NASA FRONTIER DEVELOPMENT LAB

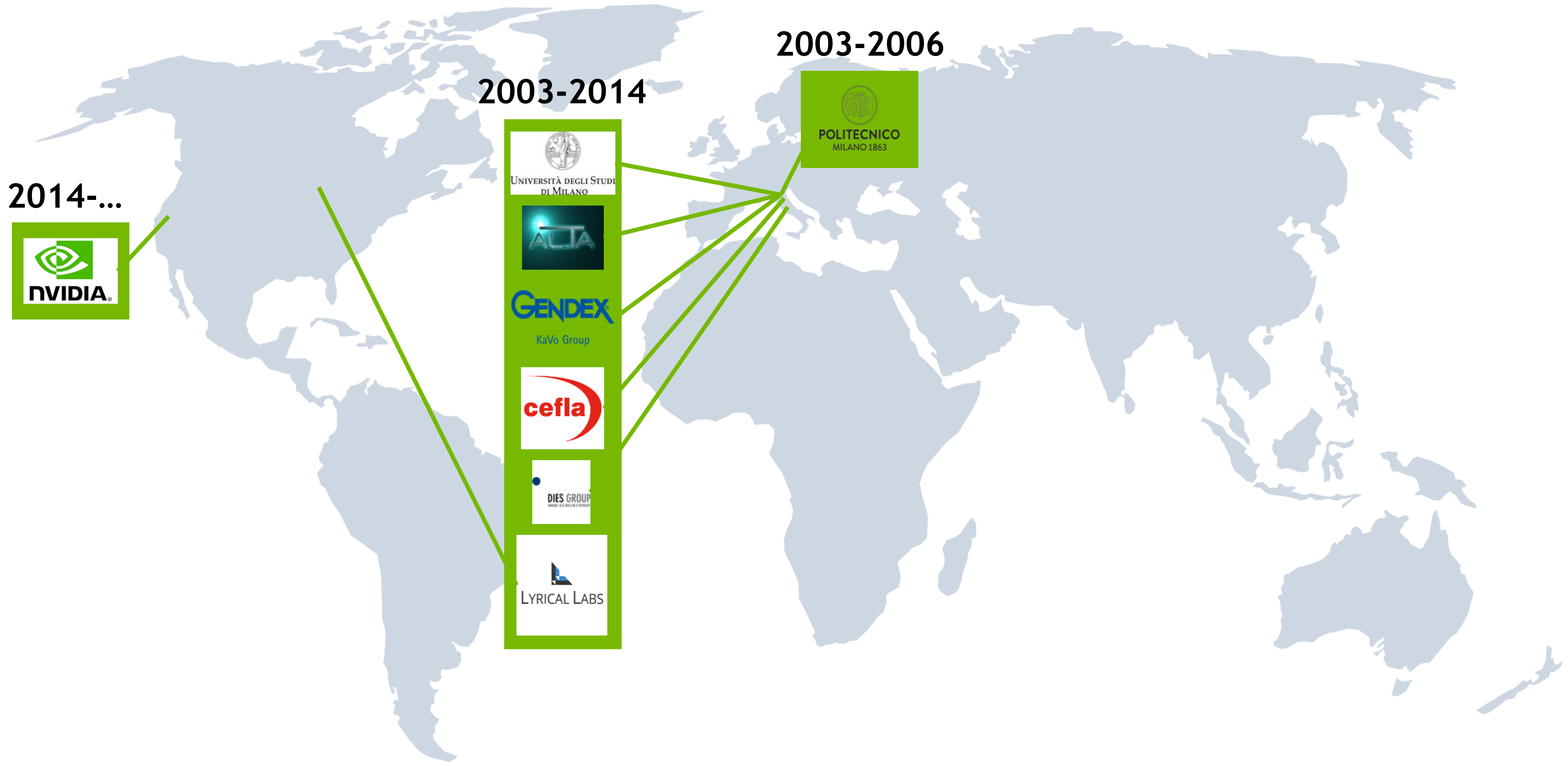
ARTIFICIAL INTELLIGENCE ACCELERATOR

NVIDIA

COMPUTATIONAL ASPECTS OF REINFORCEMENT LEARNING

iuri frosio, Jul 6-17th, 2018

ifrosio@nvidia.com


2014-...
NVIDIA
2003-2014
Università degli Studi di Milano
ALTA
GENDEX
KaVo Group
cefla
DIES GROUP
Lyrical Labs
2003-2006
POLITECNICO
MILANO 1863

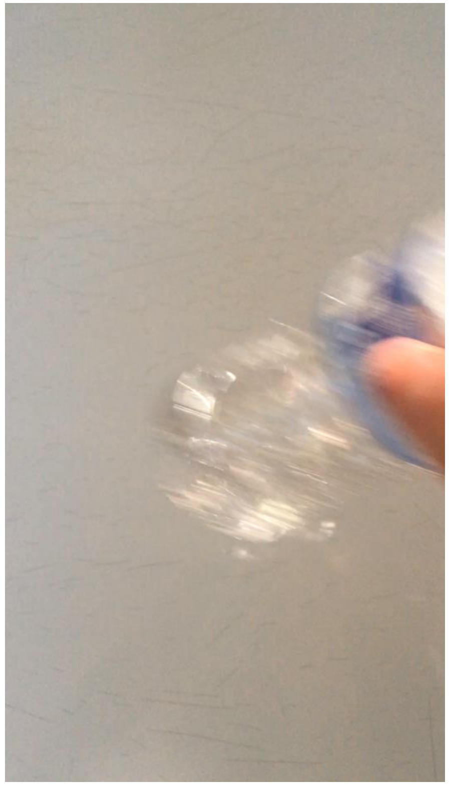
Video from https://www.youtube.com/watch?v=u0-pfzKbh2k

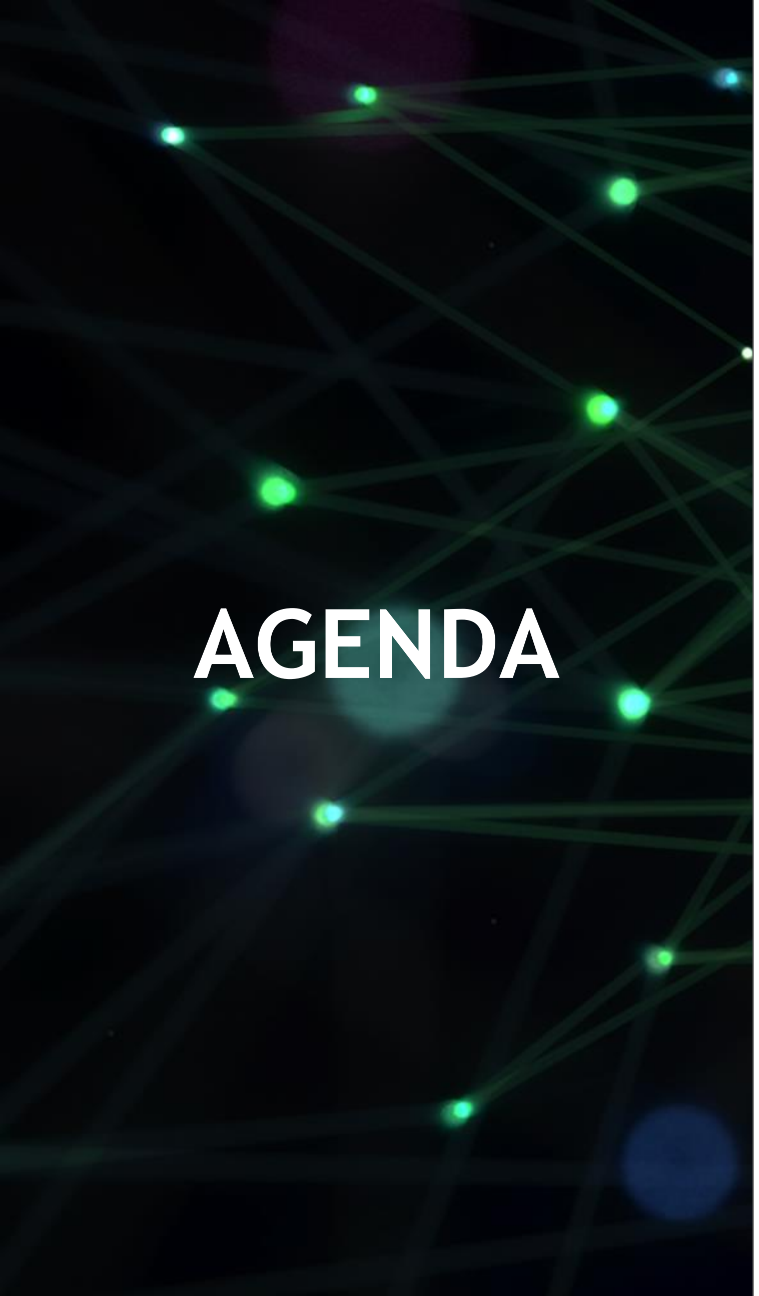
INTRODUCTION TO REINFORCEMENT LEARNING: CAN A BIOMEDICAL ENGINEER LAND A ROCKET?

LEARNING TO LAND A ROCKET?

From bottle experiments to simulation

	My attempt	Problem	Solution
Rocket model	Bottle	Not so realistic	Simulation
Training procedure	Trial and error	Time consuming	Parallelize (GPU)
How to improve	Experience (learn motion pattern)	Slow, no hints about the correct pattern	Gradient on the proper cost function





AGENDA

1. GPU-based A3C for Deep Reinforcement Learning
(The basics of Reinforcement Learning on a CPU/GPU)

2. Cule*: GPU accelerated RL
(Moving Reinforcement Learning on a GPU)

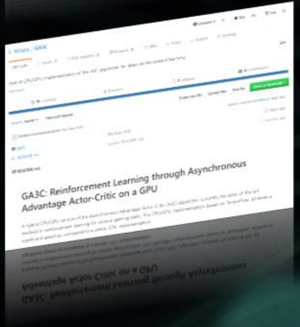
3. Sim-to-Real Transfer of Accurate Grasping with Eye-In-Hand Observations and Continuous Control
(Imitation Learning and Sample Efficiency)

4. Conclusion

OpenReview

Reinforcement Learning through Asynchronous Advantage Actor-Critic on a GPU

An ICLR 2017 paper



NVlabs / GA3C

GA3C: Reinforcement Learning through Asynchronous Advantage Actor-Critic on a GPU

A github project

GPU-BASED A3C FOR DEEP REINFORCEMENT LEARNING

M. Babaeizadeh[†,‡], I.Frosio[‡], S.Tyree[‡], J. Clemons[‡], J.Kautz[‡]

[†]University of Illinois at Urbana-Champaign, USA [‡]NVIDIA, USA

GPU-BASED A3C FOR DEEP REINFORCEMENT LEARNING

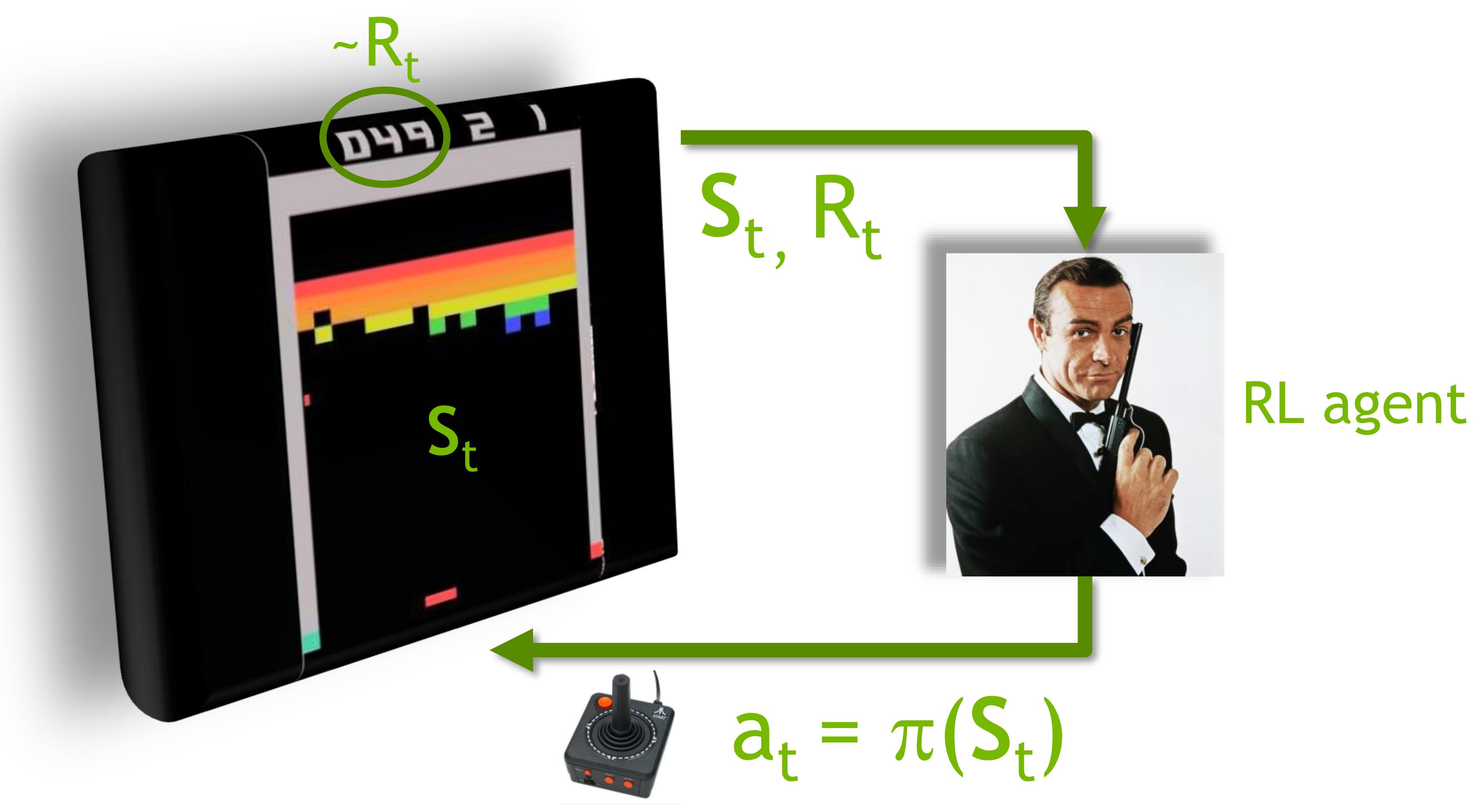
GPU-BASED A3C FOR DEEP REINFORCEMENT LEARNING

Learning to accomplish a task

Image from www.33rdsquare.com

GPU-BASED A3C FOR DEEP REINFORCEMENT LEARNING

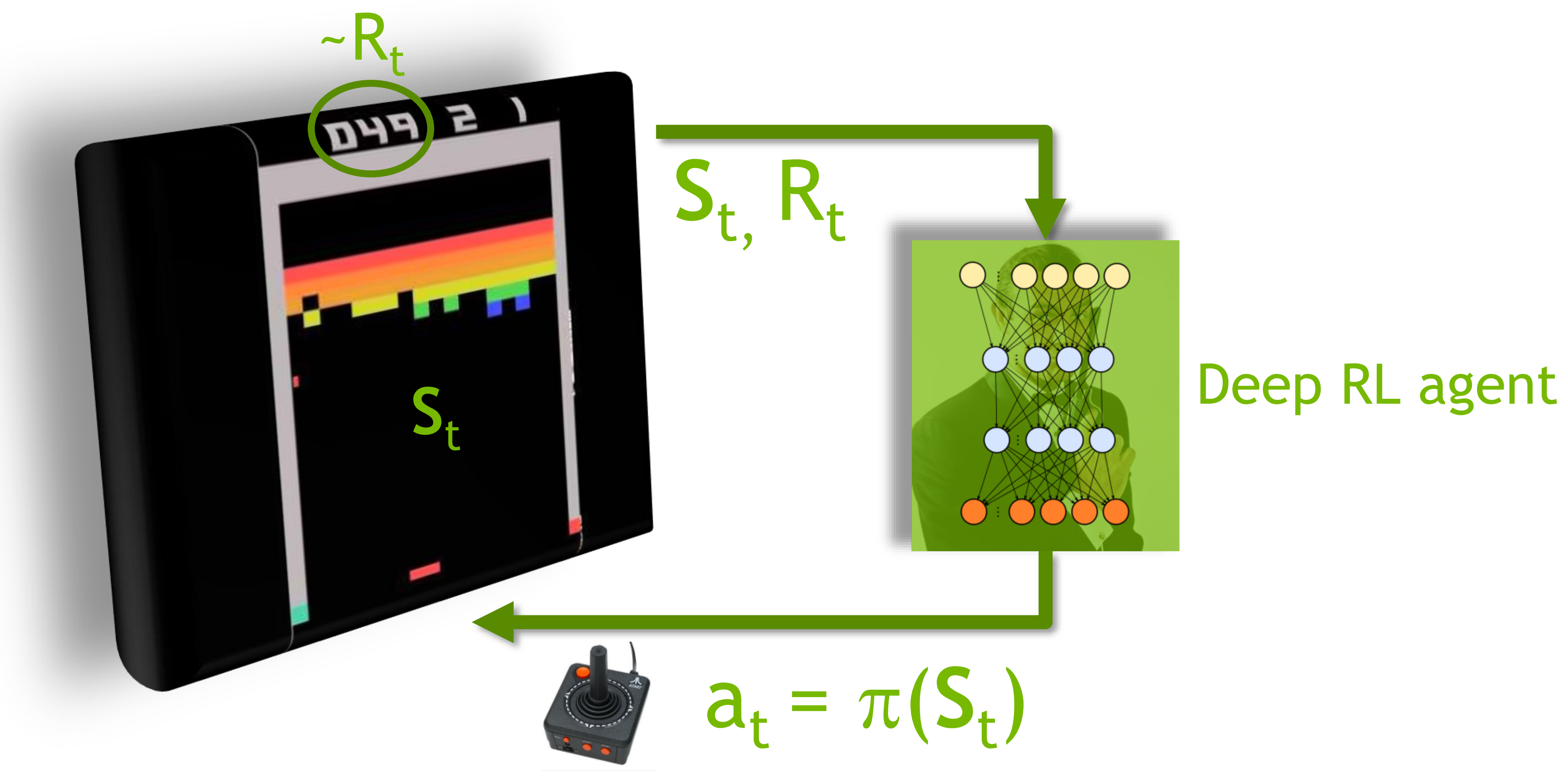
Definitions



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

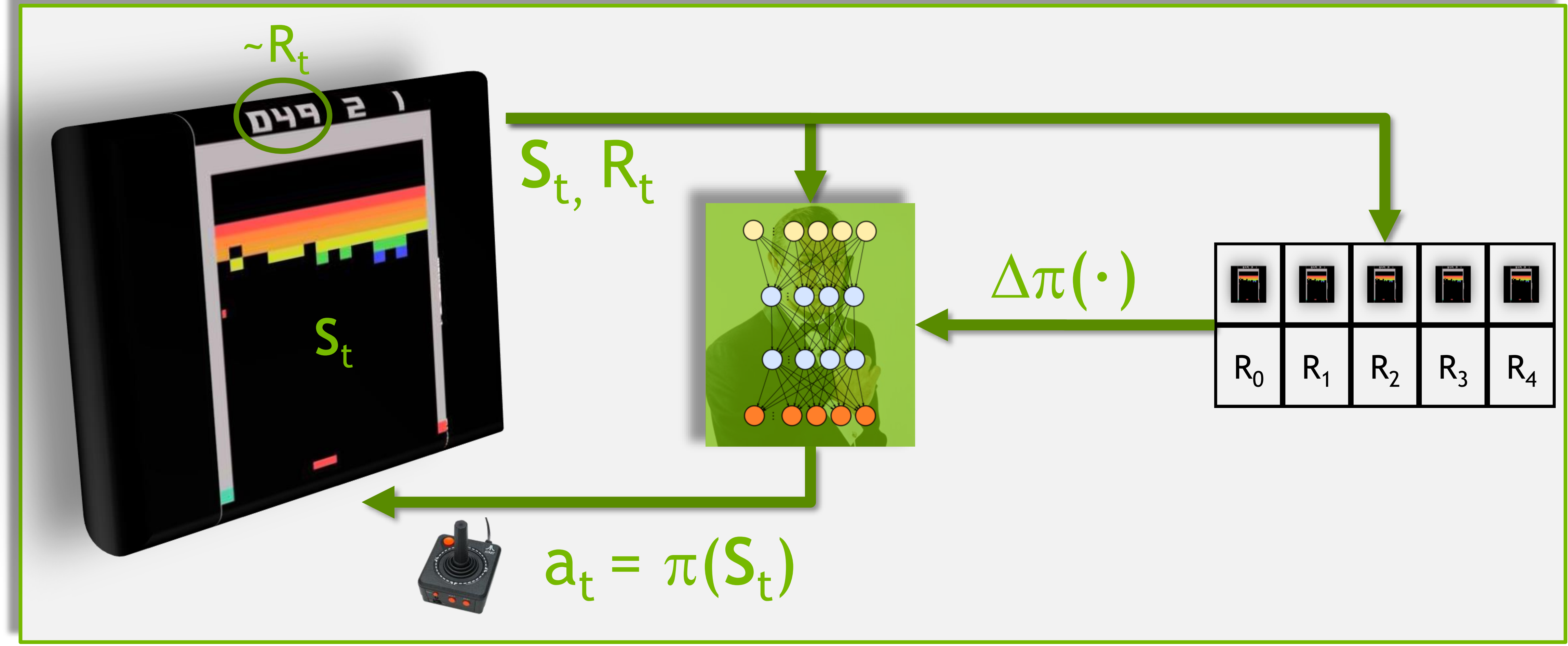
GPU-BASED A3C FOR DEEP REINFORCEMENT LEARNING

Definitions



GPU-BASED A3C FOR DEEP REINFORCEMENT LEARNING

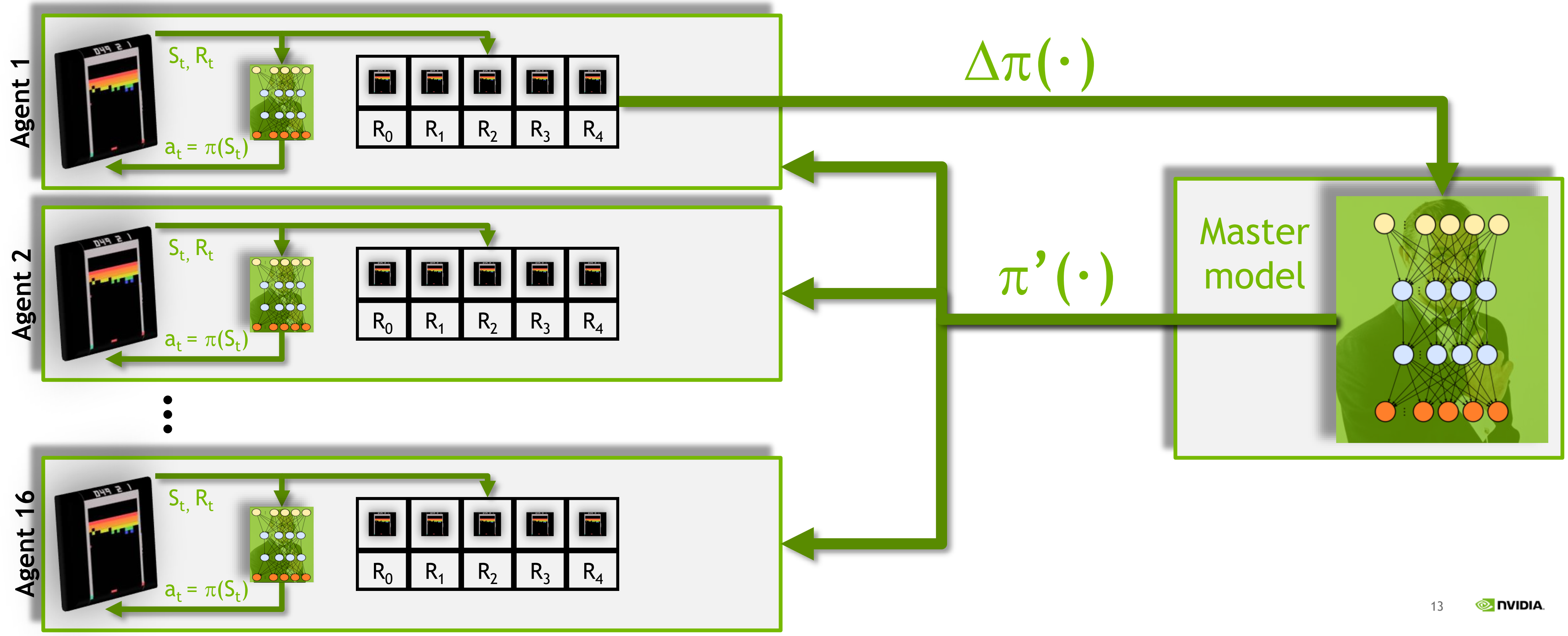
Definitions



GPU-BASED A3C FOR DEEP REINFORCEMENT LEARNING

GPU-BASED A3C FOR DEEP REINFORCEMENT LEARNING

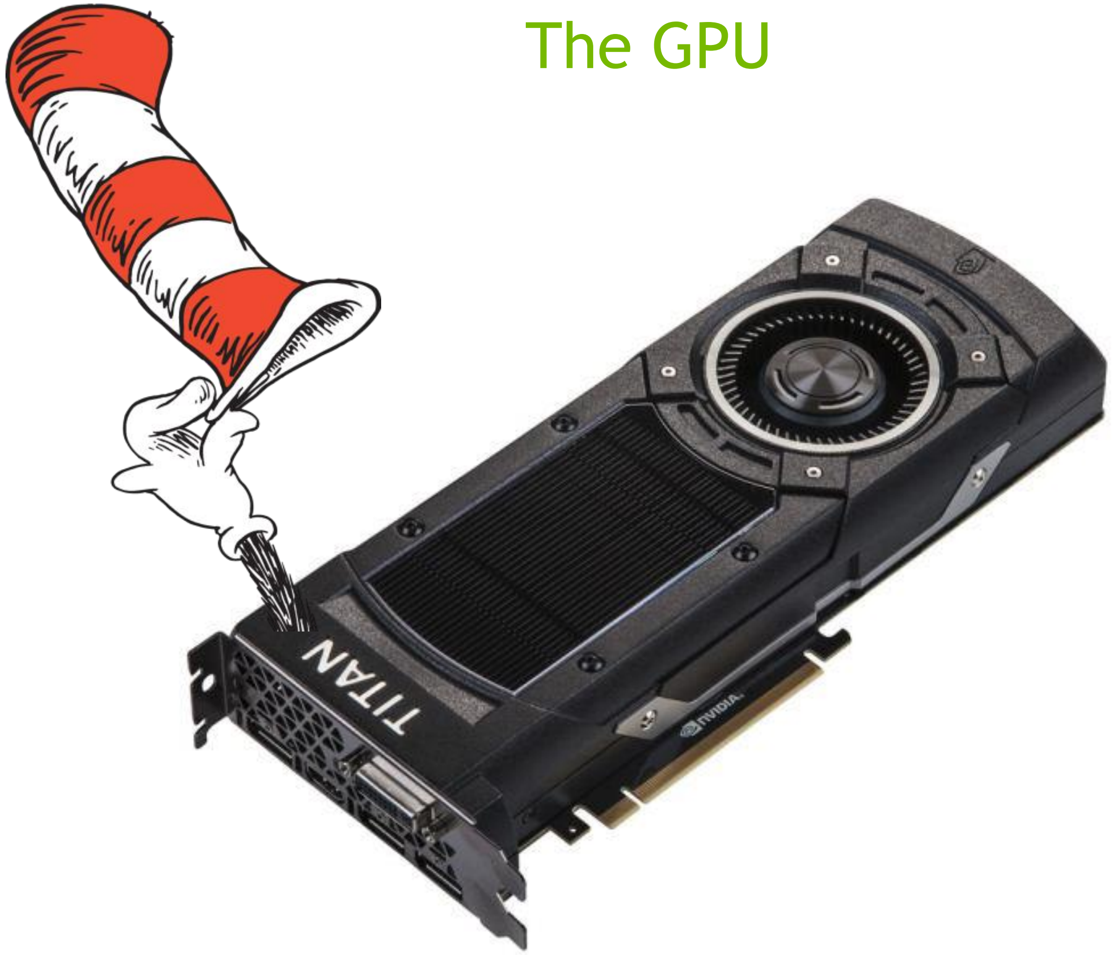
Asynchronous Advantage Actor-Critic (Mnih et al., arXiv:1602.01783v2, 2015)



GPU-BASED A3C FOR DEEP REINFORCEMENT LEARNING

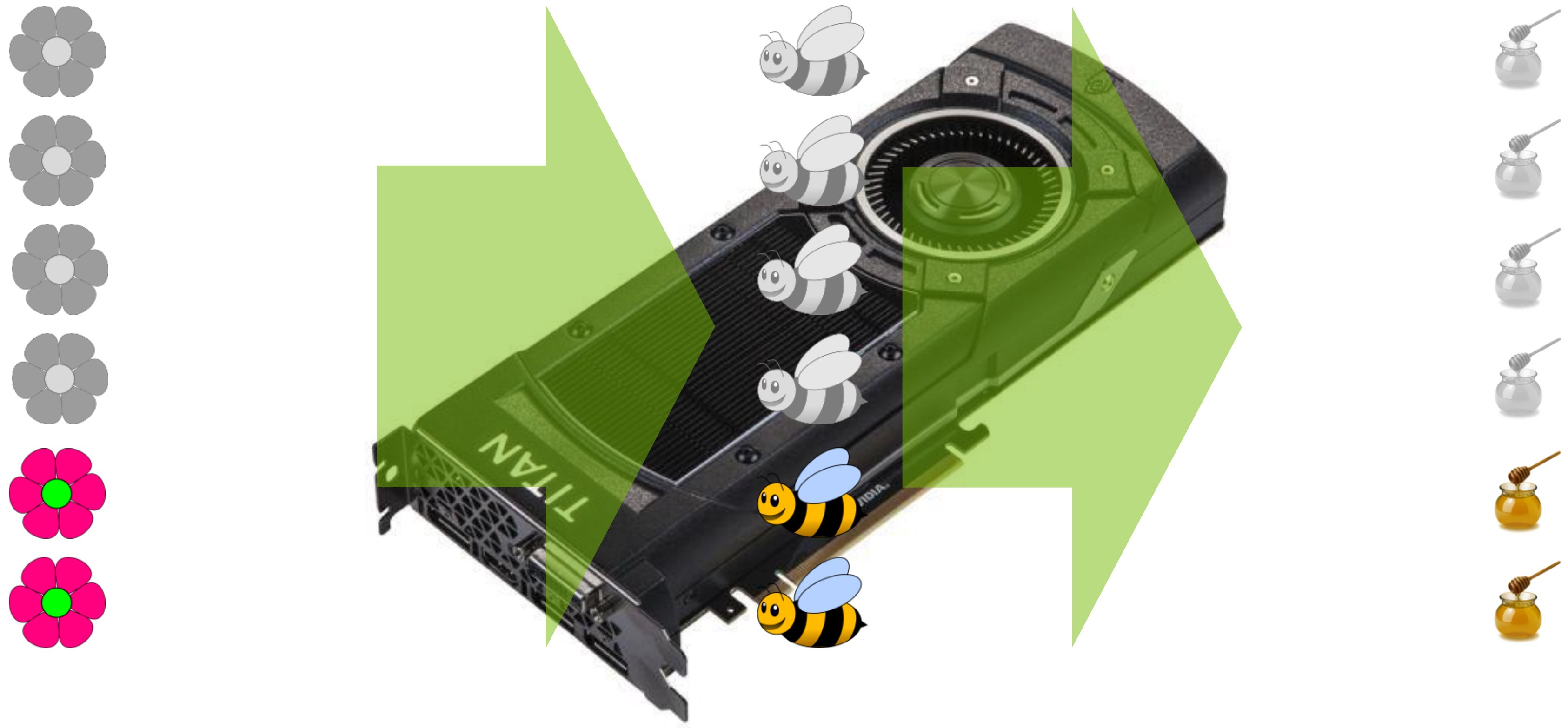
GPU-BASED A3C FOR DEEP REINFORCEMENT LEARNING

The GPU



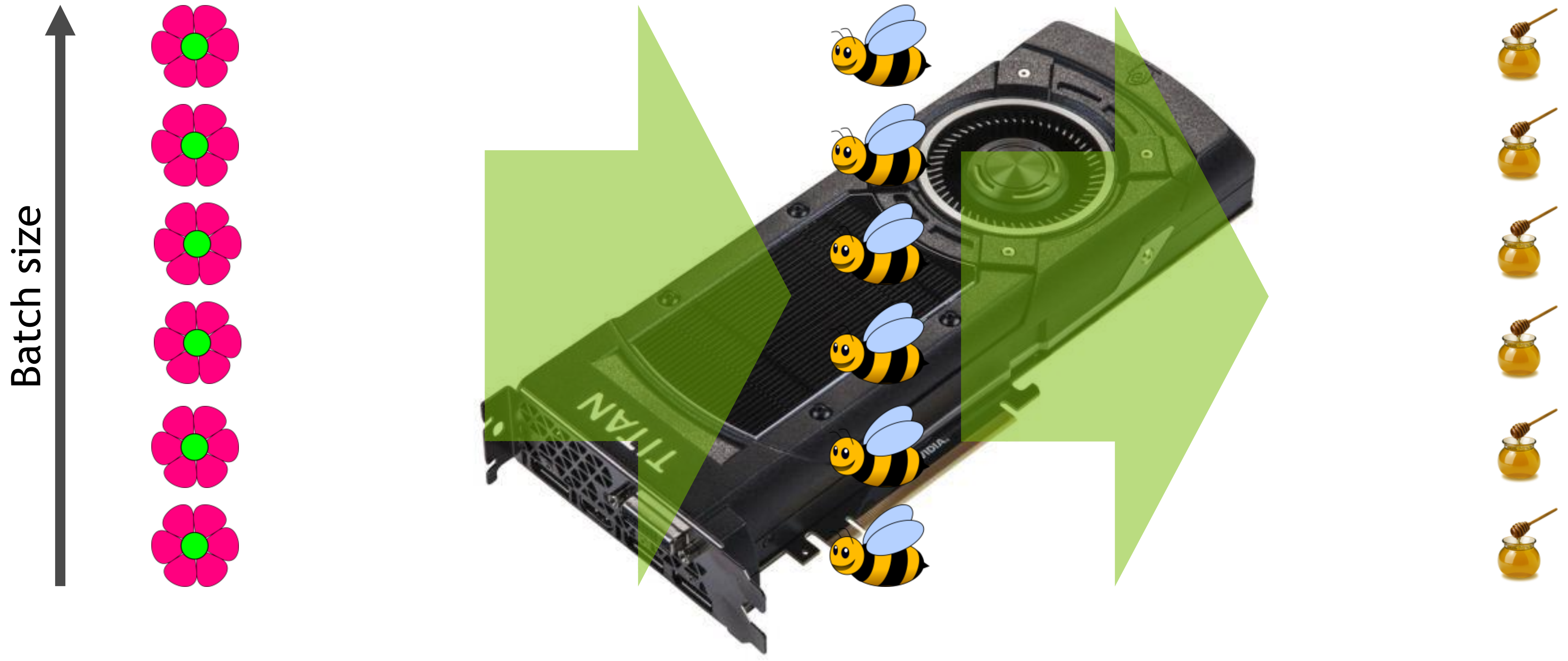
GPU-BASED A3C FOR DEEP REINFORCEMENT LEARNING

LOW OCCUPANCY (33%)



