Yu[1], Hao Wang[1], Devesh Tiwari[2], Jian Li[3], Seung-Jong Park[4]

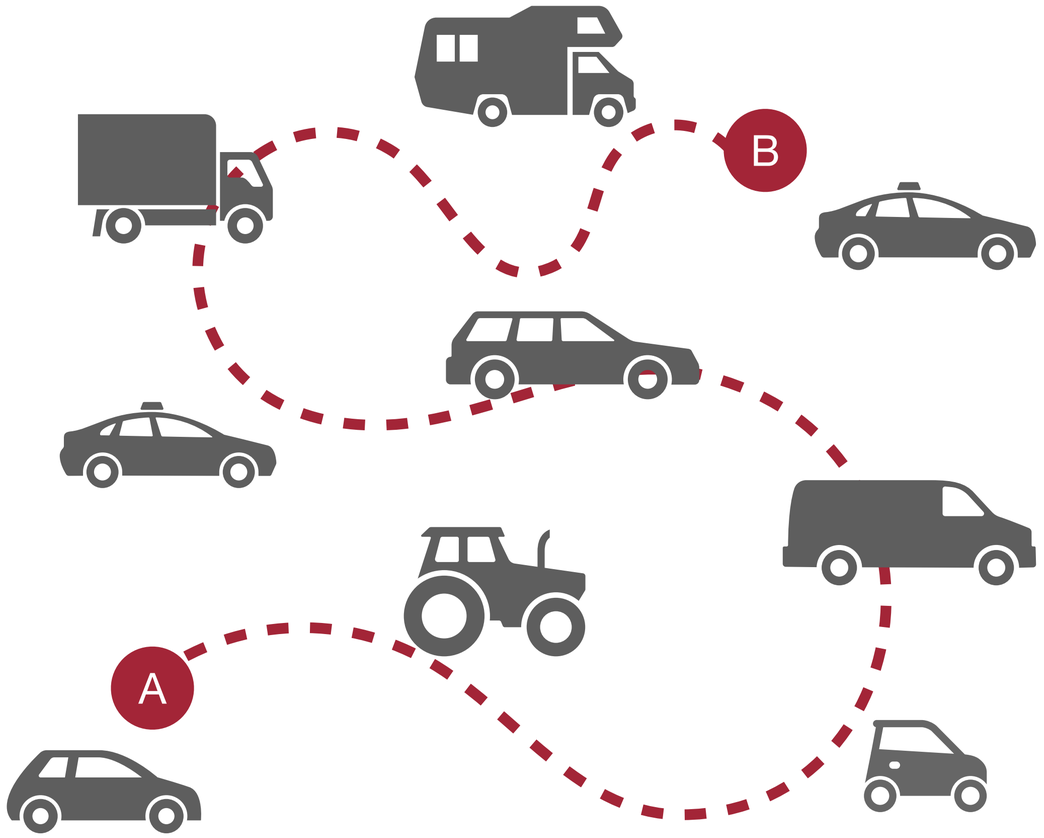
Stevens Institute of Technology[1], Northeastern University[2], Stony Brook University[3], Missouri University of Science & Technology[4]

IntelliSys Lab

Cloud / HPC → Serverless

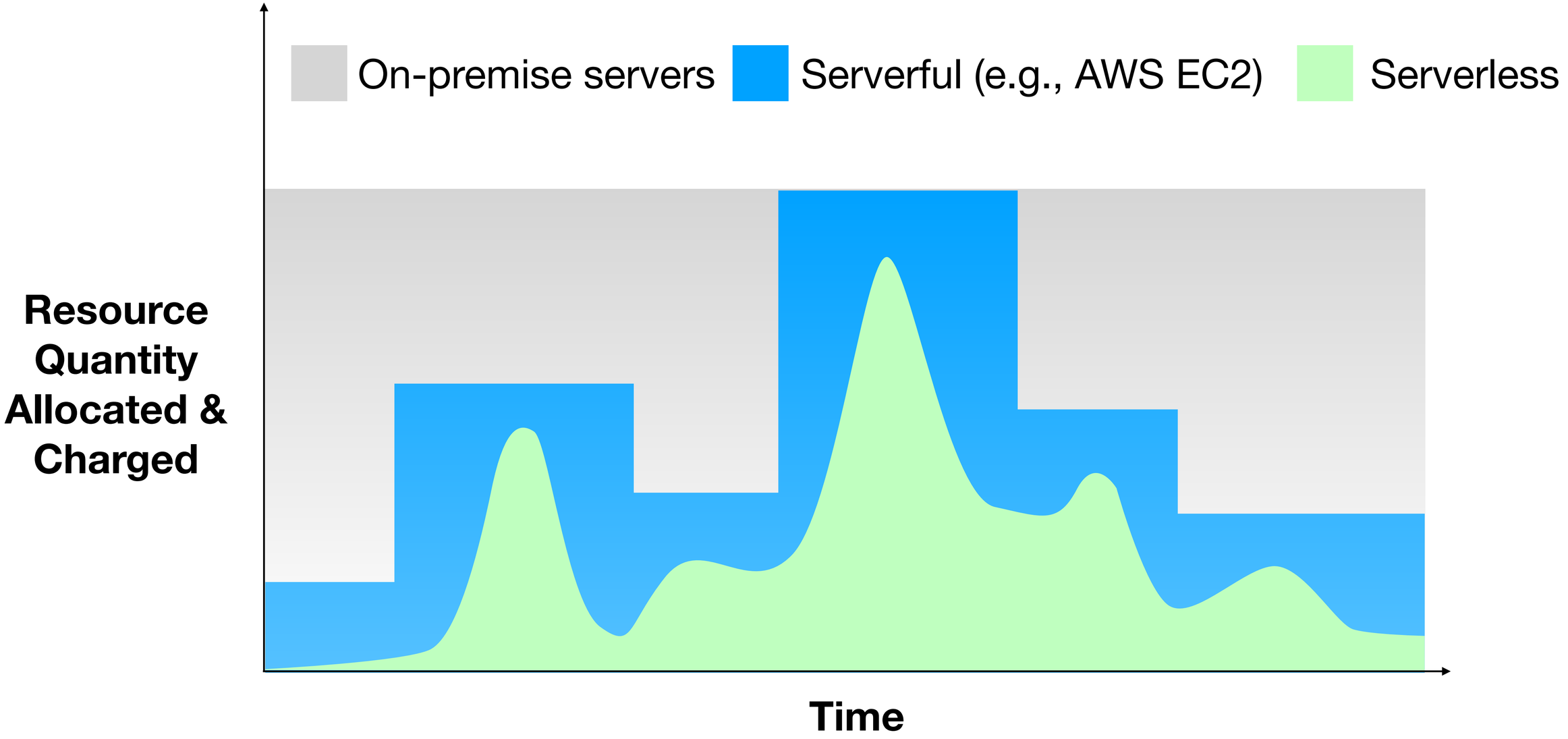


Car rental



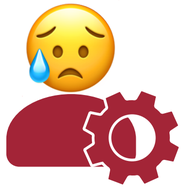
Uber/Lyft

From *A Serverless Vision for Cloud Computing*
by Prof. Ana Klimovic



Schleier-Smith, Johann, et al. "What serverless computing is and should become: The next phase of cloud computing." Communications of the ACM 64.5 (2021): 76-84.

Cloud / HPC

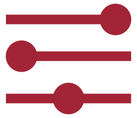


Provider

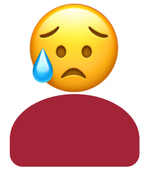
Server / VM hosting

Software Installation

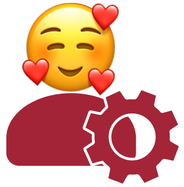
Workload management



Resource configuration



User



Provider

Software standardization

Workload scheduling

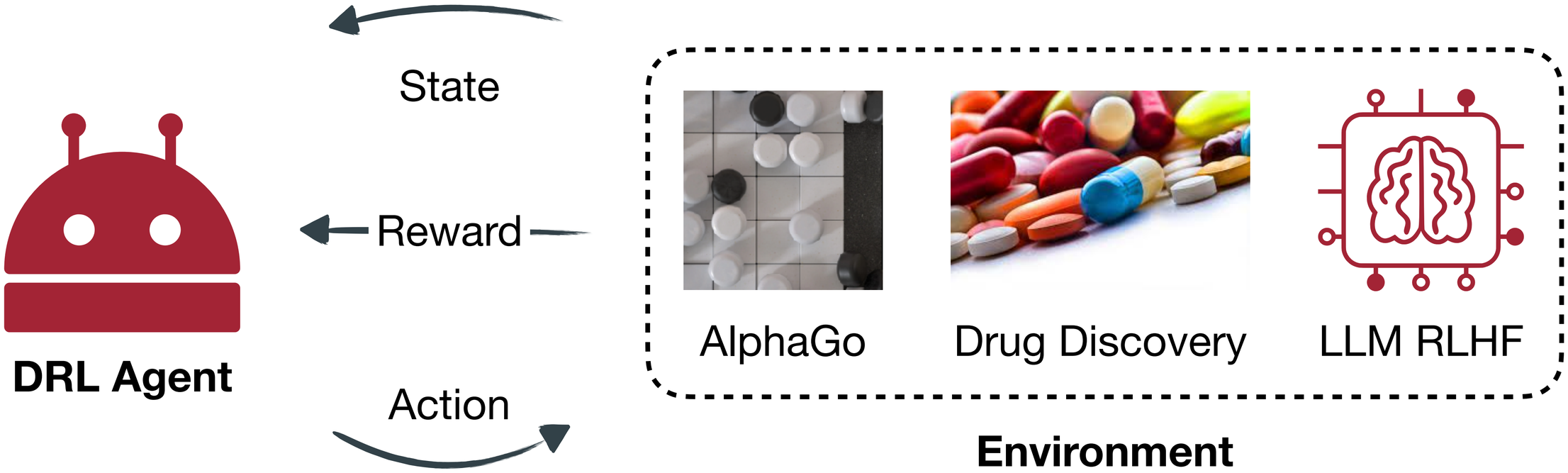
Resource optimization

User

Function invocation

Serverless

Deep Reinforcement Learning (DRL)



