Distributed Meta Optimization of Reinforcement Learning Agents

Greg Heinrich, Iuri Frosio - GTC San Jose, March 2019

AGENDA

Contents
Introduction to Reinforcement Learning
Introduction to Metaoptimization (on distributed systems) / Maglev
Metaoptimization and Reinforcement Learning (on distributed systems)
HyperTrick
Results
Conclusion

GPU-Based A3C for Deep Reinforcement Learning (RL)

keywords: GPU, A3C, RL

M. Babaeizadeh, I. Frosio, S. Tyree, J. Clemons, J. Kautz, **Reinforcement Learning through Asynchronous Advantage Actor-Critic on a GPU**, ICLR 2017 (available at). Open source implementation: .

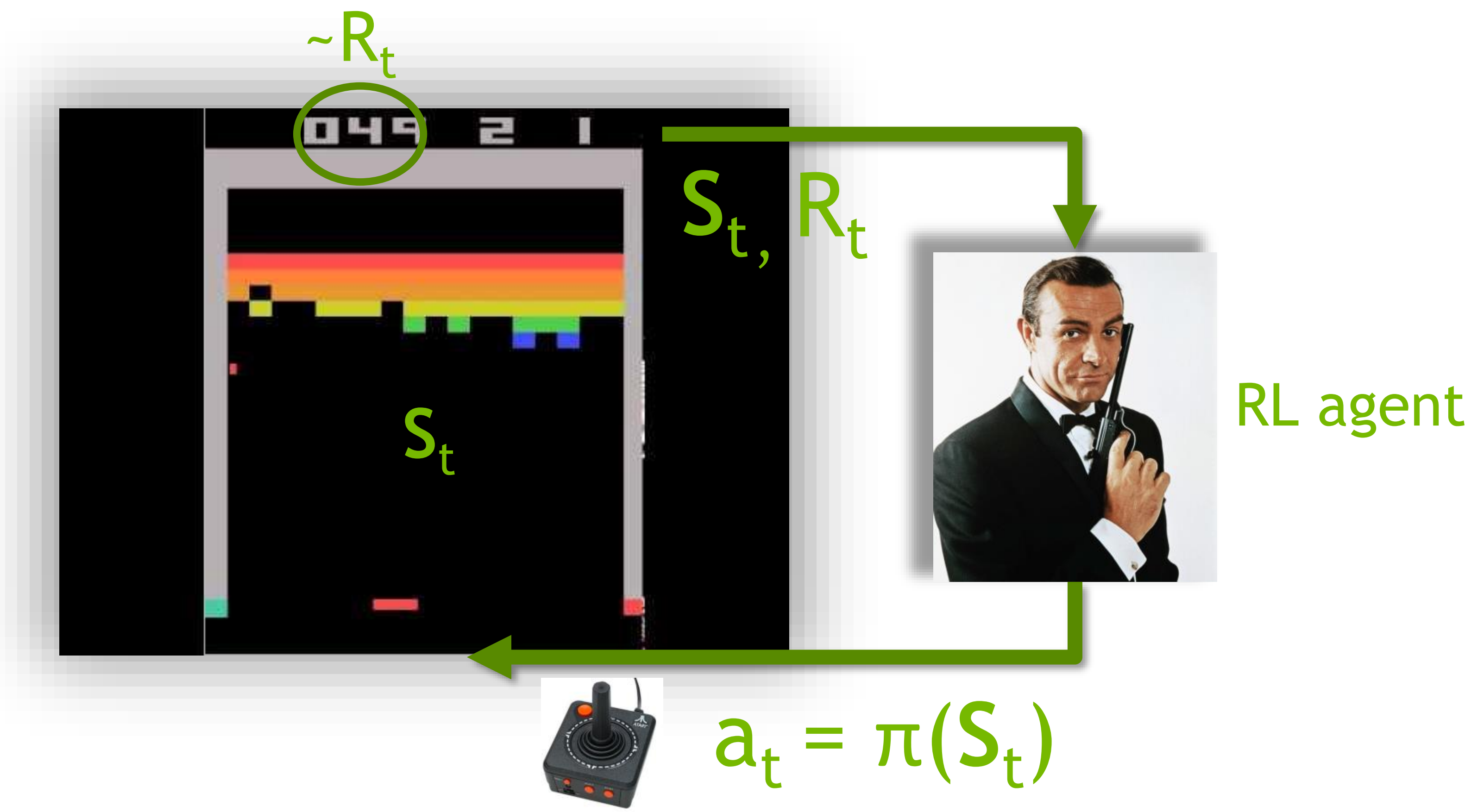
GPU-BASED A3C FOR DEEP REINFORCEMENT LEARNING

Learning to accomplish a task

Image from www.33rdsquare.com

GPU-BASED A3C FOR DEEP REINFORCEMENT LEARNING

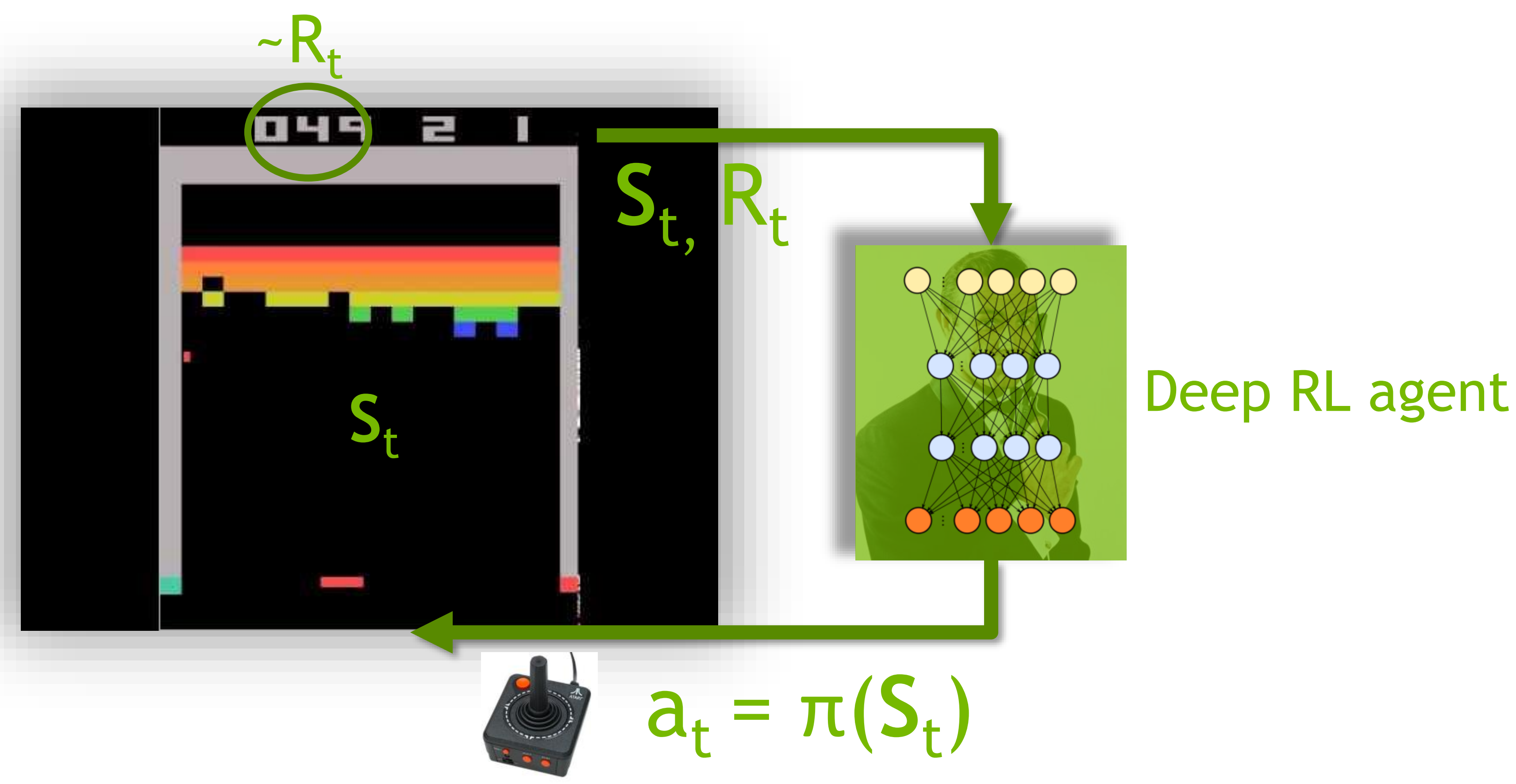
Definitions



- ✓ Environment
- ✓ Agent
- ✓ Observable status S_t
- ✓ Reward R_t
- ✓ Action a_t
- ✓ Policy $a_t = \pi(S_t)$

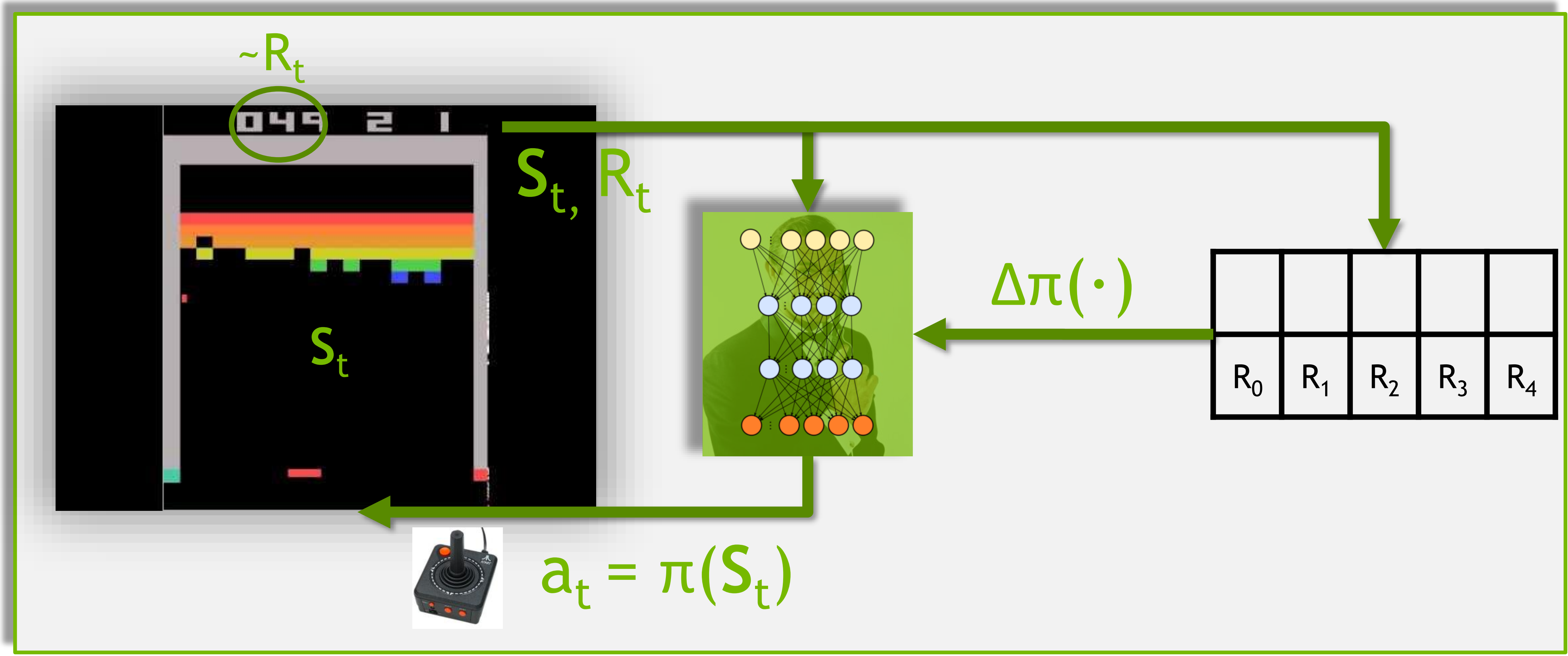
GPU-BASED A3C FOR DEEP REINFORCEMENT LEARNING

Definitions



GPU-BASED A3C FOR DEEP REINFORCEMENT LEARNING

Definitions



GPU-BASED A3C FOR DEEP REINFORCEMENT LEARNING

Objective: maximize expected discounted rewards

Value of a state

$$v_\pi(s) = \mathbb{E}_\pi \left[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots \mid S_t = s \right]$$