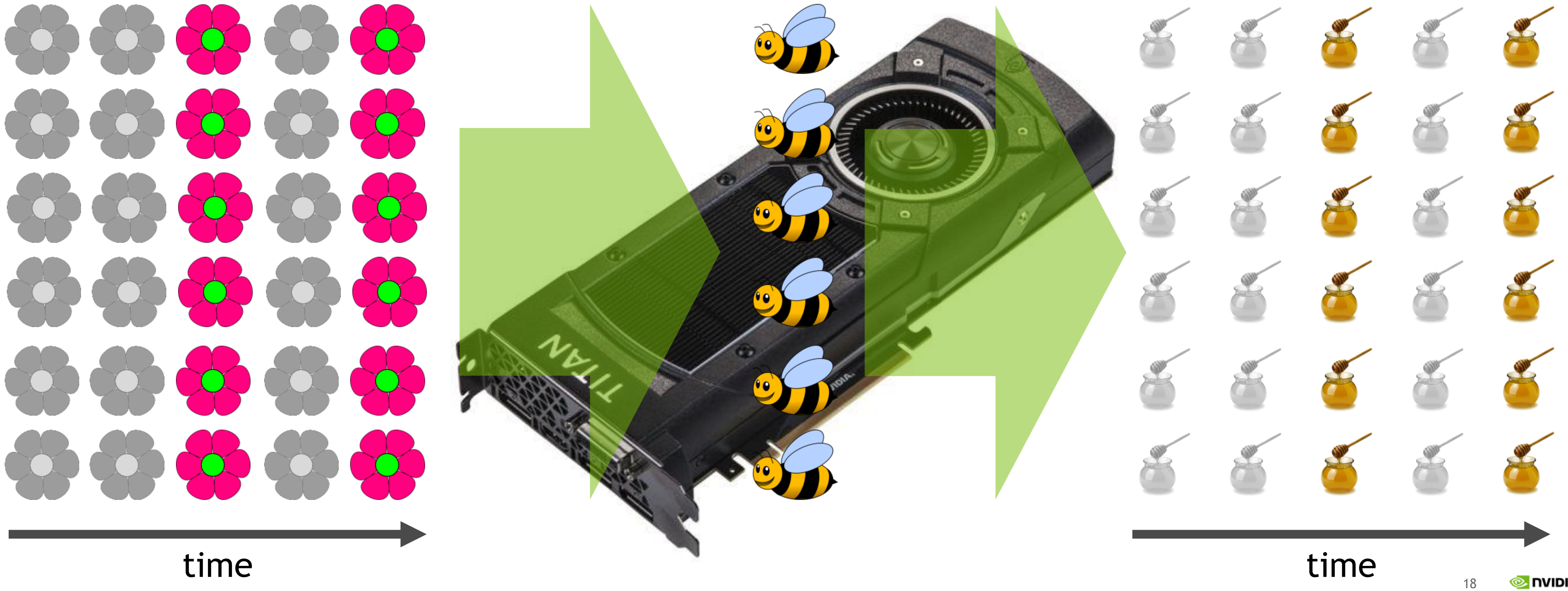
GPU-BASED A3C FOR DEEP REINFORCEMENT LEARNING

HIGH OCCUPANCY (100%)



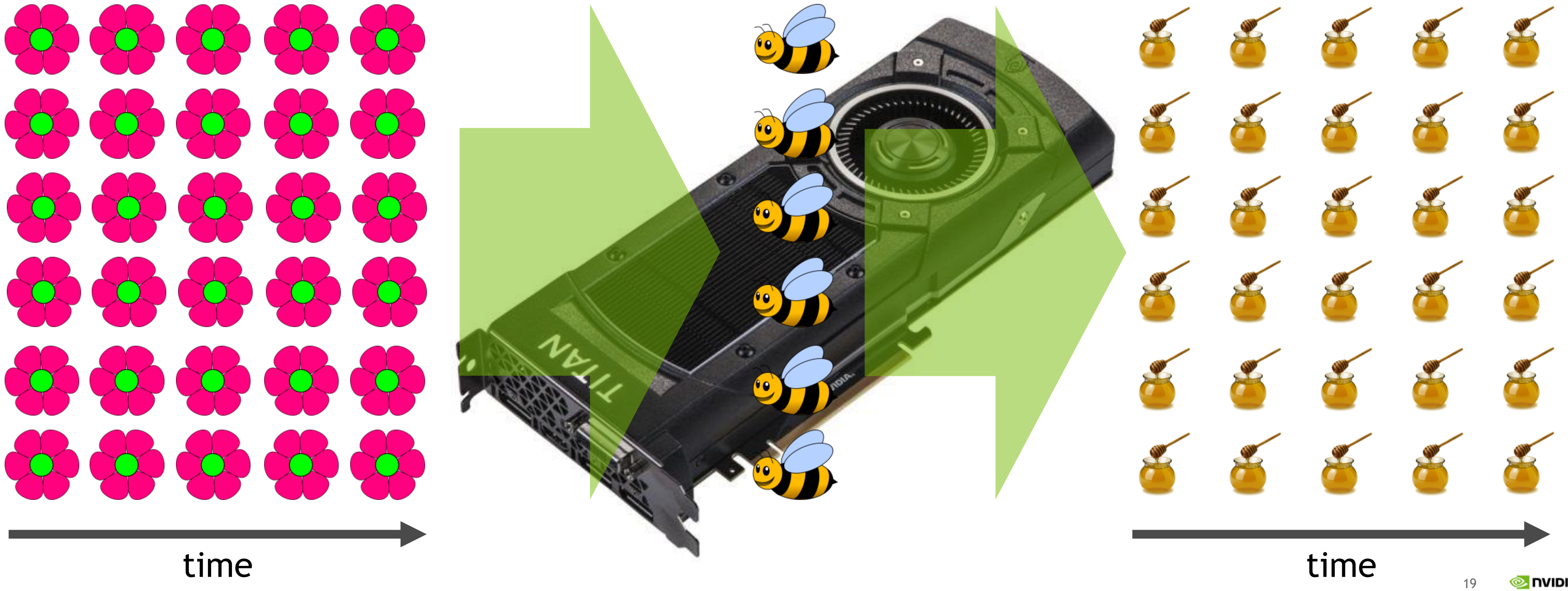
GPU-BASED A3C FOR DEEP REINFORCEMENT LEARNING

HIGH OCCUPANCY (100%), LOW UTILIZATION (40%)



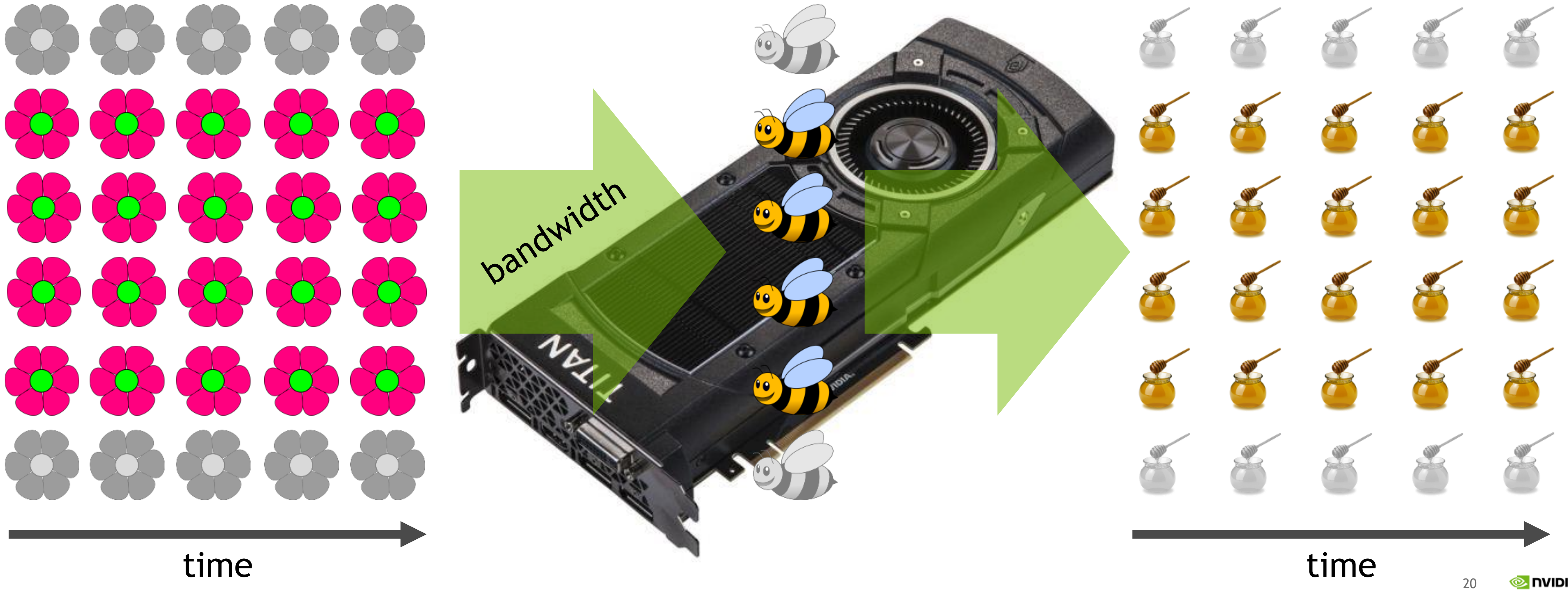
GPU-BASED A3C FOR DEEP REINFORCEMENT LEARNING

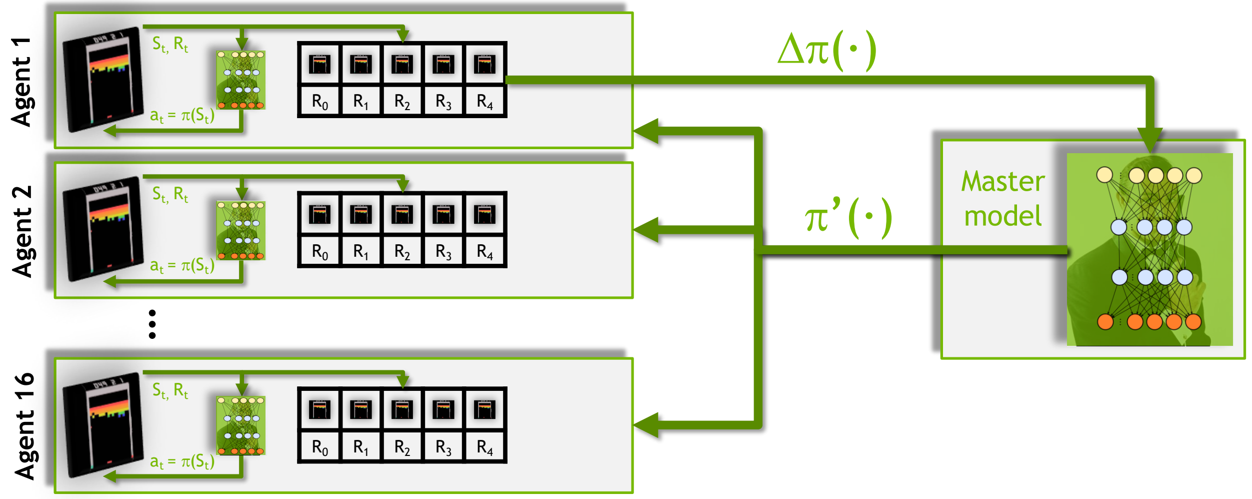
HIGH OCCUPANCY (100%), HIGH UTILIZATION (100%)



GPU-BASED A3C FOR DEEP REINFORCEMENT LEARNING

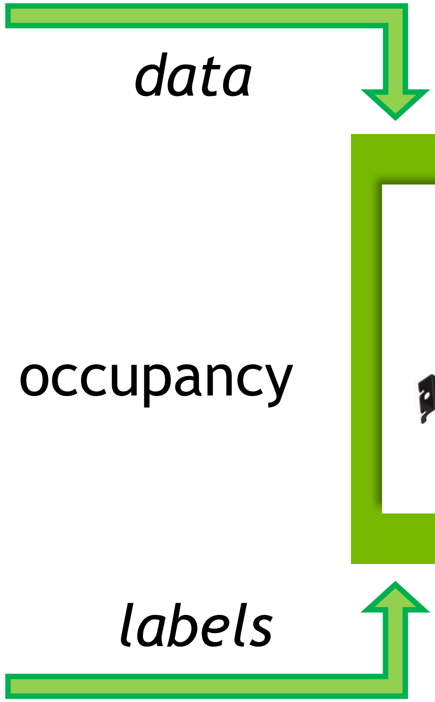
BANDWIDTH LIMITED




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

MAPPING DEEP PROBLEMS TO A GPU

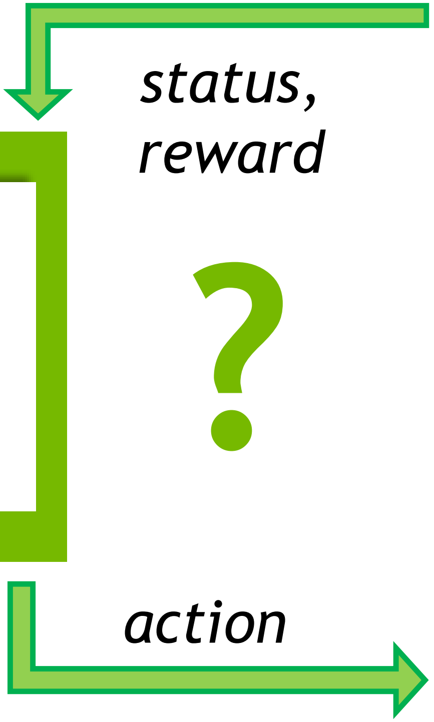
REGRESSION, CLASSIFICATION, …

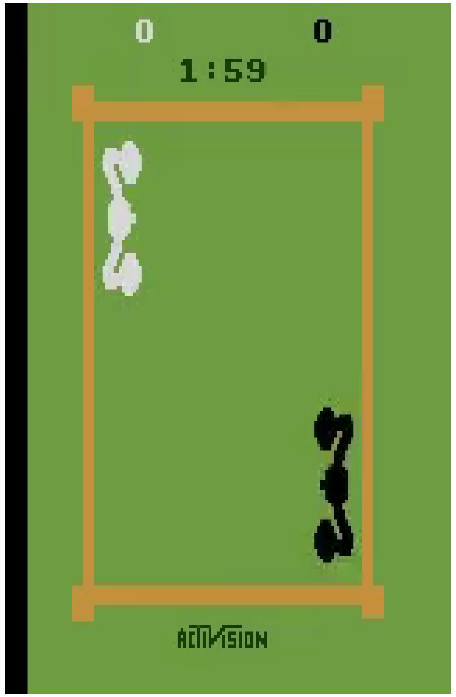


100% utilization / occupancy

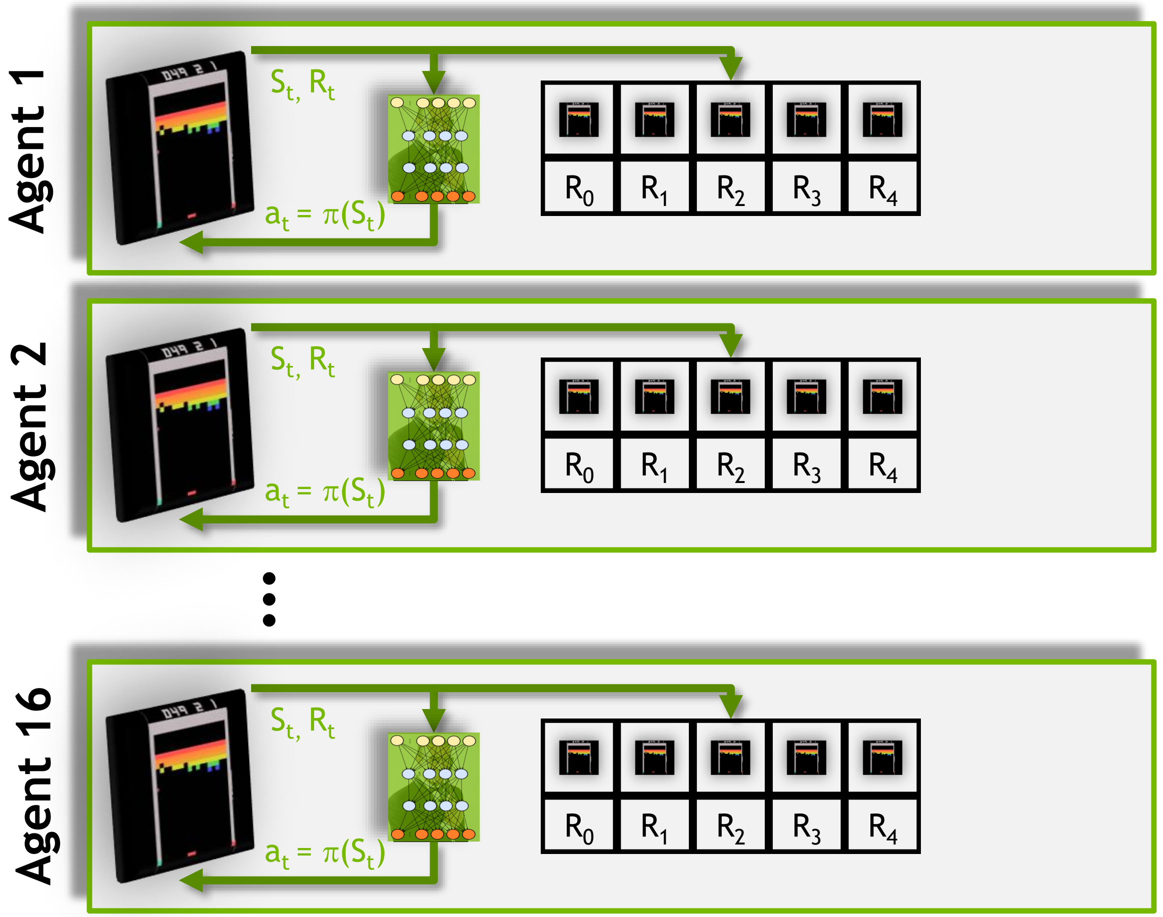
Pear, pear, pear, pear, …
Empty, empty, …
Fig, fig, fig, fig, fig, fig,
Strawberry, Strawberry,
Strawberry, …
…

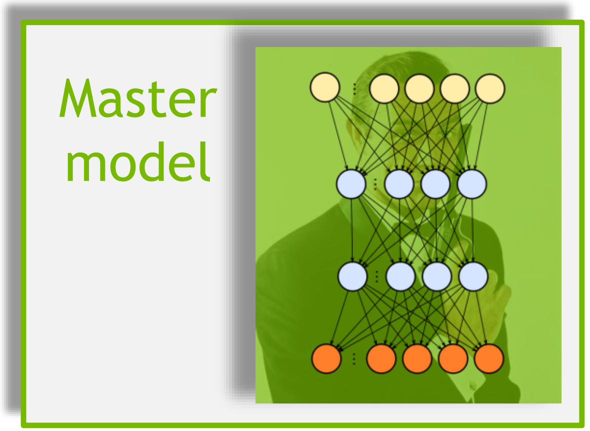
REINFORCEMENT LEARNING



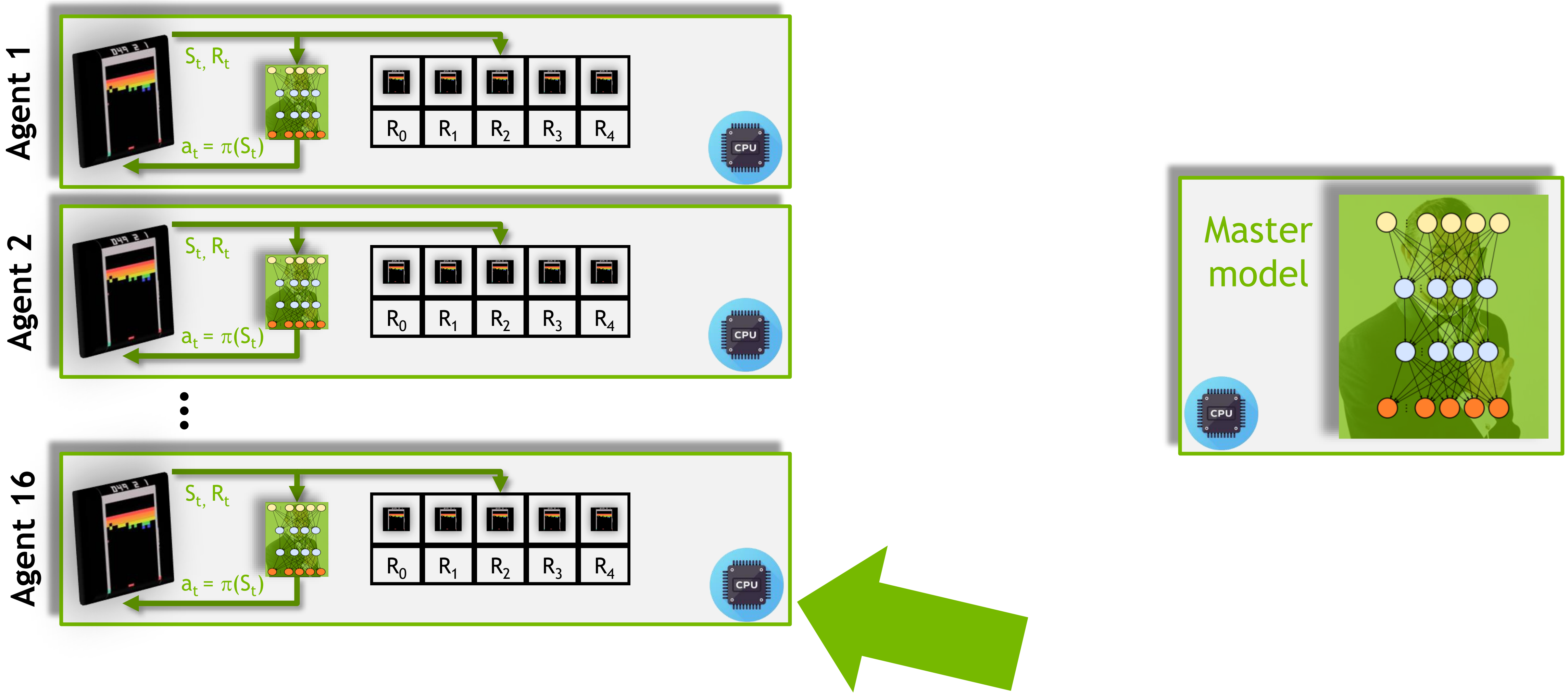


A3C

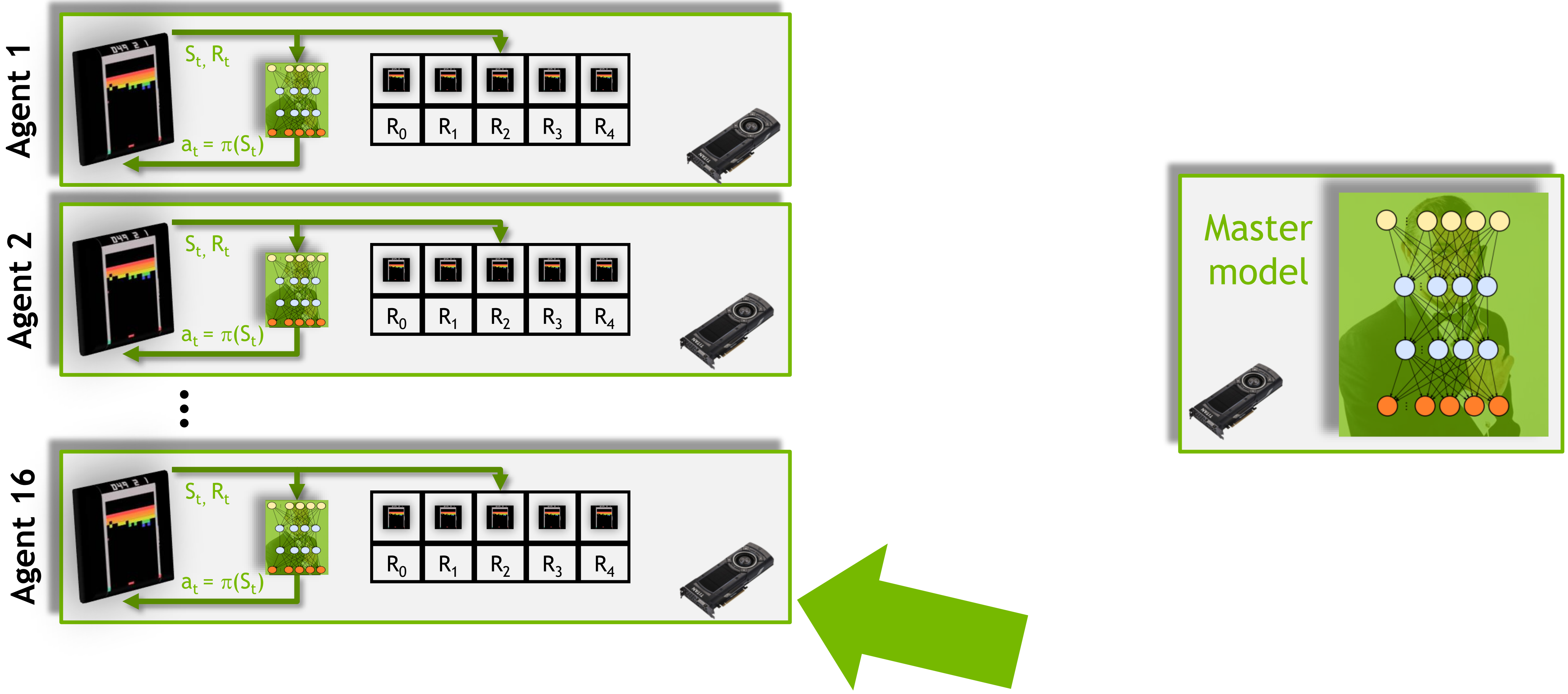

Agent 1
S_t, R_t
$a_t = \pi(S_t)$
R_0
R_1
R_2
R_3
R_4
Agent 2
S_t, R_t
$a_t = \pi(S_t)$
R_0
R_1
R_2
R_3
R_4
Agent 16
S_t, R_t
$a_t = \pi(S_t)$
R_0
R_1
R_2
R_3
R_4


Master model

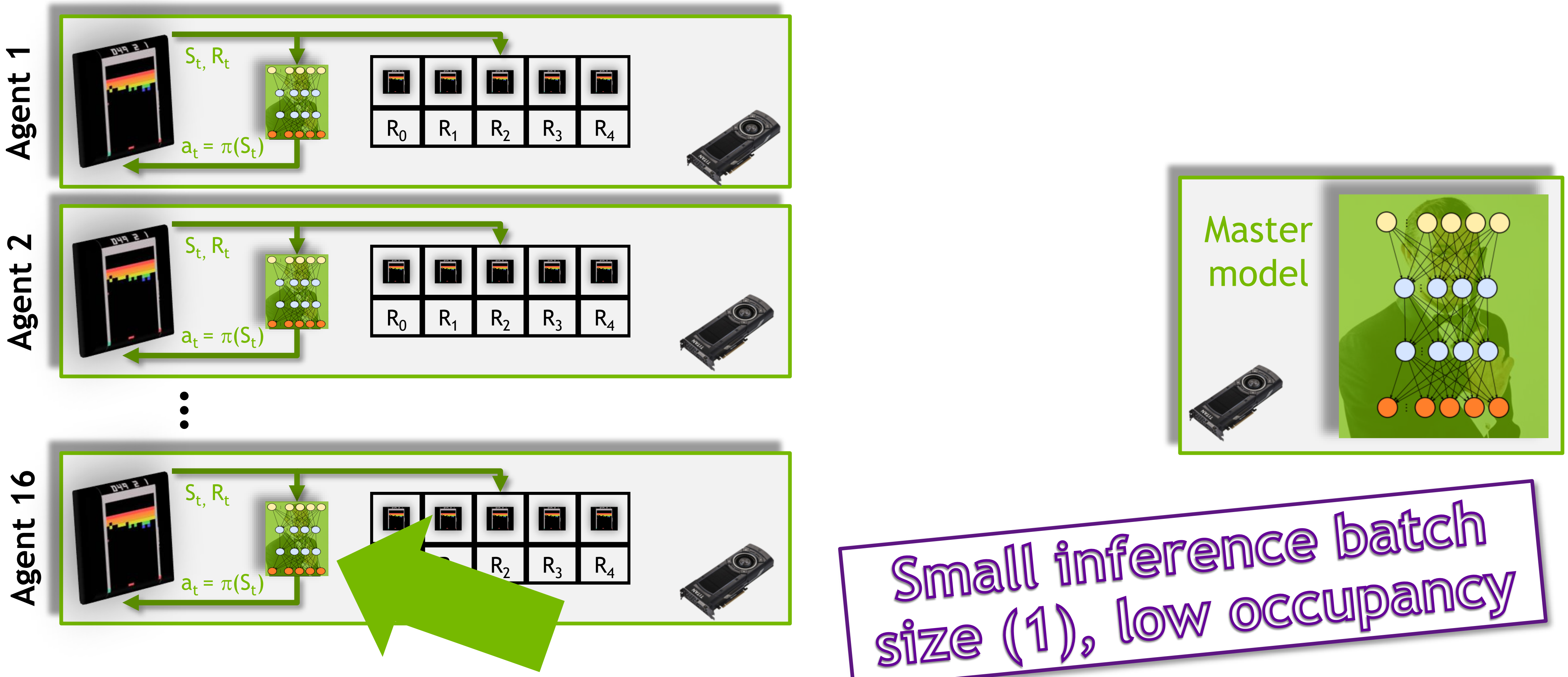
A3C


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

A3C

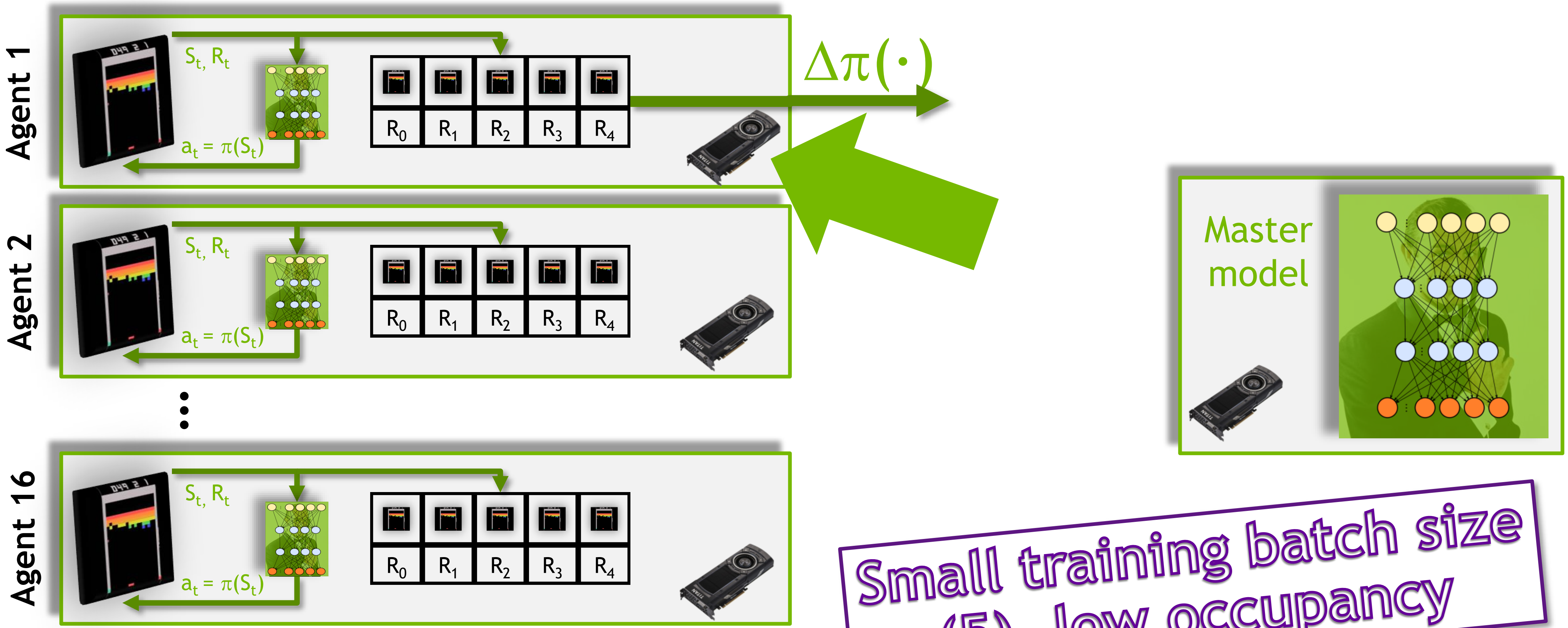

Agent 1
S_t, R_t
$a_t = \pi(S_t)$
R_0 R_1 R_2 R_3 R_4
Agent 2
S_t, R_t
$a_t = \pi(S_t)$
R_0 R_1 R_2 R_3 R_4
Agent 16
S_t, R_t
$a_t = \pi(S_t)$
R_0 R_1 R_2 R_3 R_4
Master model