DRL in HPC Community

RLScheduler: An Automated HPC Batch Job Scheduler Using Reinforcement Learning

Di Zhang[1], Dong Dai[1], Youbiao He[2], Forrest Sheng Bao[2], and Bing Xie[3]

[1]Computer Science Department, University of North Carolina at Charlotte, {dzhang16, ddai}@uncc.edu
[2]Computer Science Department, Iowa State University, {yh54, fsb}@iastate.edu
[3]Oak Ridge Leadership Computing Facility, Oak Ridge National Laboratory. xieb@ornl.gov

Reinforcement Learning Strategies for Compiler Optimization in High level Synthesis

Hafsah Shahzad*, Ahmed Sanaullah**, Sanjay Arora**, Robert Munafo*, Xiteng Yao*, Ulrich Drepper**, and Martin Herbordt*

*CAAD Lab, ECE Department, Boston University
**Red Hat Inc.

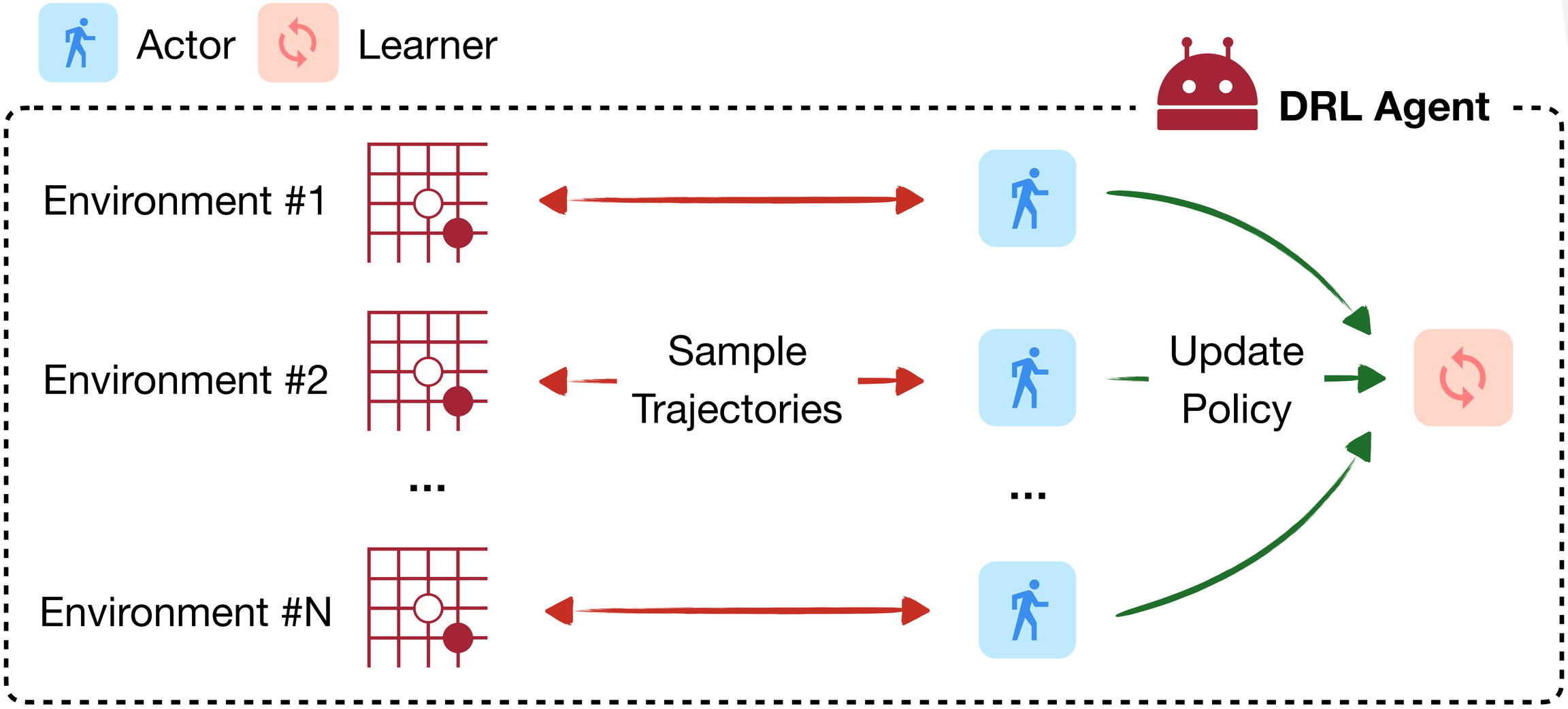
Optimizing and Extending the Functionality of EXARL for Scalable Reinforcement Learning

SAI CHENNA*, KATHERINE COSBURN*, UCHENNA EZEOBI*, MAXIM MORARU*, HYUN LIM*, JULIEN LOISEAU*, JAMAL MOHD-YUSOF*, ROBERT PAVEL*, VINAY RAMAKRISHNAIAH*, ANDREW REISNER*, and KAREN TSAI*, Los Alamos National Laboratory, USA

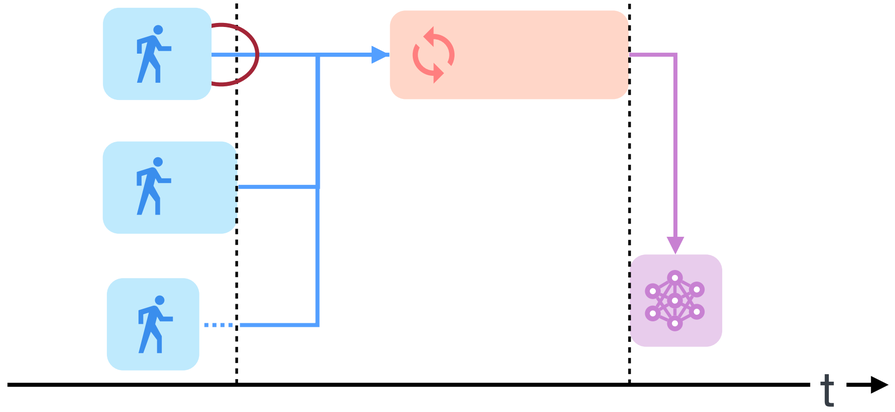
Facilitating DRL

Optimizing DRL

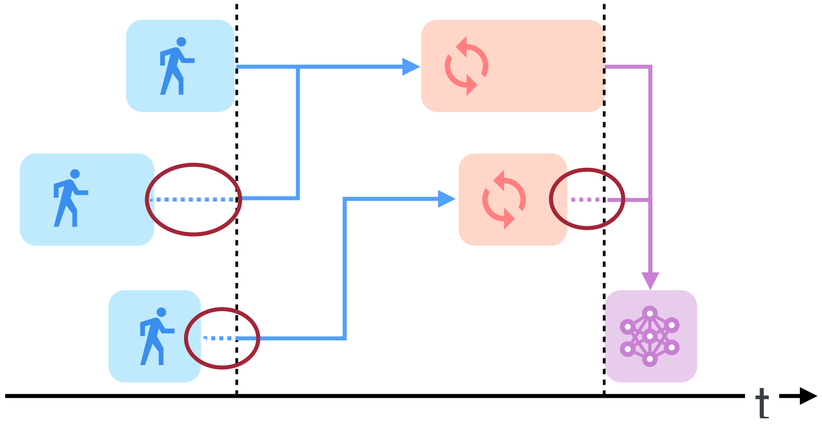
Scaling DRL Training: Actor-Learner



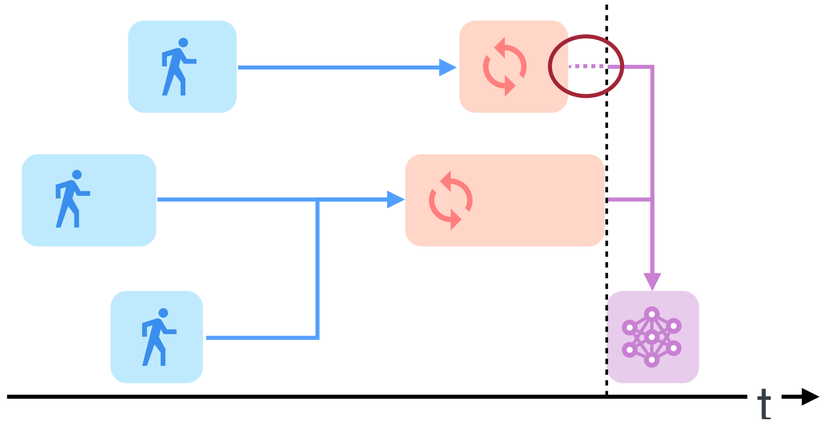
Barrier Actor Learner Policy update



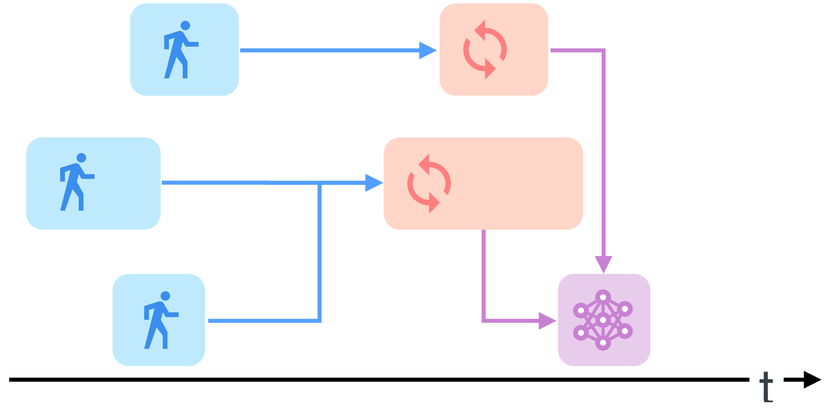
Synchronous Actors + Centralized **Sync**. Le