Expected

Reward discounted

Given that state

The role of γ: short or far-sighted agents

$0 < \gamma < 1$, usually 0.99

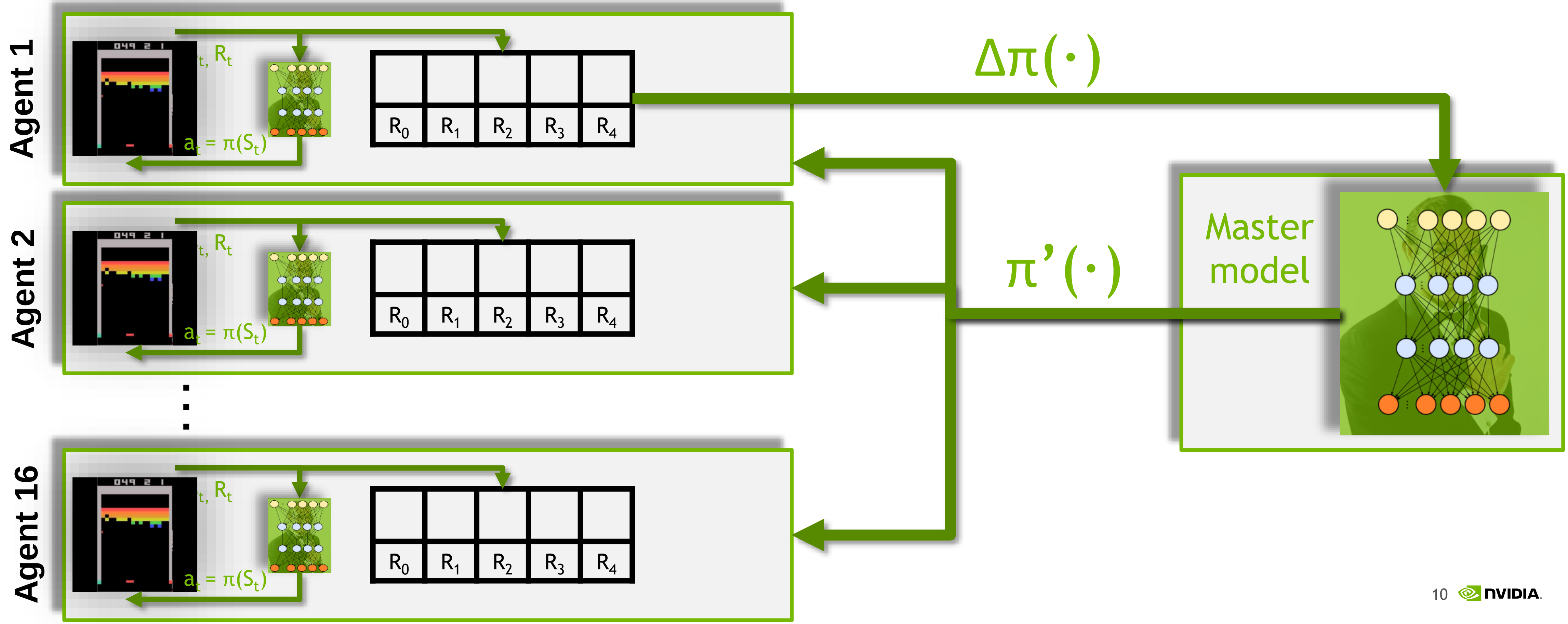
GPU-Based A3C for Deep Reinforcement Learning (RL)

keywords: GPU, A3C, RL

M. Babaeizadeh, I. Frosio, S. Tyree, J. Clemons, J. Kautz, **Reinforcement Learning through Asynchronous Advantage Actor-Critic on a GPU**, ICLR 2017 (available at). Open source implementation: .

GPU-BASED A3C FOR DEEP REINFORCEMENT LEARNING

Asynchronous Advantage Actor-Critic (Mnih et al., arXiv:1602.01783v2, 2015)



GPU-Based A3C for Deep Reinforcement Learning (RL)

keywords: GPU, A3C, RL

M. Babaeizadeh, I. Frosio, S. Tyree, J. Clemons, J. Kautz, **Reinforcement Learning through Asynchronous Advantage Actor-Critic on a GPU**, ICLR 2017 (available at). Open source implementation: .

MAPPING DEEP PROBLEMS TO A GPU

REGRESSION, CLASSIFICATION, …

data

100% utilization / occupancy

**Pear, pear, pear, pear, …
Empty, empty, …
Fig, fig, fig, fig, fig, fig,
Strawberry, Strawberry,
Strawberry, …
…**

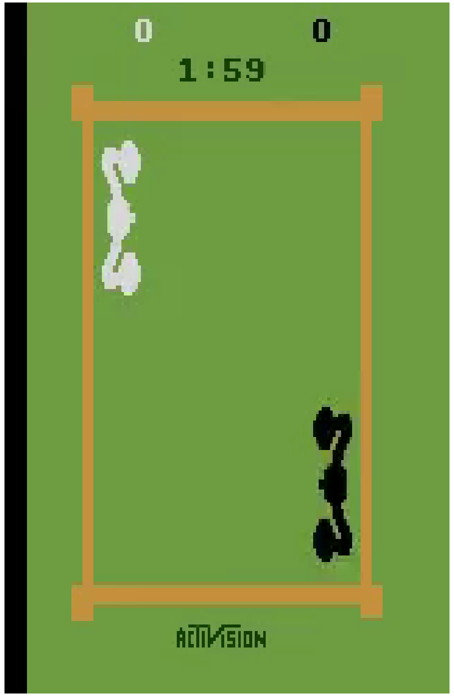
labels

REINFORCEMENT LEARNING

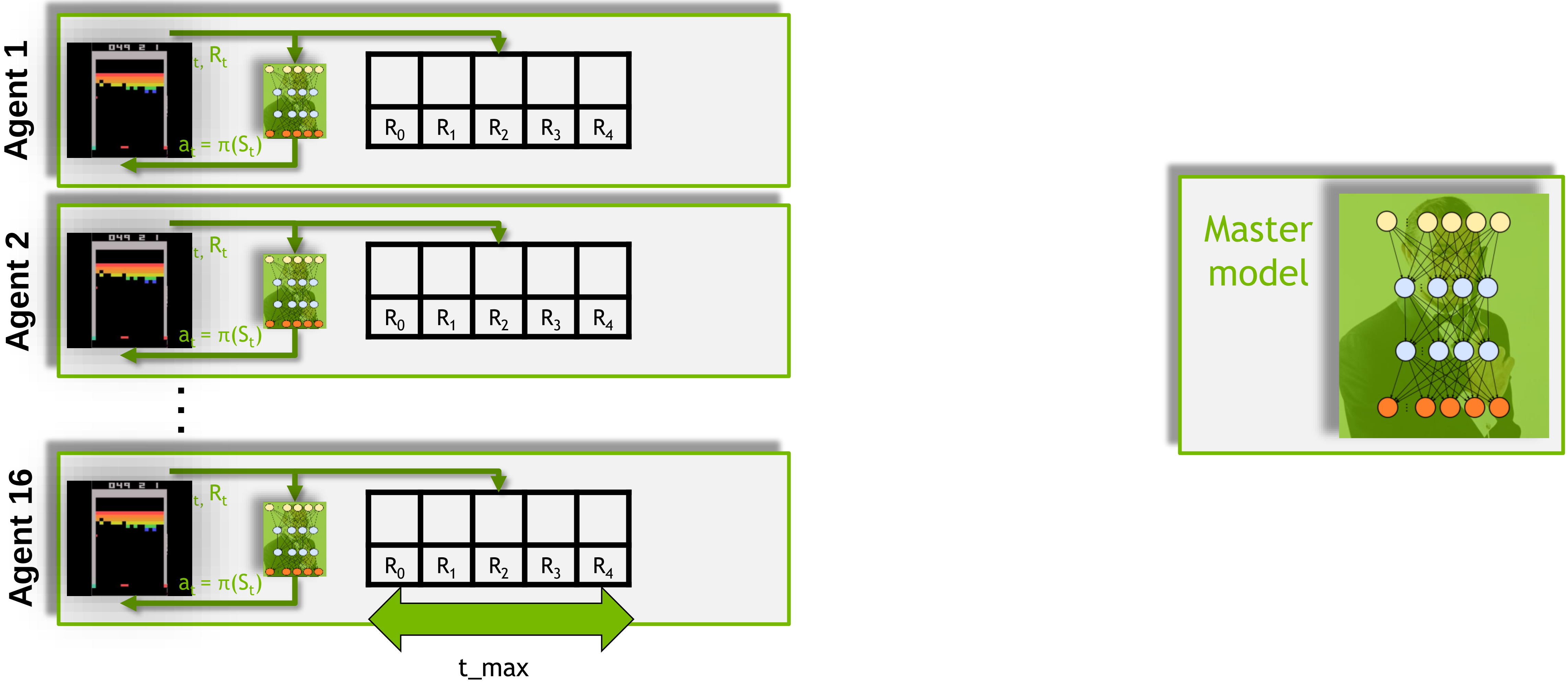
status,
reward

?