A3C



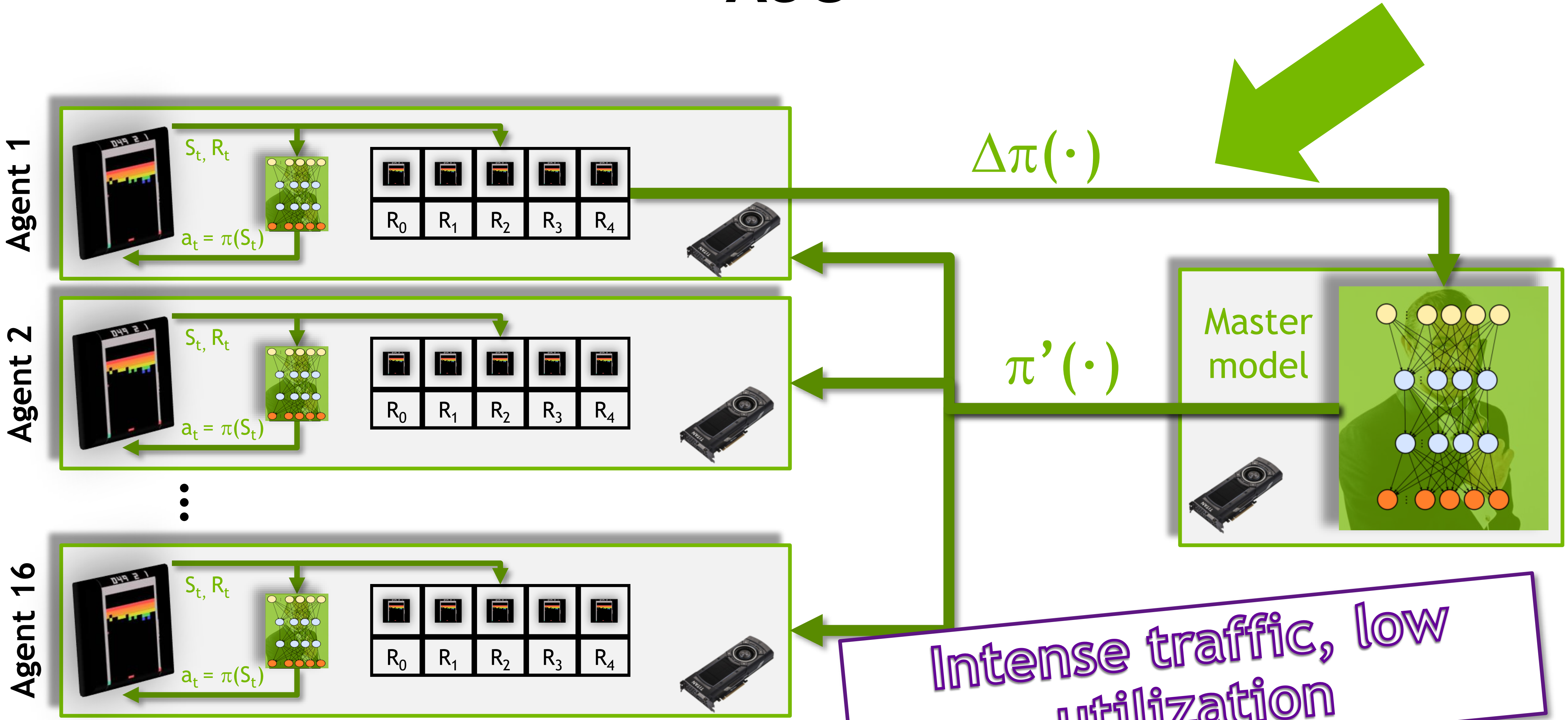
Small inference batch size (1), low occupancy

A3C



Small training batch size (5), low occupancy

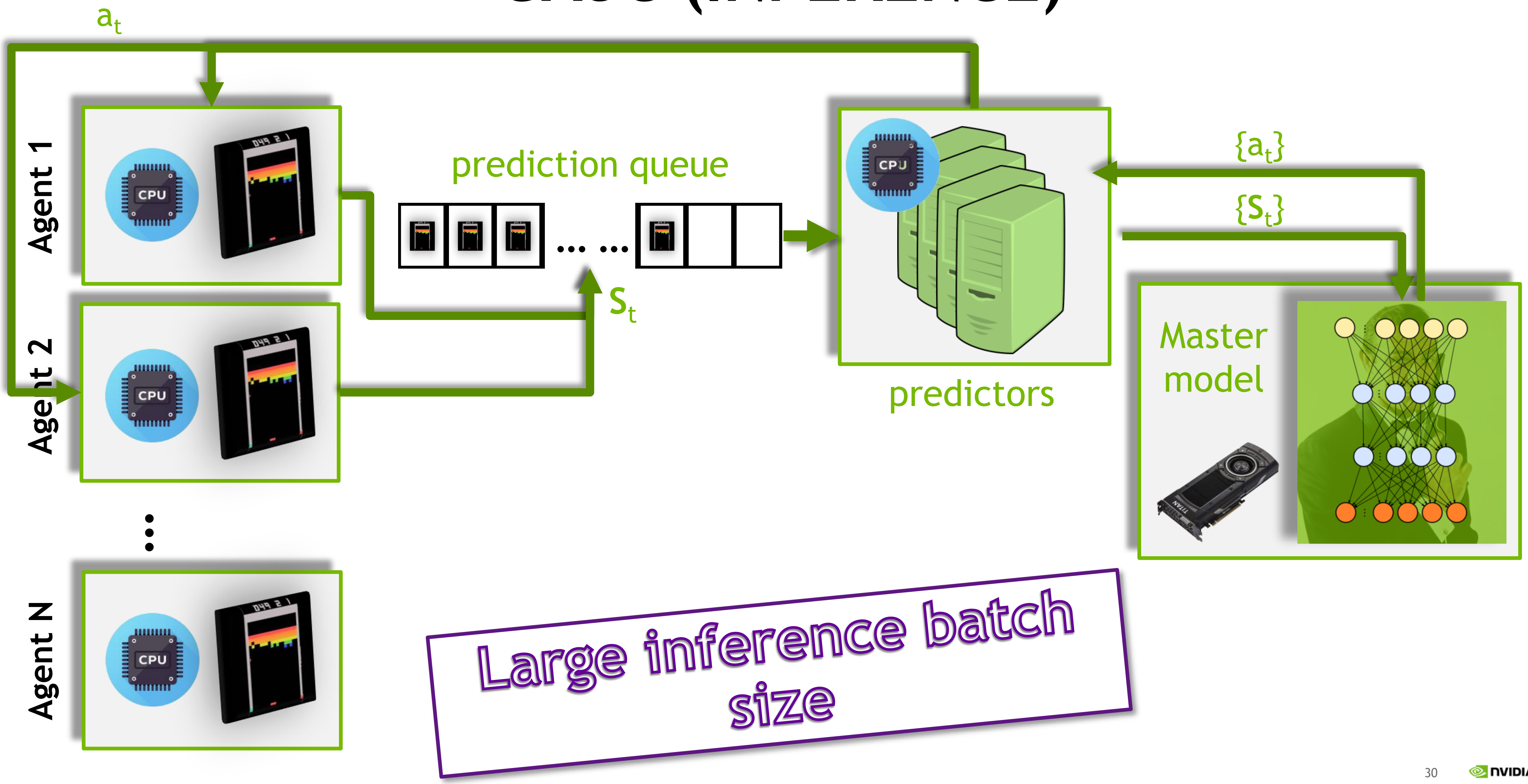
A3C



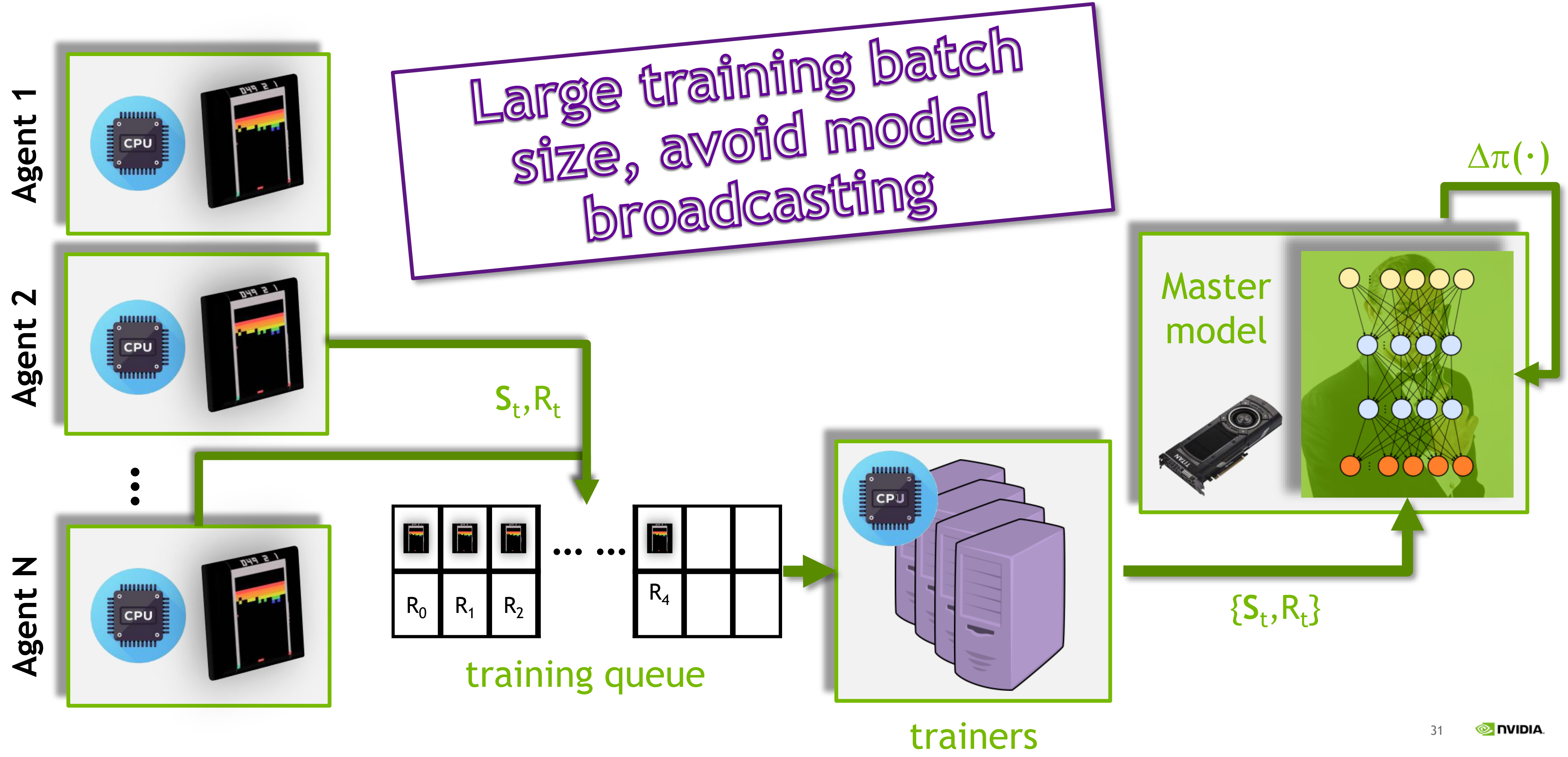
Intense traffic, low utilization

El Capitan big wall, Yosemite Valley

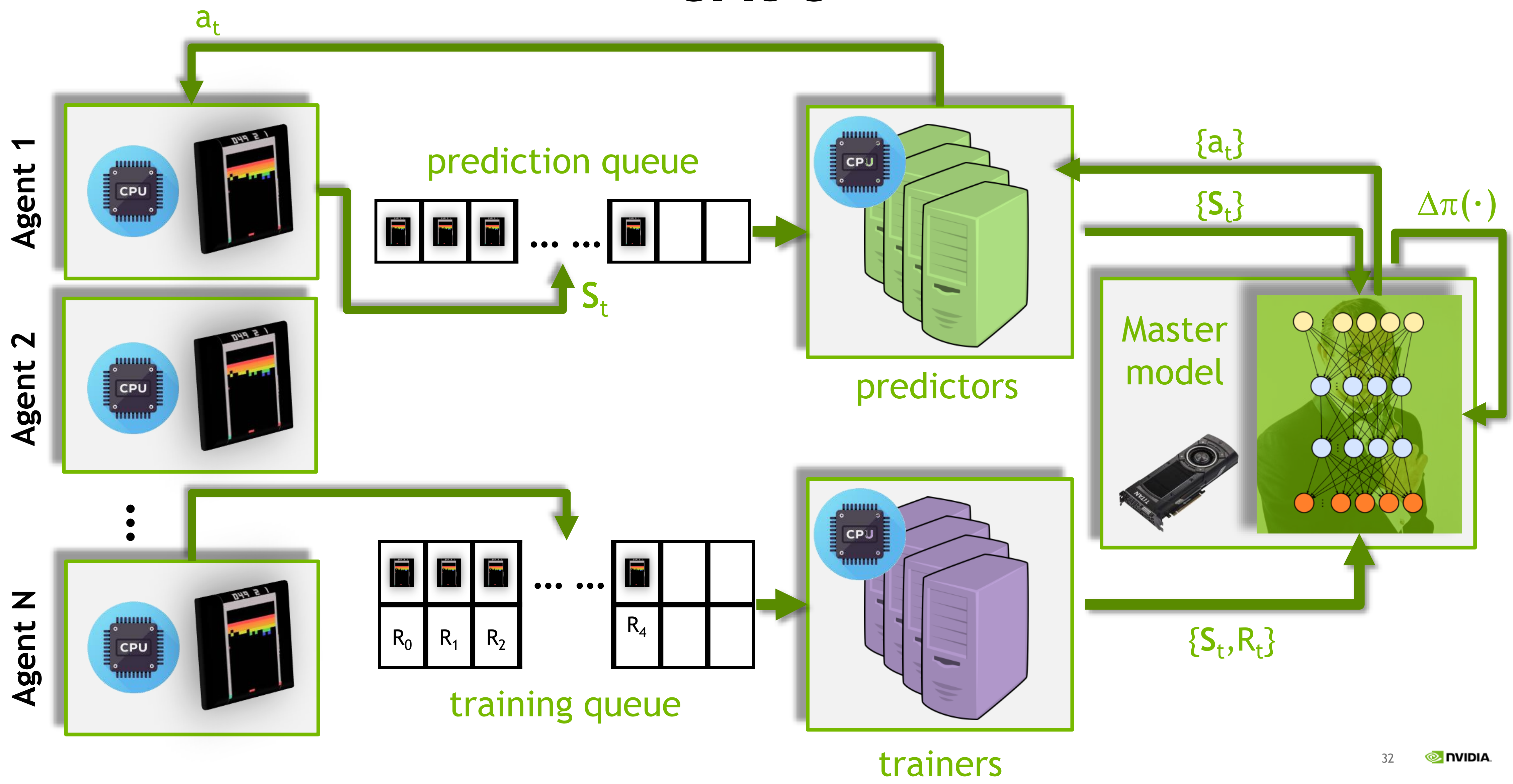
GA3C (INFERENCE)

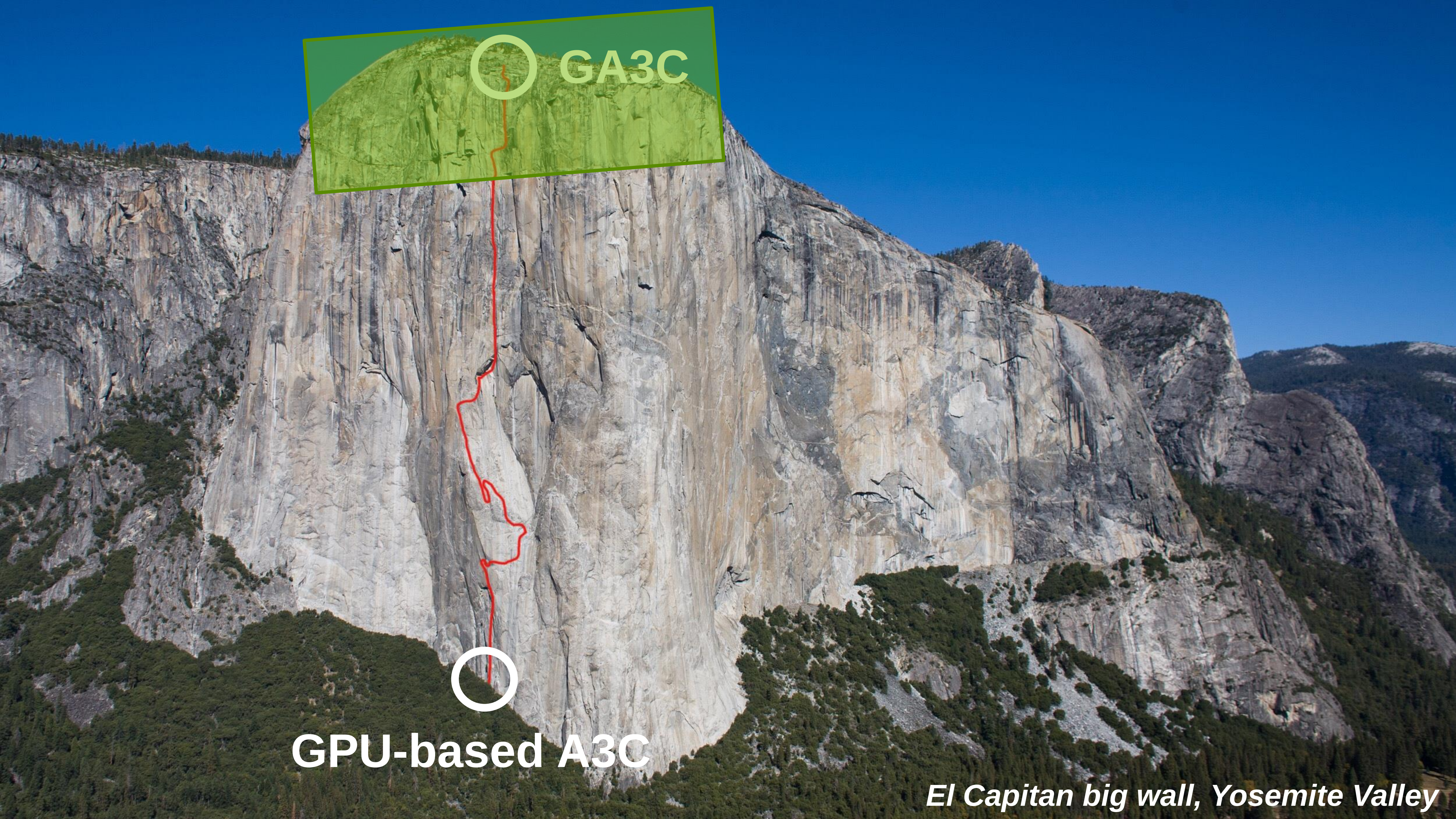


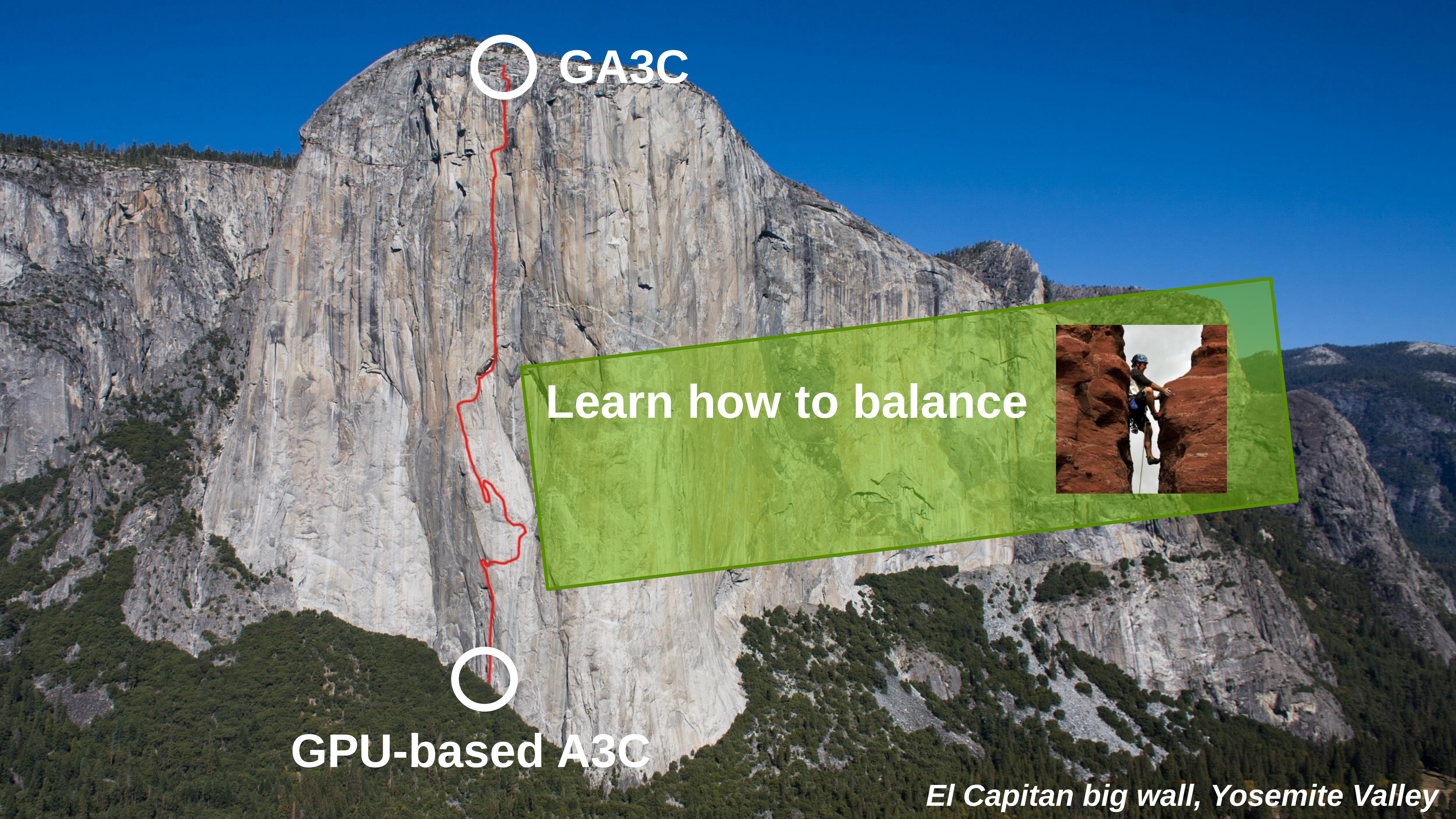
GA3C (TRAINING)



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
$\{S_t, R_t\}$

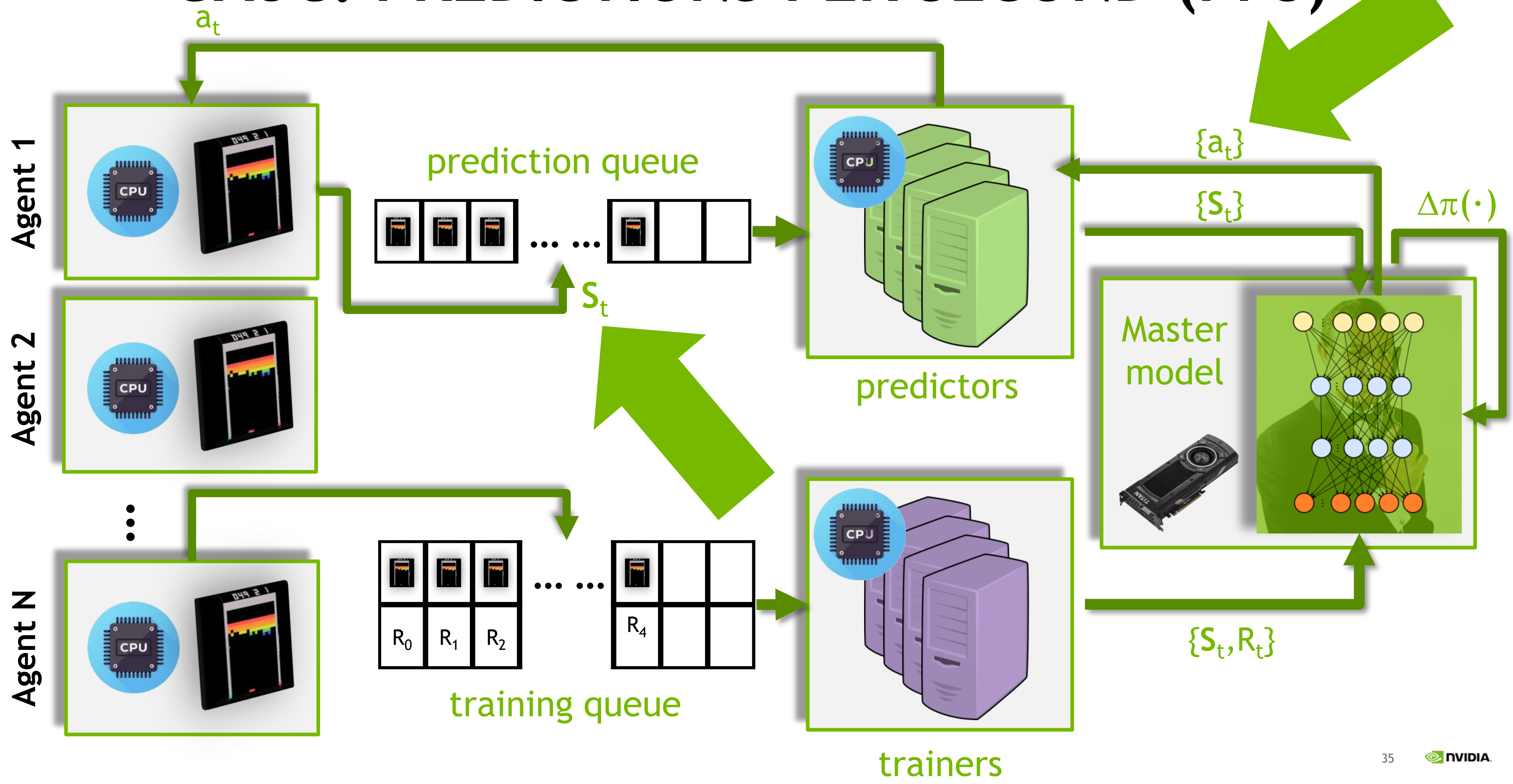

GA3C
GPU-based A3C
El Capitan big wall, Yosemite Valley



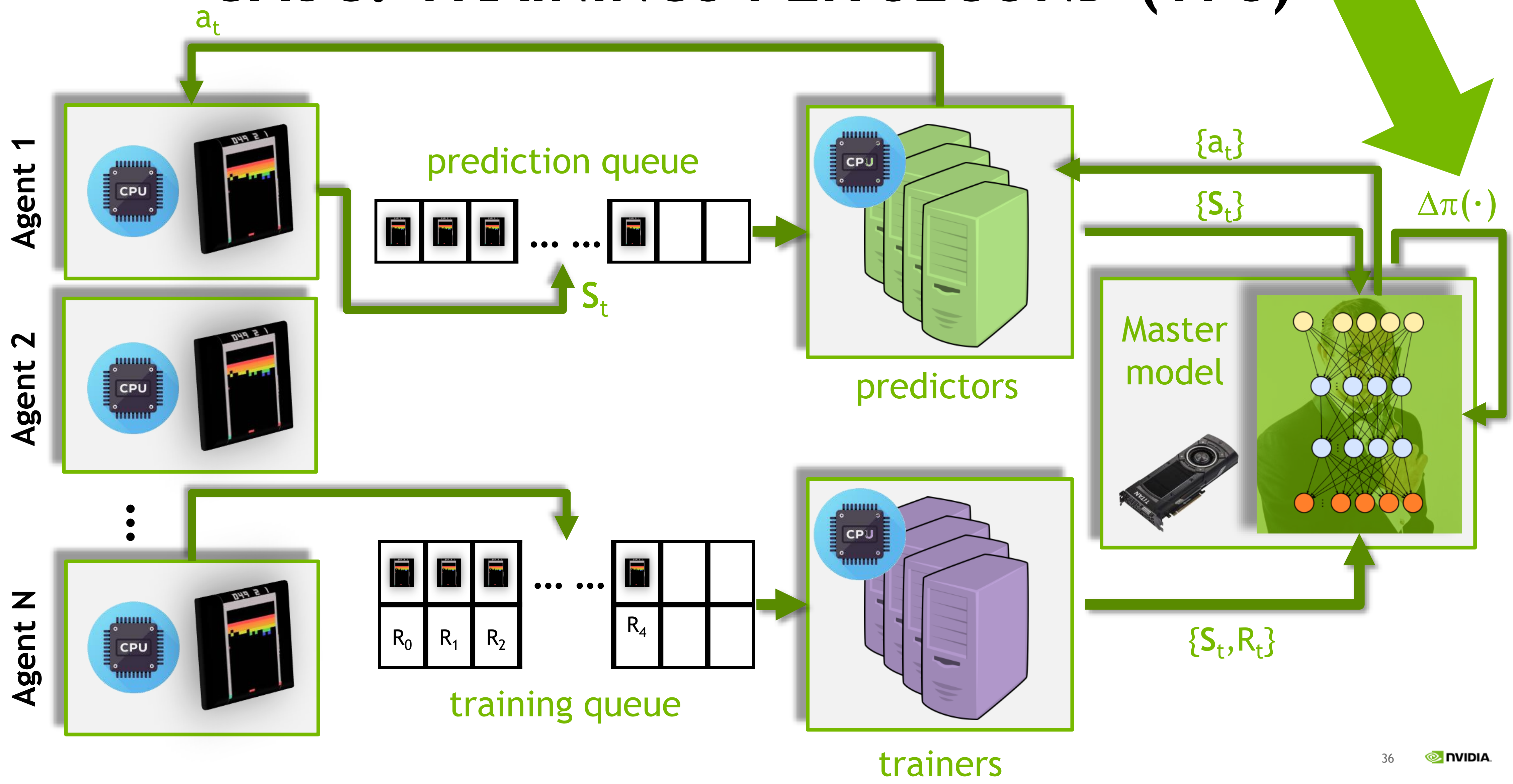
Learn how to balance

El Capitan big wall, Yosemite Valley

GA3C: PREDICTIONS PER SECOND (PPS)