Synchronous Actors + **Sync**hronous Learners



Asynchronous Actors + **Sync**hronous Lear



(**A)sync**hronous Actors + **Async**hronous Learners

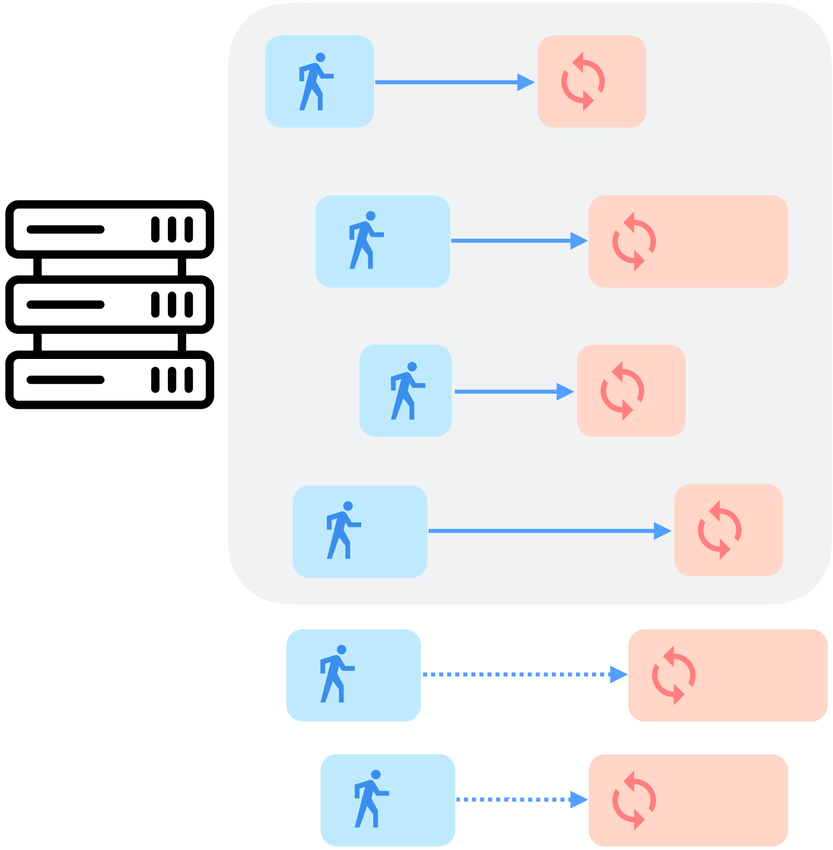
How Does Serverless Benefit DRL?