action

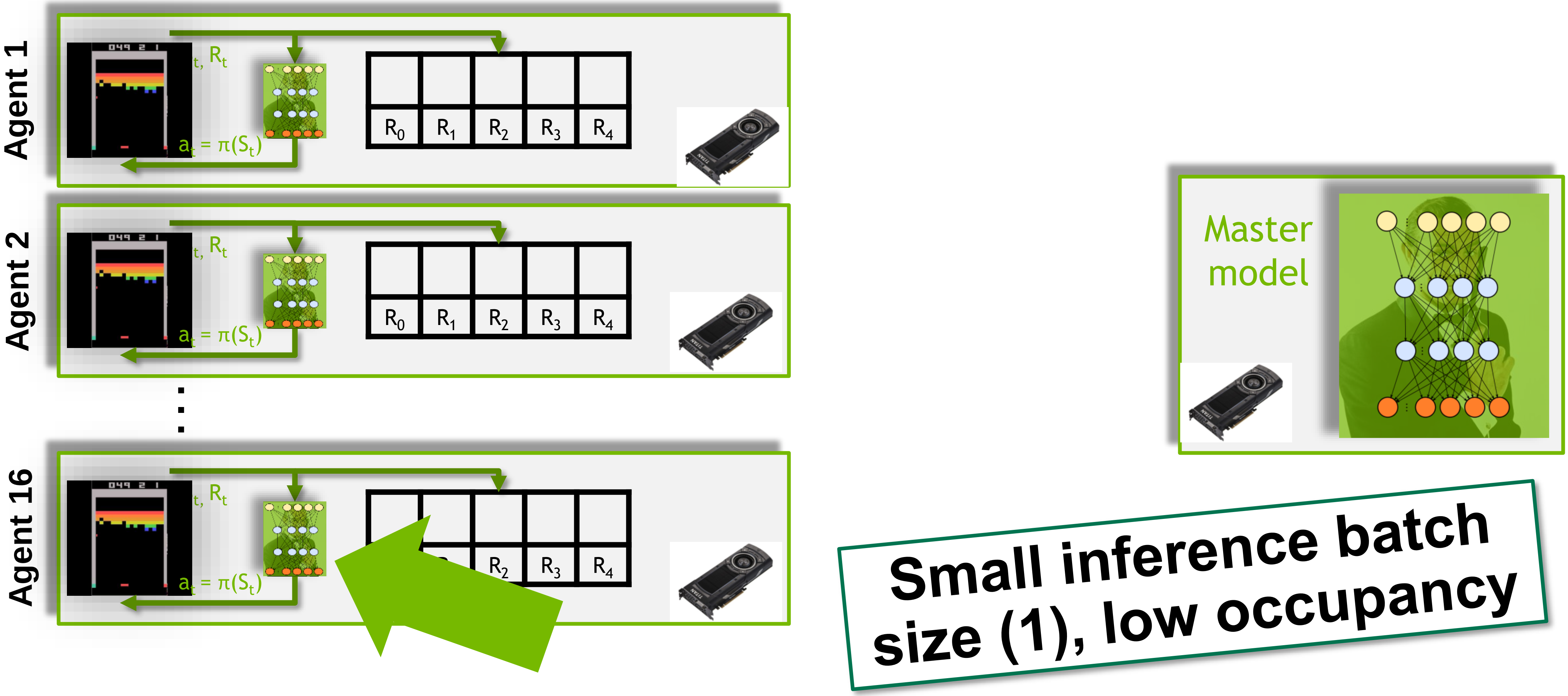


A3C

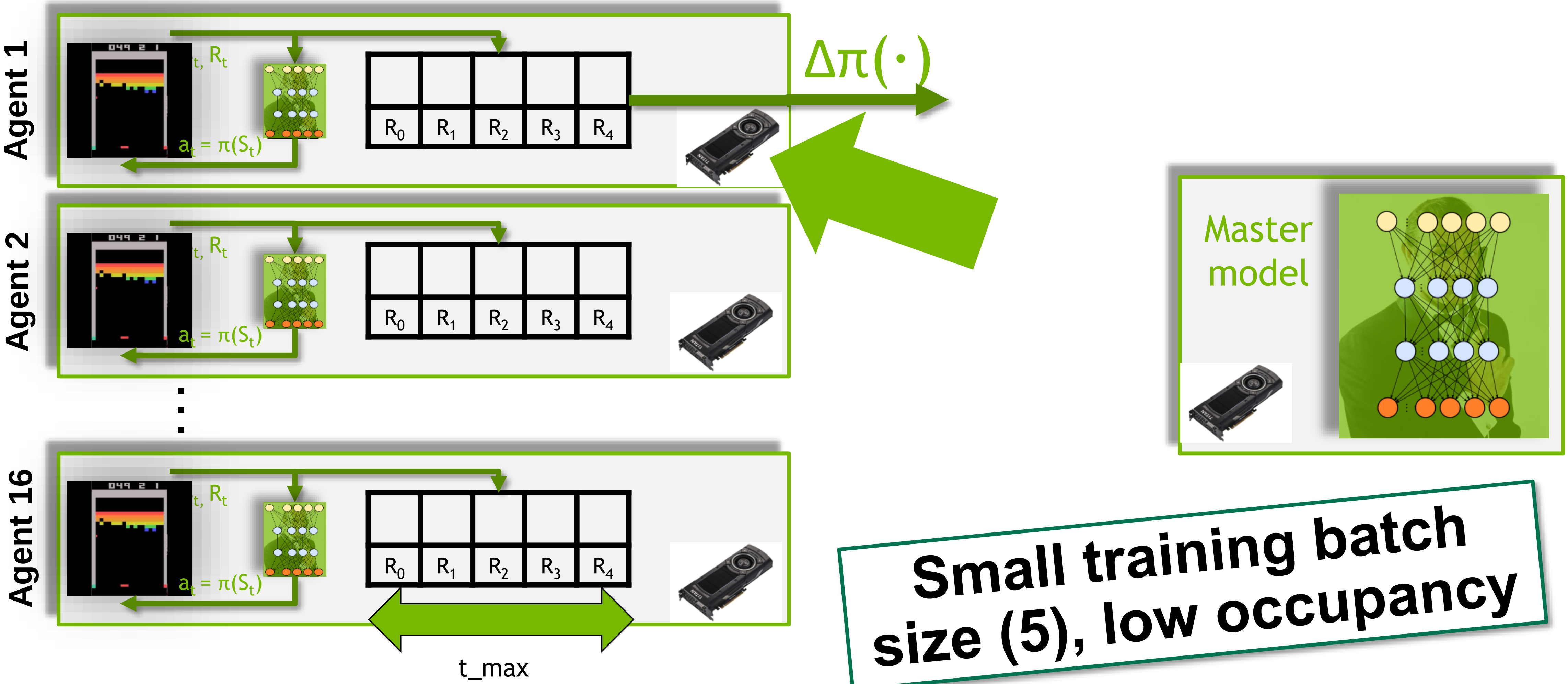


Agent 1
Agent 2
Agent 16
t, R_t
$a_t = \pi(S_t)$
R_0 R_1 R_2 R_3 R_4
t_max
Master model

A3C

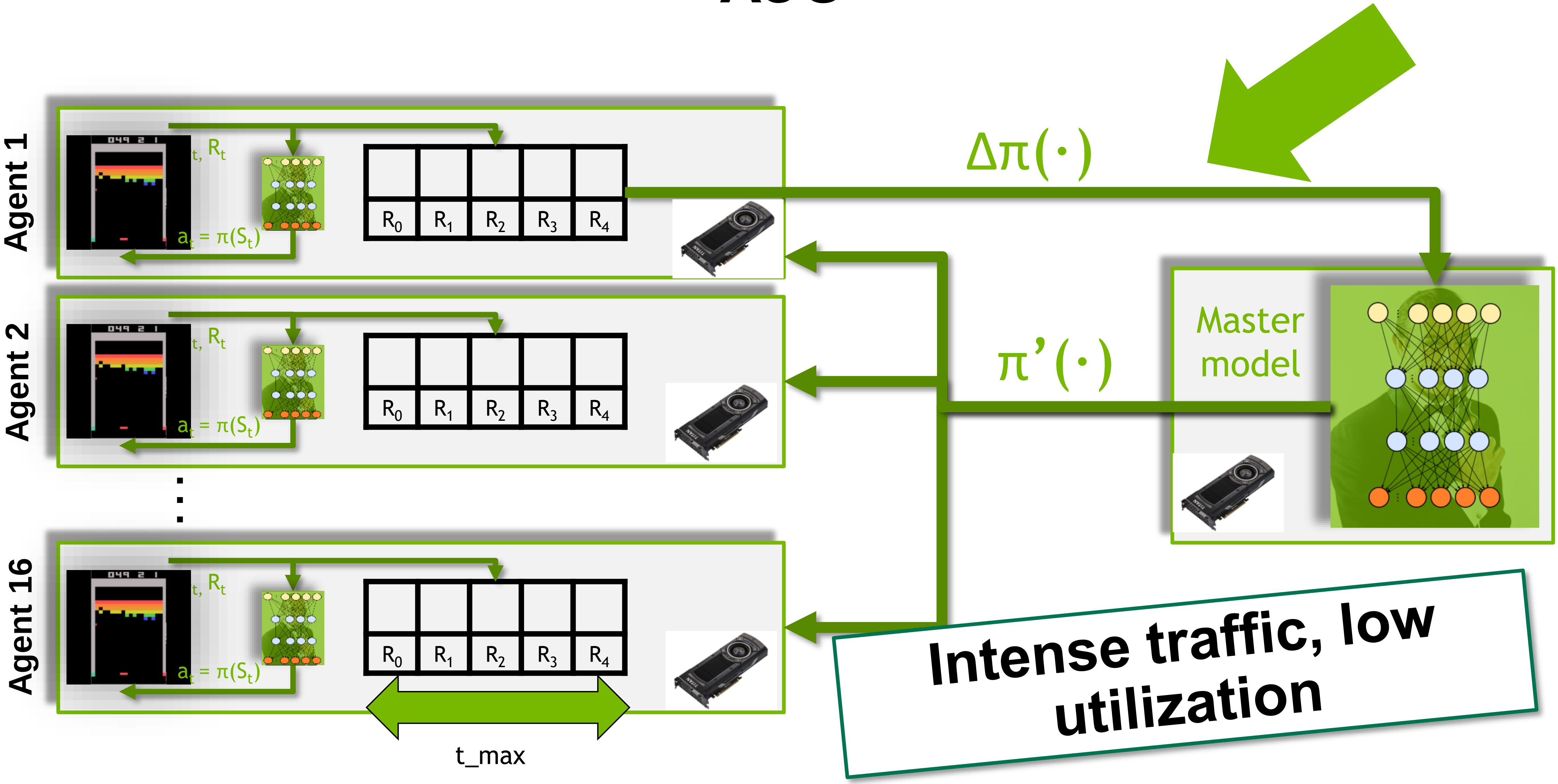

Agent 1
Agent 2
Agent 16
R_0 R_1 R_2 R_3 R_4
$a_t = \pi(S_t)$
Master model
Small inference batch size (1), low occupancy

A3C



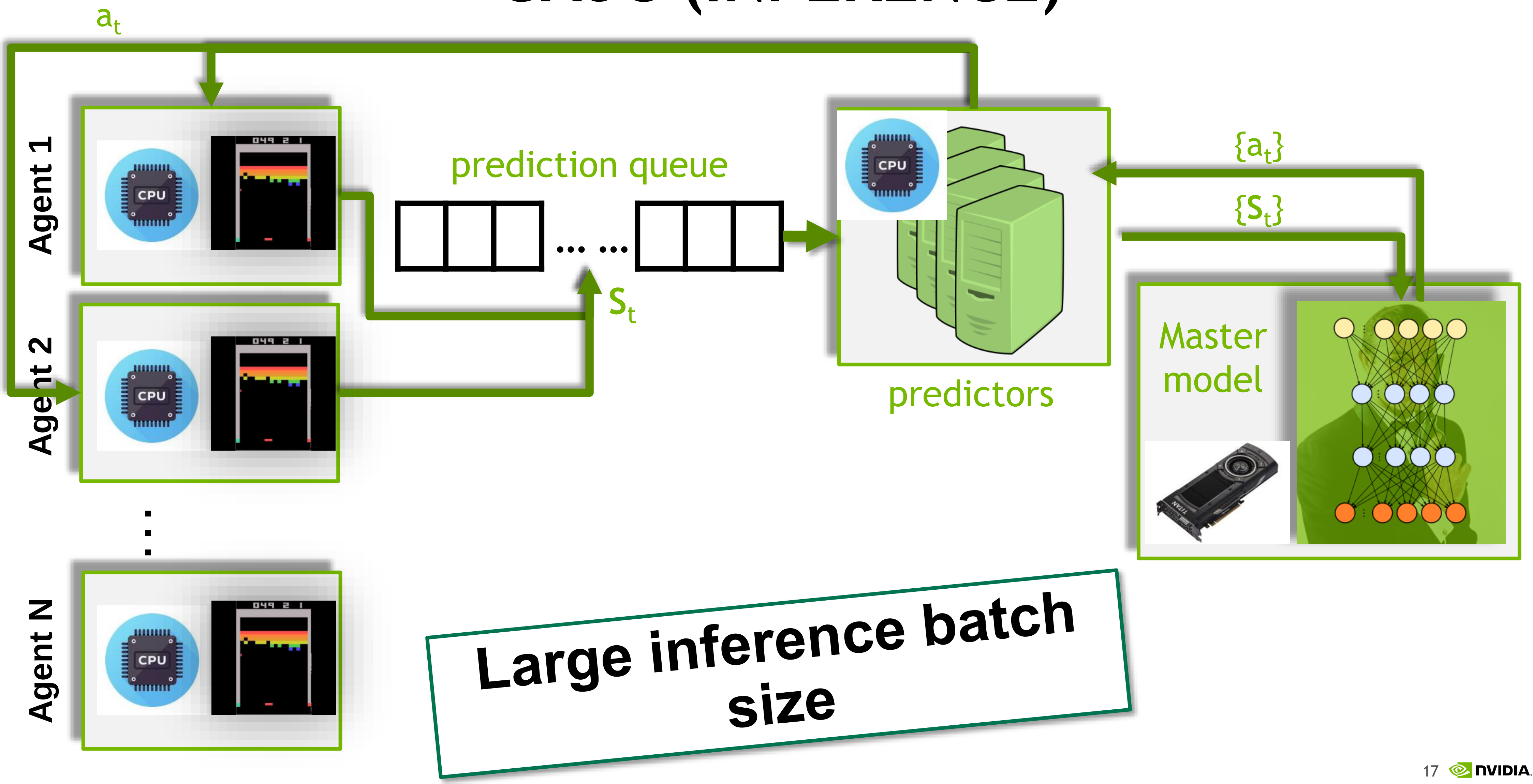
Small training batch size (5), low occupancy

A3C

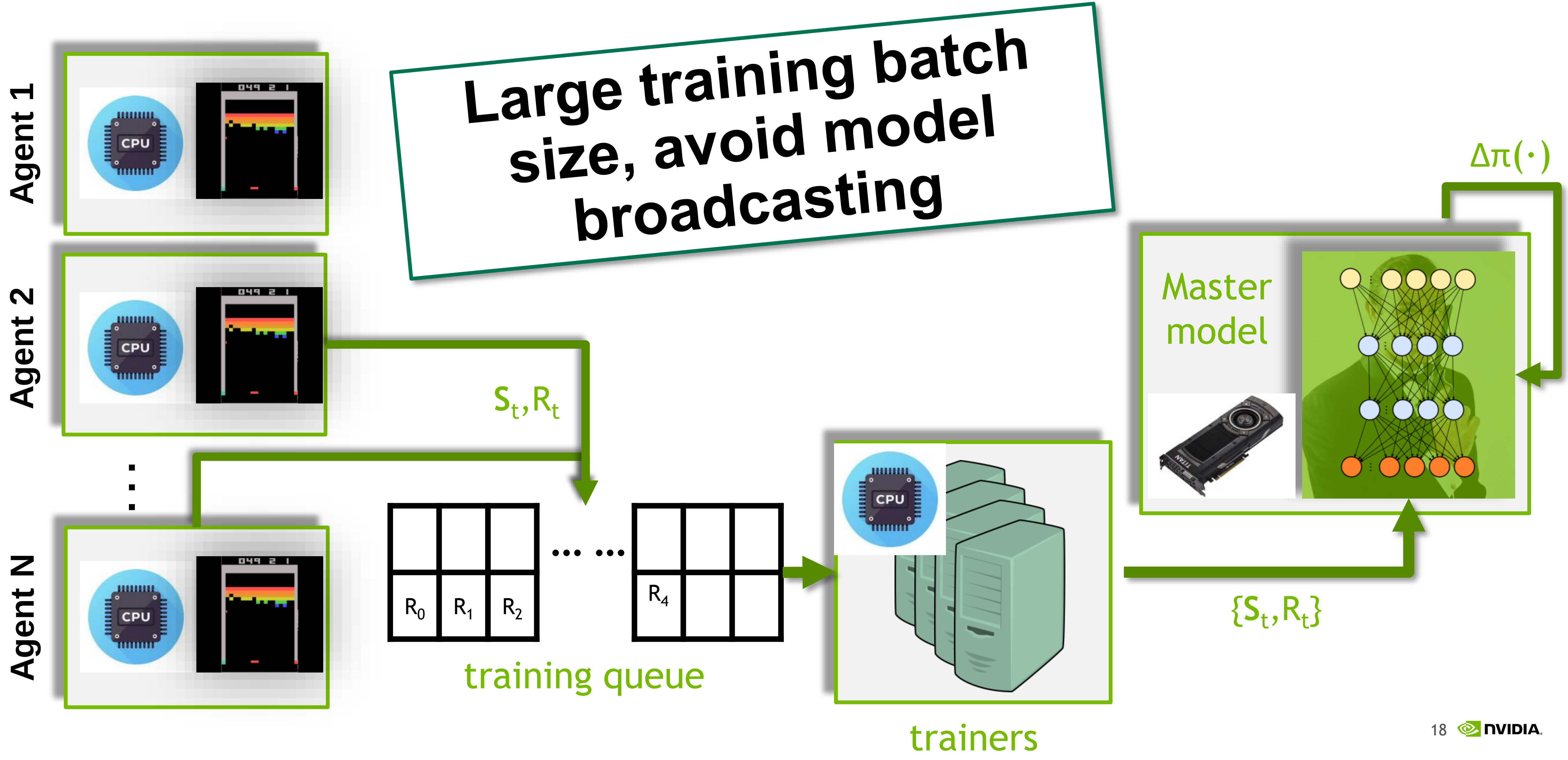


Agent 1
Agent 2
Agent 16
R_t
$a_t = \pi(S_t)$
R_0 R_1 R_2 R_3 R_4
$\Delta\pi(\cdot)$
$\pi'(\cdot)$
Master model
t_max
Intense traffic, low utilization