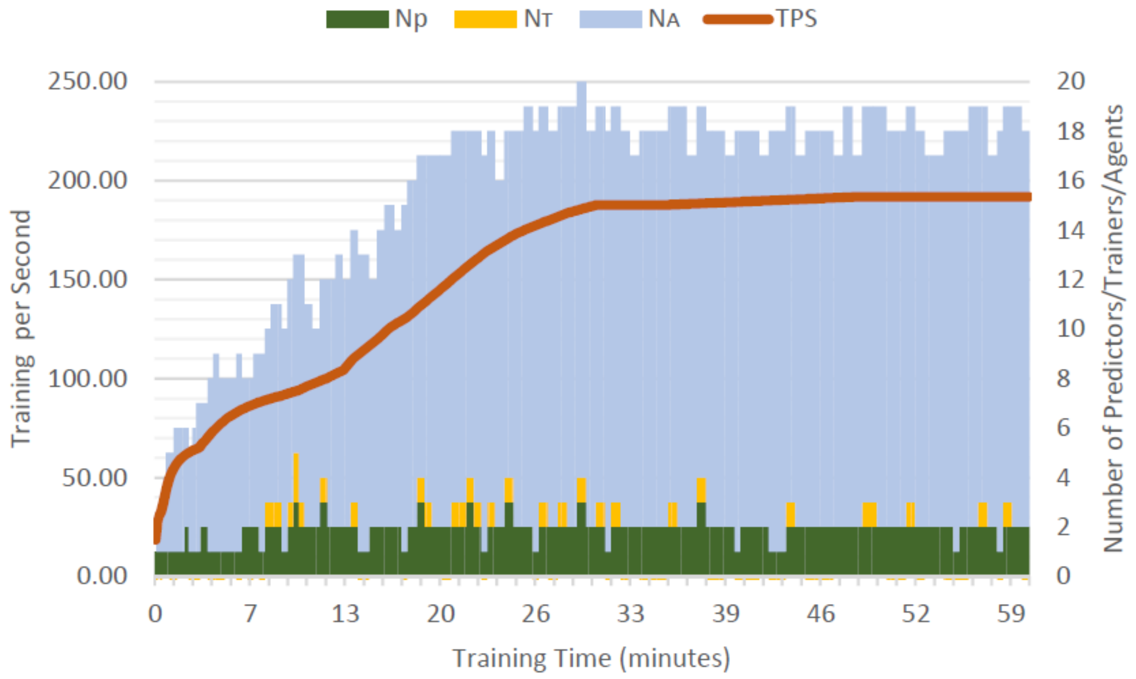
GA3C: TRAININGS PER SECOND (TPS)



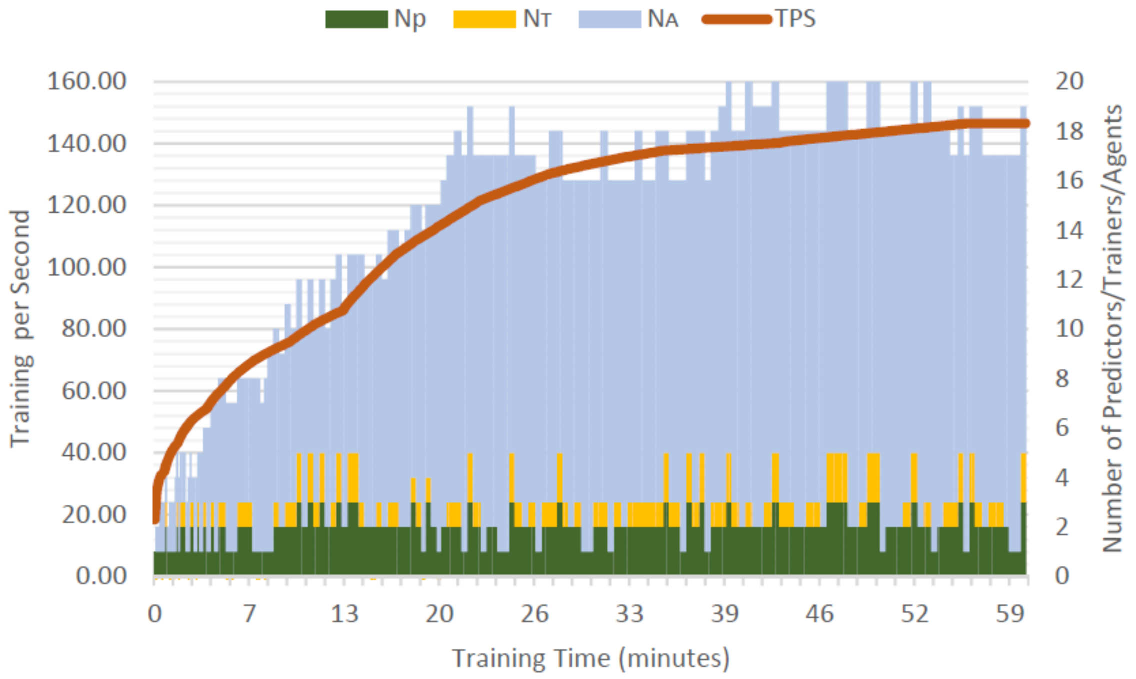
AUTOMATIC SCHEDULING

Balancing the system at run time

ATARI Boxing



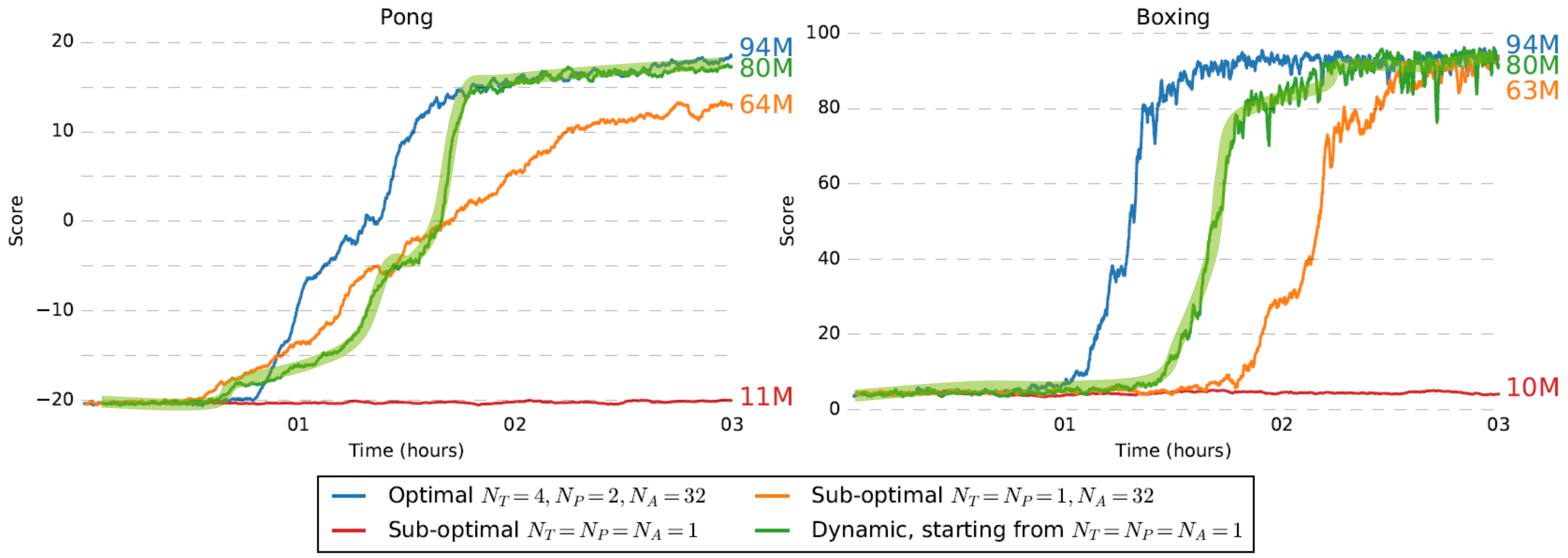
ATARI Pong



N_P = # predictors, N_T = # trainers, N_A = # agents, TPS = training per seconds

THE ADVANTAGE OF SPEED

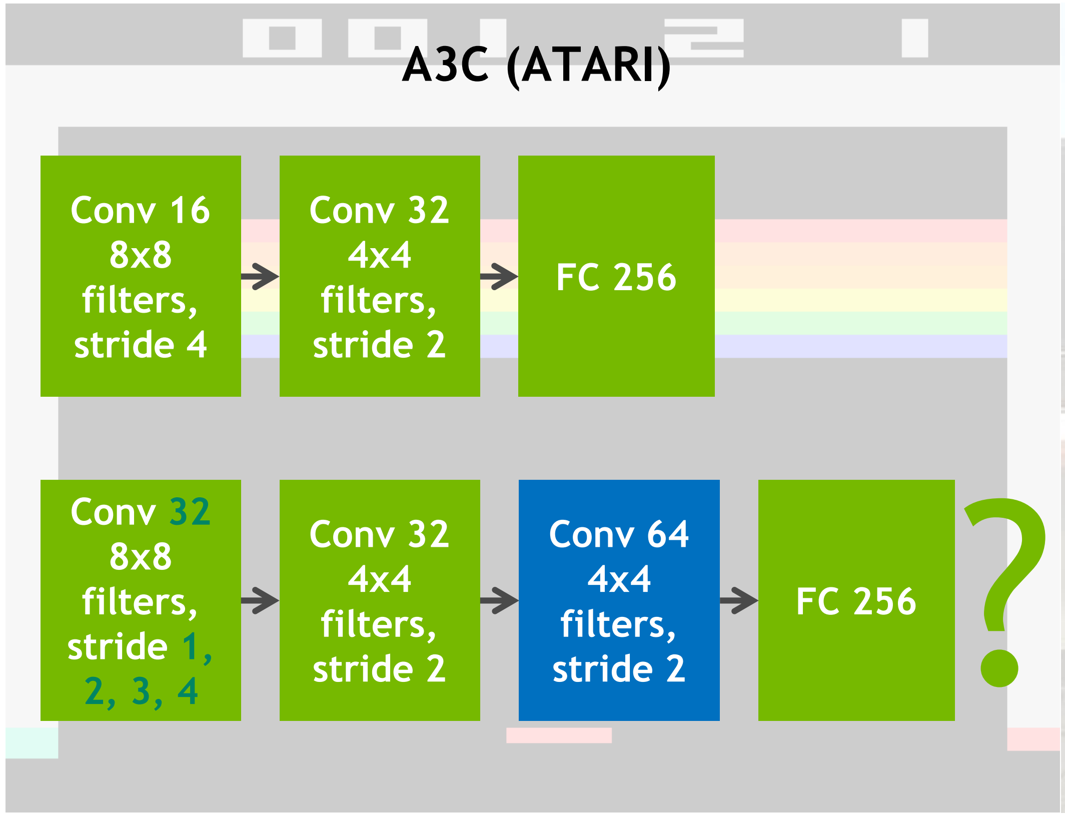
More frames = faster convergence



LARGER DNNS

For real world applications (e.g. robotics, automotive)

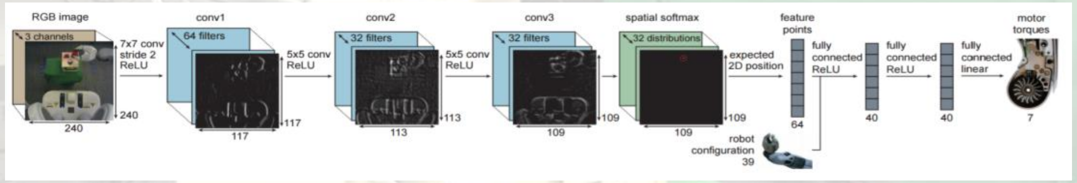
A3C (ATARI)


Conv 16 8x8 filters, stride 4
Conv 32 4x4 filters, stride 2
FC 256
Conv 32 8x8 filters, stride 1, 2, 3, 4
Conv 32 4x4 filters, stride 2
Conv 64 4x4 filters, stride 2
FC 256
?

Others (robotics)

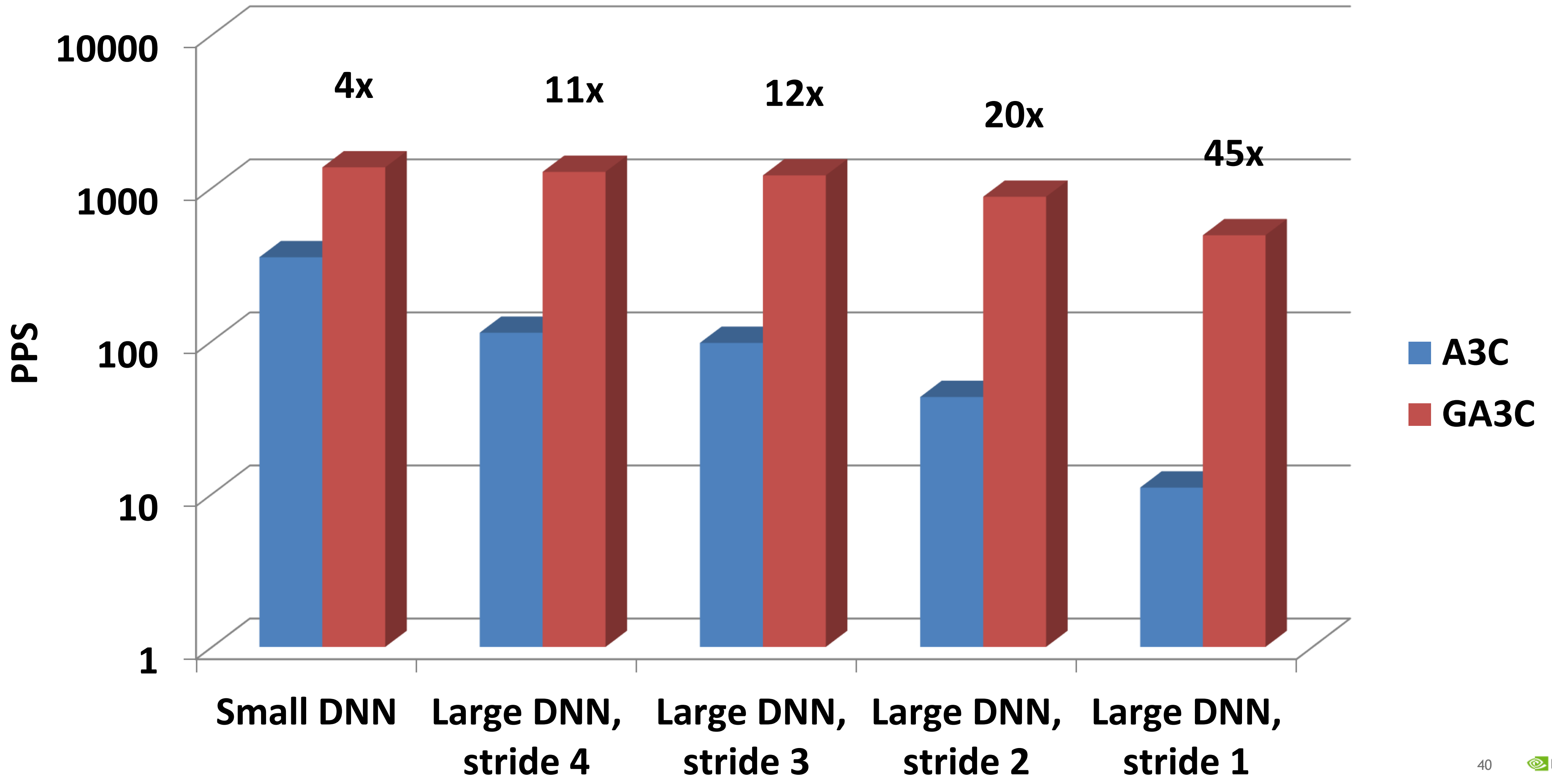
Timothy P. Lillicrap et al., Continuous control with deep reinforcement learning, International Conference on Learning Representations, 2016.

S. Levine et al., End-to-end training of deep visuomotor policies, Journal of Machine Learning Research, 17:1-40, 2016.



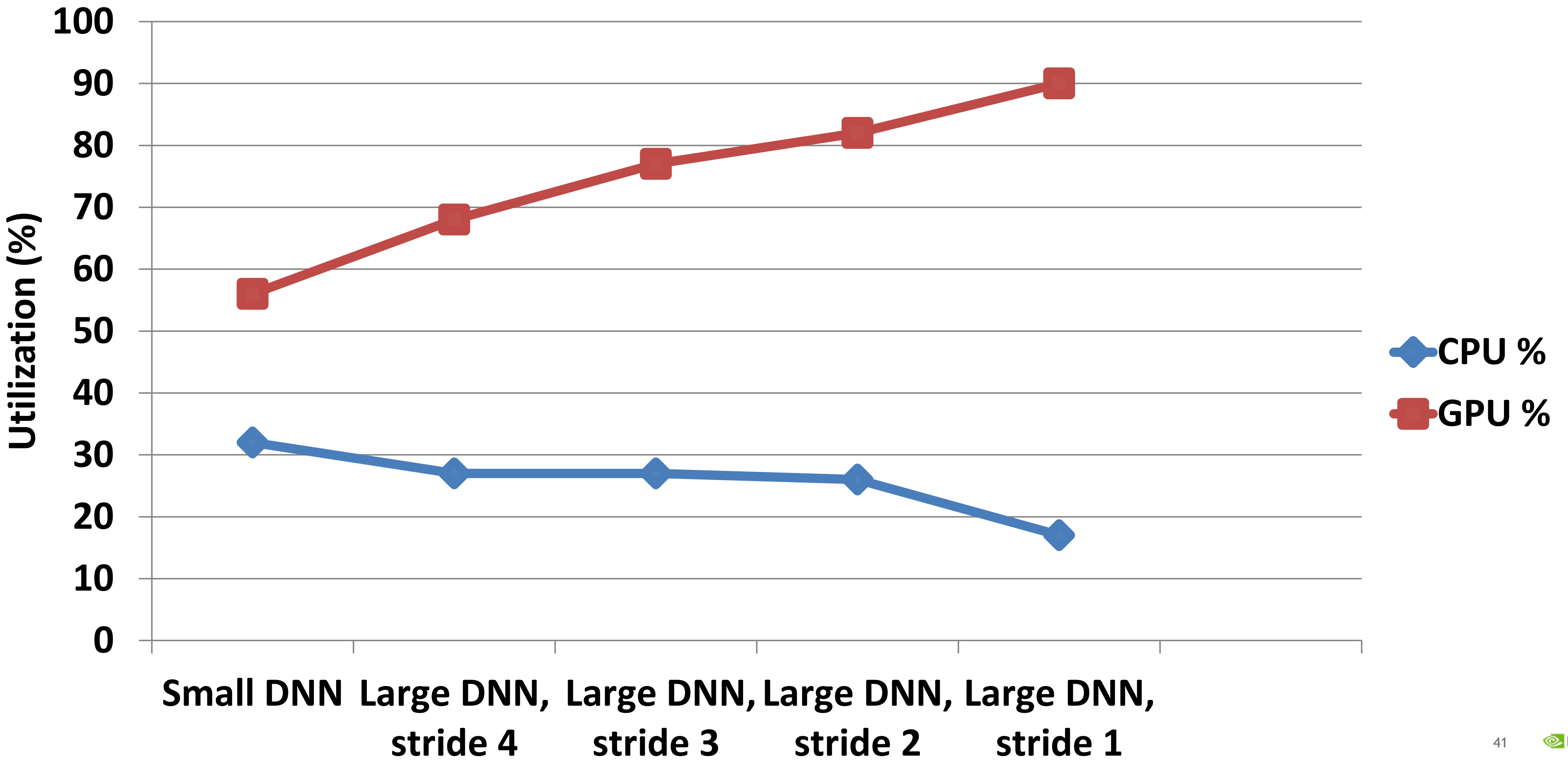
GA3C VS. A3C*: PREDICTIONS PER SECONDS

* Our Tensor Flow implementation on a CPU



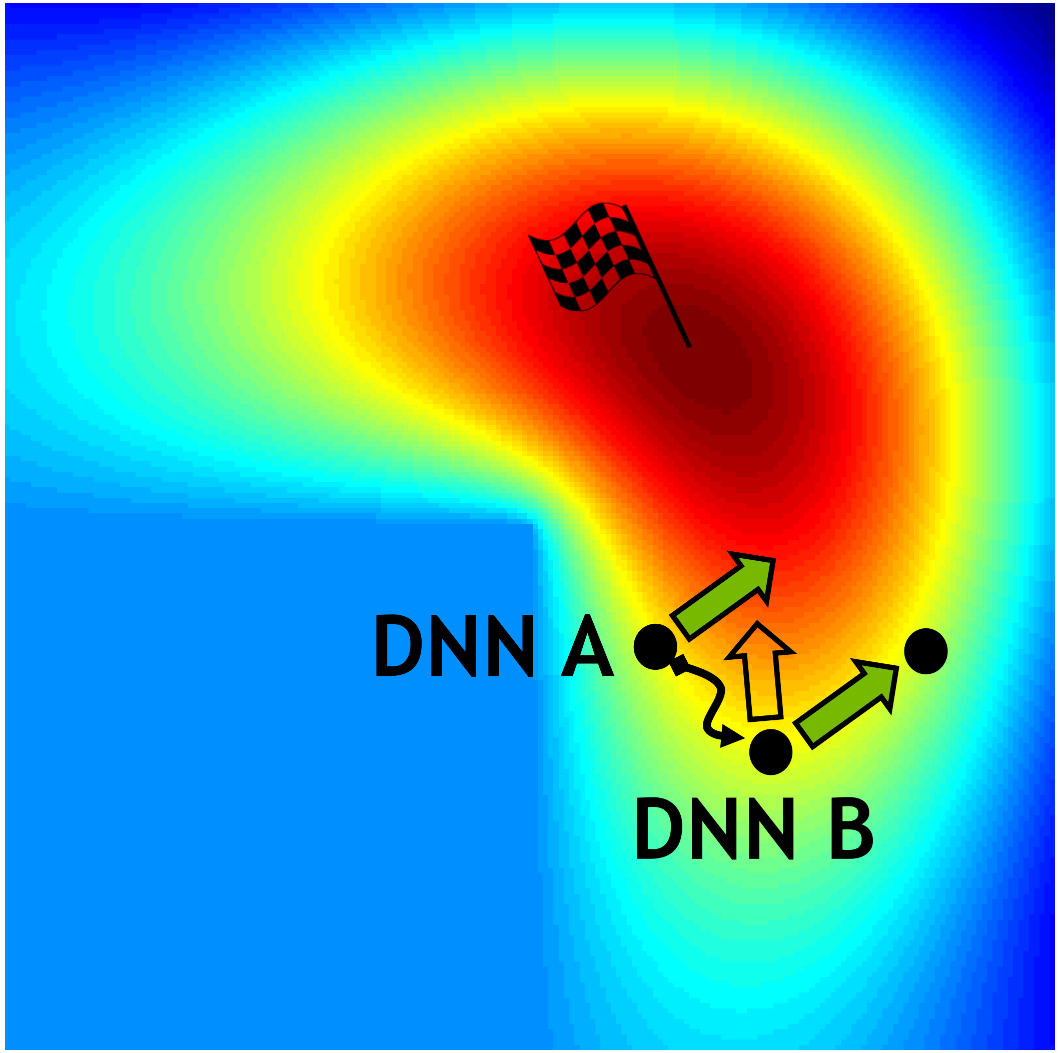
CPU & GPU UTILIZATION IN GA3C

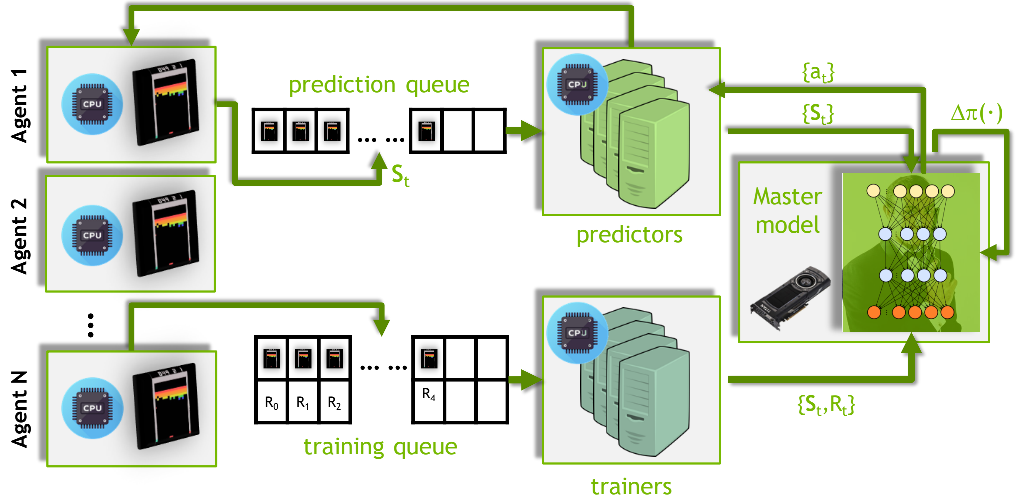
For larger DNNs



GA3C POLICY LAG

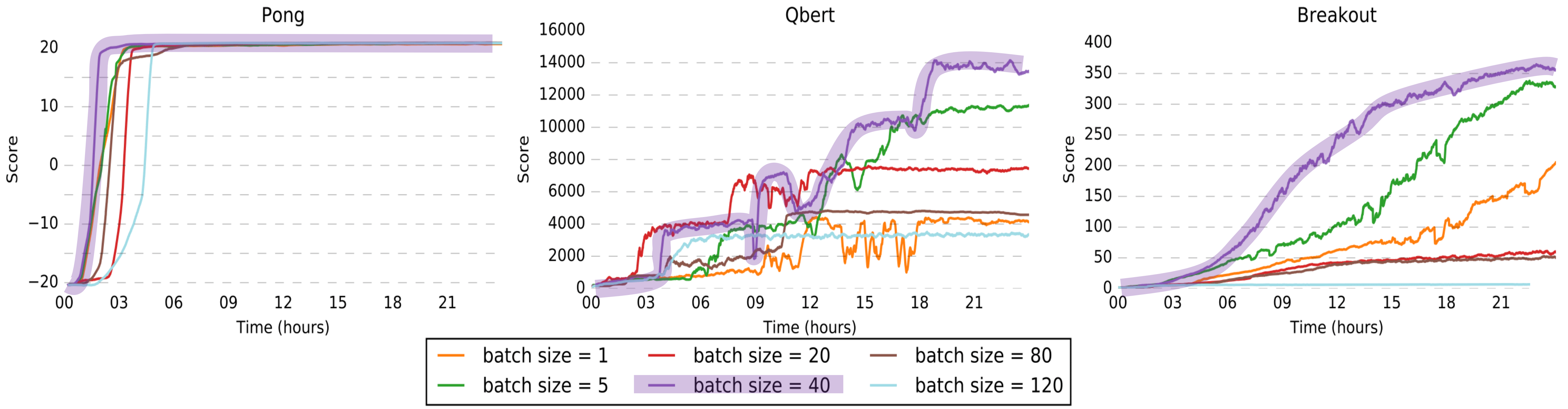
Asynchronous playing and training (off-policy updates)

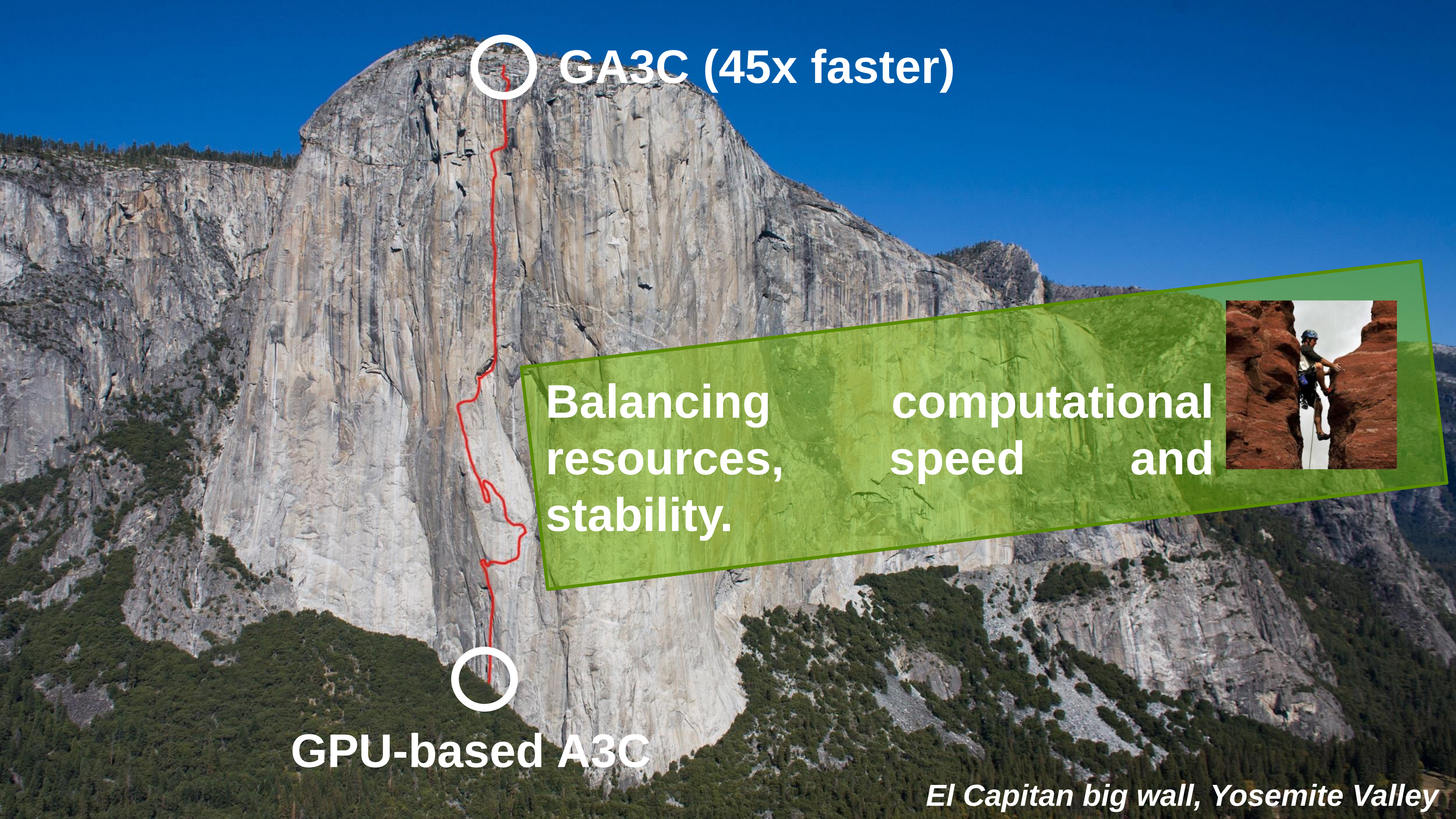




STABILITY AND CONVERGENCE SPEED

Reducing policy lag through min training batch size





Balancing computational resources, speed and stability.