Actor

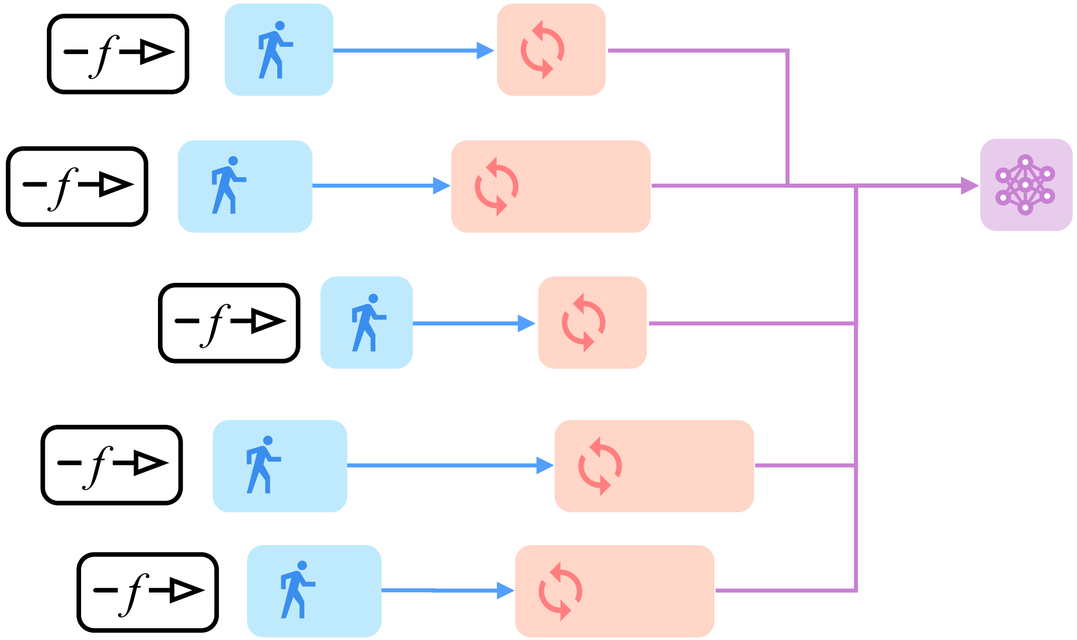
Learner

Policy update

😢 **VMs or servers** reaching capacity



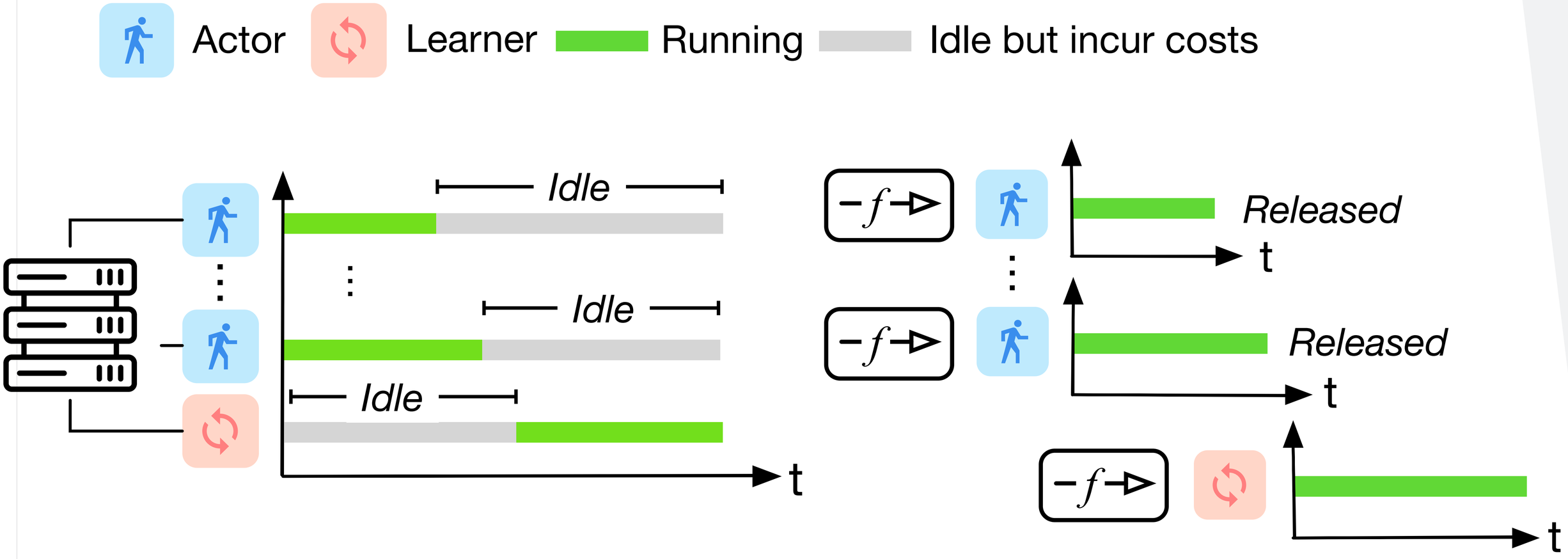
Serverful DRL



🥰 **Serverless functions** instantly scale up

Serverless DRL

Serverful vs. Serverless DRL Training

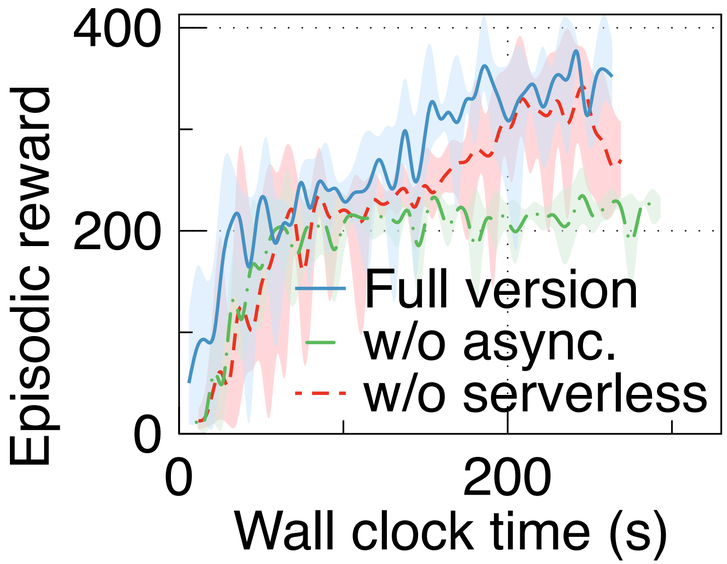


Serverful Training

Serverless Training

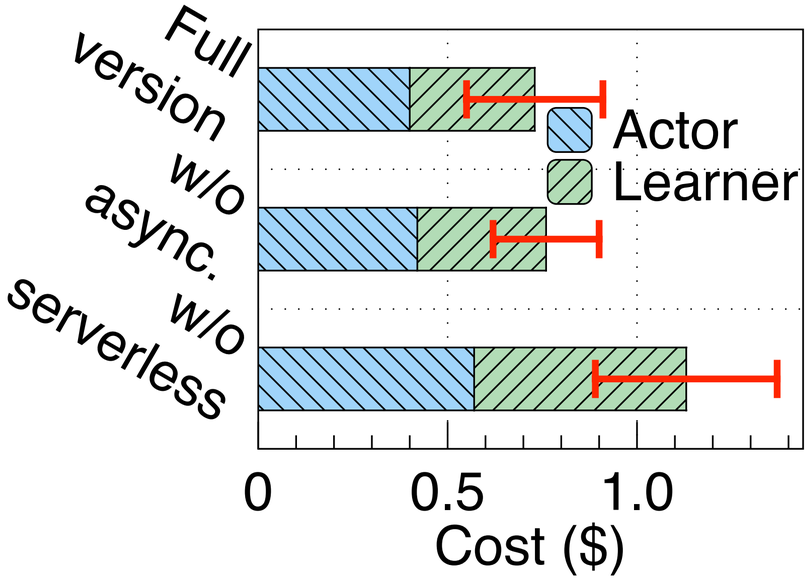
Asynchronous and Serverless DRL Training

High training efficiency



(a) Training performance

Low training cost



(b) Training cost

Proximal Policy Optimization (PPO) on MuJoCo Hopper-v4

Existing Works

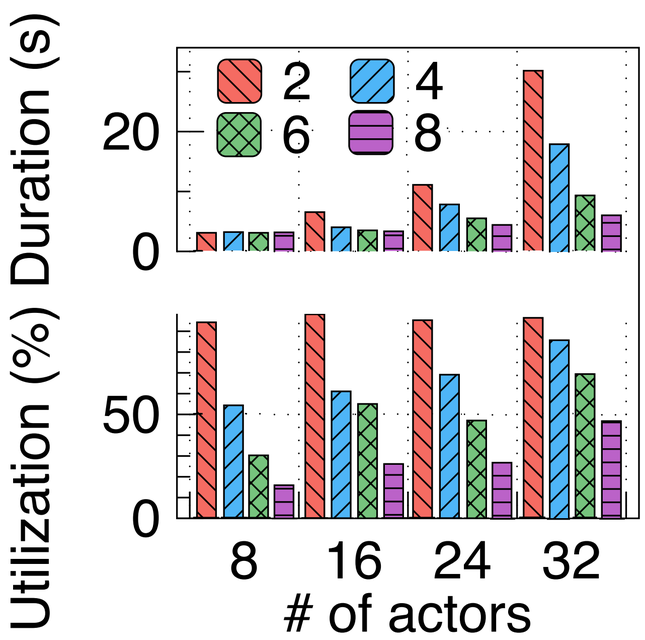
Framework	Asynchronous Learners	Scalable Actors	On-policy and Off-policy	Serverless
Ray RLlib (ICML 2018)	✗	✗	✓	✗
MSRL (ATC 2023)	✗	✗	✓	✗
SEED RL (ICLR 2020)	✗	✗	✓	✗
SRL (ICLR 2024)	✗	✗	✗	✗
MinionsRL (AAAI 2024)	✗	✓	✗	✓
Stellaris (SC 2024)	✓	✓	✓	✓