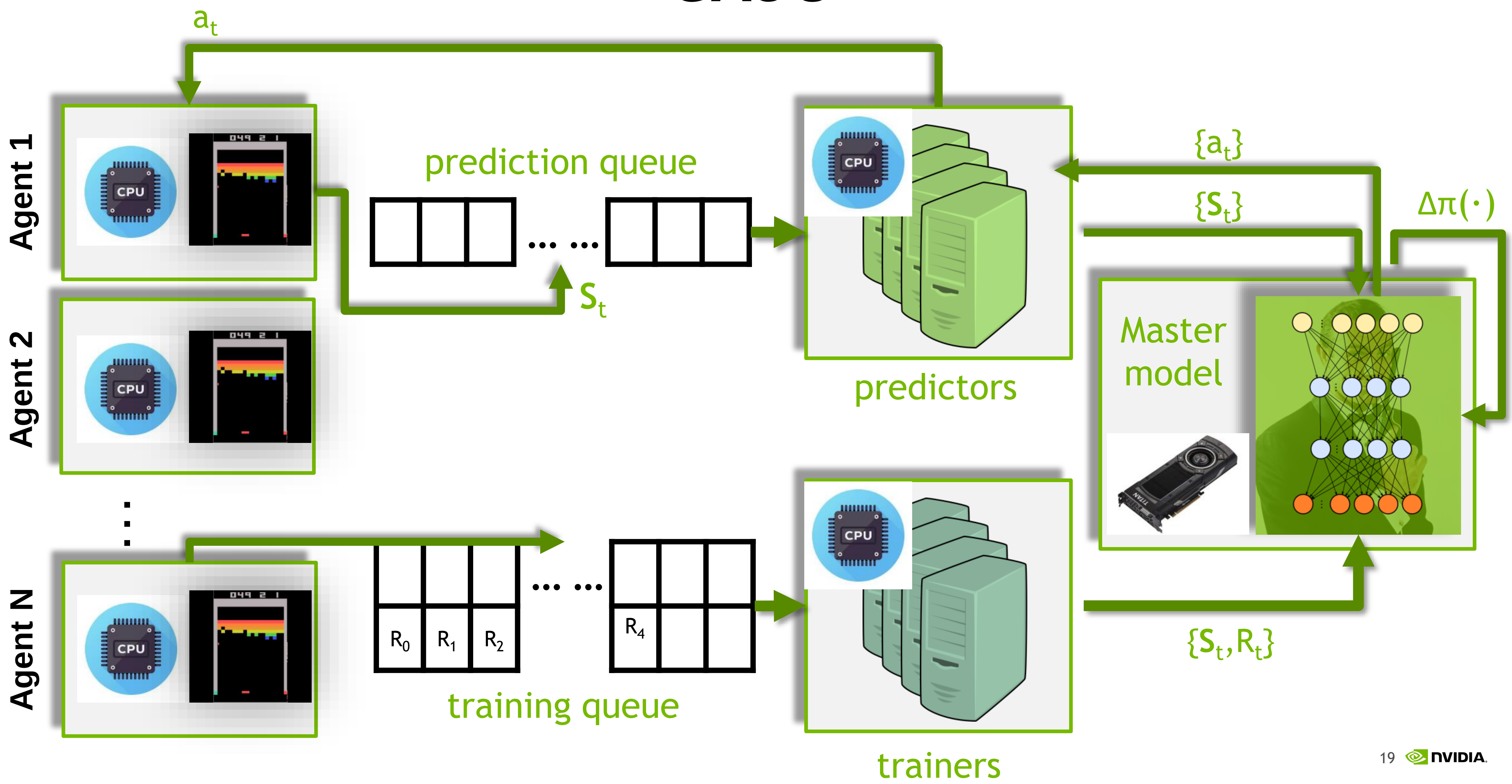
GA3C (INFERENCE)



GA3C (TRAINING)



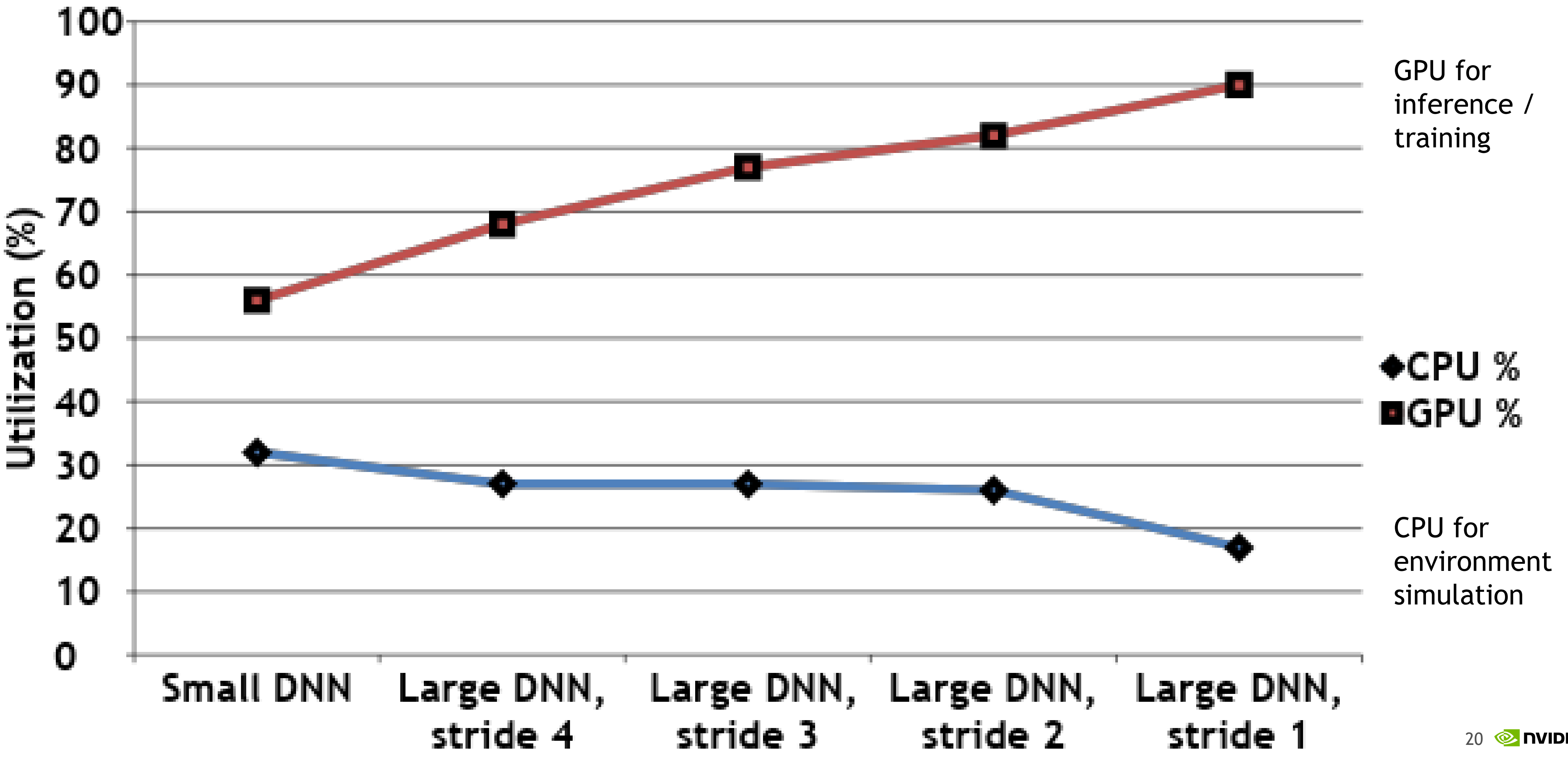
GA3C



a_t
Agent 1
Agent 2
Agent N
CPU
prediction queue
S_t
... ...
predictors
$\{a_t\}$
$\{S_t\}$
$\Delta\pi(\cdot)$
Master model
training queue
R_0 R_1 R_2 R_4
trainers
$\{S_t, R_t\}$

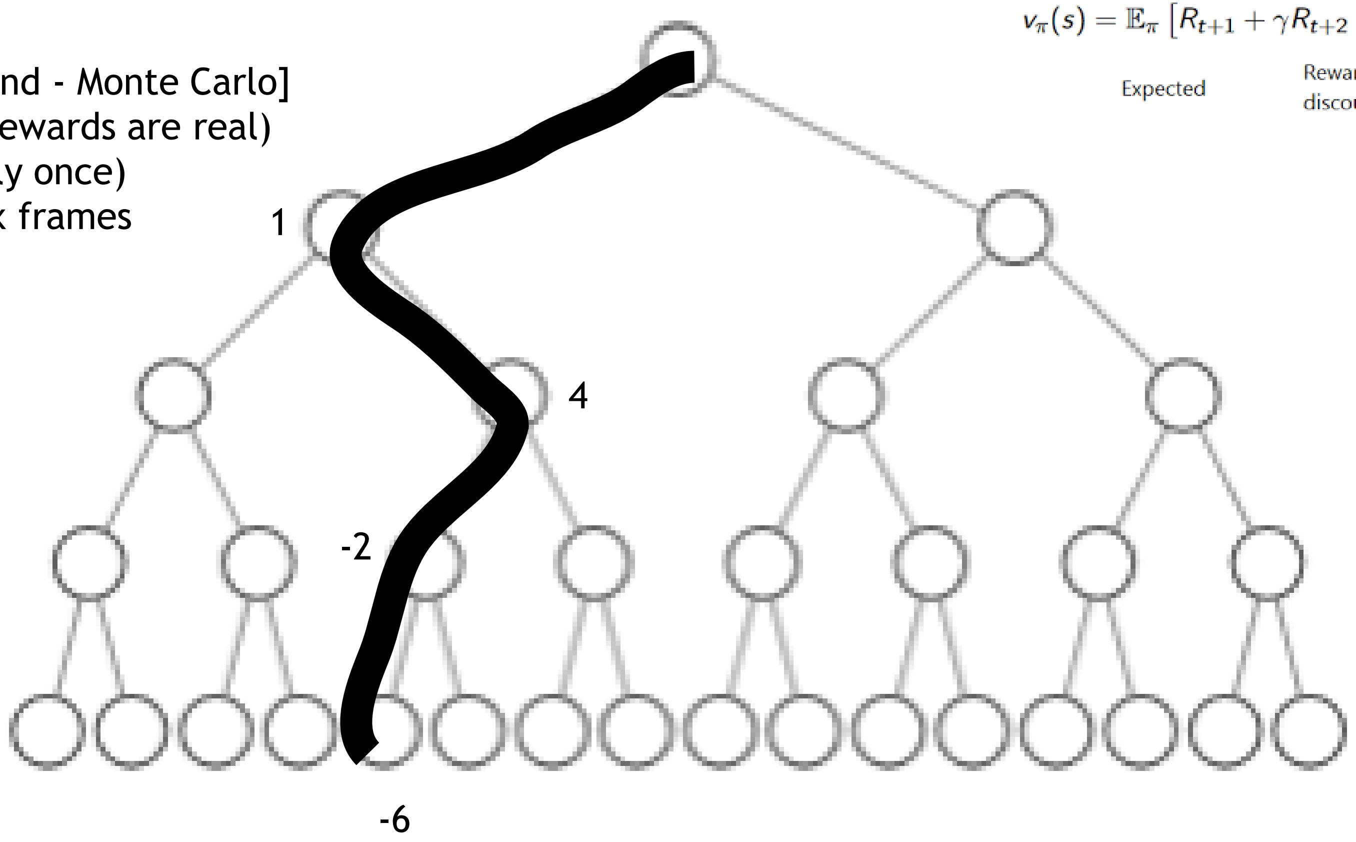
CPU & GPU UTILIZATION IN GA3C

For larger DNNs - bandwidth limited, do not scale to multiple GPUs!