El Capitan big wall, Yosemite Valley

RESOURCES

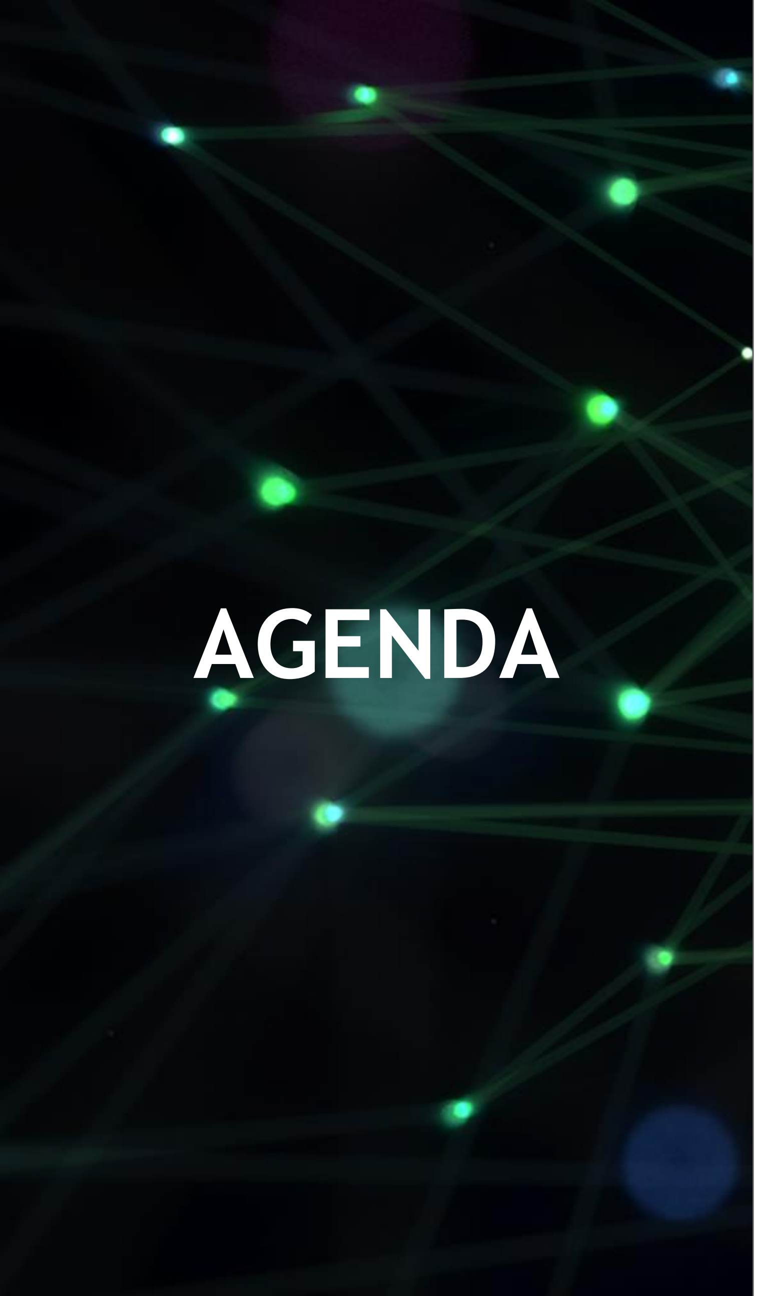
THEORY

M. Babaeizadeh, I. Frosio, S. Tyree, J. Clemons, J. Kautz, **Reinforcement Learning through Asynchronous Advantage Actor-Critic on a GPU**, ICLR 2017 (available at https://openreview.net/forum?id=r1VGvBcxl&noteId=r1VGvBcxl).

CODING

GA3C, a GPU implementation of A3C (open source at https://github.com/NVlabs/GA3C).

A general architecture to generate and consume training data.



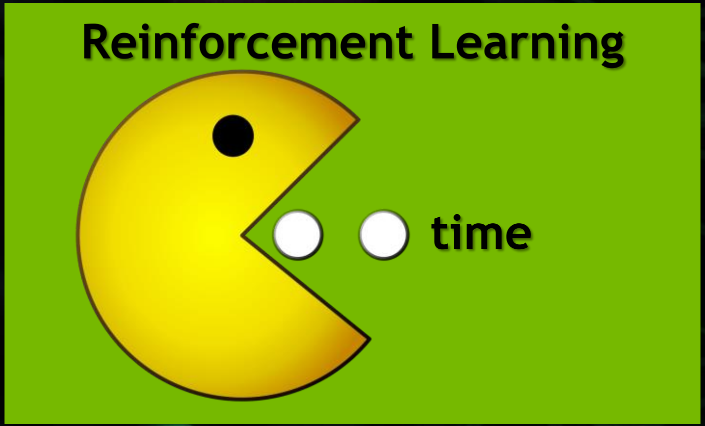
AGENDA

1. GPU-based A3C for Deep Reinforcement Learning
(The basics of Reinforcement Learning on a CPU/GPU)

2. Cule*: GPU accelerated RL
(Moving Reinforcement Learning on a GPU)

3. Sim-to-Real Transfer of Accurate Grasping with Eye-In-Hand Observations and Continuous Control
(Imitation Learning and Sample Efficiency)

4. Conclusion



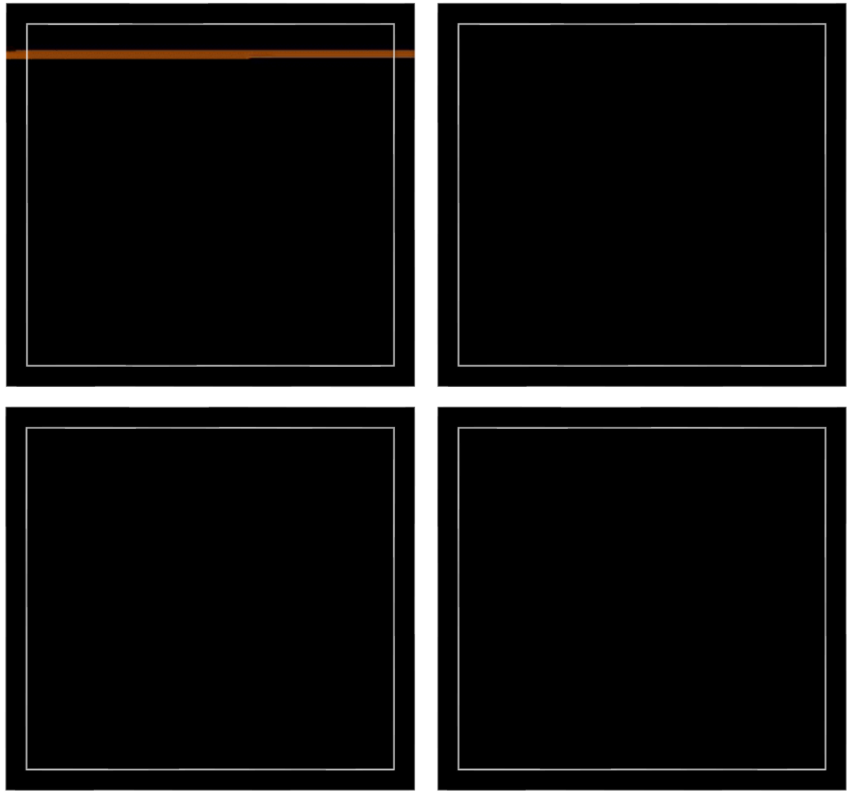
CULE*: GPU ACCELERATED RL

* CUDA Learning Environment

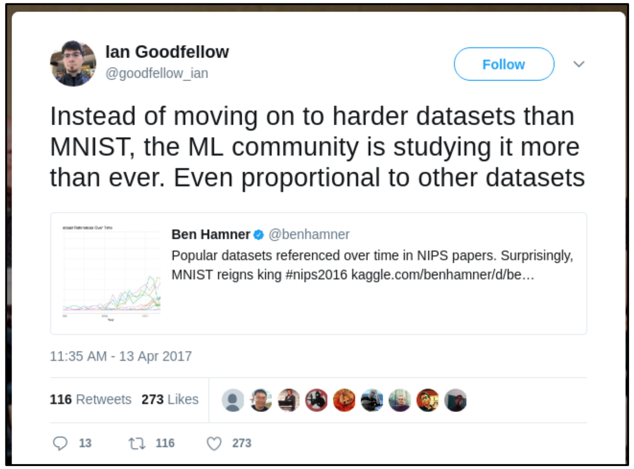
Steven Dalton, Iuri Frosio, Jared Hoberock, Jason Clemons

REINFORCEMENT LEARNING

ALE (Atari Learning Environment)



- Diverse set of tasks
- Established benchmark - MNIST of RL?



Ian Goodfellow
@goodfellow_ian

Follow

Instead of moving on to harder datasets than MNIST, the ML community is studying it more than ever. Even proportional to other datasets

Ben Hamner @benhamner
Popular datasets referenced over time in NIPS papers. Surprisingly, MNIST reigns king #nips2016 kaggle.com/benhamner/d/be…

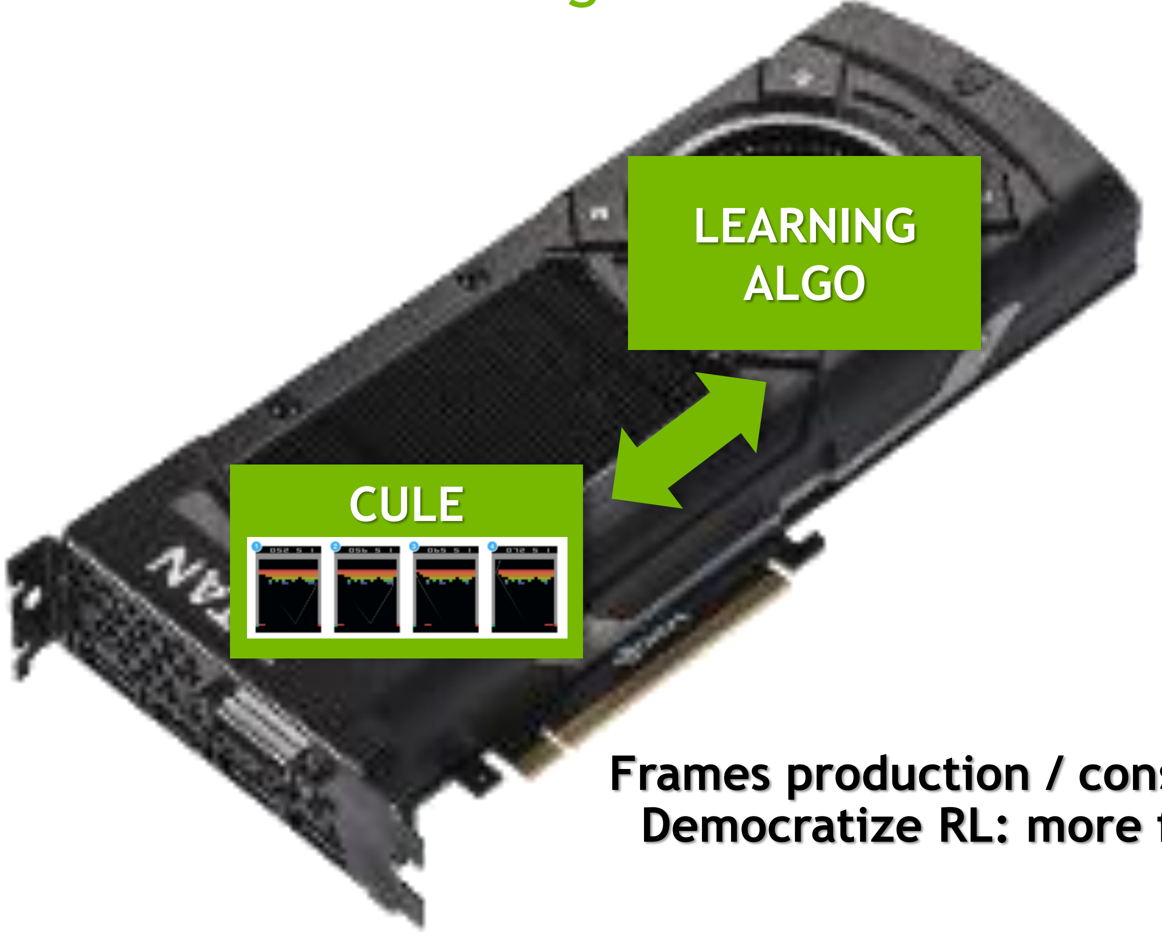
11:35 AM - 13 Apr 2017

116 Retweets 273 Likes

13 116 273

CULE

CUDA Learning Environment



Frames production / consumption rate > 10K / s
Democratize RL: more frames for less money

AGENDA

RL training: CPU, GPU

Limitations

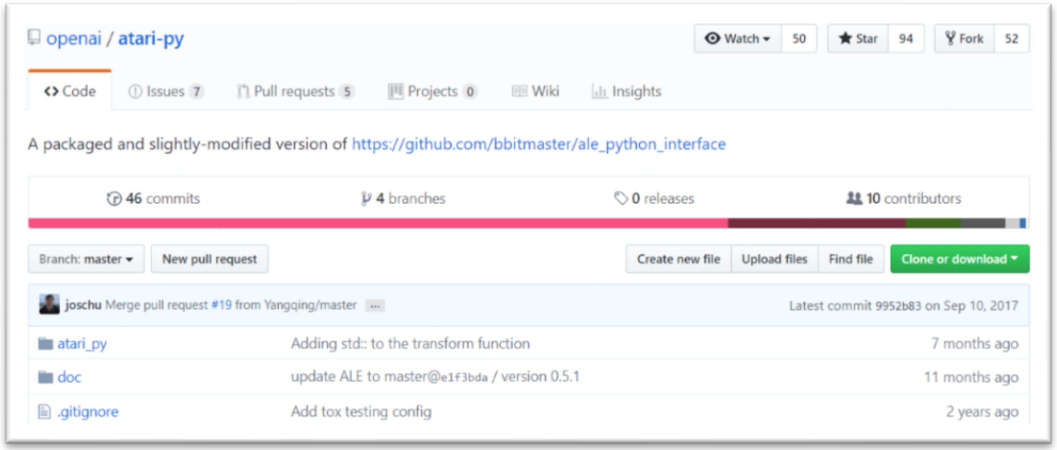
CuLE

Performance

Analysis and new scenarios

RL TRAINING

The OpenAI ATARI interface



openai / atari-py

Watch 50 Star 94 Fork 52

Code Issues 7 Pull requests 5 Projects 0 Wiki Insights

A packaged and slightly-modified version of https://github.com/bbitmaster/ale_python_interface

46 commits 4 branches 0 releases 10 contributors

Branch: master New pull request Create new file Upload files Find file Clone or download

joschu Merge pull request #19 from Yangqing/master ... Latest commit 9952b83 on Sep 10, 2017

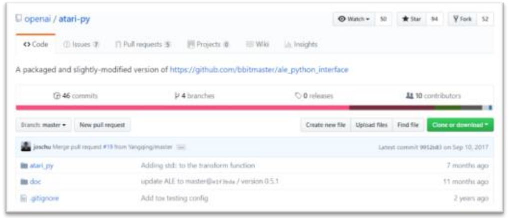
atari_py	Adding std:: to the transform function	7 months ago
doc	update ALE to master@e1f3bda / version 0.5.1	11 months ago
.gitignore	Add tox testing config	2 years ago

https://github.com/openai/atari-py (OpenAI gym)

CPU based

RL TRAINING

CPU only based training



A packaged and slightly-modified version of https://github.com/bbitmaster/ale_python_interface

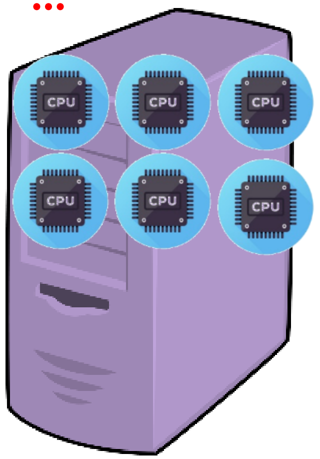
46 commits · 4 branches · 0 releases · 10 contributors

joschu Merge pull request #19 from Yangqing/master — Latest commit 9952b83 on Sep 10, 2017

atari_py	Adding std:: to the transform function	7 months ago
doc	update ALE to master@[illegible] / version 0.5.1	11 months ago
.gitignore	Add tox testing config	2 years ago

https://github.com/openai/atari-py (OpenAI gym)

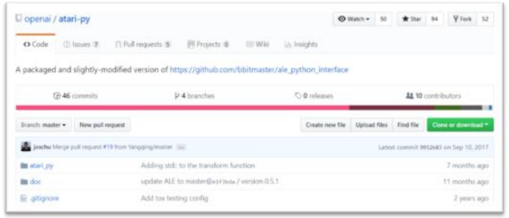
DQN
A3C
…



Mnih V. et al., Human-level control through deep reinforcement Learning, Nature, 2015
Minh V. et al., Asynchronous Methods for Deep Reinforcement Learning, ICML 2016

RL TRAINING

Hybrid CPU GPU training



openai / atari-py

Watch 50 · Star 94 · Fork 52

<> Code · Issues 7 · Pull requests 5 · Projects 0 · Wiki · Insights

A packaged and slightly-modified version of https://github.com/bbitmaster/ale_python_interface

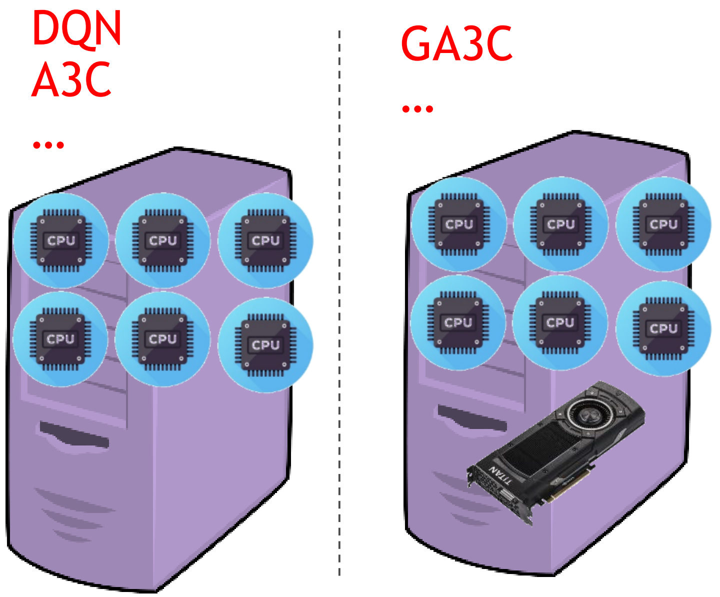
46 commits · 4 branches · 0 releases · 10 contributors

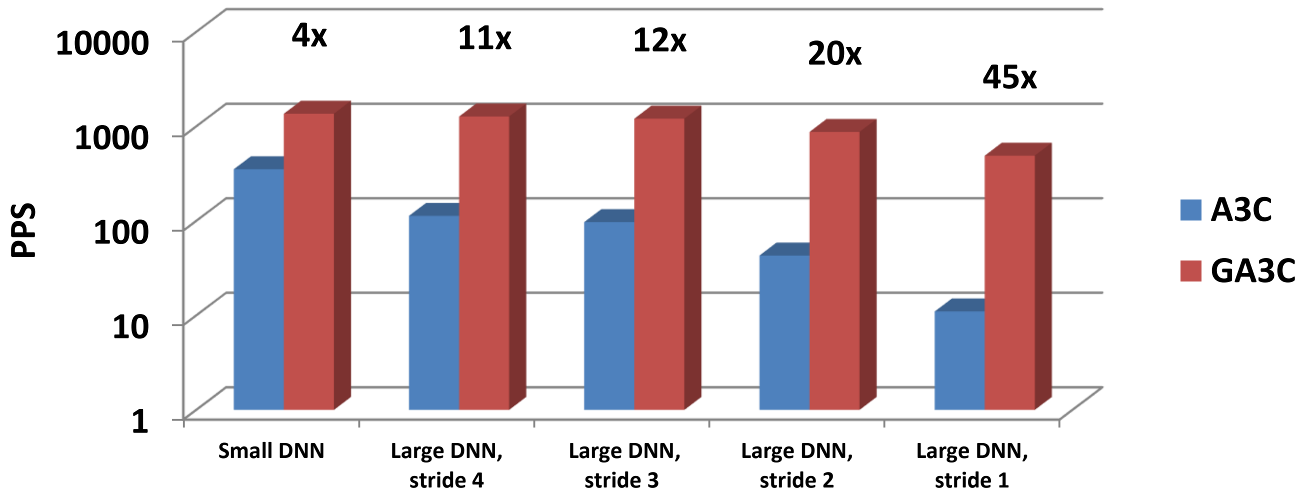
Branch: master · New pull request · Create new file · Upload files · Find file · Clone or download

joschu Merge pull request #19 from Yangqing/master … Latest commit 9952b83 on Sep 10, 2017

atari_py	Adding std:: to the transform function	7 months ago
doc	update ALE to master@w1r3eda / version 0.5.1	11 months ago
.gitignore	Add tox testing config	2 years ago

https://github.com/openai/atari-py (OpenAI gym)

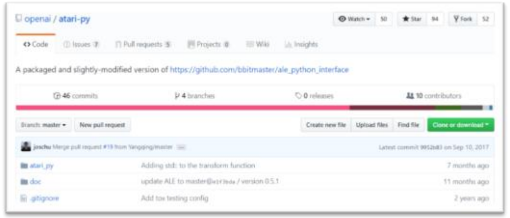




Babaeizadeh M. et al., Reinforcement Learning through Asynchronous Advantage Actor-Critic on a GPU, ICLR, 2017

RL TRAINING

Clusters


openai / atari-py

Watch 50 | Star 94 | Fork 52

Code | Issues 7 | Pull requests 5 | Projects 0 | Wiki | Insights

A packaged and slightly-modified version of https://github.com/bbitmaster/ale_python_interface

46 commits | 4 branches | 0 releases | 10 contributors

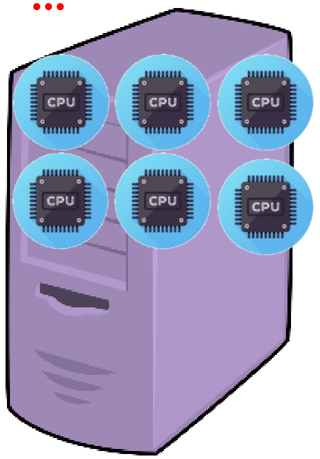
Branch: master | New pull request | Create new file | Upload files | Find file | Clone or download

joschu Merge pull request #19 from Yangqing/master … Latest commit 9952b83 on Sep 10, 2017

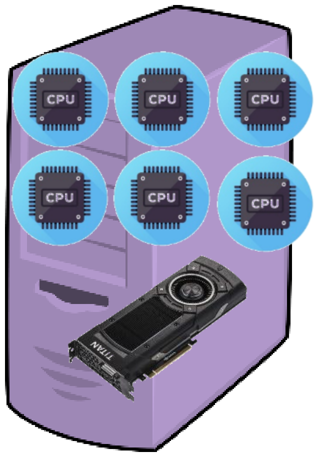
atari_py	Adding std:: to the transform function	7 months ago
doc	update ALE to master@[illegible] / version 0.5.1	11 months ago
.gitignore	Add tox testing config	2 years ago

https://github.com/openai/atari-py (OpenAI gym)

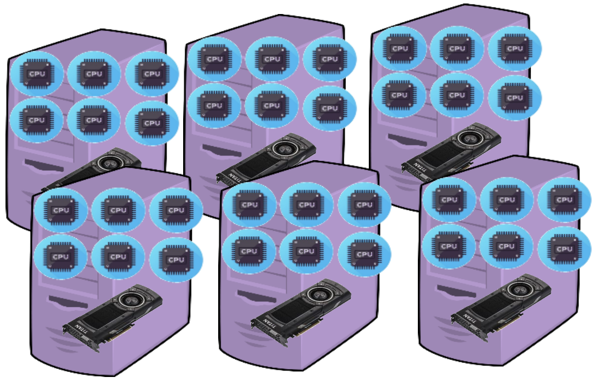
DQN
A3C
…



GA3C
…



Cluster

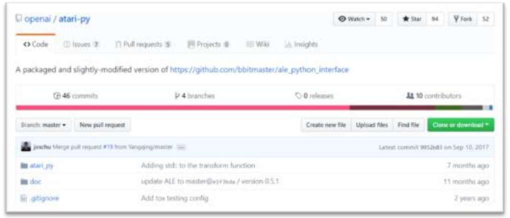


ES (GA)
A3C
A2C
IMPALA
…

Espeholt L. et al., IMPALA: Scalable Distributed Deep-RL with Importance Weighted Actor-Learner Architectures, 2018

RL TRAINING

DGX-1



openai / atari-py

Watch 50 | Star 94 | Fork 52

Code | Issues 7 | Pull requests 5 | Projects 0 | Wiki | Insights

A packaged and slightly-modified version of https://github.com/bbitmaster/ale_python_interface

46 commits | 4 branches | 0 releases | 10 contributors

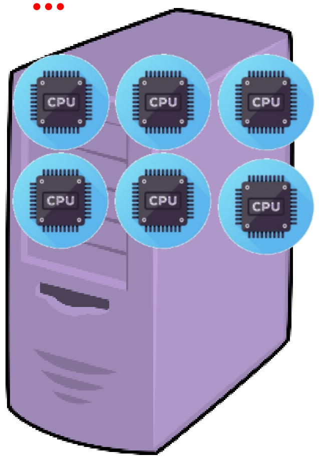
Branch: master | New pull request | Create new file | Upload files | Find file | Clone or download

joschu Merge pull request #19 from Yangqing/master — Latest commit 9952b83 on Sep 10, 2017

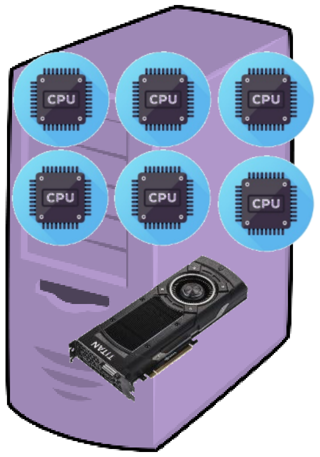
atari_py	Adding std:: to the transform function	7 months ago
doc	update ALE to master@ / version 0.5.1	11 months ago
.gitignore	Add tox testing config	2 years ago

https://github.com/openai/atari-py (OpenAI gym)

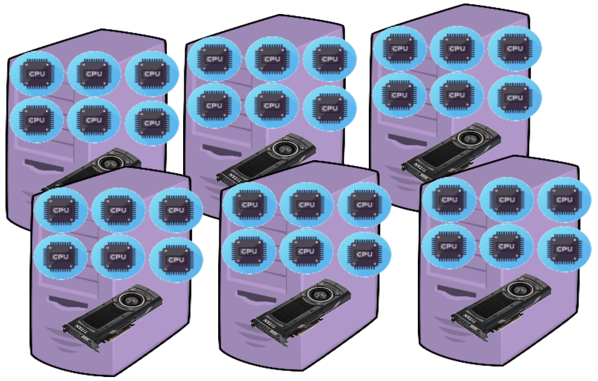
DQN
A3C
…



GA3C
…



Cluster



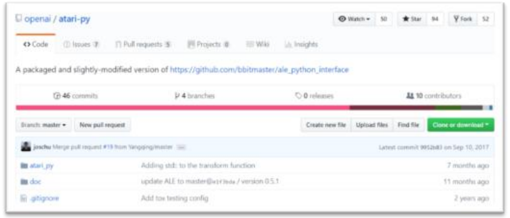
ES (GA)
A3C
A2C
IMPALA
…

CPU
CPU
TITAN

DGX-1

Policy gradient
Q-value
…

RL TRAINING
Limitations


openai / atari-py

A packaged and slightly-modified version of https://github.com/bbitmaster/ale_python_interface

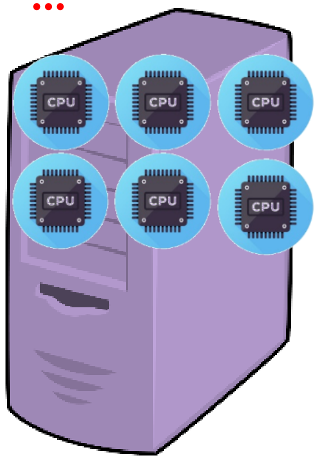
46 commits | 4 branches | 0 releases | 10 contributors

joschu Merge pull request #19 from Yangqing/master — Latest commit 9952b83 on Sep 10, 2017

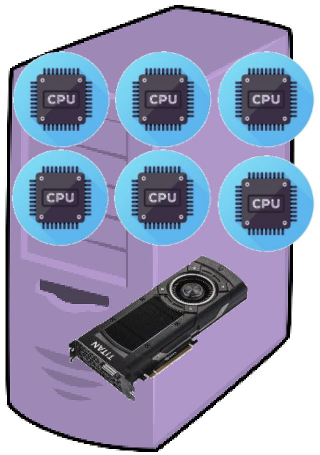
atari_py	Adding std:: to the transform function	7 months ago
doc	update ALE to master@a141feda / version 0.5.1	11 months ago
.gitignore	Add tox testing config	2 years ago

https://github.com/openai/atari-py (OpenAI gym)

DQN
A3C
...



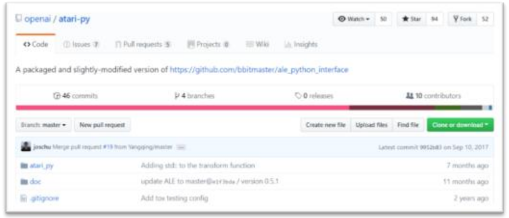
GA3C
...