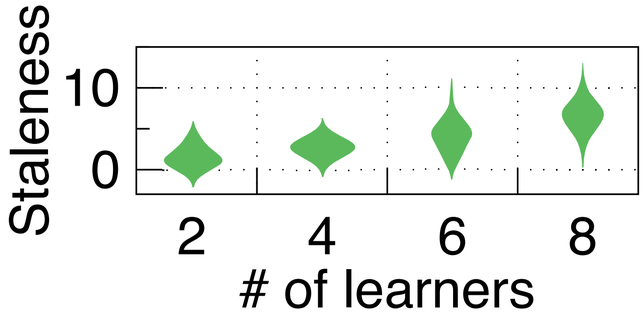
Challenges

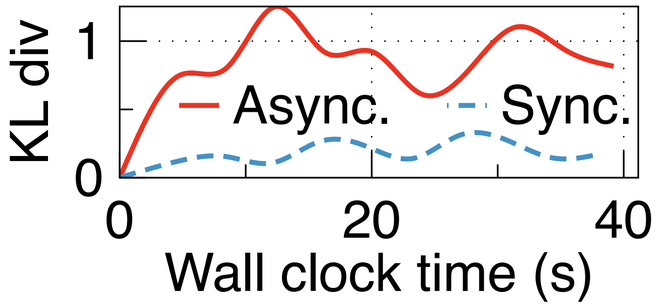
Dynamic learner orchestration and usage

Dynamic staleness

Unstable policy updates







KL div: Kullback–Leibler divergence

Design Goals

Dynamic learner orchestration

On-Demand Serverless Learners

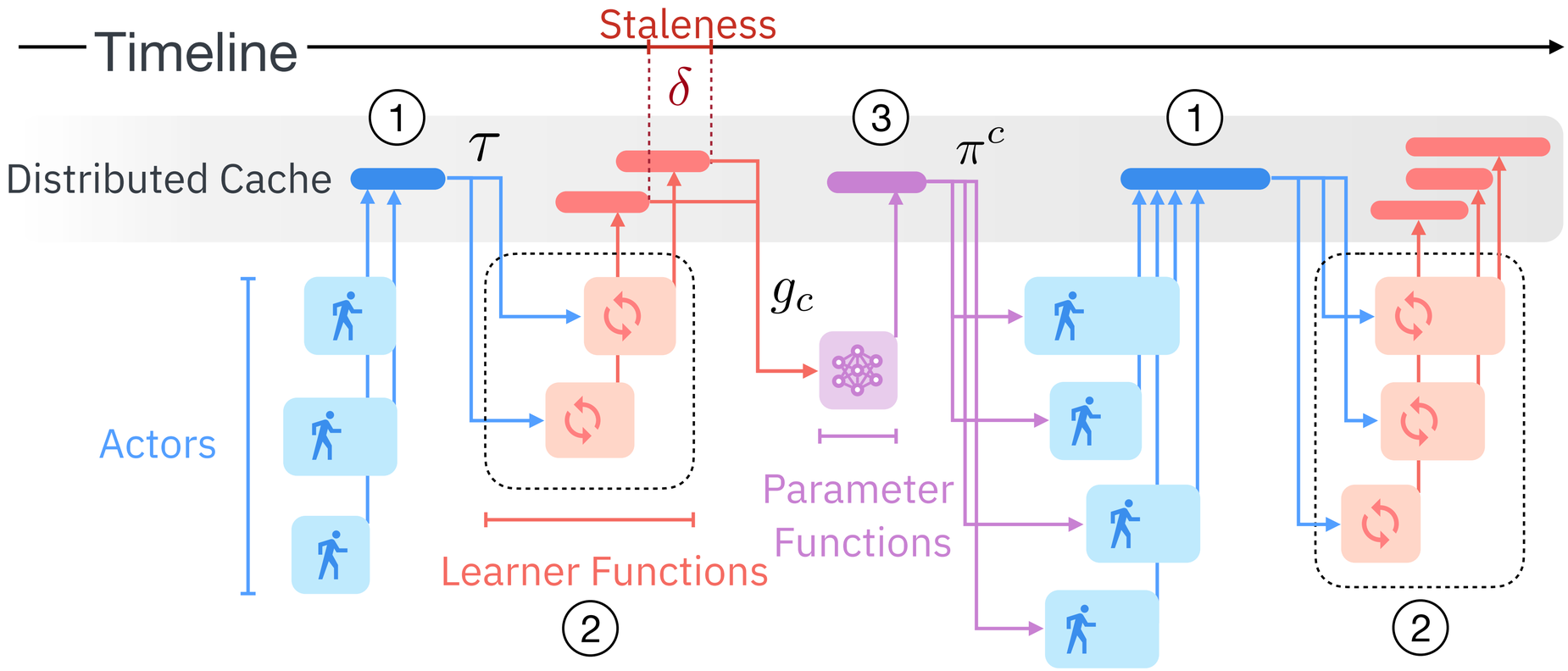
Dynamic staleness

Staleness-Aware Gradient Aggregation

Unstable policy updates

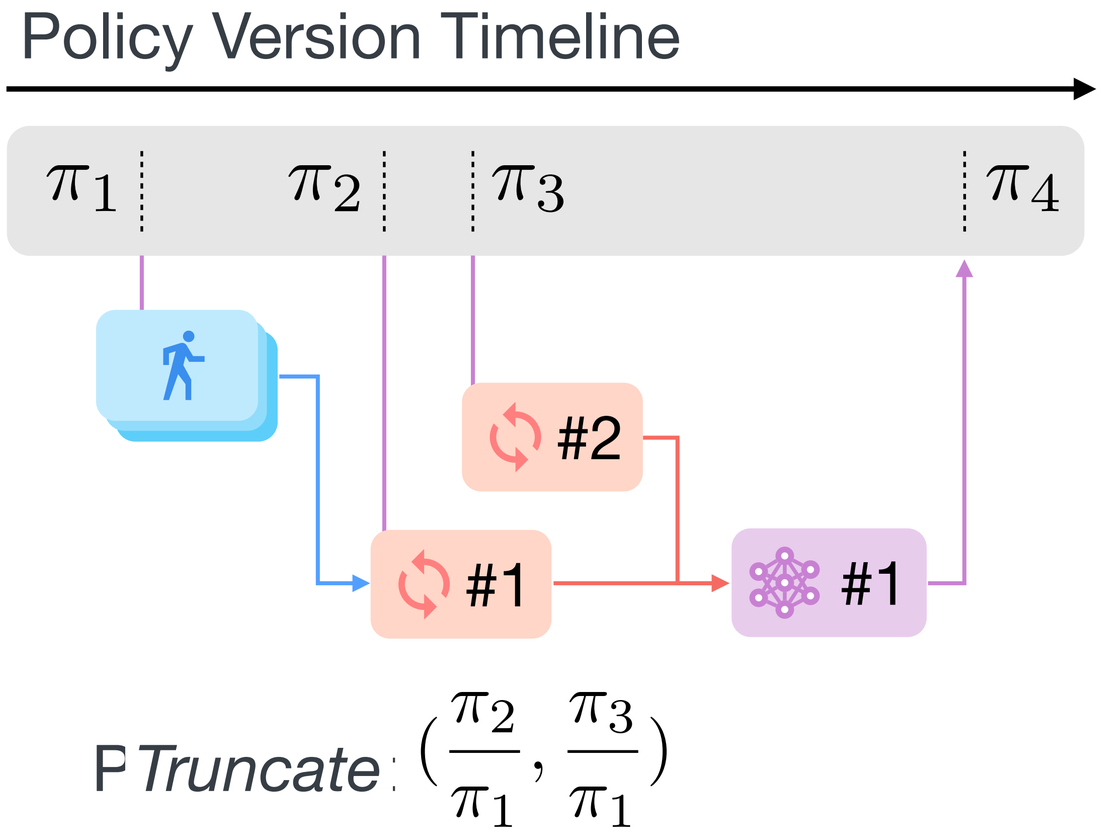
Global Importance Sampling Truncation

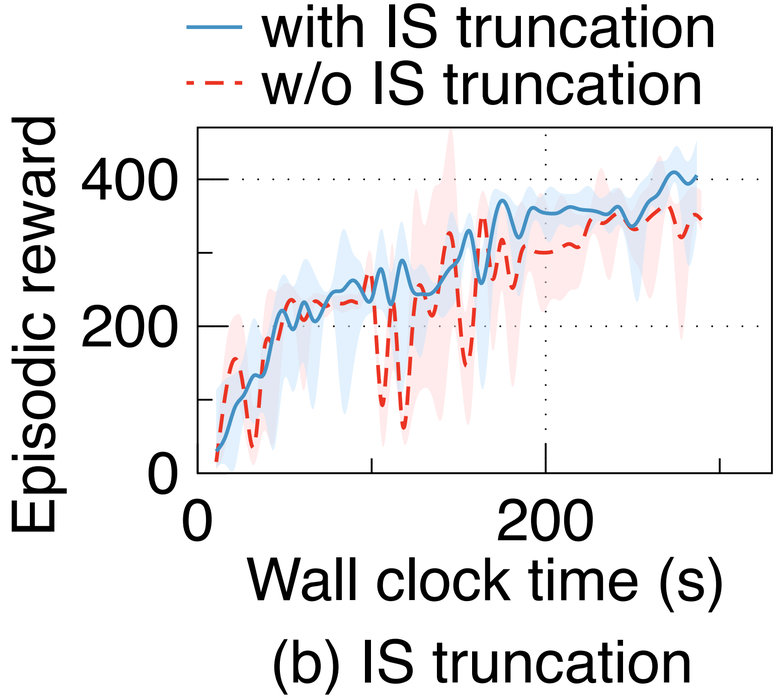
Stellaris



1. Importance sampling driven trajectory collection
2. On-demand gradient calculation
3. Staleness-aware gradient aggregation

Global Importance Sampling Truncation

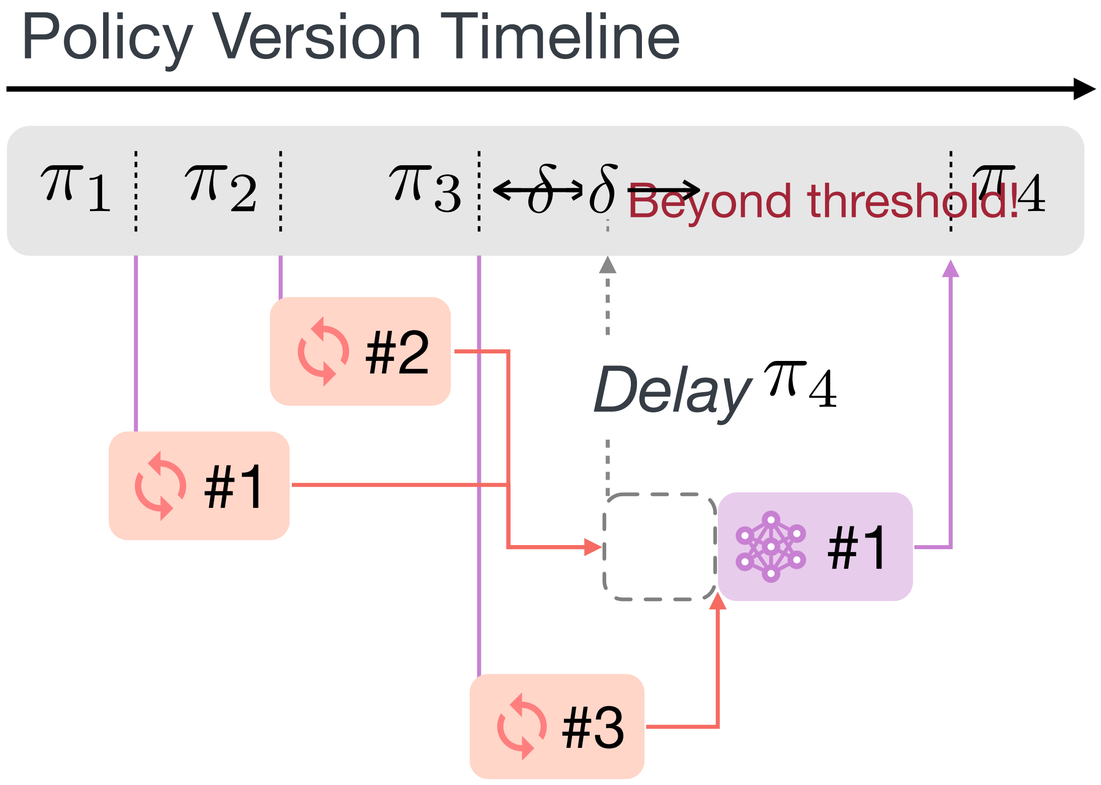


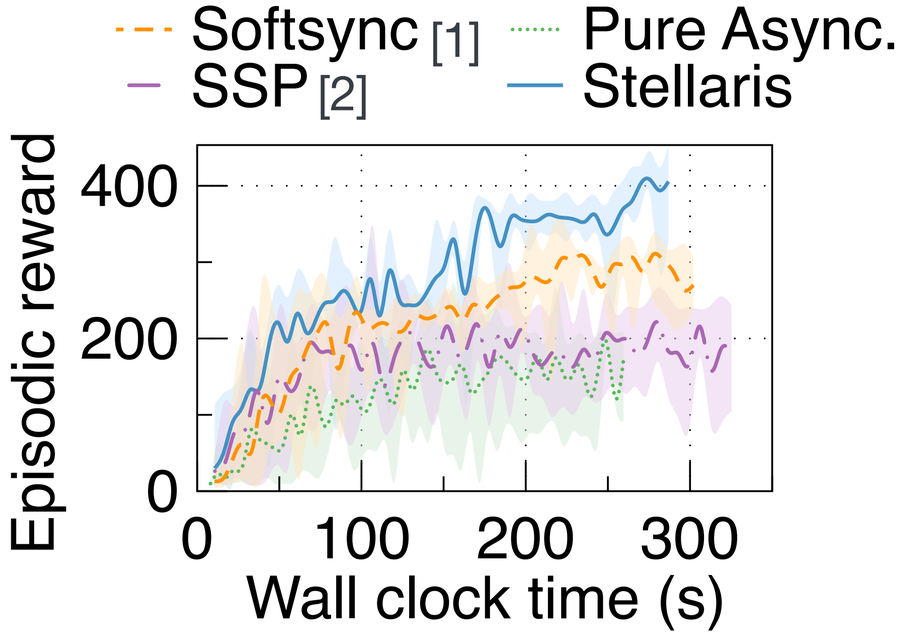


(b) IS truncation

Actor Learner Function Parameter Function Delay

Staleness-Aware Gradient Aggregation





(a) Gradient aggregation

Actor Learner Function Parameter Function Delay

[1] Zhang, W. et al. “Staleness-Aware Async-SGD for Distributed Deep Learning”. IJCAI. 2016.
[2] Ho, Qirong, et al. "More Effective Distributed ML via a Stale Synchronous Parallel Parameter Server." NeurIPS. 2013.