Role of t_max

t_max = 4 [Play to the end - Monte Carlo]
No variance (collected rewards are real)
High bias (we played only once)
One update every t_max frames



$$v_\pi(s) = \mathbb{E}_\pi\left[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots \mid S_t = s\right]$$

Expected
Reward discounted
Given that state

Role of t_max

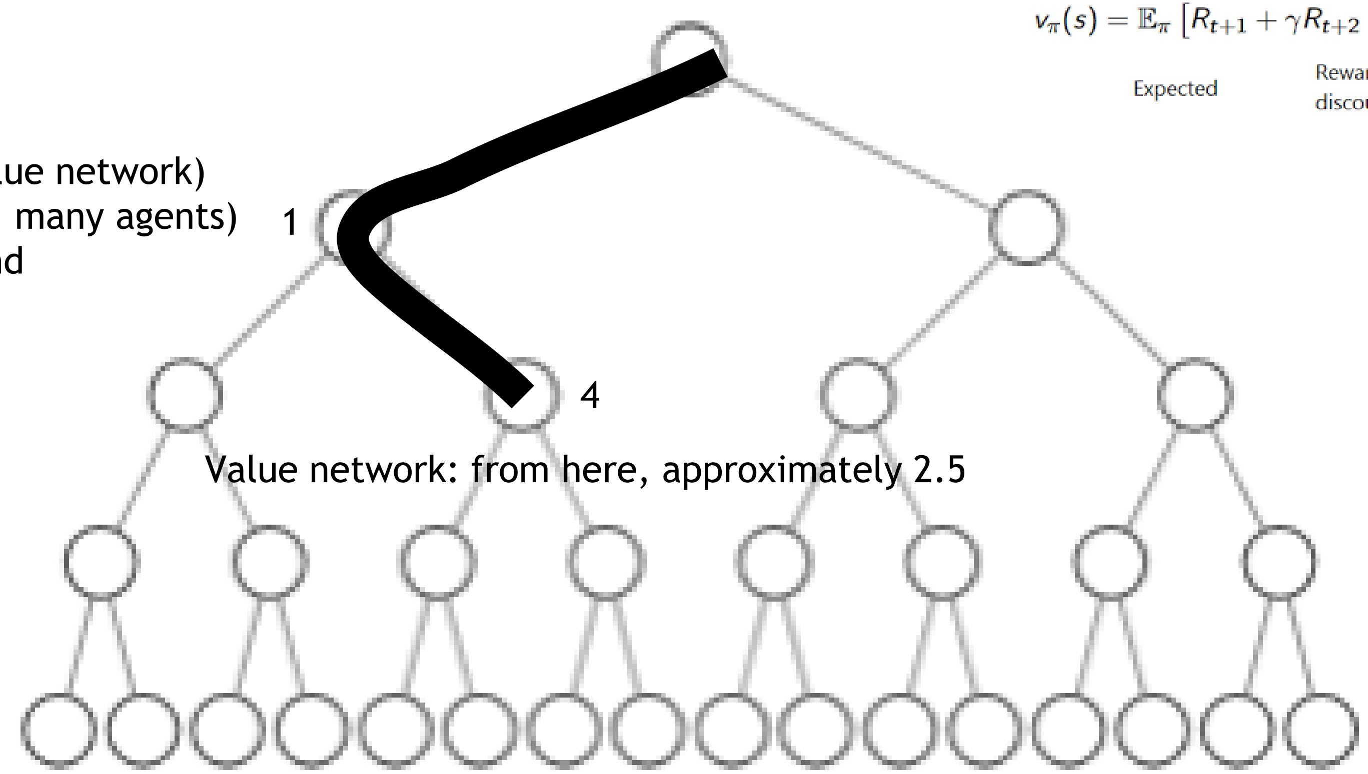
$$v_\pi(s) = \mathbb{E}_\pi \left[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots \mid S_t = s \right]$$

Expected

Reward discounted

Given that state

t_max = 2
Value network
High variance (noisy value network)
Low bias (unbiased net, many agents)
More updates per second



t_max affects bias, variance, computational cost (number of updates per second, batch size)

Other parameters and stability

Hyperparameter search in 2015:

The search for the optimal learning rate:

Asynchronous Methods for Deep Reinforcement Learning

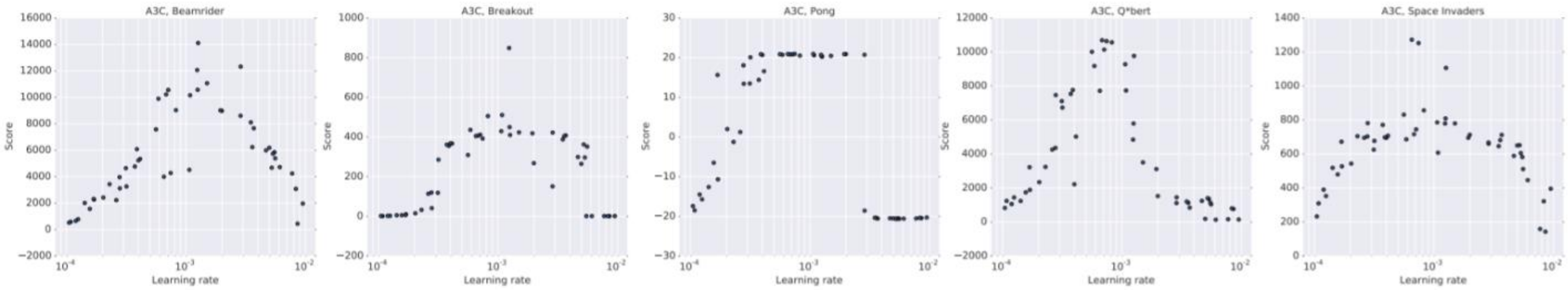
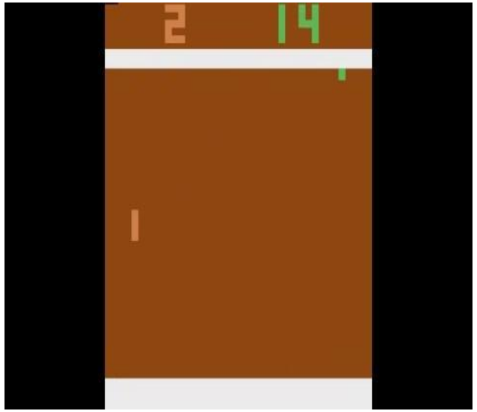


Figure 2. Scatter plots of scores obtained by asynchronous advantage actor-critic on five games (Beamrider, Breakout, Pong, Q*bert, Space Invaders) for 50 different learning rates and random initializations. On each game, there is a wide range of learning rates for which all random initializations acheive good scores. This shows that A3C is quite robust to learning rates and initial random weights.

https://arxiv.org/pdf/1602.01783.pdf

GA3C on distributed systems

- RL is unstable, **metaoptimization** for optimal hyperparameters search
- E.g. learning rate may affect the **stability** and **speed of convergence**
- GA3C does not scale to multiple GPUs (bandwidth limited), but … We can run **parallel instances** of GA3C on a **distributed system**
- The discount factor γ affects the **final aim** (short or far-sighted agent)
- The t_max factor affects the **computational cost** and **stability*** of GA3C



*** See G. Heinrich, I. Frosio, Metaoptimization on a Distributed System forDeep Reinforcement Learning, [ttps://arxiv.org/abs/1902.02725](https://arxiv.org/abs/1902.02725).**

AGENDA

Contents
Introduction to Reinforcement Learning
Introduction to Metaoptimization (on distributed systems) / Maglev
Metaoptimization and Reinforcement Learning (on distributed systems)
HyperTrick
Results
Conclusion

META OPTIMIZATION

It is as easy as flying a Concorde.

GA3C Agent

- Topology parameters
  - Number of layers and their width
  - Choice of activations
- Training parameters
  - Learning rate
  - Reward decay rate (γ)
  - back-propagation window size (tmax)
  - Choice of optimizer
  - Number of training episodes
- Data parameters
  - Environment model.

➡ Exhaustive search is intractable

Concorde

Source: Christian Kath

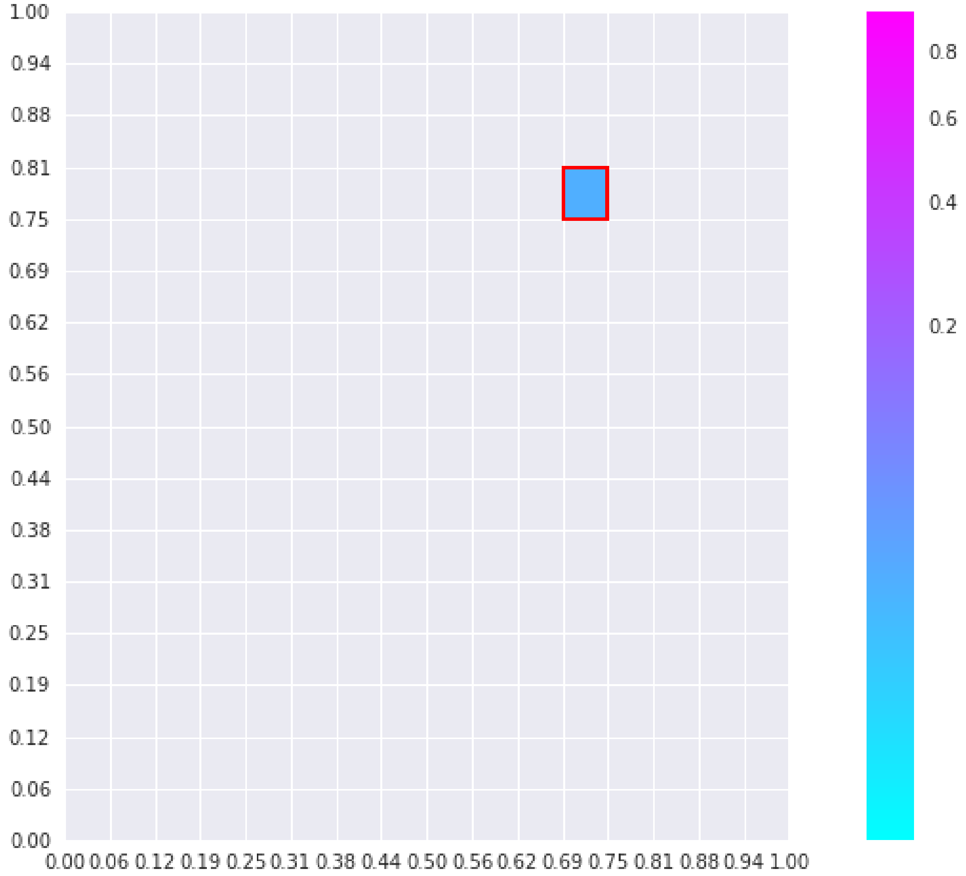
META OPTIMIZATION

How does a standard optimization algorithm fare?

Example: Tree of Parzen Estimators

- Two Parameters, one Metric to minimize.
- Optimization Trade-offs:
  - Exploitation v.s. exploration.
  - Wall time v.s. resource efficiency.
- Optimization packages start with a Random Search.
- Tens of experiments are needed before historical records can be leveraged.

➡ Warm Starts are needed to cut down complexity over time.