TIME

RL TRAINING

Limitations


openai / atari-py

A packaged and slightly-modified version of https://github.com/bbitmaster/ale_python_interface

46 commits | 4 branches | 0 releases | 10 contributors

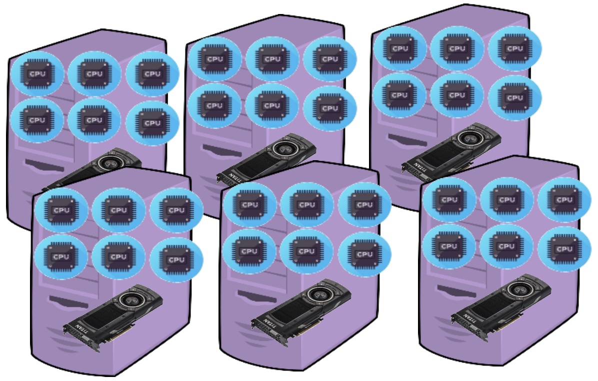
joschu Merge pull request #19 from Yangqing/master — Latest commit 9952b83 on Sep 10, 2017

atari_py	Adding std:: to the transform function	7 months ago
doc	update ALE to master@e31f3eda / version 0.5.1	11 months ago
.gitignore	Add tox testing config	2 years ago

https://github.com/openai/atari-py (OpenAI gym)

$$$

Cluster



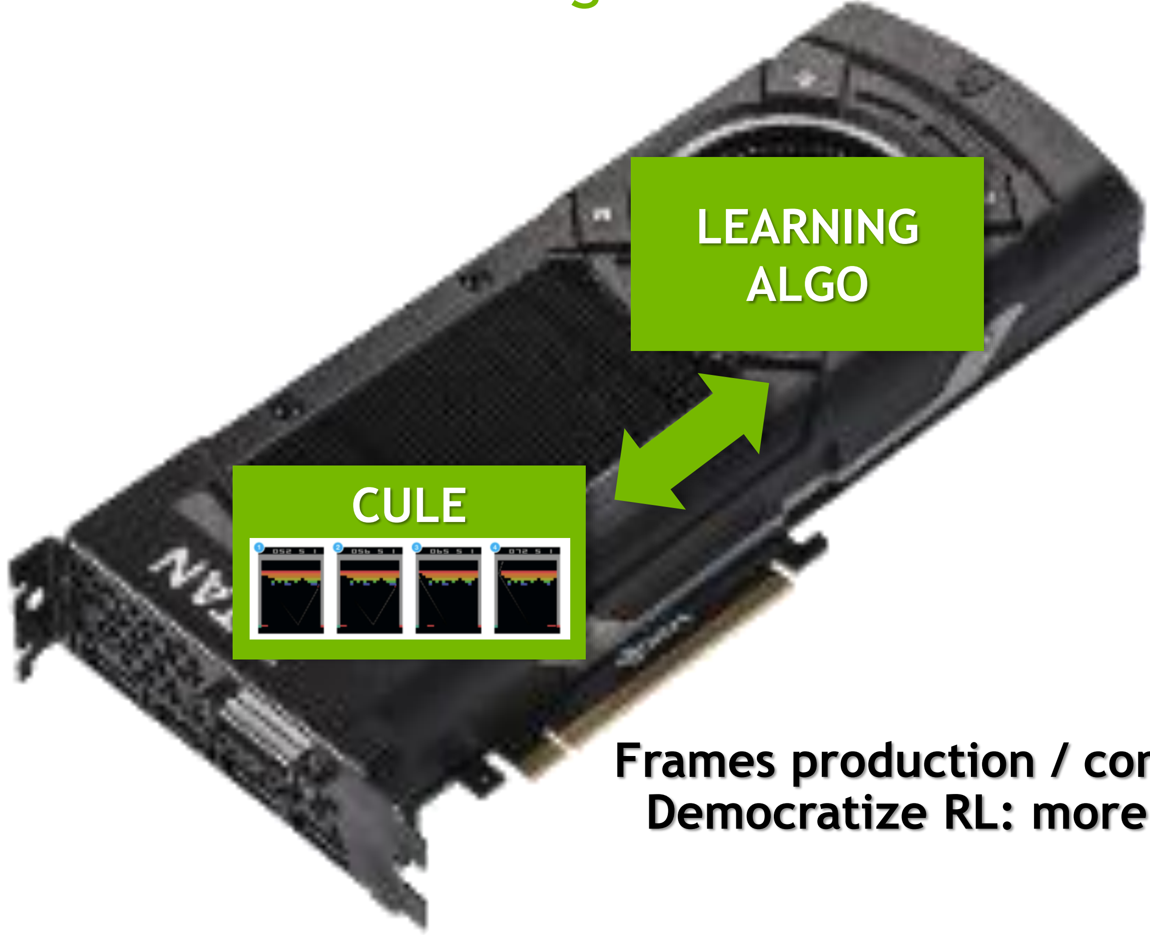
ES (GA)
A3C
A2C
IMPALA
…

DGX-1

Policy gradient
Q-value
…

CULE

CUDA Learning Environment



Frames production / consumption rate > 10K / s
Democratize RL: more frames for less money

AGENDA

RL training: CPU, GPU

Limitations

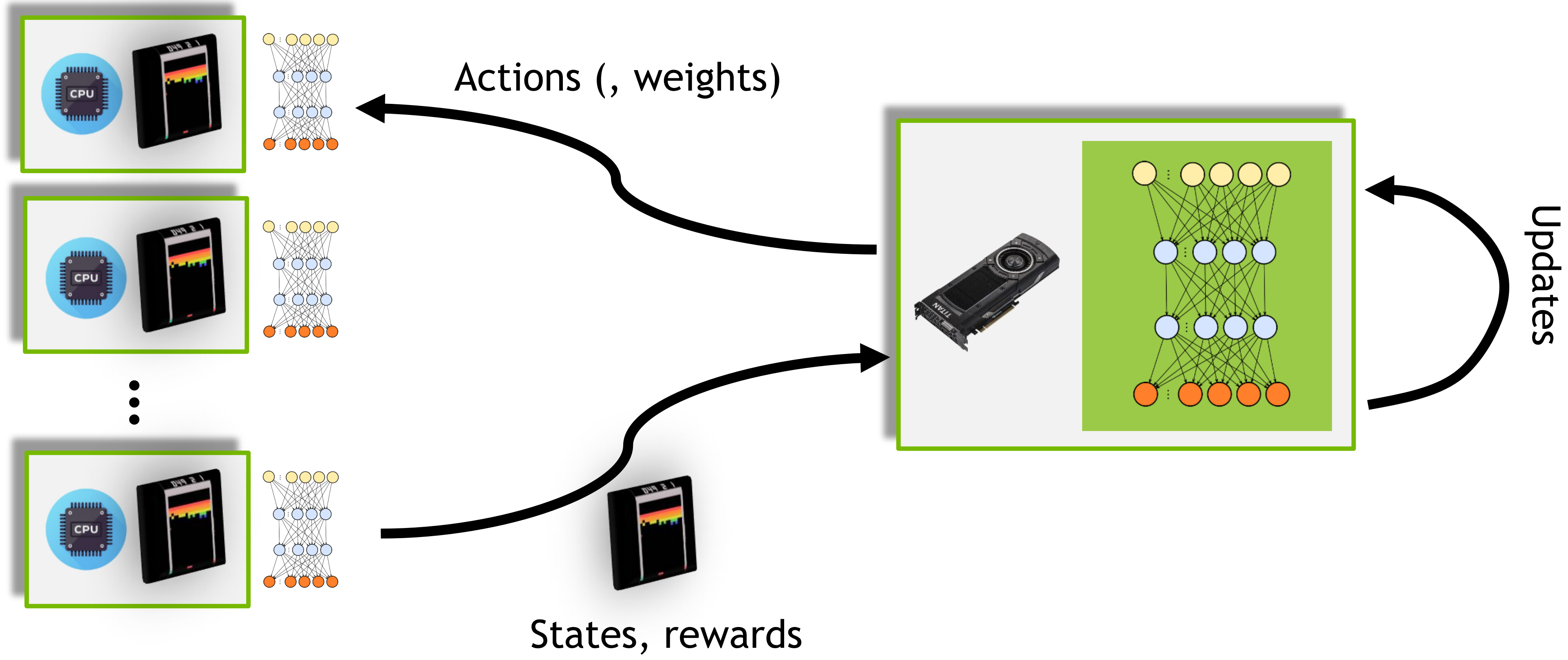
CuLE

Performance

Analysis and new scenarios

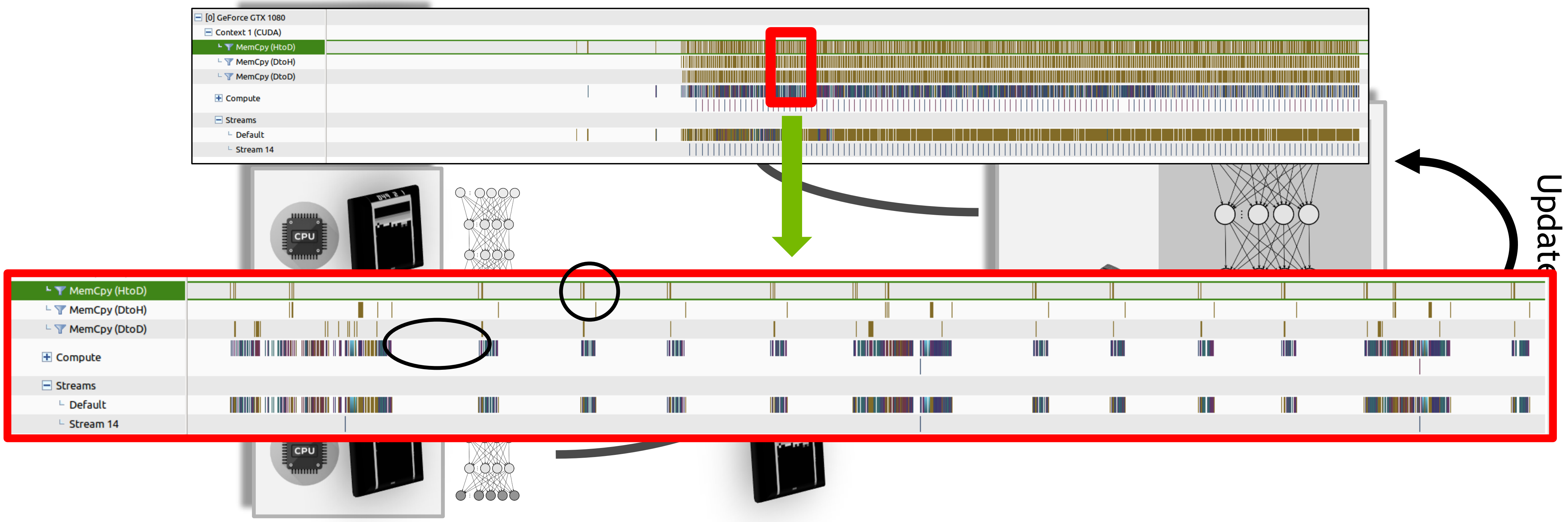
RL TRAINING (CPU SIMULATION)

Standard training scenario



RL TRAINING (CPU SIMULATION)

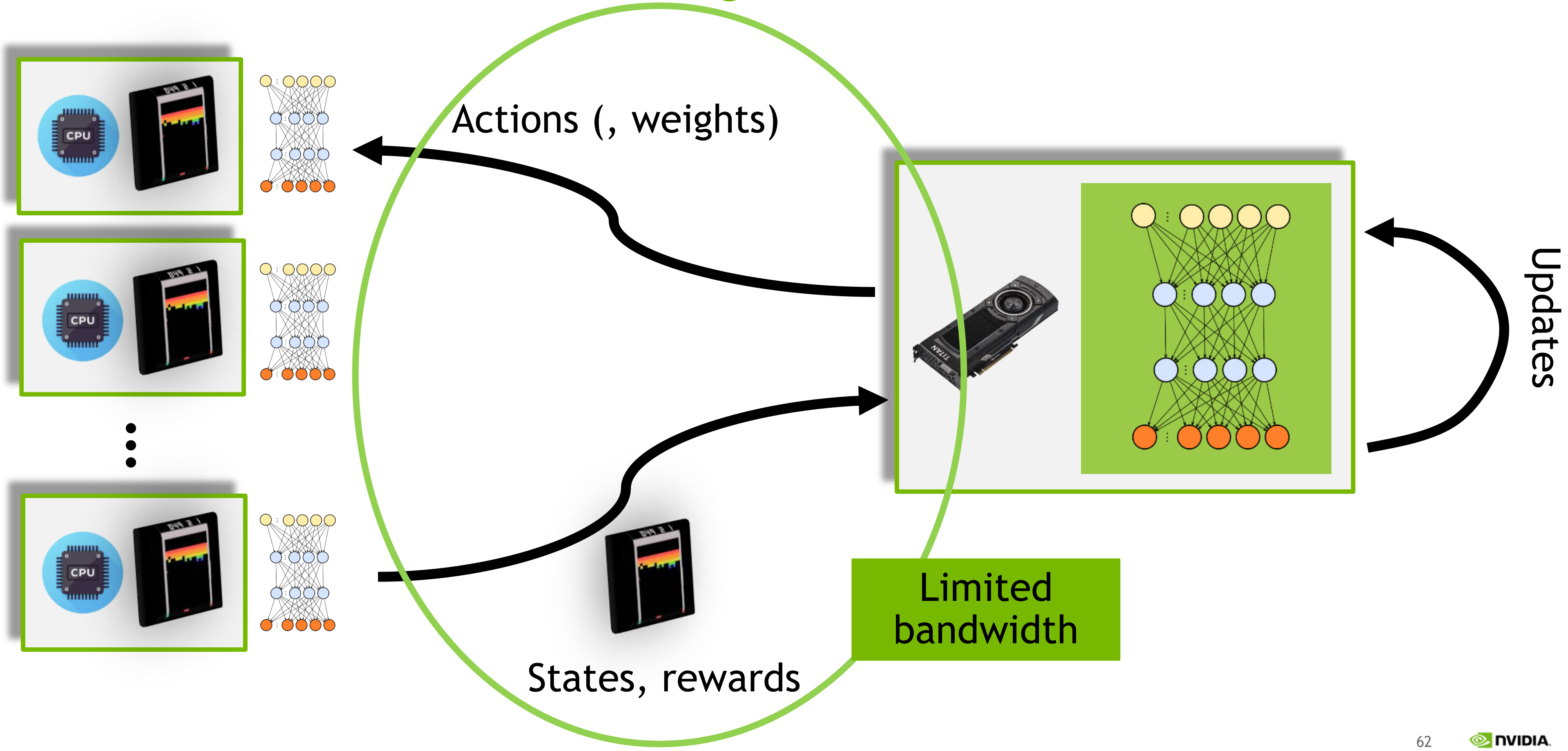
Standard training scenario



States, rewards

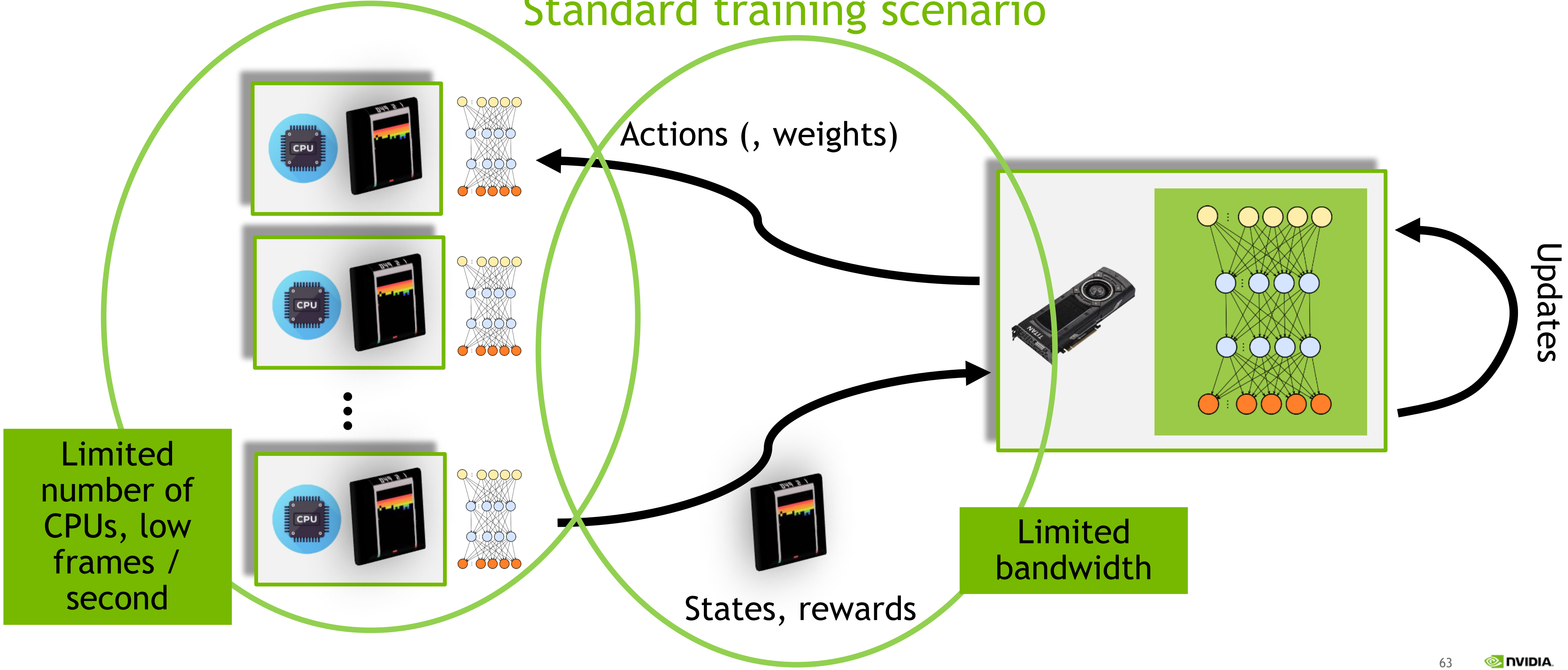
RL TRAINING (CPU SIMULATION)

Standard training scenario



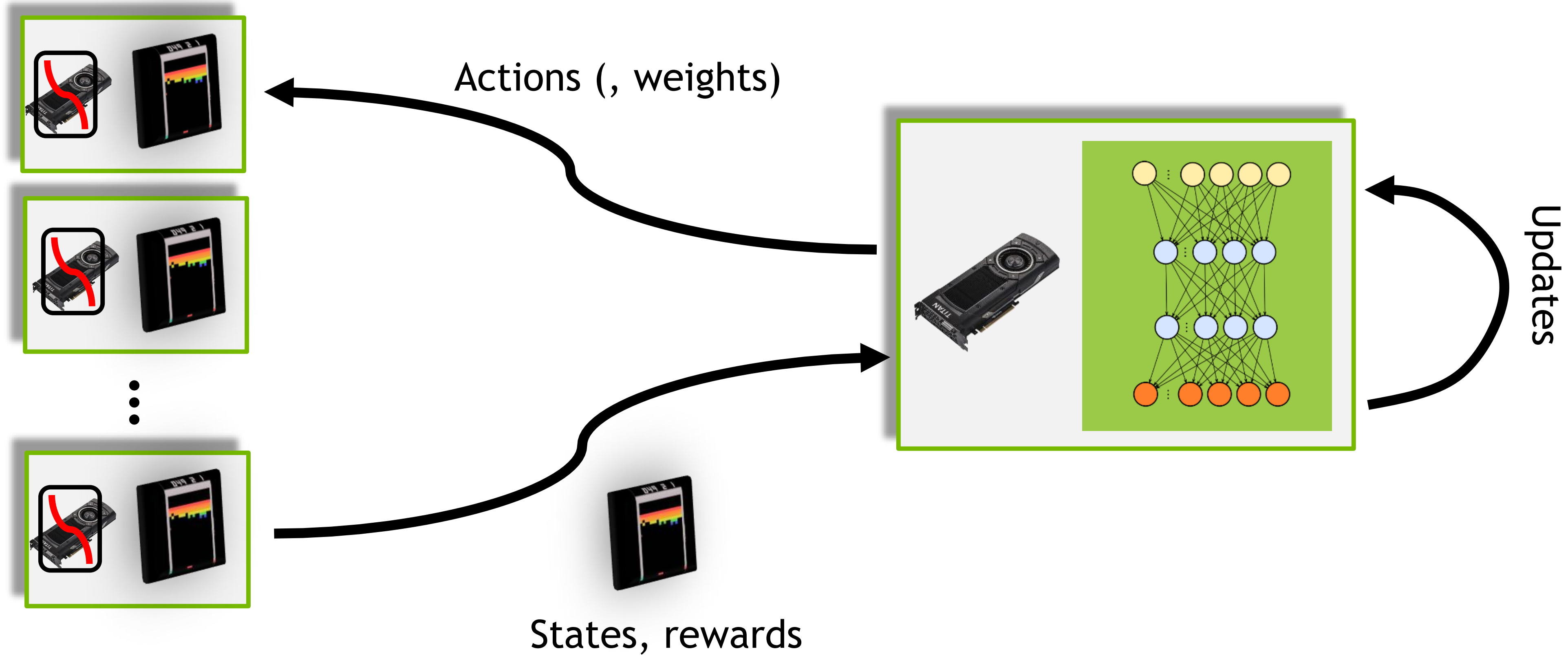
RL TRAINING (CPU SIMULATION)

Standard training scenario



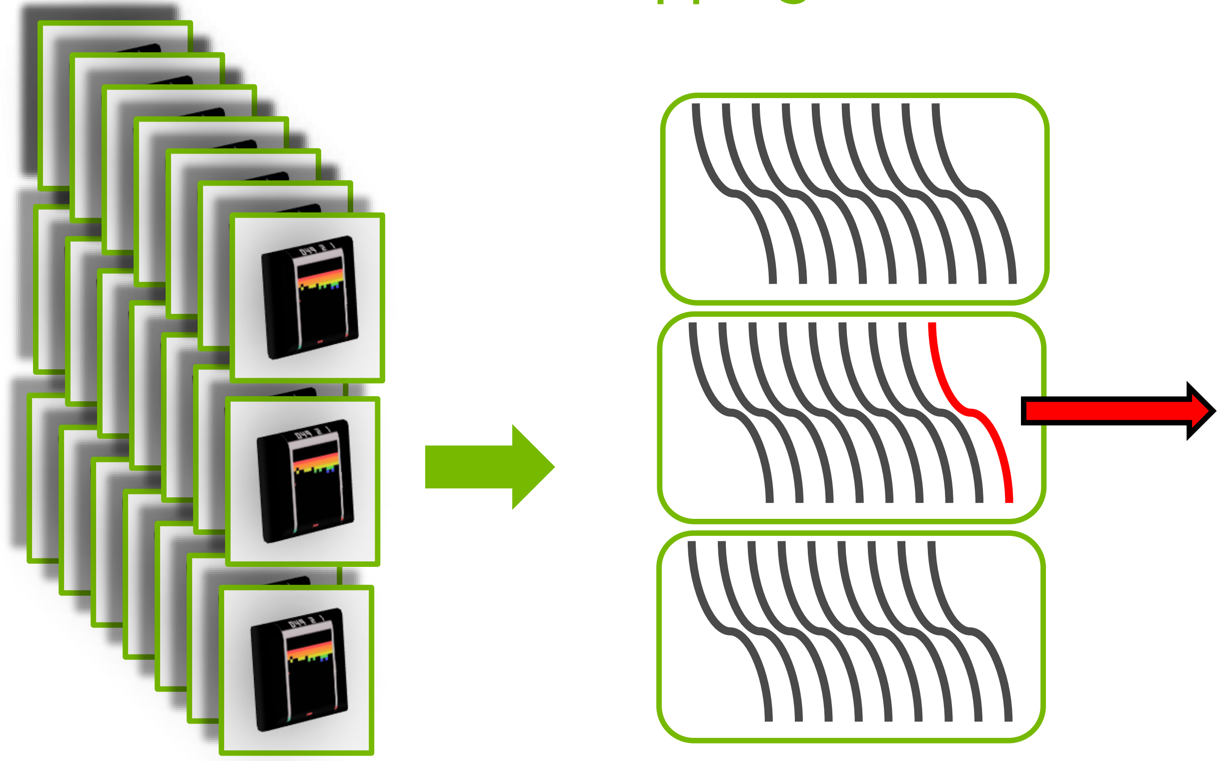
RL TRAINING (CULE)

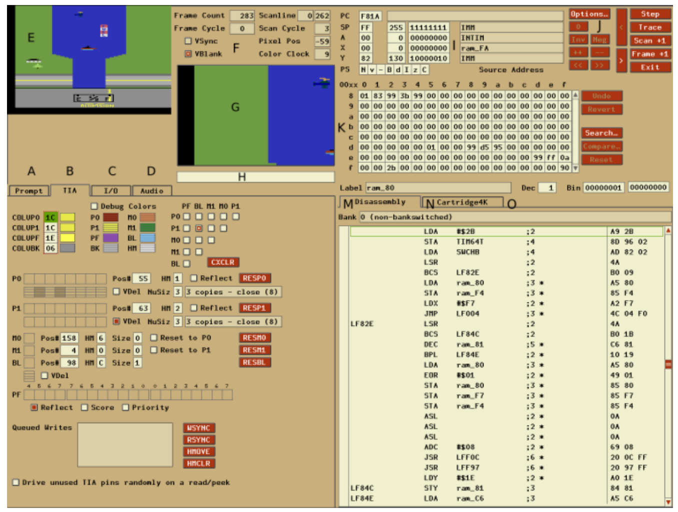
Porting ATARI to the GPU



RL TRAINING (GPU)

1-to-1 mapping of ALEs to threads





ALE ATARI simulator

AGENDA

GYM COMPATIBLE (MOSTLY)

AtariPy

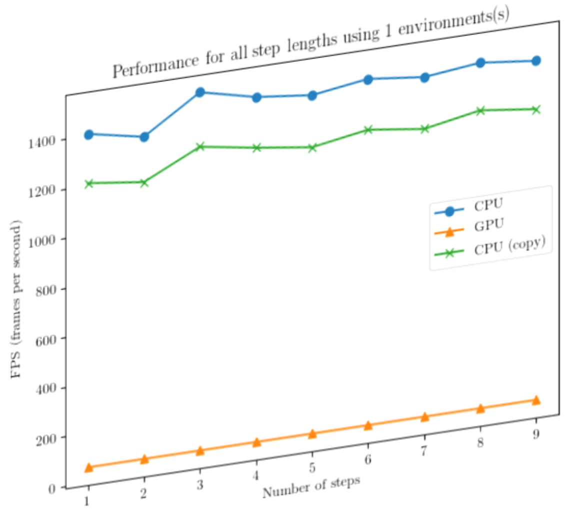
```
for agent in (0, agents):
    action.cpu() # transfer to CPU
    observation, reward, done, info = env.step(action.numpy()) # execute
    observation.cuda() # transfer back GPU
    reward.cuda()
```

CuLE

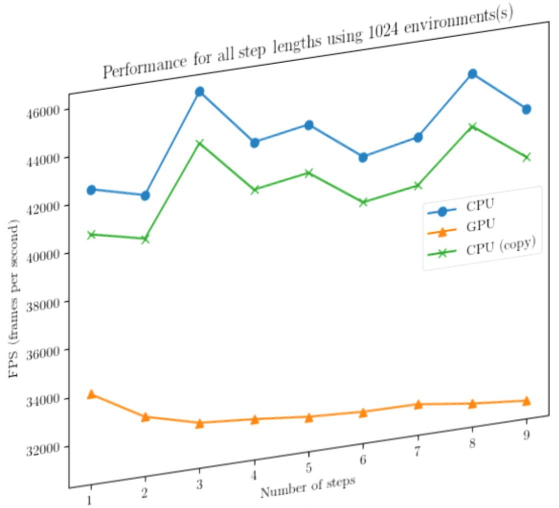
```
# parallel call to all agents
observations, rewards, dones, infos = env.step(actions) # execute
```

FRAMES PER SECOND

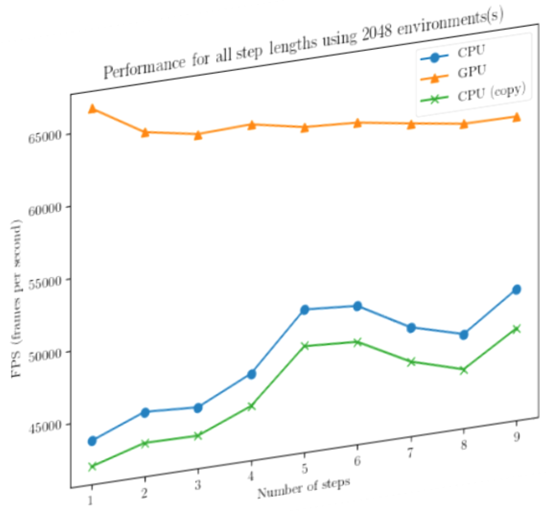
Breakout, inference only (no training)



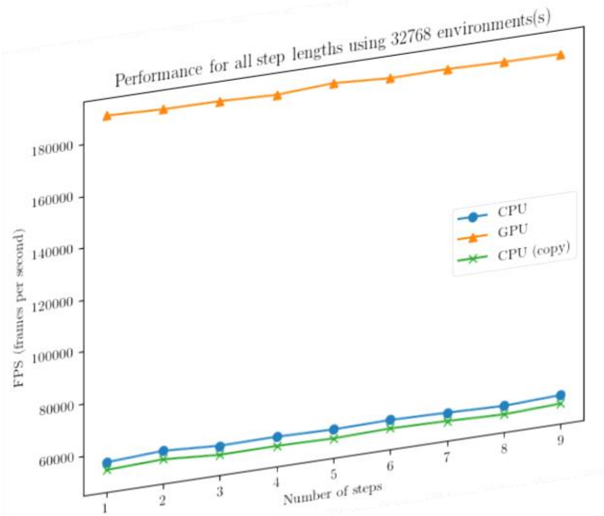
1 environment



1024 environments



4096 environments



32768 environments

GPU occupancy

GYM COMPATIBLE (MOSTLY)

AtariPy

```
for agent in (0, agents):
    action.cpu() # transfer to CPU
    observation, reward, done, info = env.step( action.numpy()) # execute
    observation.cuda() # transfer back GPU
    reward.cuda()

train()
```

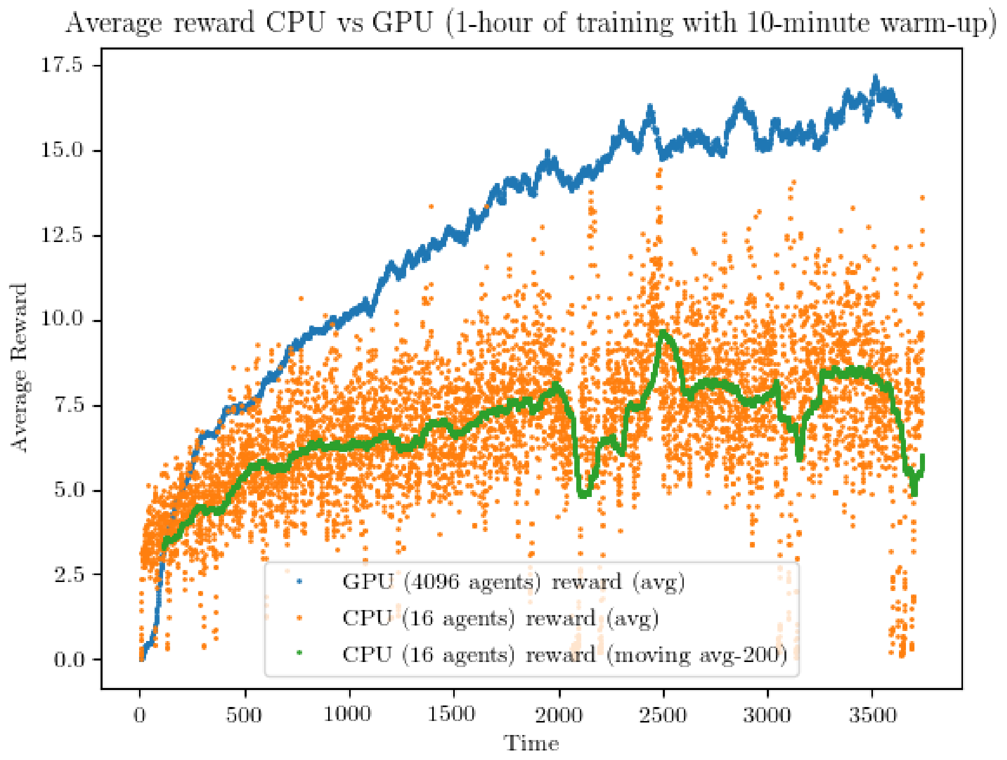
CuLE

```
# parallel call to all agents
observations, rewards, dones, infos = env.step(actions) # execute

train()
```

REINFORCEMENT LEARNING

Breakout – A2C (preliminary result)