Theoretical Guarantees

Importance Sampling Truncation

Lower bound on monotonic reward improvement [1]

$$J(\pi_i) - J(\mu) \geq -\frac{\gamma \epsilon^{\pi_i} \sqrt{2 \log \rho}}{(1-\gamma)^2}$$

Gradient Aggregation

Near-linear convergence rate [2]

$$\frac{1}{T} \sum_{t=1}^{T} \mathbb{E}(||\nabla J(\theta_t)||^2) \leq 2\sqrt{\frac{2C_1C_2}{Tb}}$$

[1] Joshua, A. et al. "Constrained Policy Optimization." *ICML*. 2017.
[2] Zhang, W. et al. “Staleness-Aware Async-SGD for Distributed Deep Learning”. IJCAI. 2016.

Evaluation

Implementation

Ray RLlib
Docker Containers
AWS EC2

Baselines

Ray RLlib [1]
MinionsRL [2]

Metrics

Episodic reward
Training cost

Benchmarks

MuJoCo [3]

Hopper
Humanoid
Walker2d

Atari [4]

Gravitar
SpaceInvaders
Qbert

[1] Liang, E.; et al. RLlib: Abstractions for Distributed Reinforcement Learning. ICML 2018
[2] H. Yu; et al. Cheaper and Faster: Distributed Deep Reinforcement Learning with Serverless Computing. AAAI 2024
[3] Todorov, E.; et al. Mujoco: A Physics Engine for Model-based Control. IROS 2012
[4] Marc, B.; et al. The Arcade Learning Environment: An Evaluation Platform for General Agents. JAIR 2013