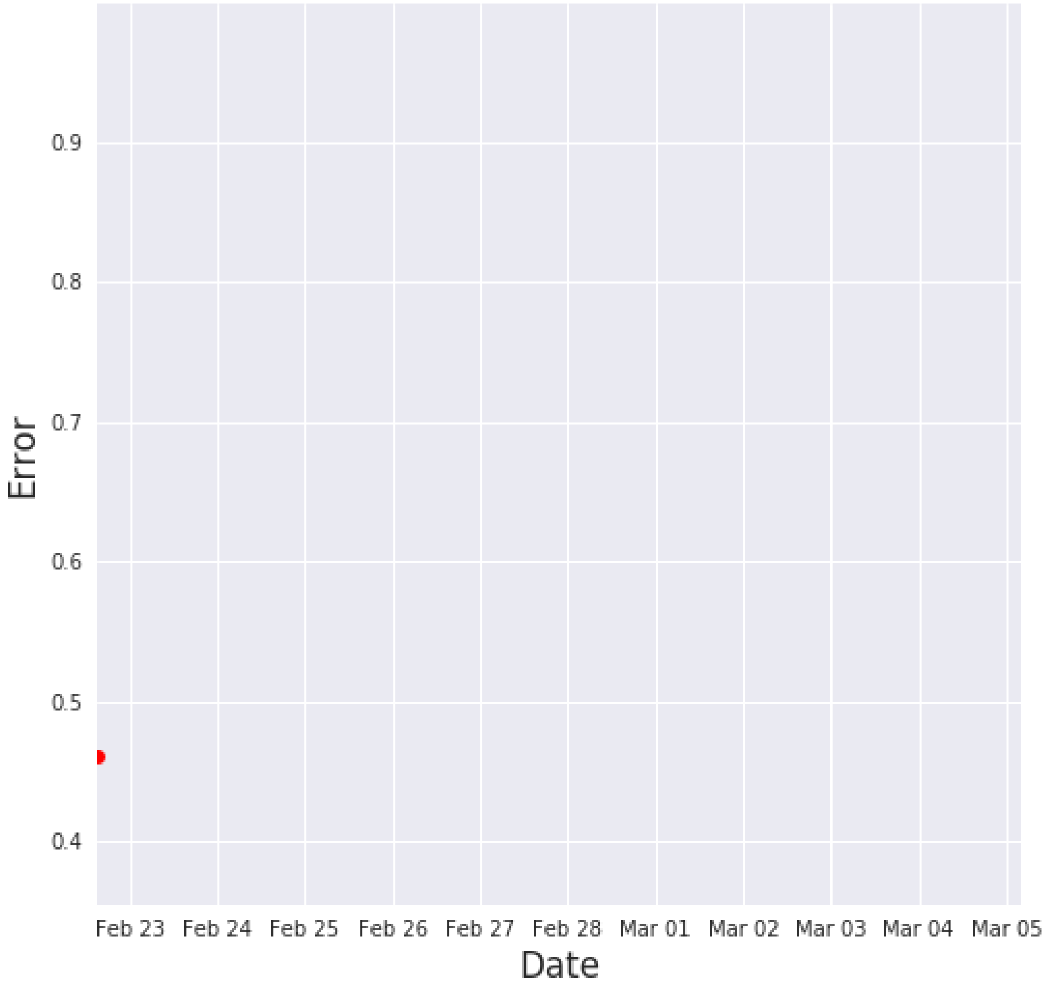
META OPTIMIZATION

The Need for Diversity

Metric Variance

- Non-determinism makes individual experiments inconclusive.
- A change can only be considered an improvement if it works under a variety of conditions.

➡ Meta Optimization should be part of data scientists' daily routine.



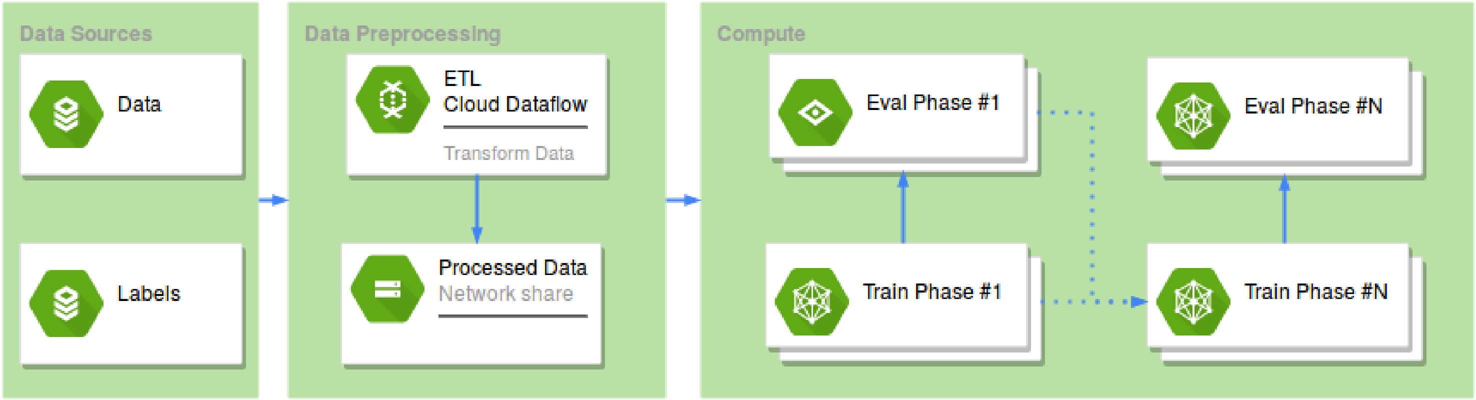
META OPTIMIZATION

The Complexity of Evaluating Models

Complex Pipelines

- Evaluation cannot be reduced to a single Python function, or Docker container.

➡ Meta Optimization must be independent of task scheduling.

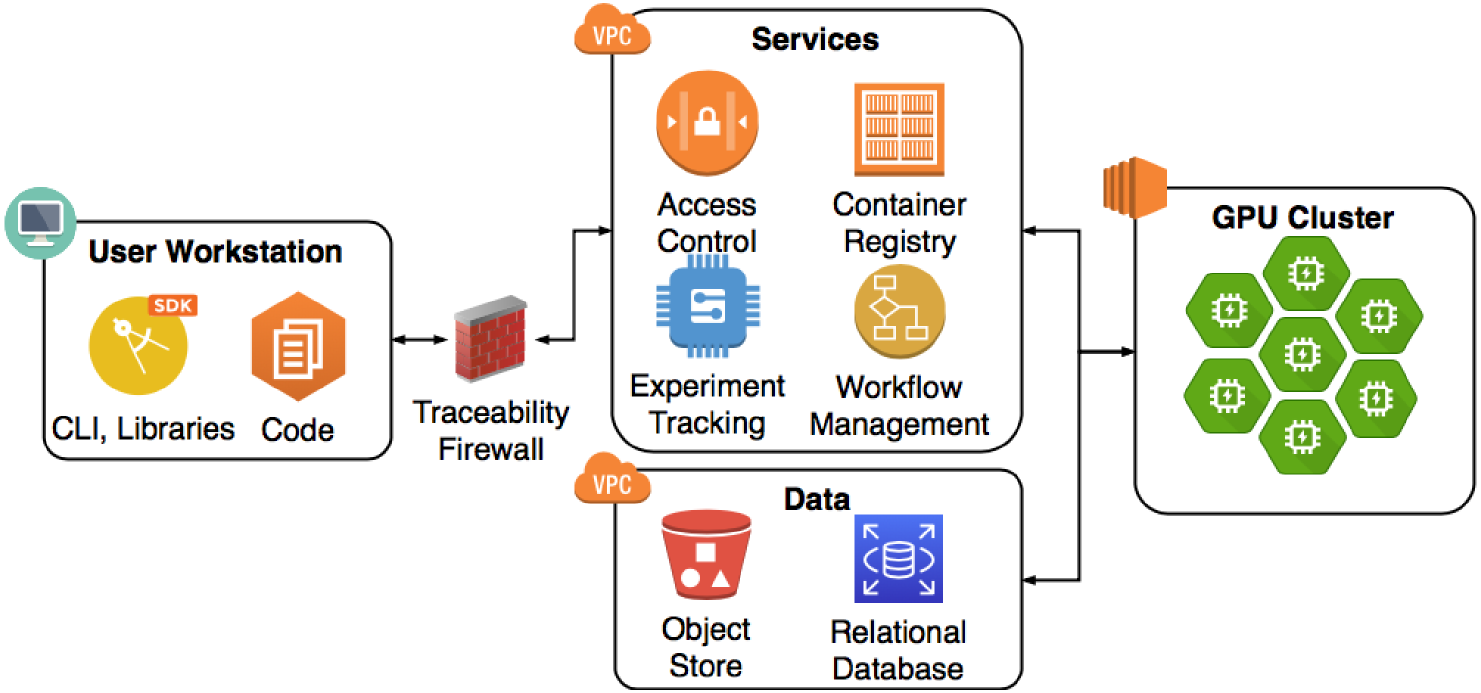


META OPTIMIZATION

Project MagLev: Machine Learning Platform

Architecture

- Scalable Platform for Traceable Machine Learning Workflows
- Self-Documented Experiments
- Services can be used in isolation, or combined for maximum traceability.



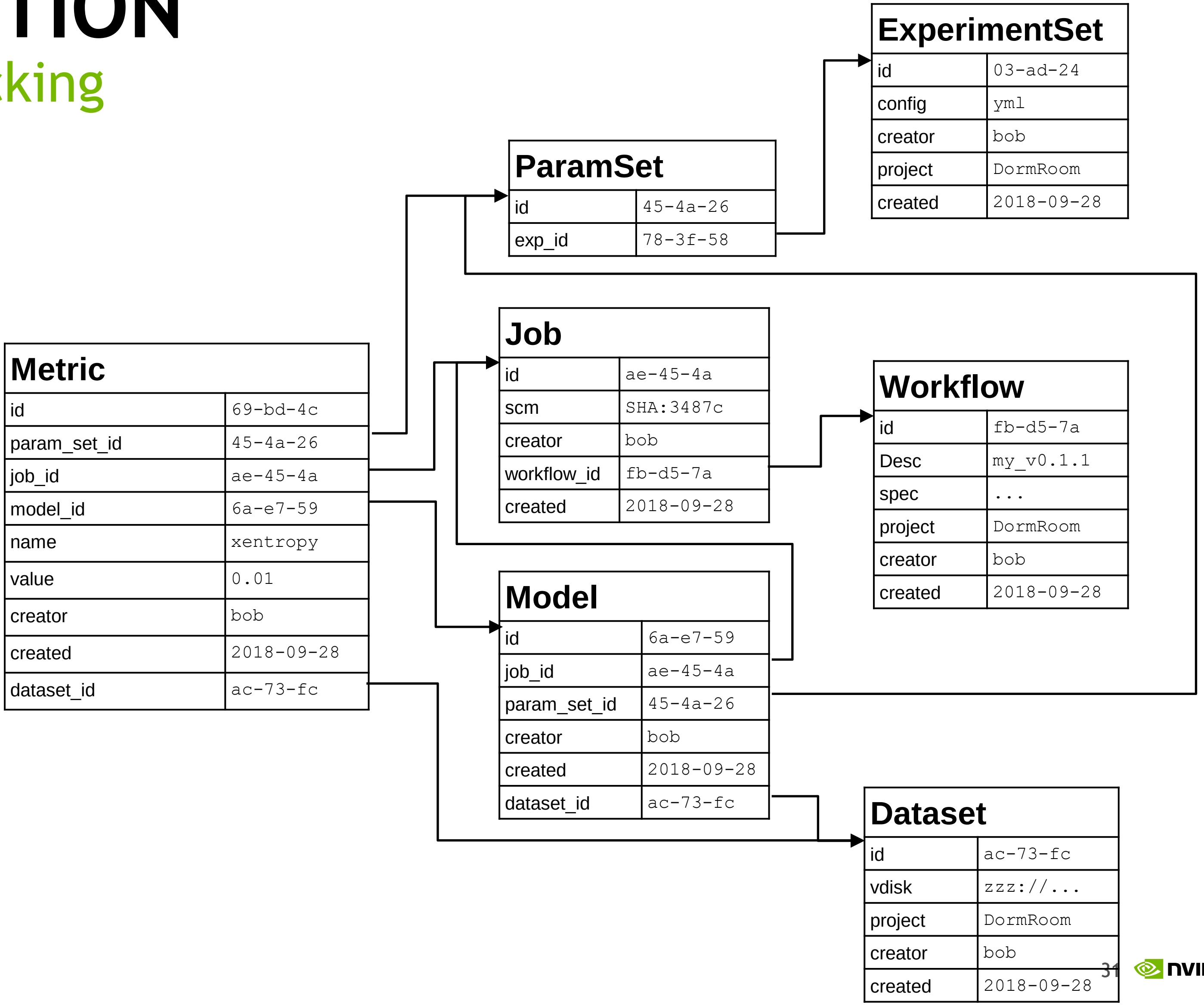
META OPTIMIZATION

MagLev Experiment Tracking

Knowledge Base

- Experiment Data is fully connected.
- Objects are searched through their relationships with others.

➡ No Information silo.