AGENDA

RL training: CPU, GPU

Limitations

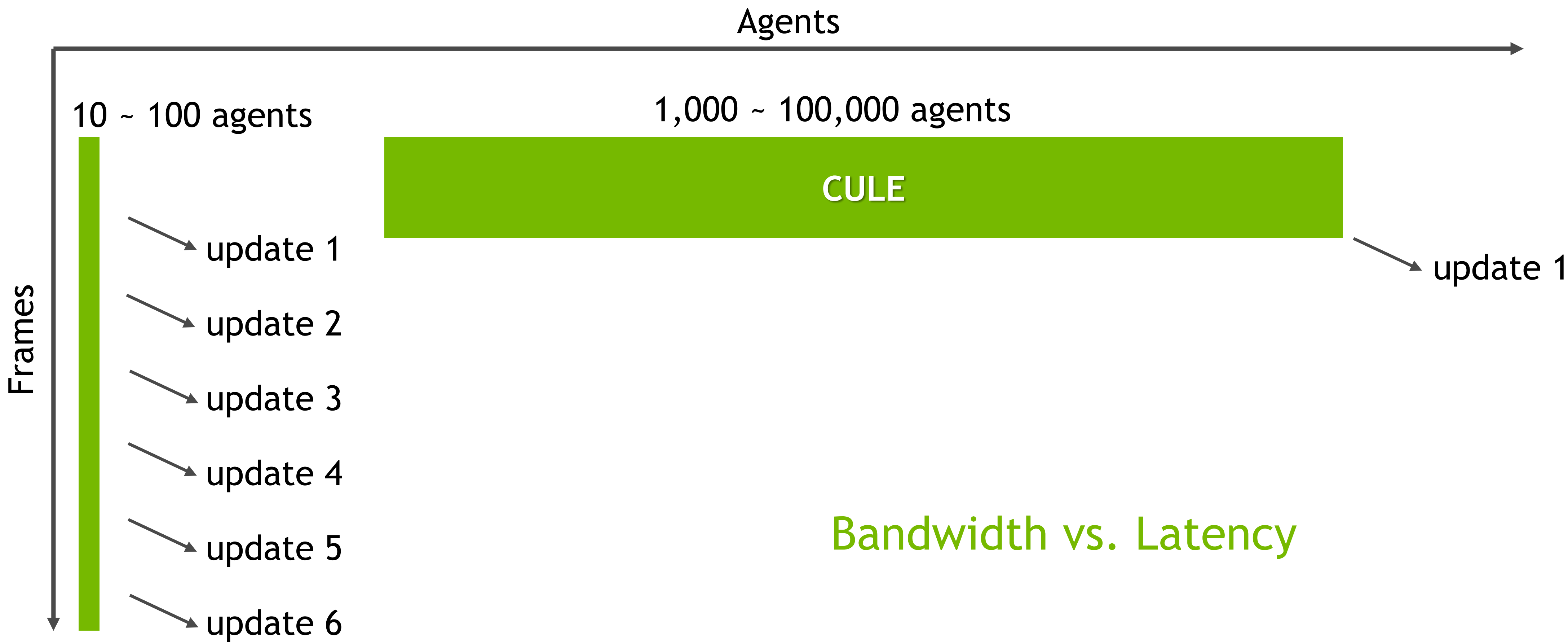
CuLE

Performance

Analysis and new scenarios

TRADE-OFF

Same amount of time: CuLE vs. non CuLE

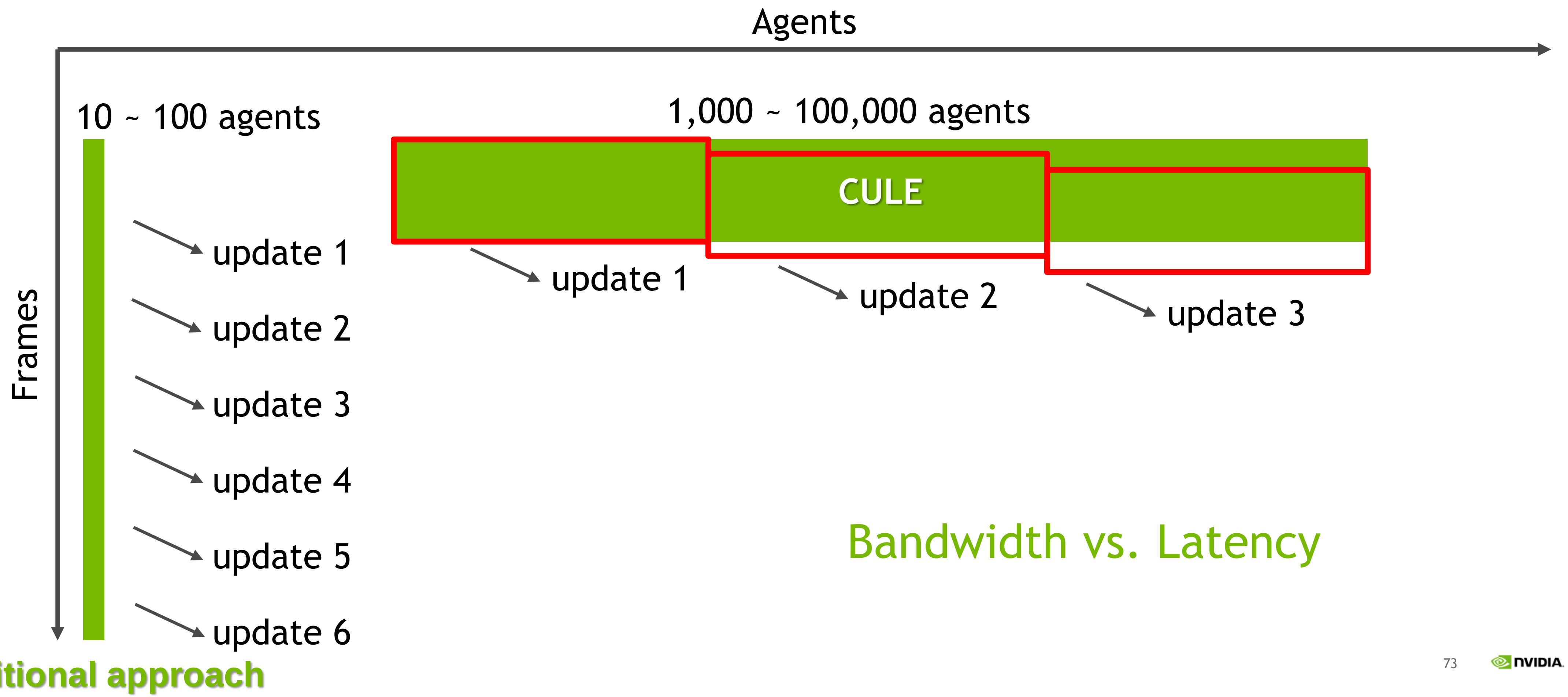


Bandwidth vs. Latency

Traditional approach

TRADE-OFF

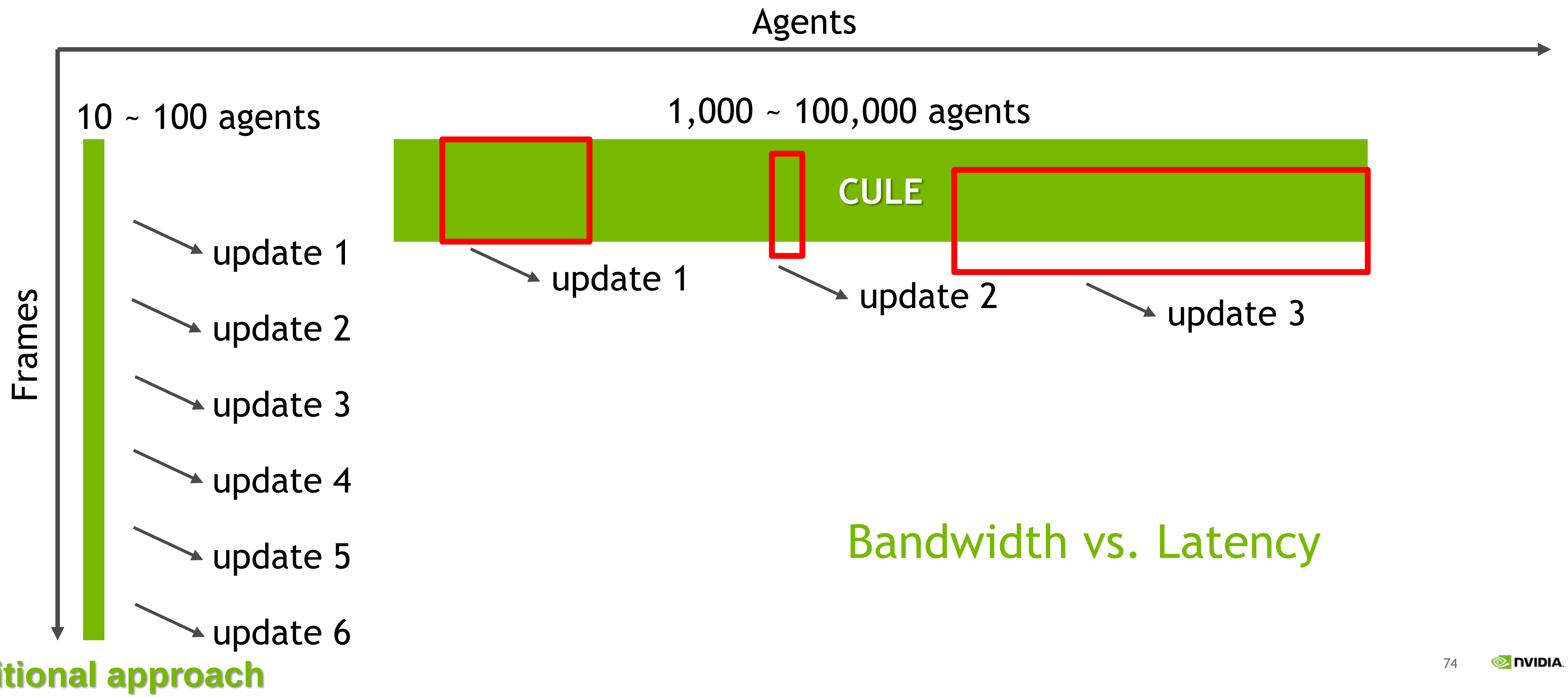
Same amount of time: CuLE vs. non CuLE



Bandwidth vs. Latency

TRADE-OFF

Same amount of time: CuLE vs. non CuLE



Bandwidth vs. Latency

GYM COMPATIBLE (MOSTLY)

AtariPy / CuLE

```
for time in (0, np.inf):
    action.cpu() # transfer to CPU
    observation, reward, done, info = env.step( action.numpy()) # execute
    cpu_state = cule.get_state() # get state

train()
```

CuLE

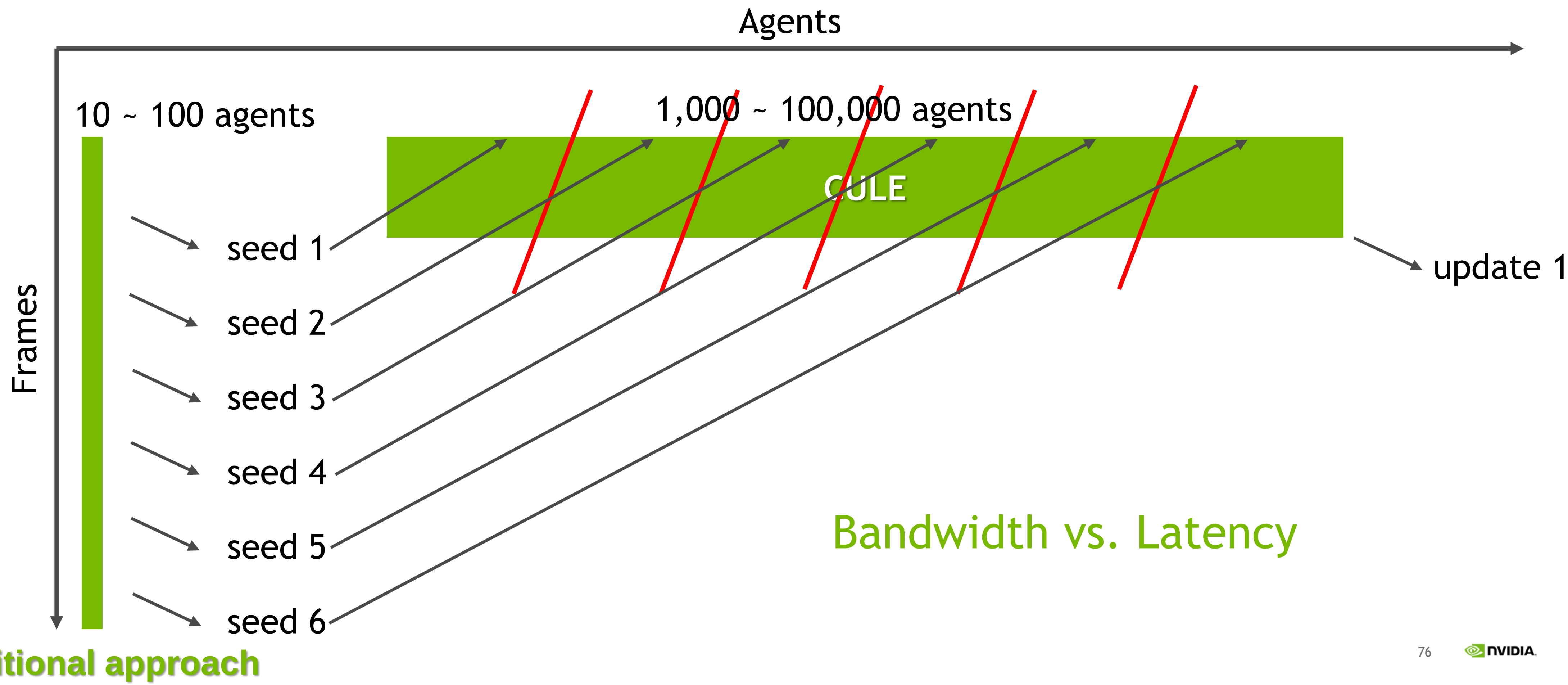
```
# seed, cule::set_state(gpuState, cpuState)
env.seed(cpu_state, first_agent = 0, last_agent = 100)

# parallel call to all agents
observations, rewards, dones, infos = env.step(actions) # execute

# …
```

SEEDING

Same amount of time: CuLE vs. non CuLE



Bandwidth vs. Latency

CONCLUSION

CuLE

More frames for less money (democratizing RL)

New scenarios

How to use large batches?

Seeding from the CPU, ES, …

Soon released on https://github.com/NVlabs/

RESOURCES

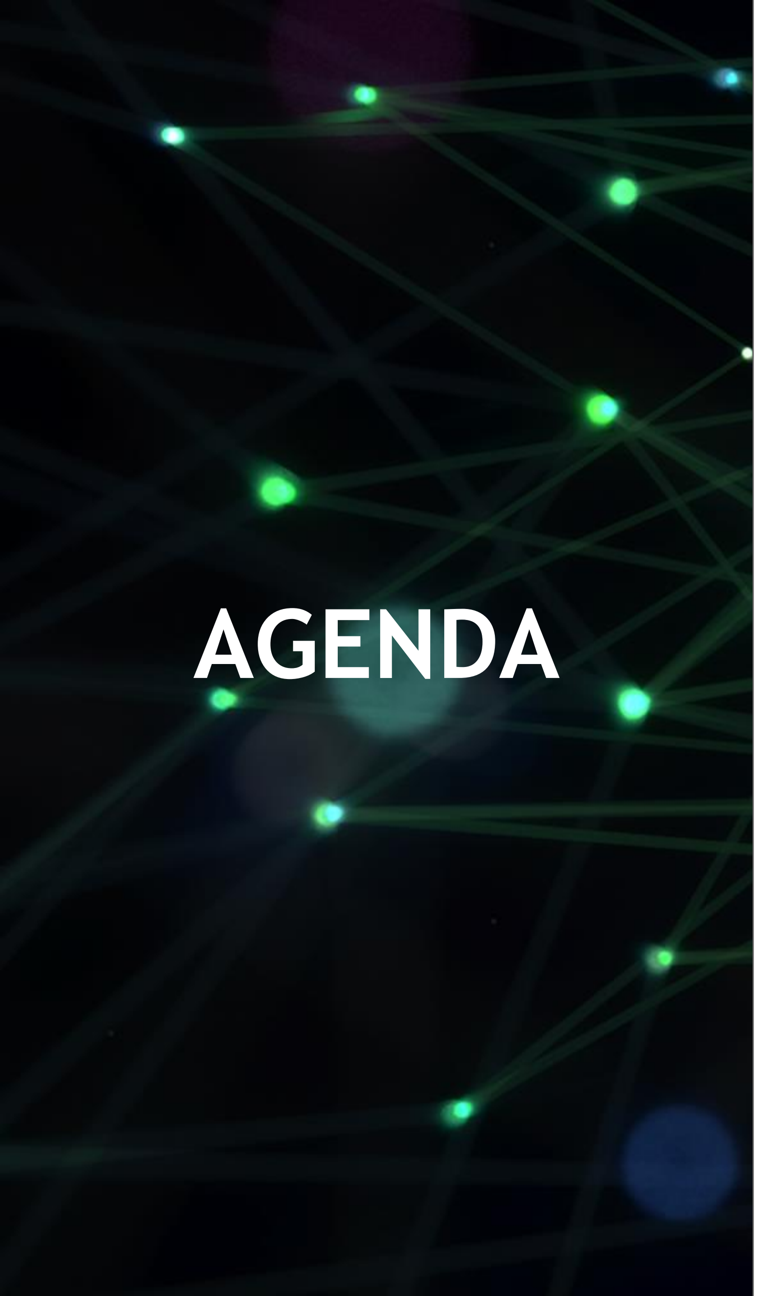
THEORY

S. Dalton, I. Frosio, J. Hoberok, J. Clemons, CULE, a companion library for accelerated RL training, GTC 2018, (http://on-demand.gputechconf.com/gtc/2018/presentation/s8440-cule-a-companion-library-for-accelerated-rl-training.pdf).

CODING

Releasing soon:

https://github.com/NVlabs



AGENDA

1. GPU-based A3C for Deep Reinforcement Learning
(The basics of Reinforcement Learning on a CPU/GPU)

2. Cule*: GPU accelerated RL
(Moving Reinforcement Learning on a GPU)

3. Sim-to-Real Transfer of Accurate Grasping with Eye-In-Hand Observations and Continuous Control
(Imitation Learning and Sample Efficiency)

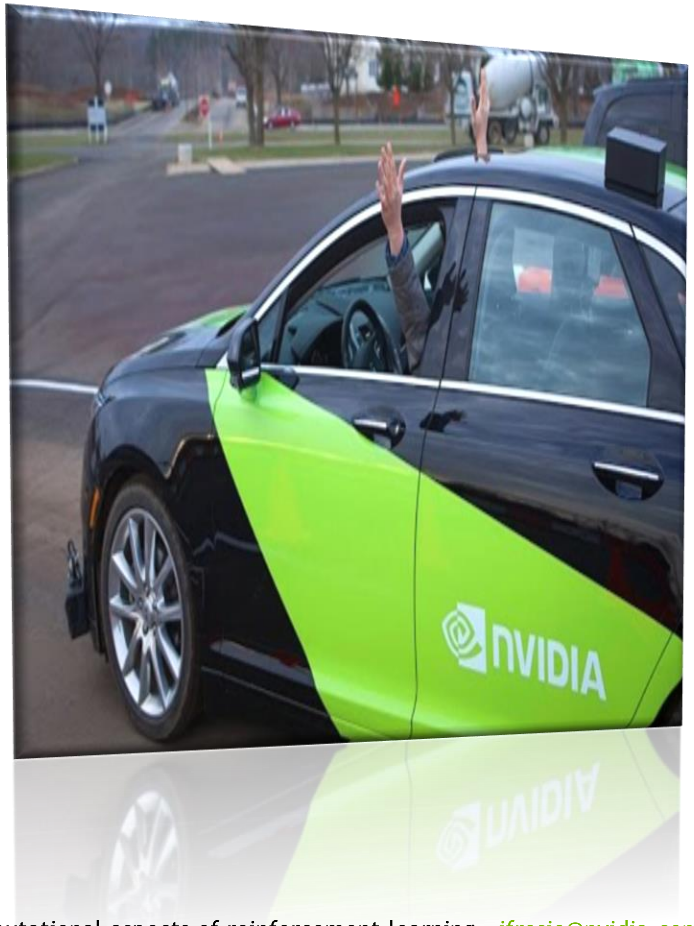
4. Conclusion

SIM-TO-REAL TRANSFER OF ACCURATE GRASPING WITH EYE-IN-HAND OBSERVATIONS AND CONTINUOUS CONTROL

Mengyuan Yan, Iuri Frosio, Stephen Tyree, Jan Kautz

ROBOTS

ROBOTS GOING INTO THE WILD

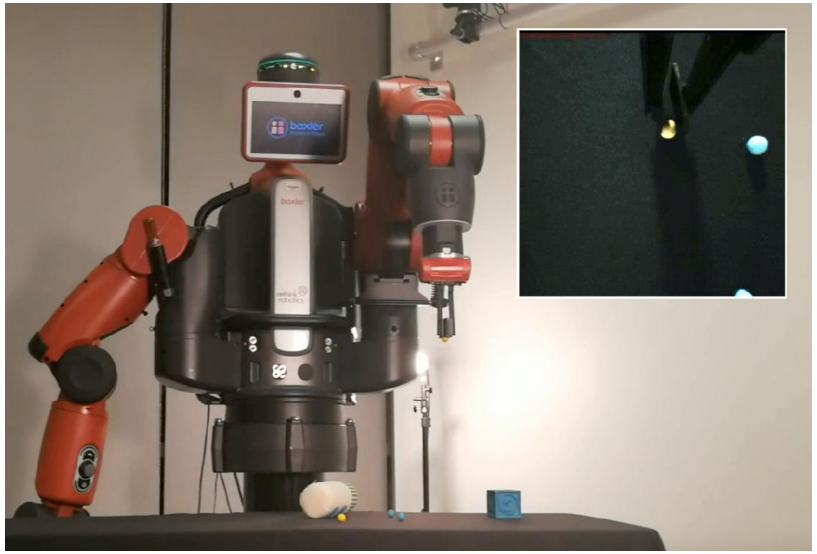


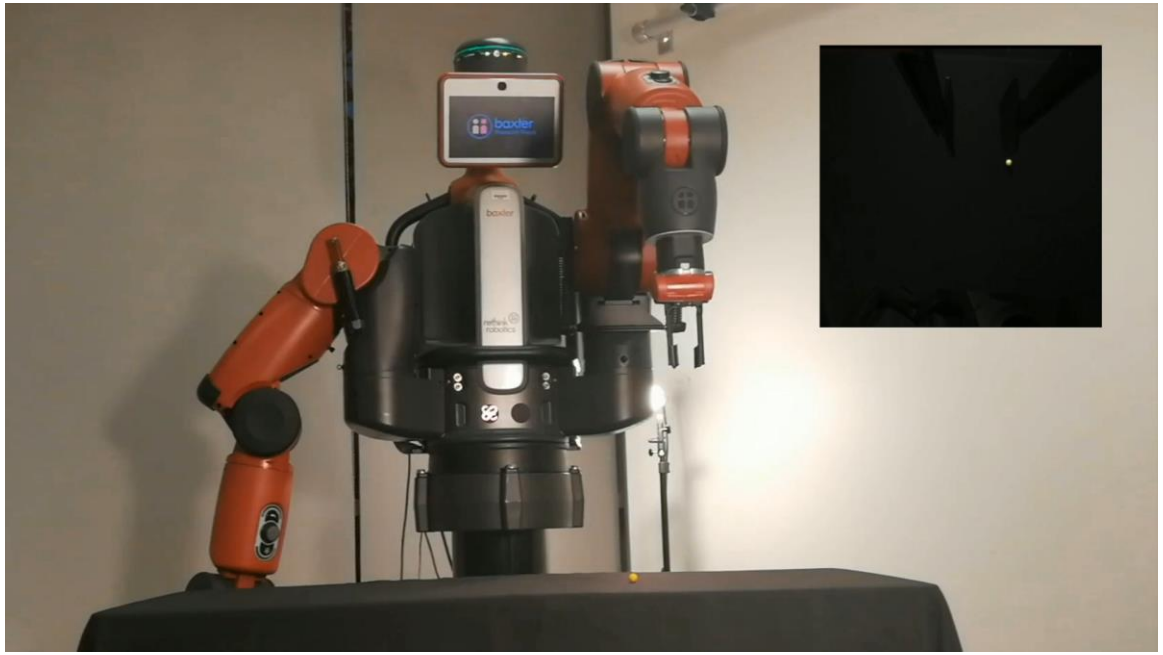
CHALLENGES & SOLUTION

Challenge	Solution
Safety	Simulation
Data collection	Simulation
Sample efficiency	Imitation Learning
Simulation ☺	Domain transfer

THE PROBLEM OF GRASPING

"Elementary" thus significant





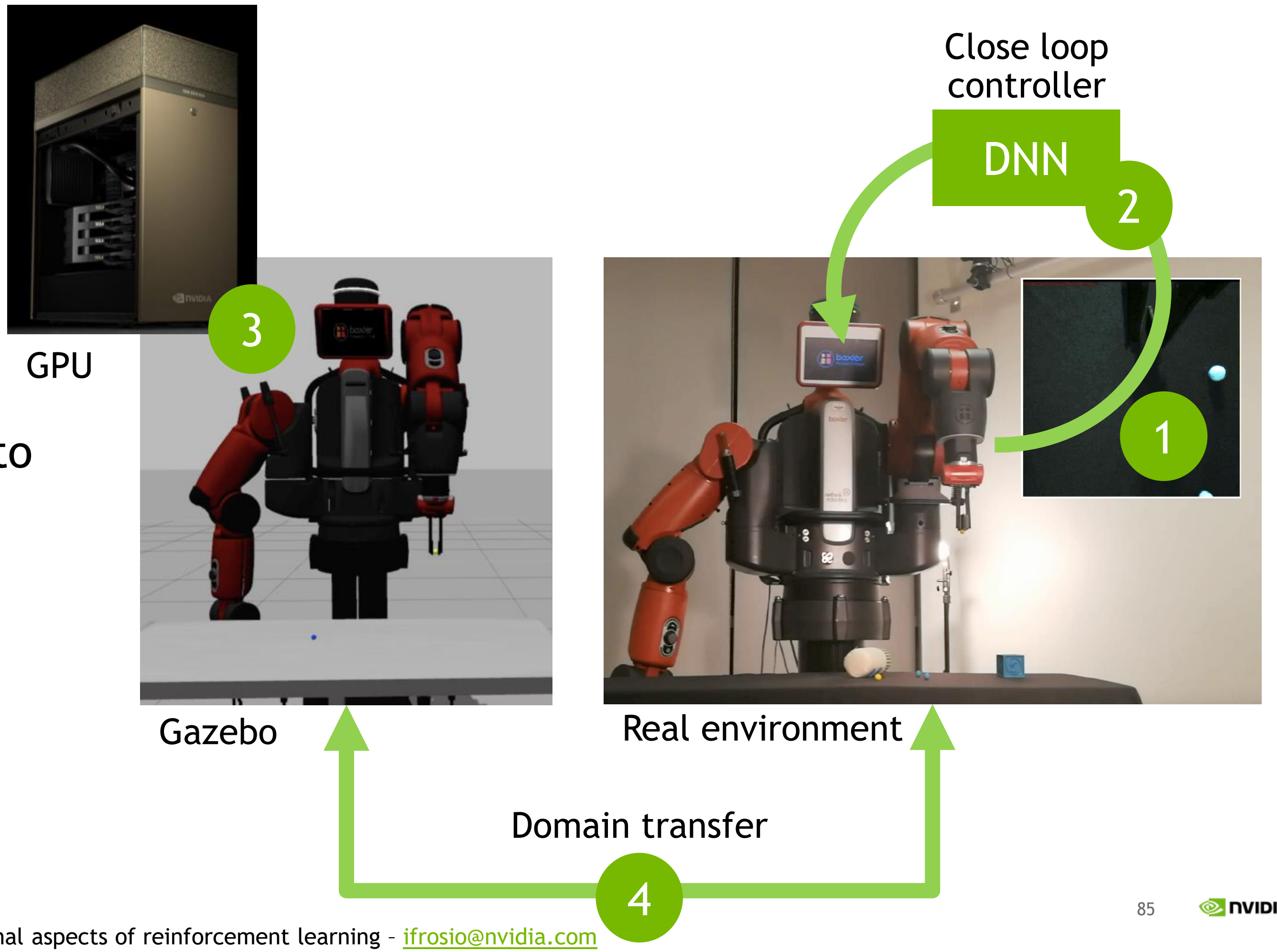
LEARNING AND TOOLS

1) Visual input from wrist camera

2) Closed-loop continuous position control

3) Use imitation learning (on a GPU) to learn in simulation (Gazebo)

4) Visual domain transfer to real environments



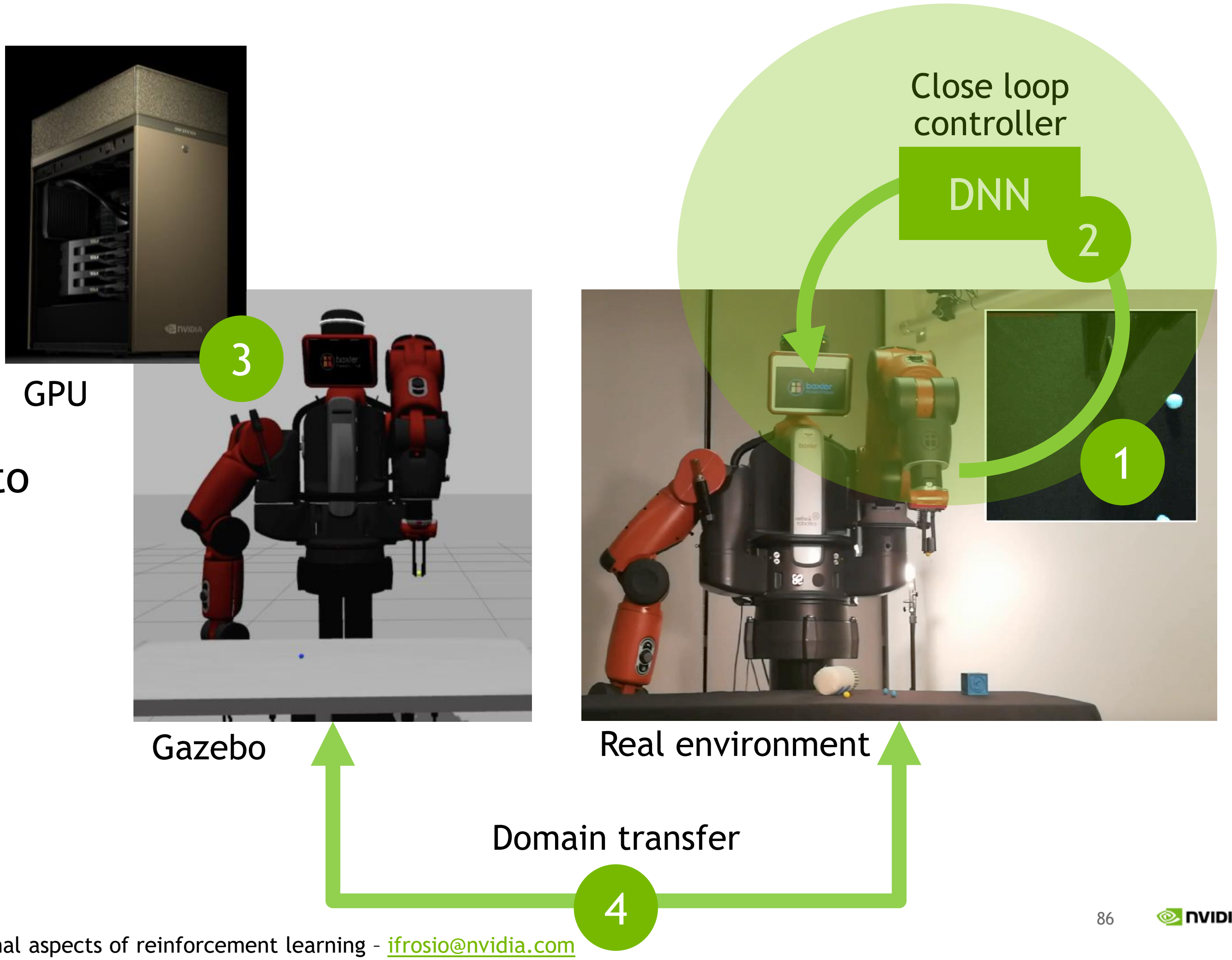
LEARNING AND TOOLS

1) Visual input from wrist camera

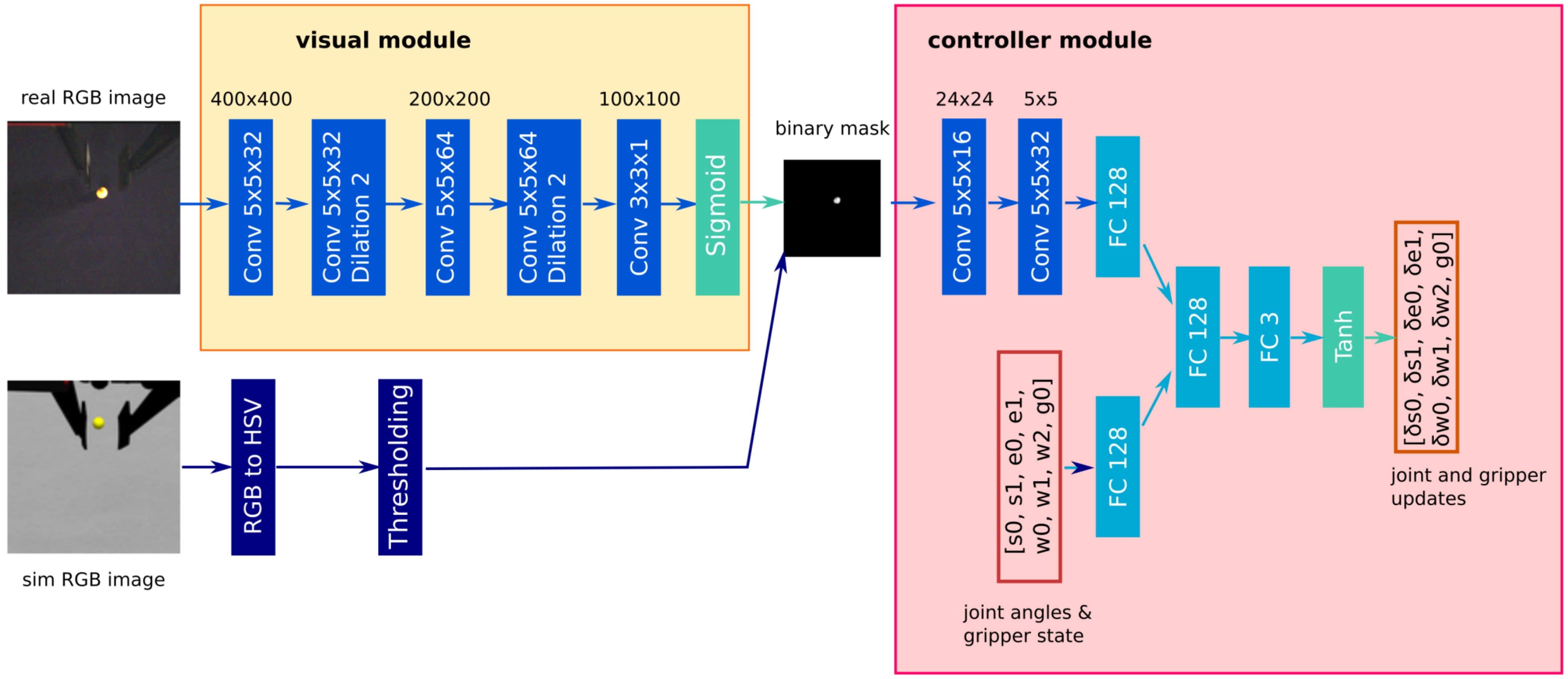
2) Closed-loop continuous position control

3) Use imitation learning (on a GPU) to learn in simulation (Gazebo)