Testbed Clusters

GPU Testbed

3 nodes

128 AMD EPYC 7R13 CPU cores

2 V100 GPUs

HPC Testbed

7 nodes

960 AMD EPYC 9R14 CPU cores

16 V100 GPUs

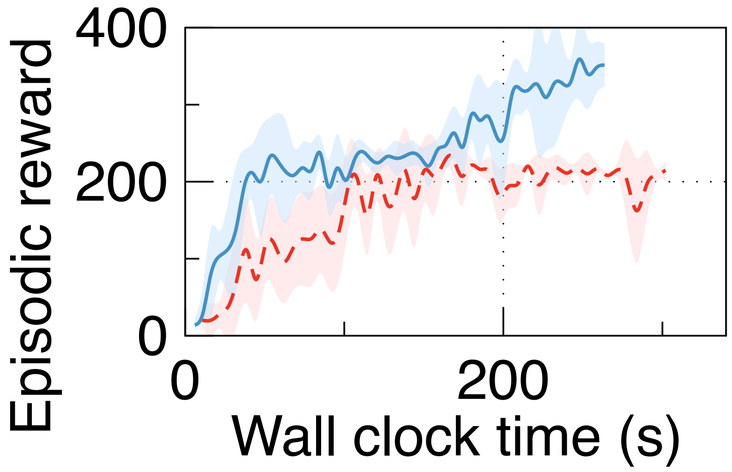
Training Performance

Faster Training

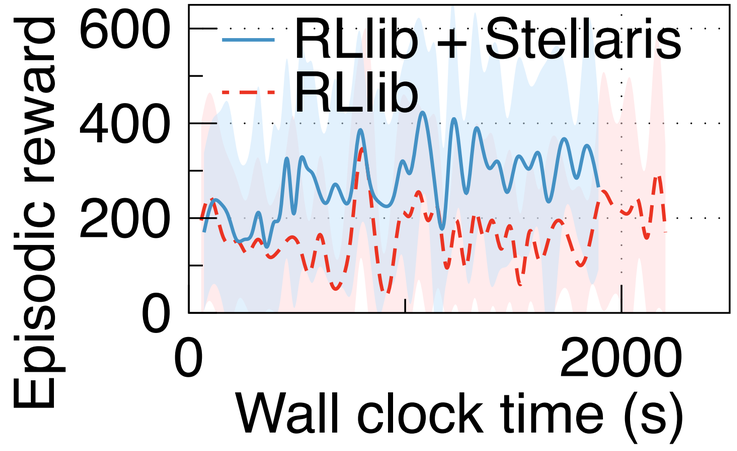
Higher Rewards

2.2x

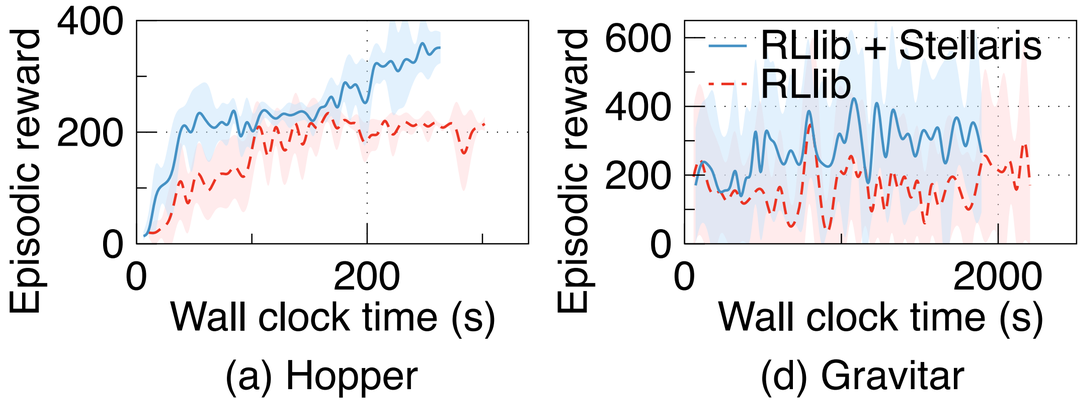
Training performance improvement



(a) Hopper

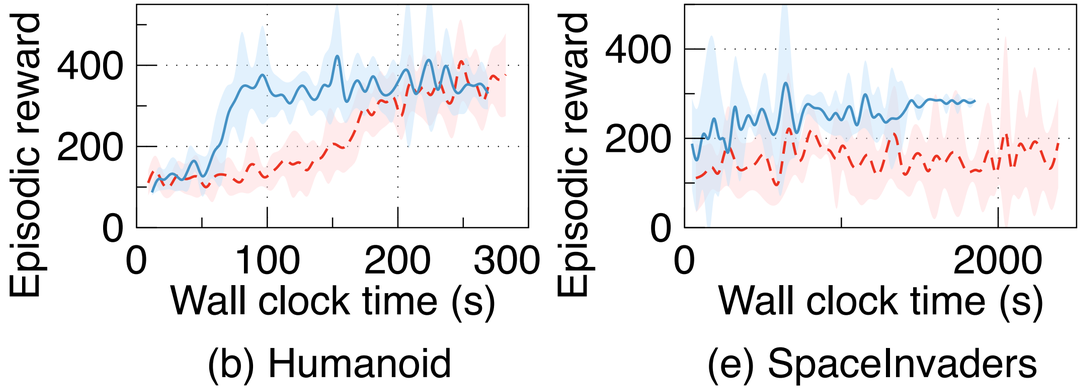


(b) Gravitar



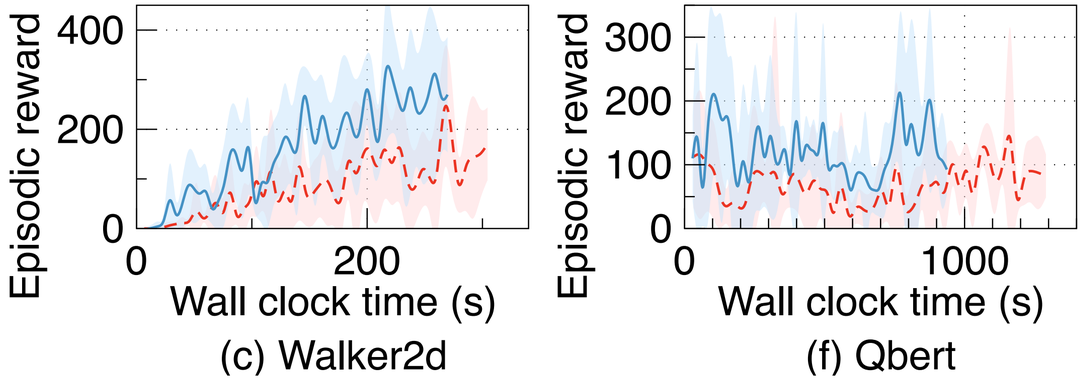
(a) Hopper

(d) Gravitar



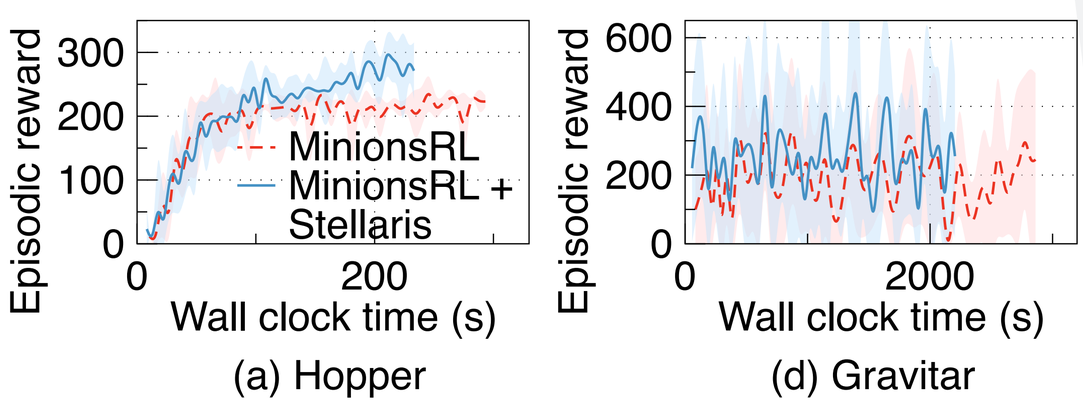
(b) Humanoid

(e) SpaceInvaders



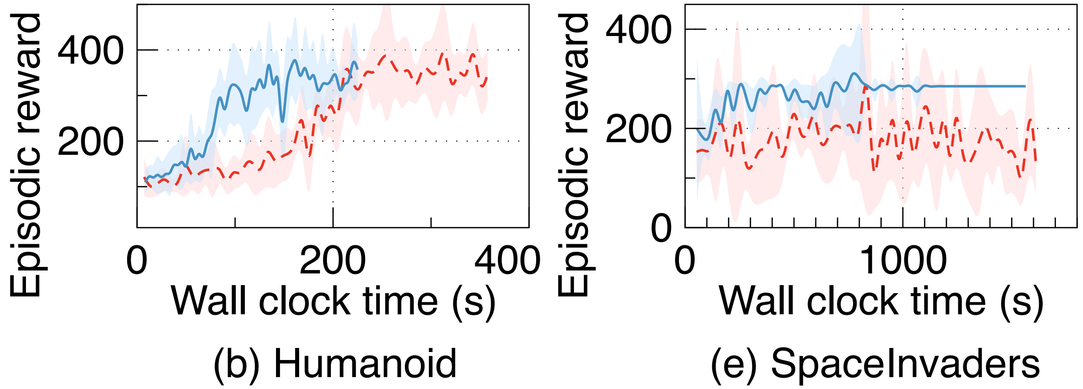
(c) Walker2d

(f) Qbert



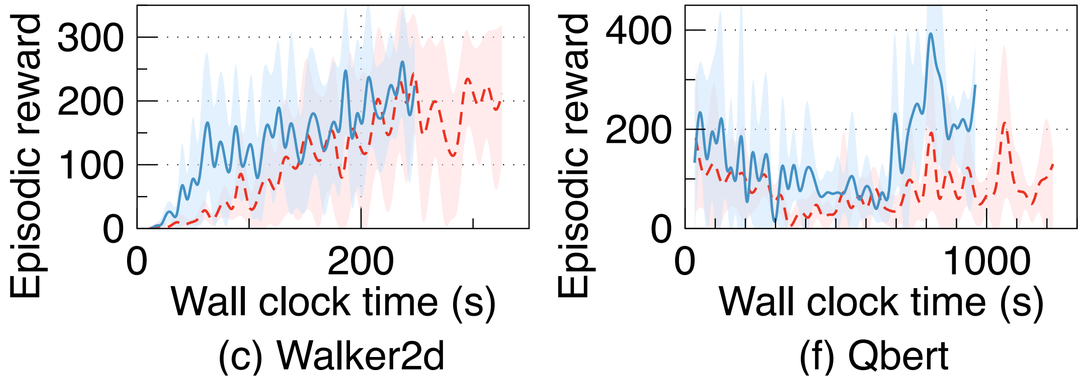
(a) Hopper

(d) Gravitar



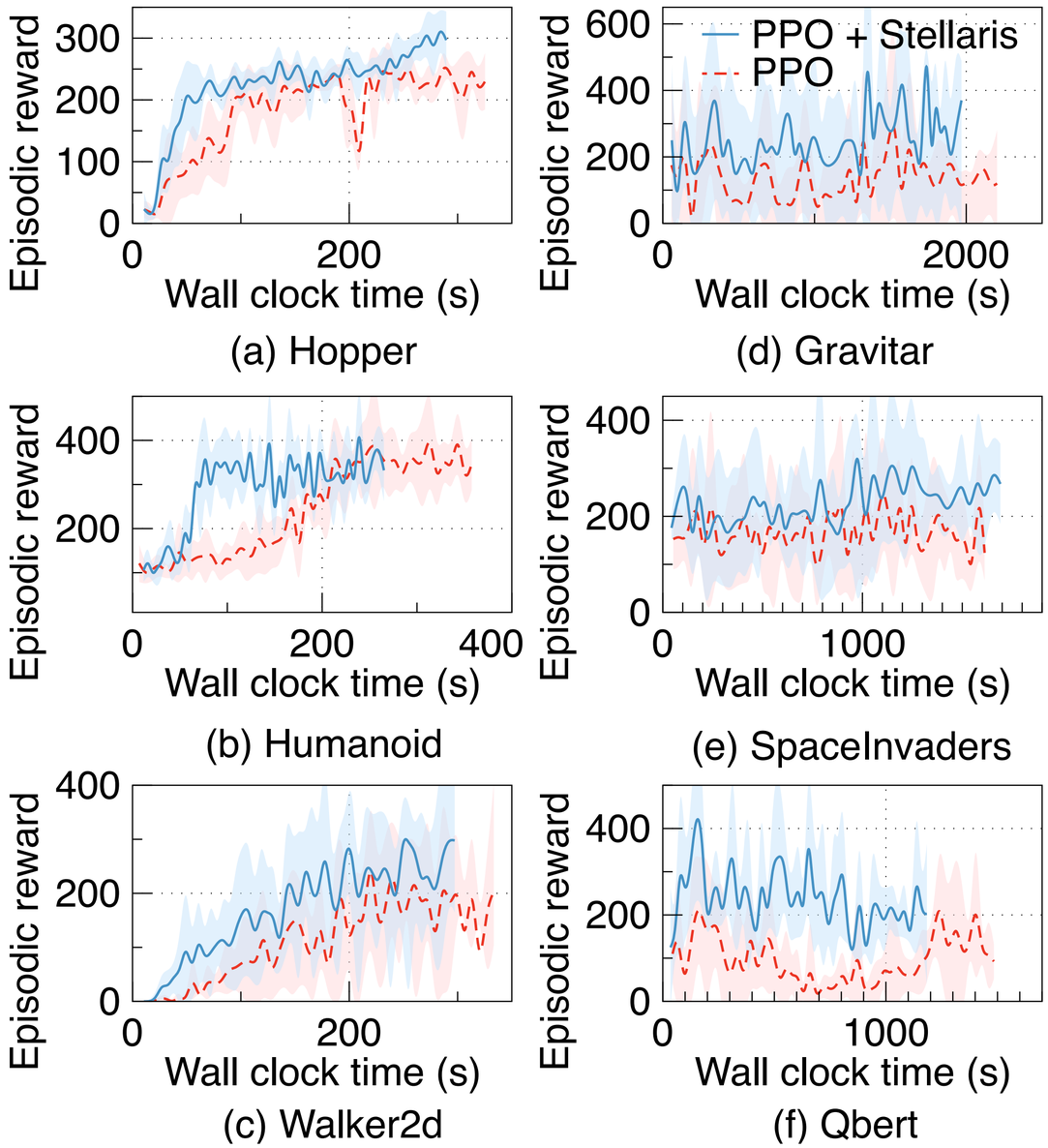
(b) Humanoid

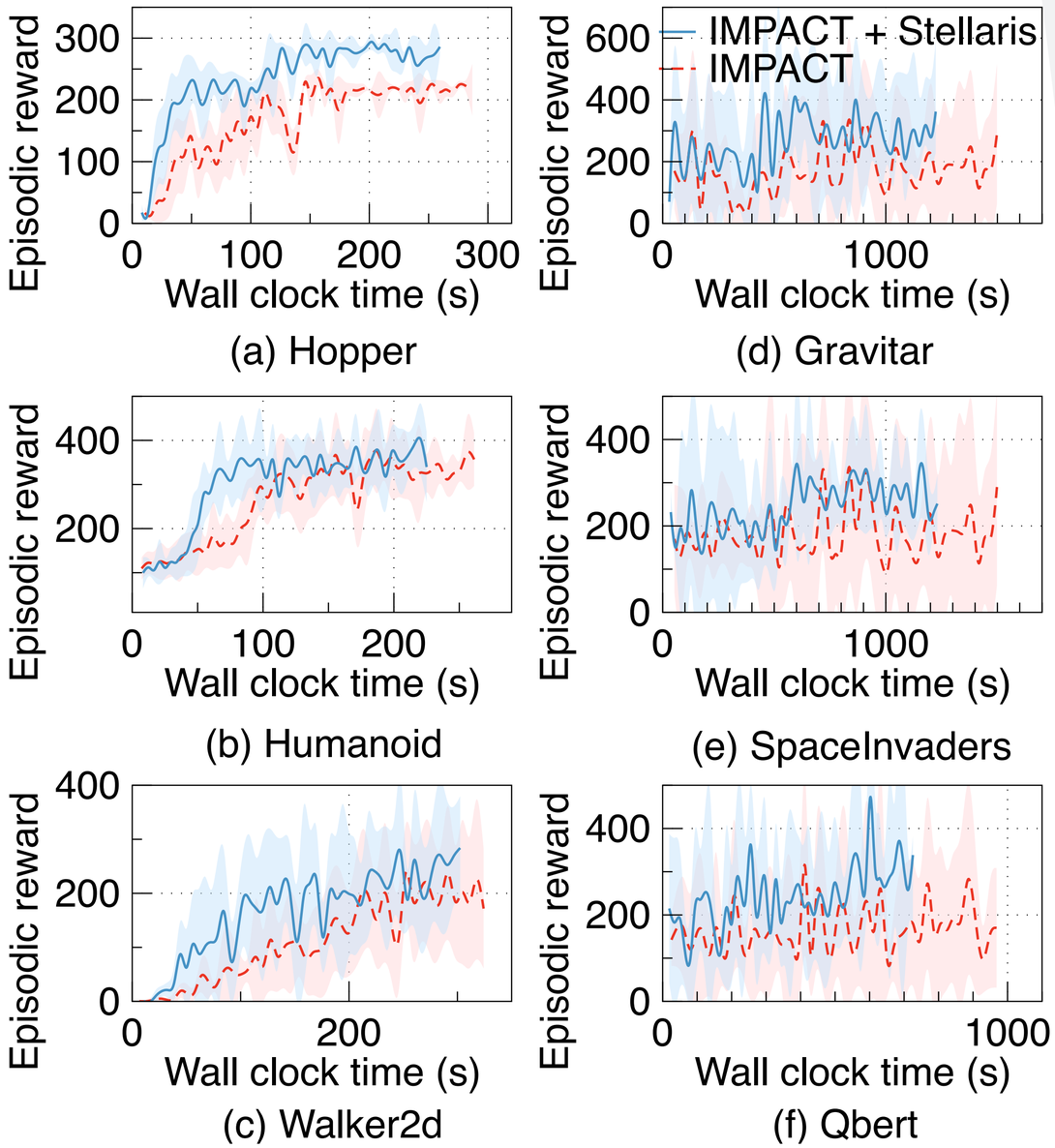
(e) SpaceInvaders



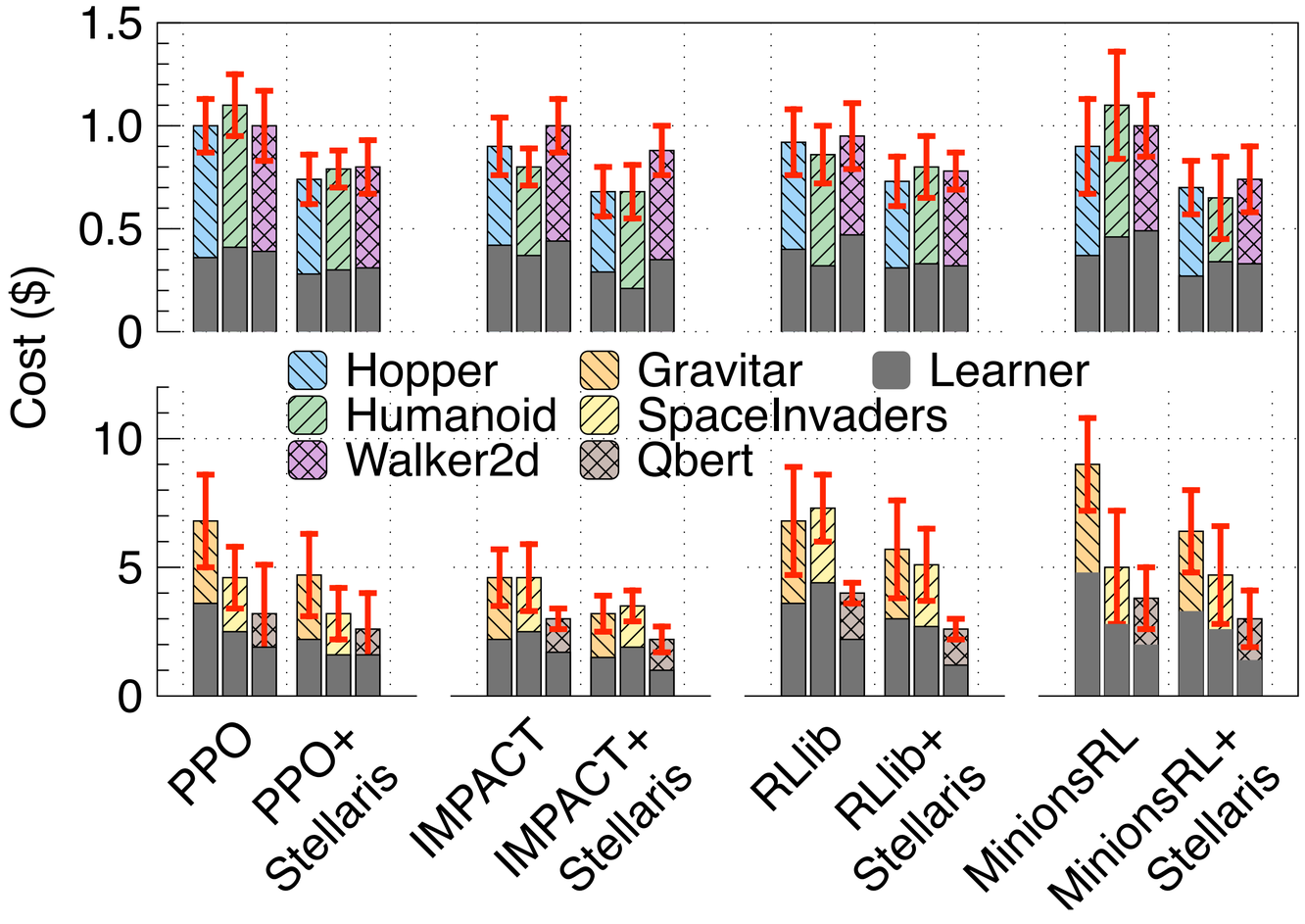
(c) Walker2d

(f) Qbert


Episodic reward
Wall clock time (s)
(a) Hopper
(b) Humanoid
(c) Walker2d
PPO + Stellaris
PPO
(d) Gravitar
(e) SpaceInvaders
(f) Qbert