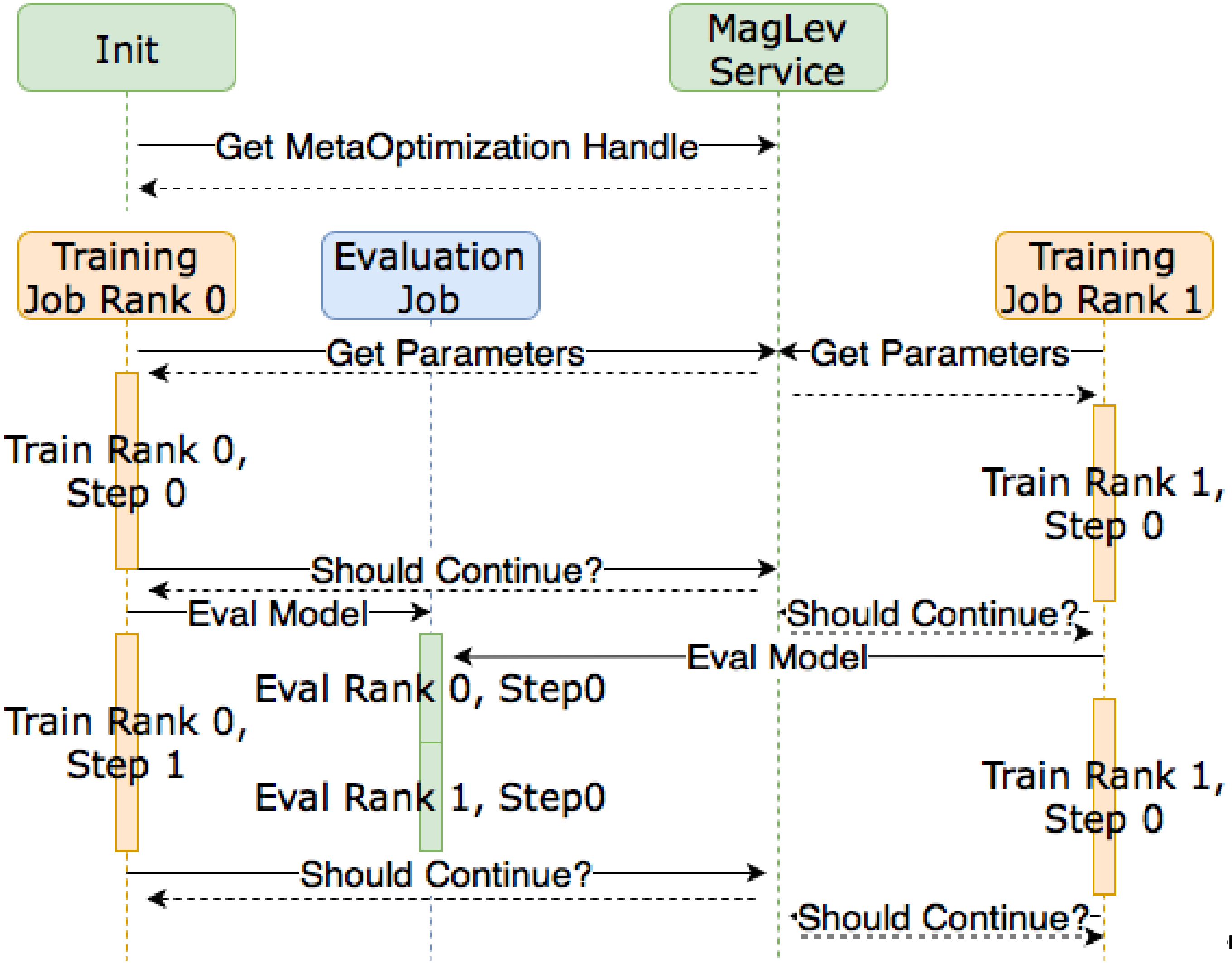


META OPTIMIZATION

Typical Setup

Main SDK Features

- All common parameter types.
- Early-termination methods.
- Standard + custom parameter picking methods.



AGENDA

Contents
Introduction to Reinforcement Learning
Introduction to Metaoptimization (on distributed systems) / Maglev
Metaoptimization and Reinforcement Learning (on distributed systems)
HyperTrick
Results
Conclusion

META OPTIMIZATION + GA3C

Hyper Parameters and Preview of results.

- Learning Rate: log uniform distribution over [1e-5, 1e-2] interval.
- tmax: quantized (q=1) log uniform distribution over [2, 100] interval.
- γ: one of {0.9, 0.95, 0.99, 0.995, 0.999, 0.9995, 0.9999}

Game	Episodes per Phase	N_p	r	α (min[α], E[α])	Score (GA3C)	Score (HyperTrick)
Boxing	2500	10	25%	48.2% (18.87%, 37.75%)	92	98
Centipede	2500	10	25%	52.2% (18.87%, 37.75%)	7386	8707
Ms Pacman	2500	10	25%	46.1% (18.87%, 37.75%)	1978	2112
Pong	2500	5	25%	59.1% (30.51%, 61.02%)	18	18

AGENDA

Contents
Introduction to Reinforcement Learning
Introduction to Metaoptimization (on distributed systems) / Maglev
Metaoptimization and Reinforcement Learning (on distributed systems)
HyperTrick
Results
Conclusion

HYPERTRICK

Early Termination Without Compromise

Successive Halving (SH) [1]

Terminate ½ of workers every $N*2^P$ units of work (P is a phase index).

- ⇨ Requires synchronization between workers.
- ⇨ Assumes relative perf over time is constant.

[1] https://arxiv.org/abs/1502.07943

HYPERTRICK

Early Termination Without Compromise

Successive Halving (SH) [1]
Terminate ½ of workers every $N*2^P$ units of work (P is a phase index). ⇨ Requires synchronization between workers. ⇨ Assumes relative perf over time is constant.
HyperBand [2]
Run several instances of SH in parallel with different values of N. ⇨ Does not assume constant relative perf but still requires synchronization.

[1] https://arxiv.org/abs/1502.07943 [2] http://arxiv.org/abs/1603.06560

HYPERTRICK

Early Termination Without Compromise

Successive Halving (SH) [1]

Terminate ½ of workers every $N*2^P$ units of work (P is a phase index).

- ➡ Requires synchronization between workers.
- ➡ Assumes relative perf over time is constant.

HyperBand [2]

Run several instances of SH in parallel with different values of N.

- ➡ Does not assume constant relative perf but still requires synchronization.

HyperTrick [3]

Parameters:

- N: total number of workers.
- r: eviction rate per phase P.

Worker:

```
while maglev.should_continue():
        work()
        maglev.report_metrics()
```

MagLev:

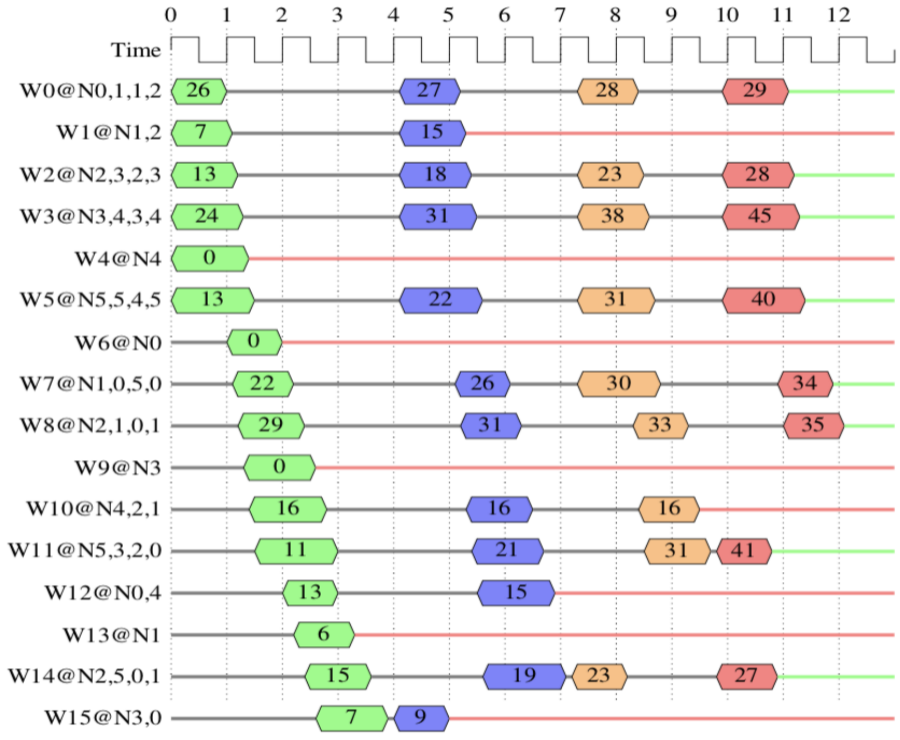
- Let earliest $N(1-\sqrt{r})(1-r^P)$ run.
- Others are terminated, if in bottom $\sqrt{r}$ quantile.
- ➡ Expected number of workers at end of phase P is $N(1-r)^P$

[1] https://arxiv.org/abs/1502.07943 [2] http://arxiv.org/abs/1603.06560 [3] https://arxiv.org/abs/1902.02725

HYPERTRICK v.s. SUCCESSIVE HALVING

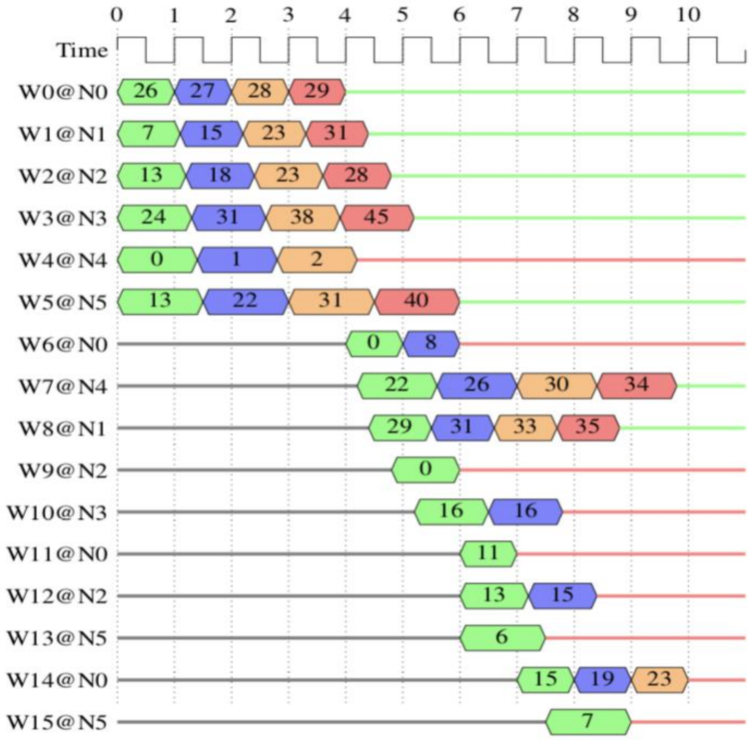
16 workers on 6 nodes running up to 4 iterations

Successive Halving



Must support preemption

HyperTrick



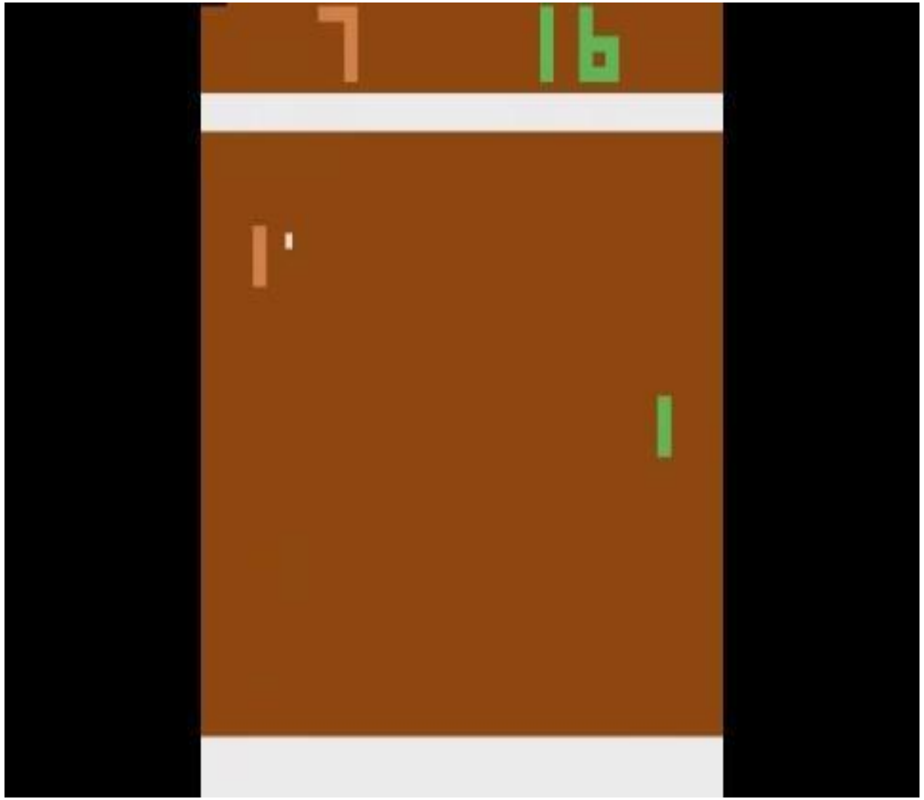
No context switches, shorter wall time

AGENDA

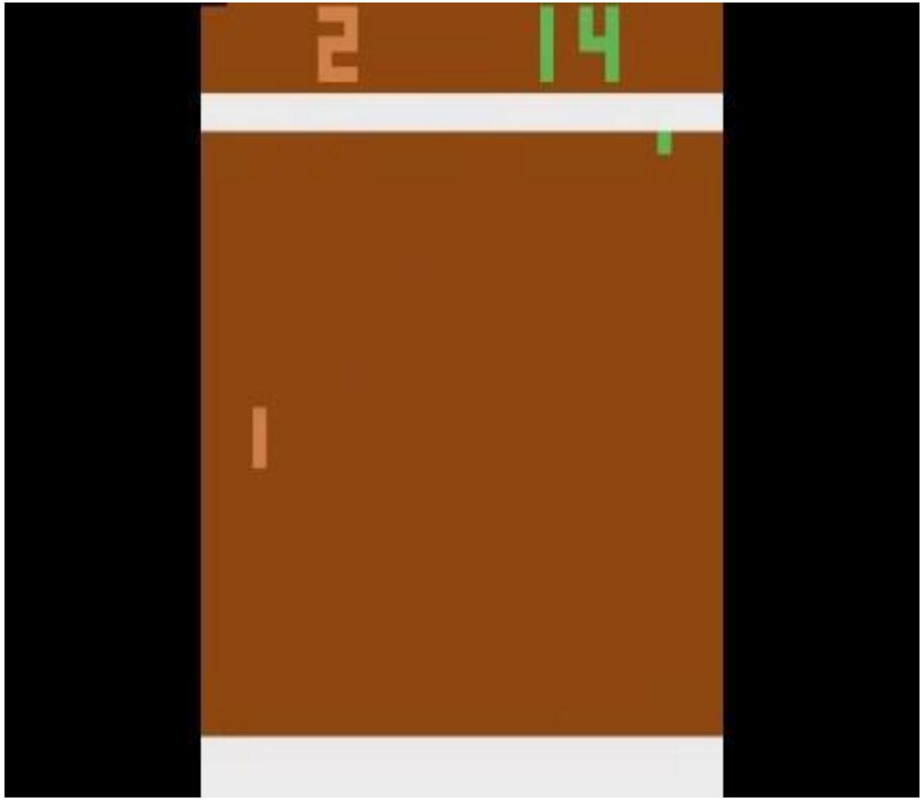
Contents
Introduction to Reinforcement Learning
Introduction to Metaoptimization (on distributed systems) / Maglev
Metaoptimization and Reinforcement Learning (on distributed systems)
HyperTrick
Results
Conclusion

PONG

Videos of Trained Agents



γ=0.9 (short-sighted)

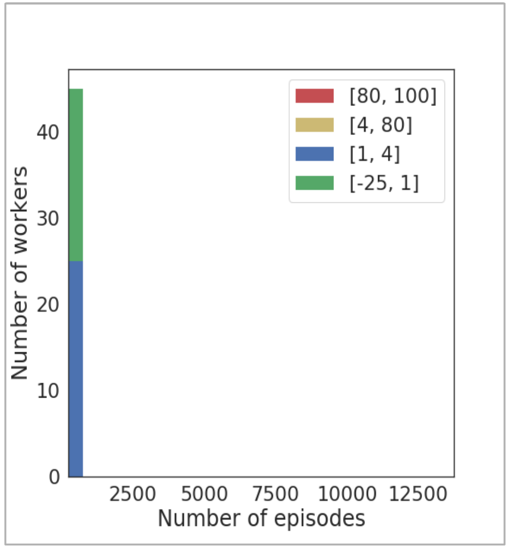


γ=0.995 (far-sighted)

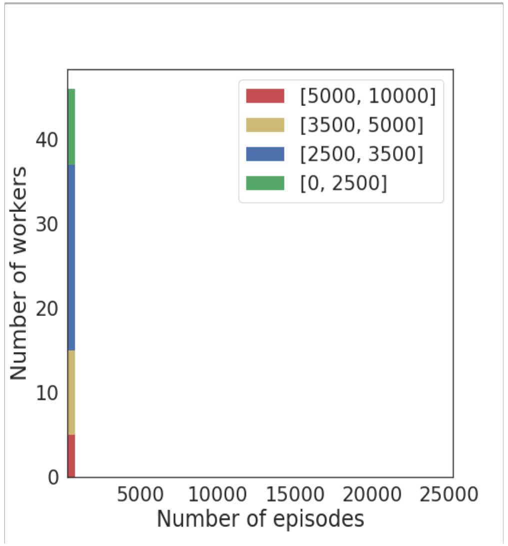
HYPERTRICK

Terminate Underperformers

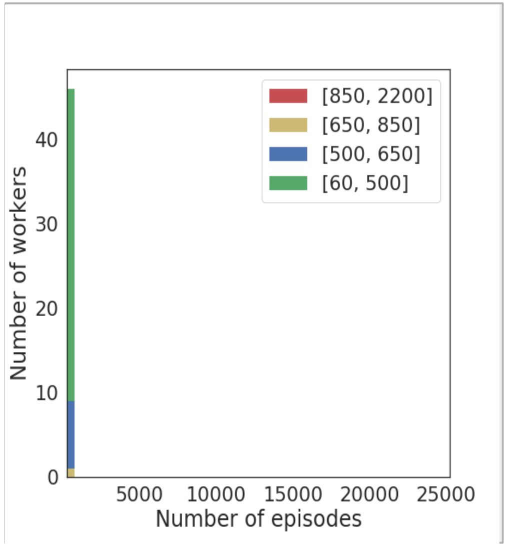
unpainted area = saved resources



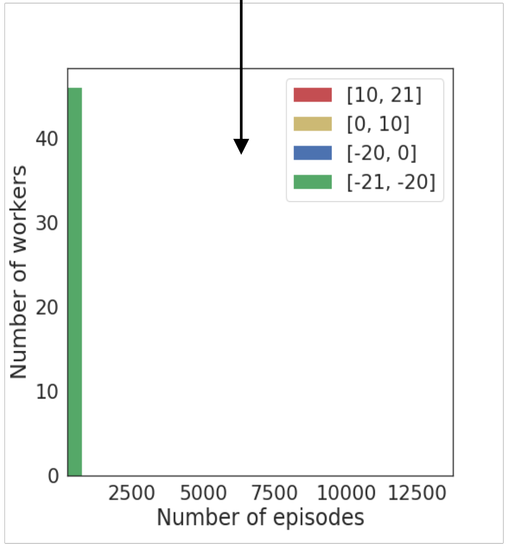
Boxing



Centipede



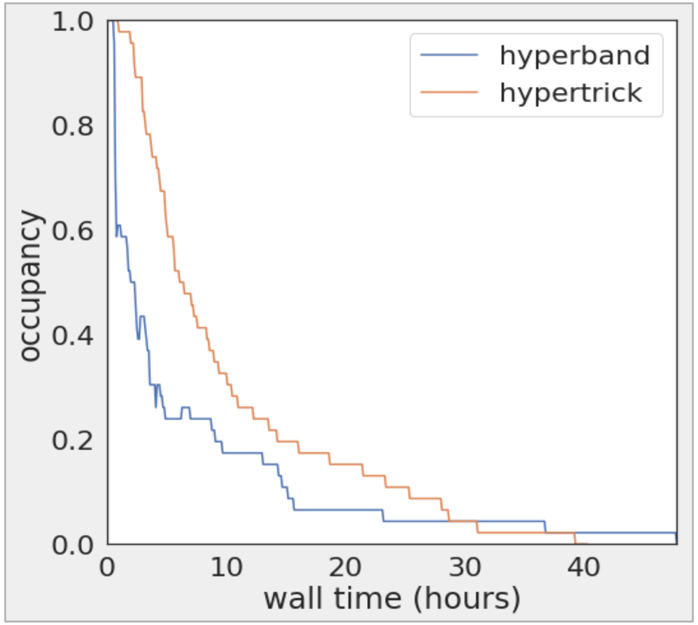
Pacman



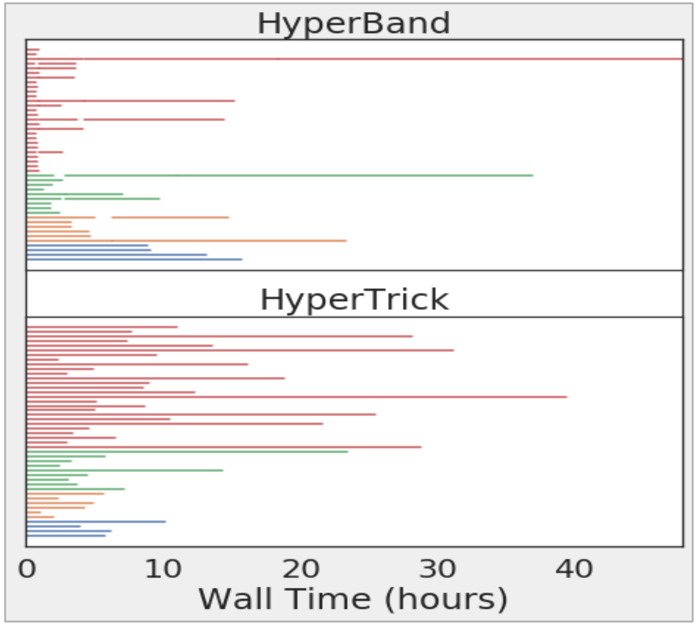
Pong

HYPERTRICK

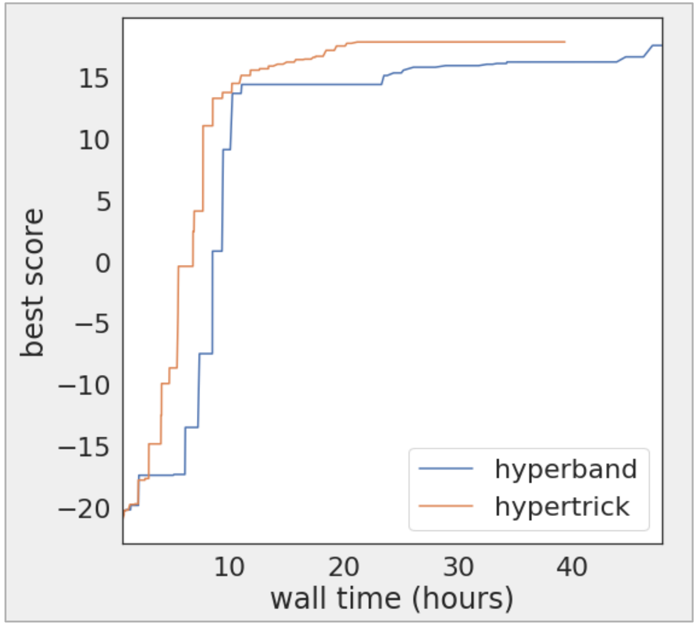
Comparison Against HyperBand



Cluster occupancy
(Pong)



Timelines
(Pong)



Best Score v.s. Wall Time (Pong)

META OPTIMIZATION

Experimental Comparison of HyperTrick v.s. HyperBand

Game	Method	Best Score	Total Wall Time	Time To Best Score	Best Config		
					LR	γ	T_{max}
Boxing	HyperBand	96	51h	29h	$3.3e^{-4}$	0.99	13
	HyperTrick	95	38h	13h	$3.3e^{-4}$	0.99	13
Centipede	HyperBand	8521	42h	2h	$5.4e^{-3}$	0.9995	72
	HyperTrick	8667	38h	29h	$1.2e^{-4}$	0.9999	33
Pacman	HyperBand	2456	31h	26h	$1.6e^{-4}$	0.95	73
	HyperTrick	2243	27h	16h	$1.6e^{-4}$	0.95	73
Pong	HyperBand	17.5	48h	47h	2.0^{-3}	0.95	64
	HyperTrick	17.8	39h	22h	$5.9e^{-4}$	0.995	6

Table 3: HyperBand vs HyperTrick results on four Atari games.

META OPTIMIZATION + GA3C

Conclusion

Take Away
RL is unstable => meta optimization is useful.
Hypertrick, a new algorithm for metaoptimization.
GA3C + HyperTrick + Maglev is effective.
Our paper: https://arxiv.org/abs/1902.02725
GA3C: https://github.com/NVlabs/GA3C
MagLev info: Yehia Khoja (ykhoja@nvidia.com)

Related Talks			
S9649	Wed 9:00am	NVIDIA's AI Infrastructure for Self-Driving Cars	Clement Farabet
S9613	Wed 10:00am	Deep Active Learning	Adam Lesnikowski
S9911	Wed 2:00pm	Determinism In Deep Learning	Duncan Riach
S9630	Thu 2:00pm	Scaling Up DL for Autonomous Driving	Jose Alvarez
S9987	Thu 9:00am	MagLev: production-grade AI platform...	Divya Vavili