Episodic reward
Wall clock time (s)
(a) Hopper
(b) Humanoid
(c) Walker2d
IMPACT + Stellaris
IMPACT
(d) Gravitar
(e) SpaceInvaders
(f) Qbert

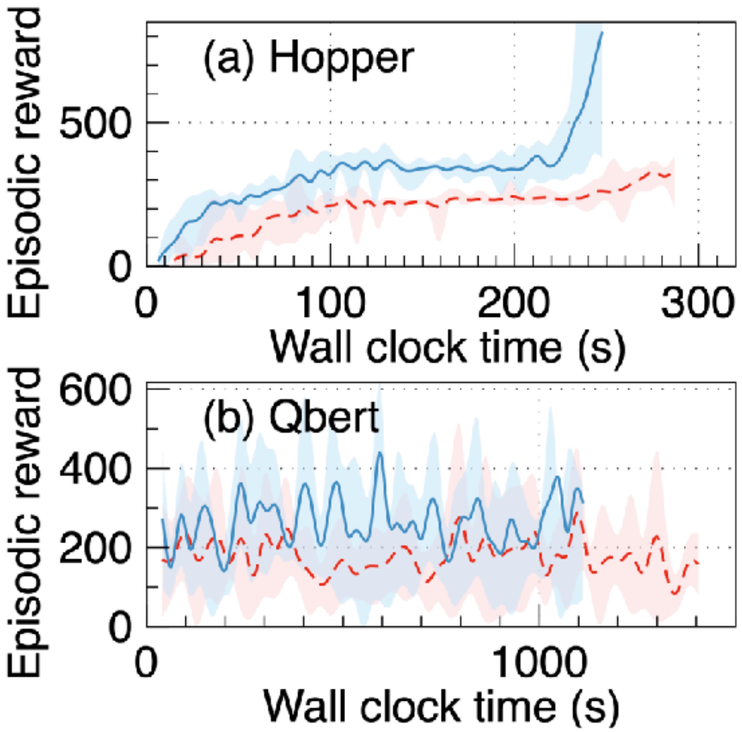
Training Cost

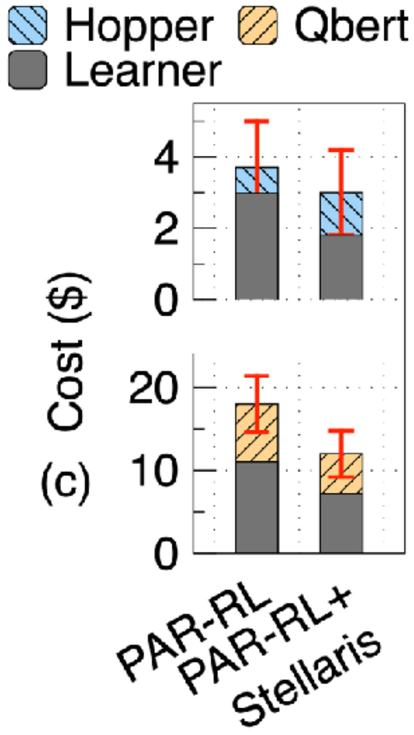


41%

Training cost reduction

Improving PAR-RL [1] on HPC Testbeds





2.4x

Training performance improvement

34%

Training cost reduction

[1] https://github.com/Romit-Maulik/PAR-RL. Argonne Leadership Computing Facility. 2022.

Asynchronous Serverless Learners

Global Importance Sampling Truncation

Staleness-Aware Gradient Aggregation

Stellaris

2.2x

Training performance improvement

41%

Training cost reduction

Stellaris Code Repository:

https://github.com/IntelliSys-Lab/Stellaris-SC24

GitHub Code

Corresponding Author:

Hanfei Yu <hyu42@stevens.edu>

Hao Wang <hwang9@stevens.edu>

I'm looking for a research internship in summer 2025!

aws

IntelliSys Lab

About Me

STEVENS
INSTITUTE OF TECHNOLOGY
1870

IntelliSys Lab

THANK YOU

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