4) Visual domain transfer to real environments



CONTROLLER AND VISUAL DOMAIN TRANSFER



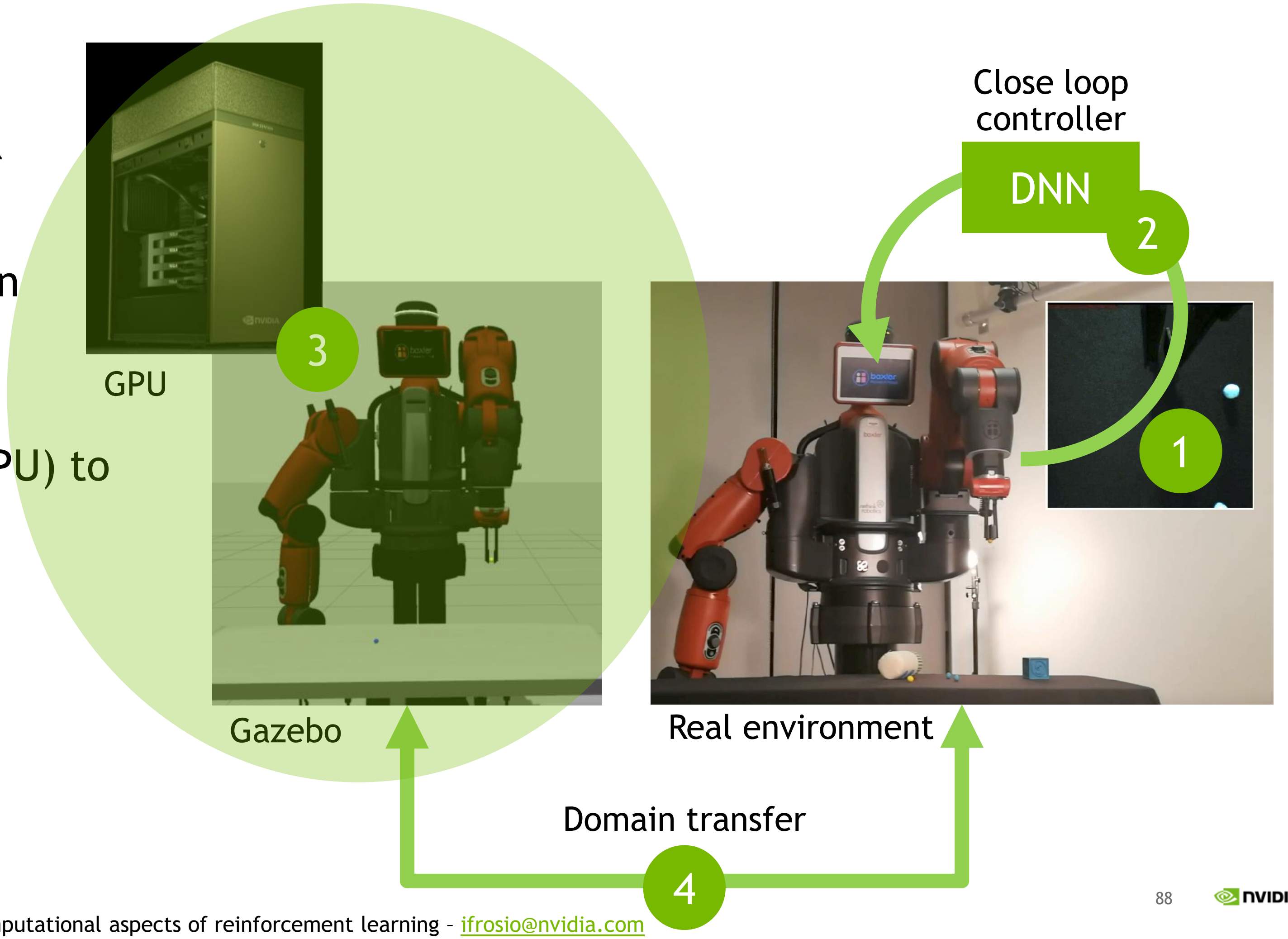
LEARNING AND TOOLS

1) Visual input from wrist camera

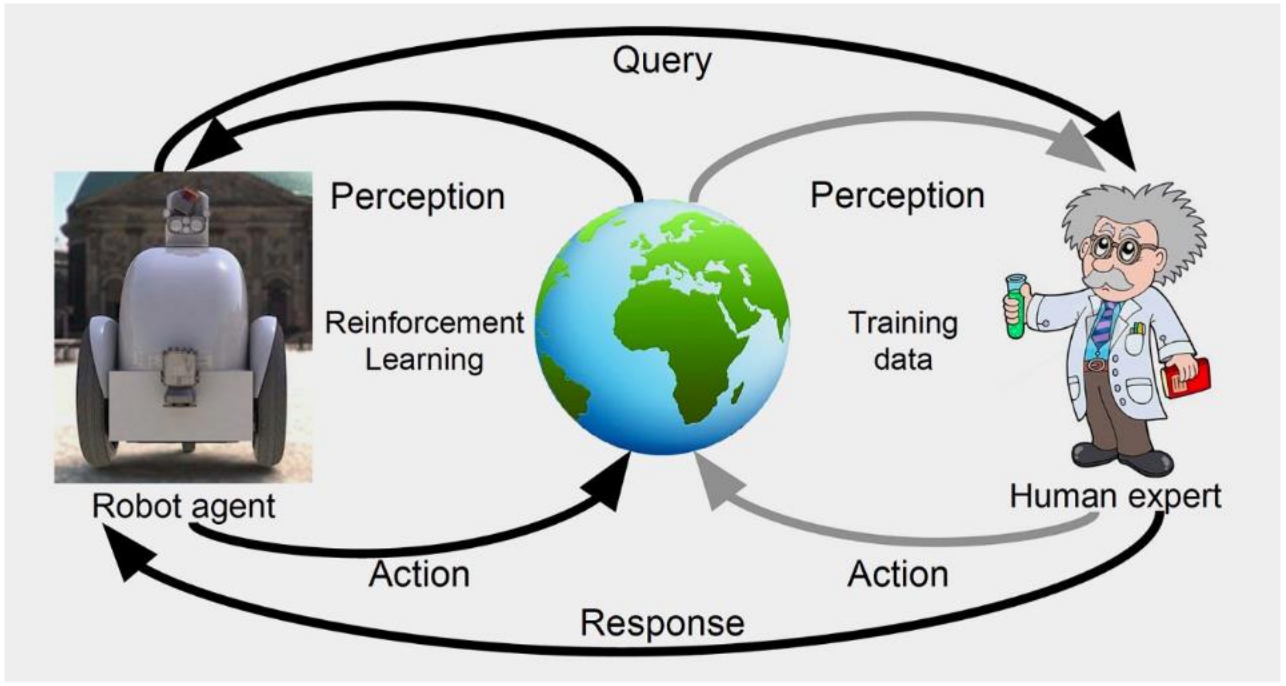
2) Closed-loop continuous position control

3) Use imitation learning (on a GPU) to learn in simulation (Gazebo)

4) Visual domain transfer to real environments



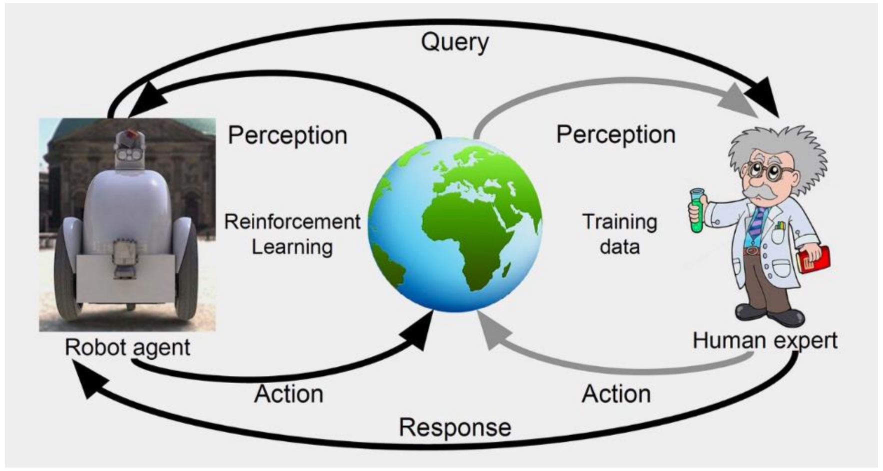
IMITATION LEARNING

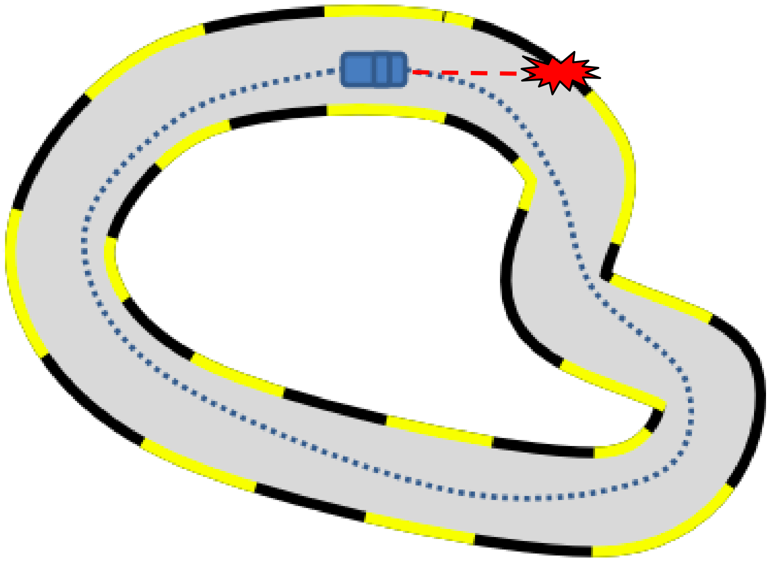


The expert is not necessarily solving our problem.

It can solve the problem by having a complete access to the status of the system, which is not available to our agent!

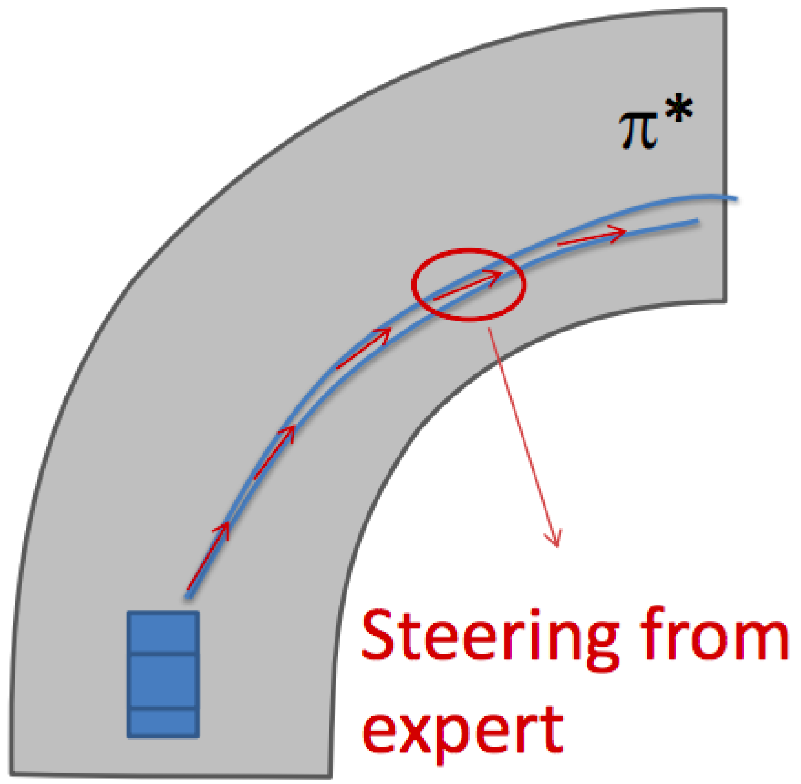
COVARIANCE SHIFT





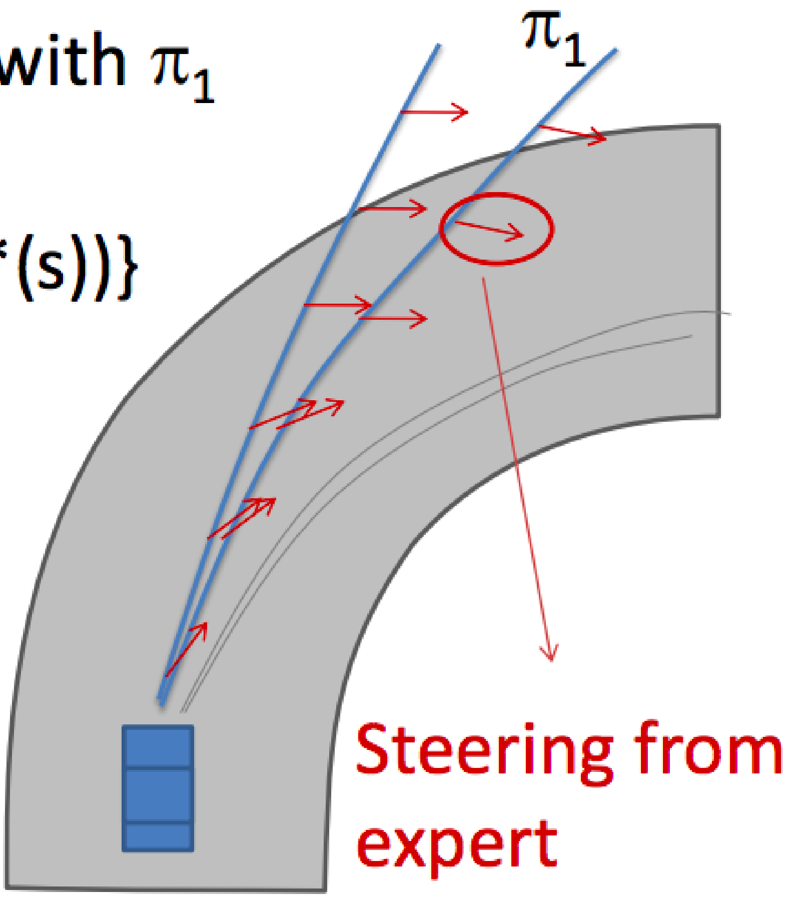
DAgger: Dataset Aggregation

- Collect trajectories with expert π^*
- Dataset $D_0 = \{(s, \pi^*(s))\}$
- Train π_1 on D_0



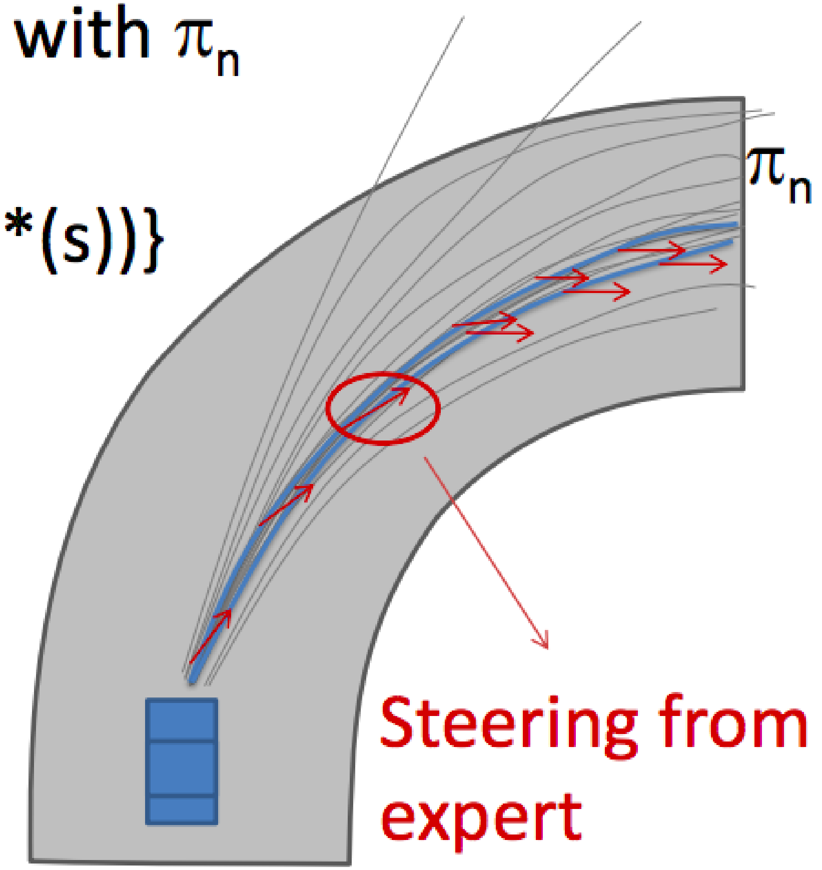
DAgger: Dataset Aggregation

- Collect new trajectories with π_1
- New Dataset $D_1' = \{(s, \pi^*(s))\}$
- Aggregate Datasets:
  $D_1 = D_0 \cup D_1'$
- Train π_2 on D_1



DAgger: Dataset Aggregation

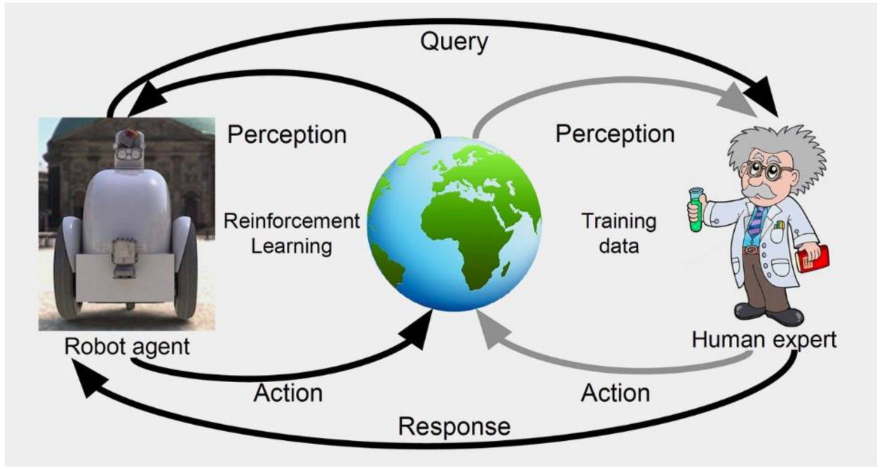
- Collect new trajectories with π_n
- New Dataset $D_n' = \{(s, \pi^*(s))\}$
- Aggregate Datasets:
  $D_n = D_{n-1} \cup D_n'$
- Train π_{n+1} on D_n



IMITATION LEARNING VS REINFORCEMENT LEARNING

Pros and cons

Imitation learning pros and cons



+ Sample efficient (data reuse)

+ Less time spent for exploration

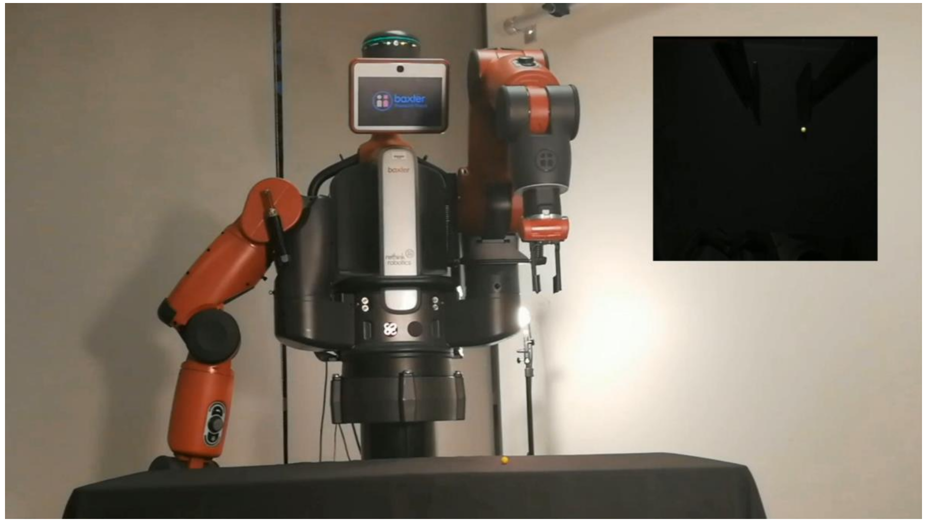
+ Faster convergence

- Need for an expert

- Lower generalization capabilities

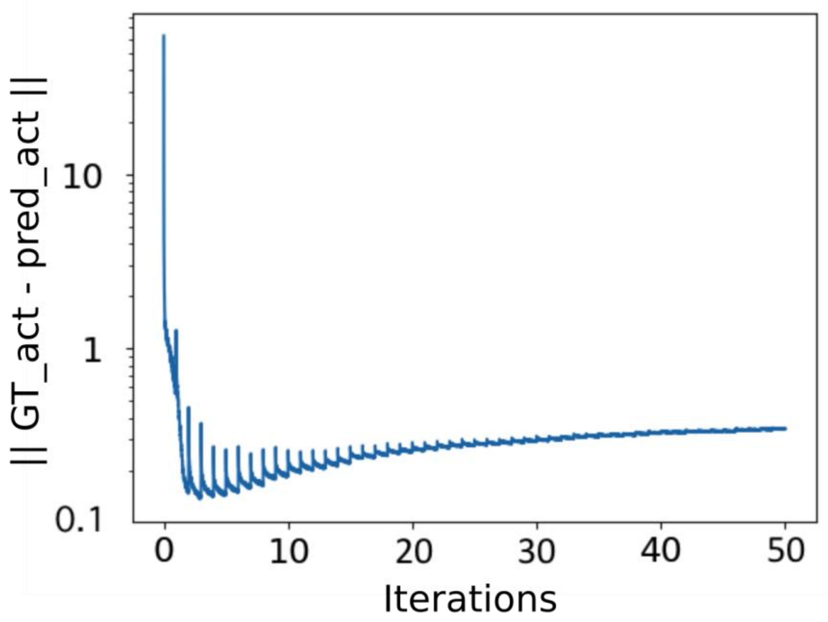
- Optimize metric?

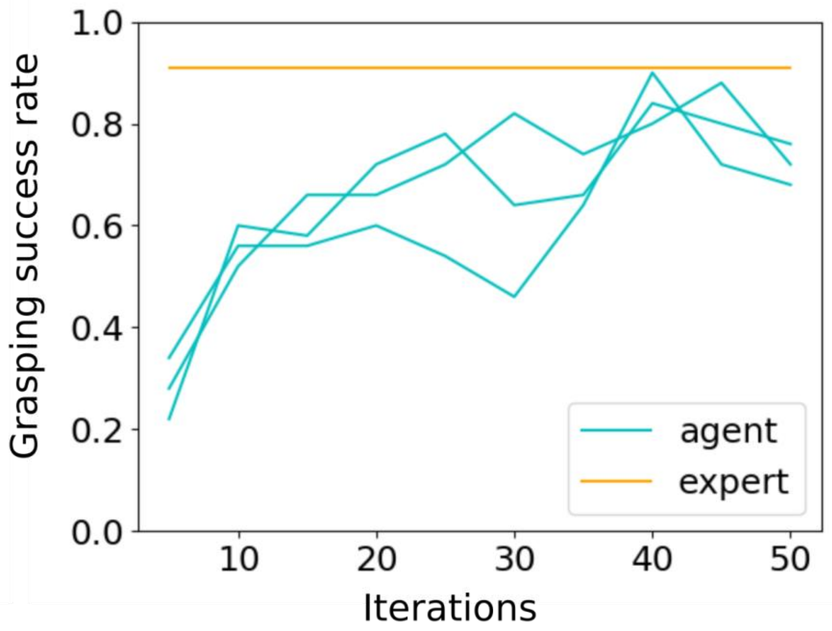
FSM EXPERT



$$\begin{tikzcd}
\text{Reset} \arrow[r, bend right] & \text{Approach} \arrow[l, bend right] \arrow[r, bend right] & \text{Grasp} \arrow[l, bend right]
\end{tikzcd}$$

DAGGER: LEARNING CURVE





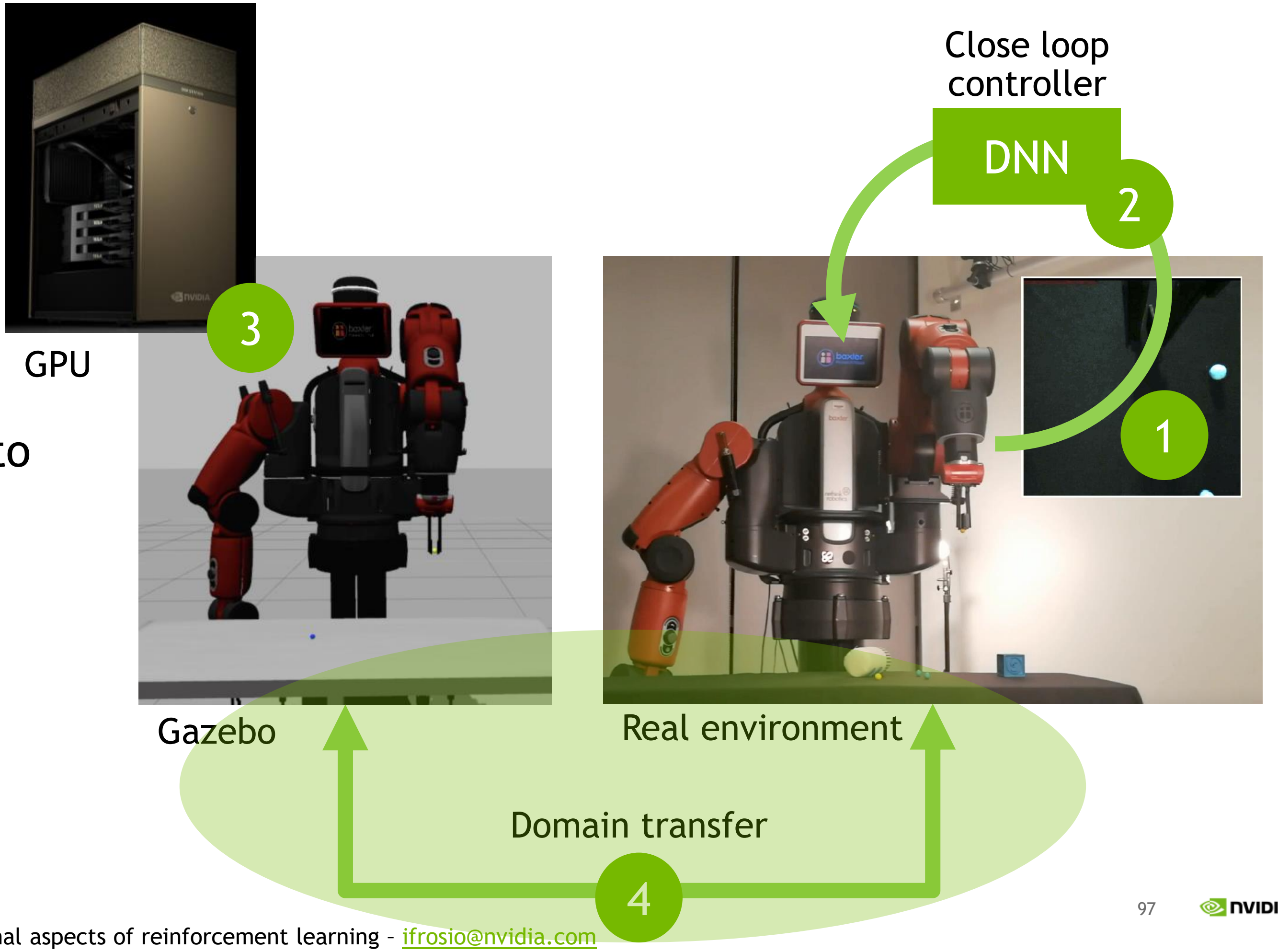
LEARNING AND TOOLS

1) Visual input from wrist camera

2) Closed-loop continuous position control

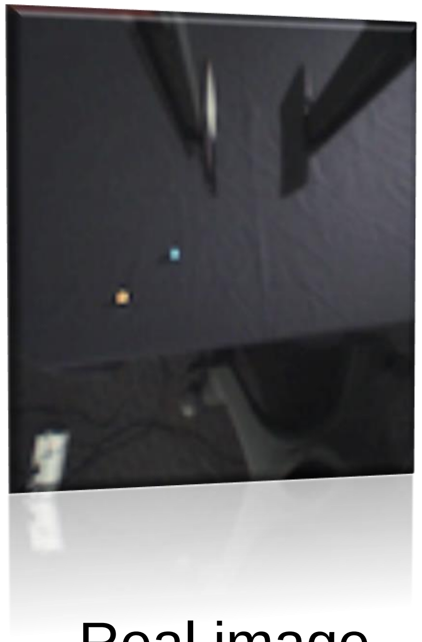
3) Use imitation learning (on a GPU) to learn in simulation (Gazebo)

4) Visual domain transfer to real environments



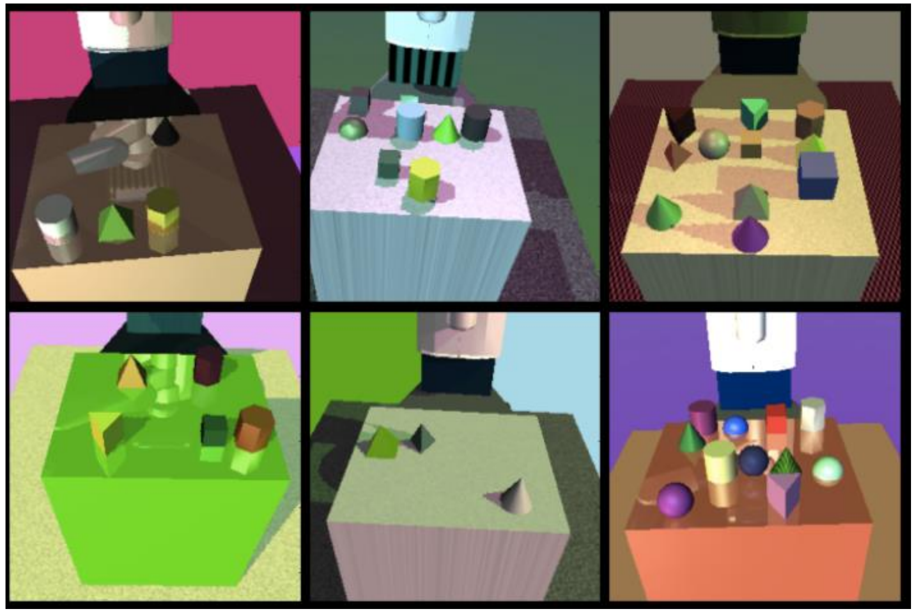
TRANSFERRING TO THE REAL ROBOT

Domain randomization



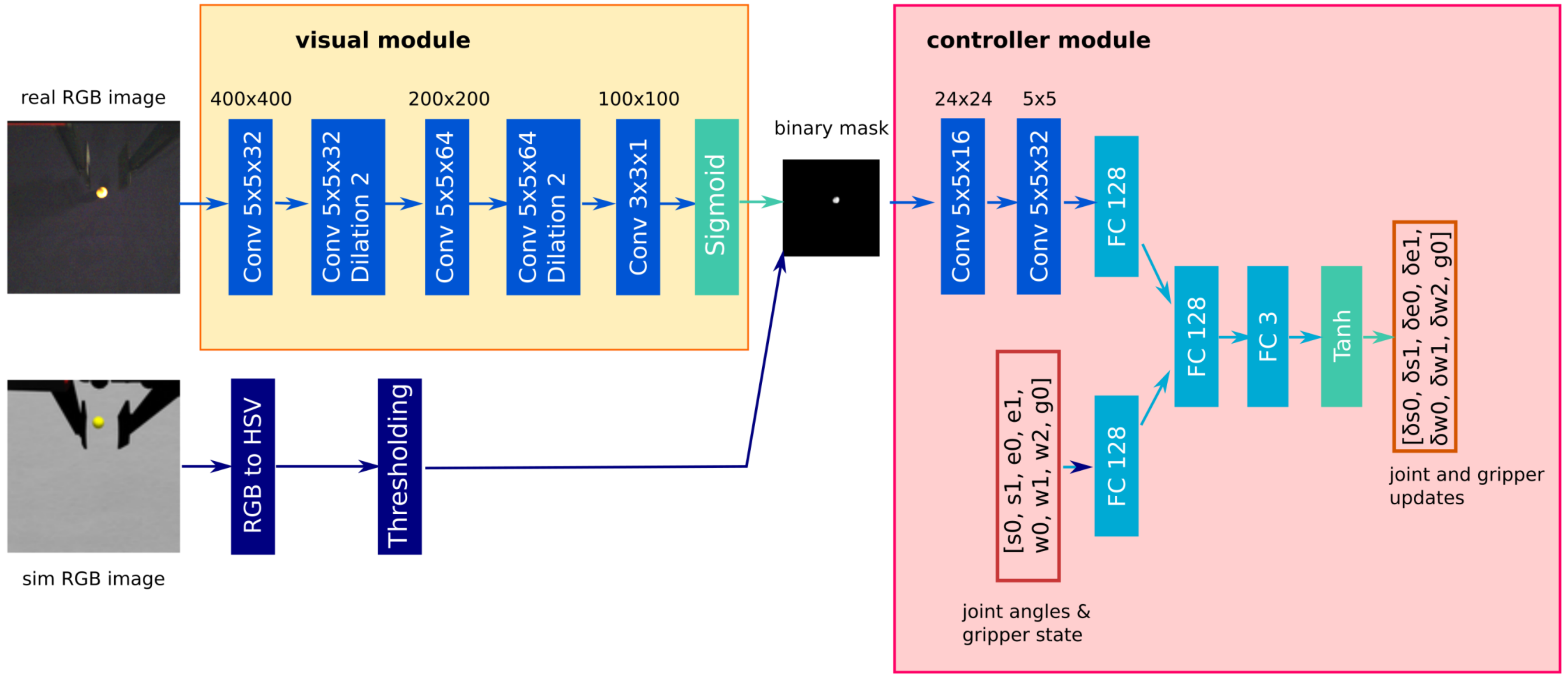
Real image

Synthetic image

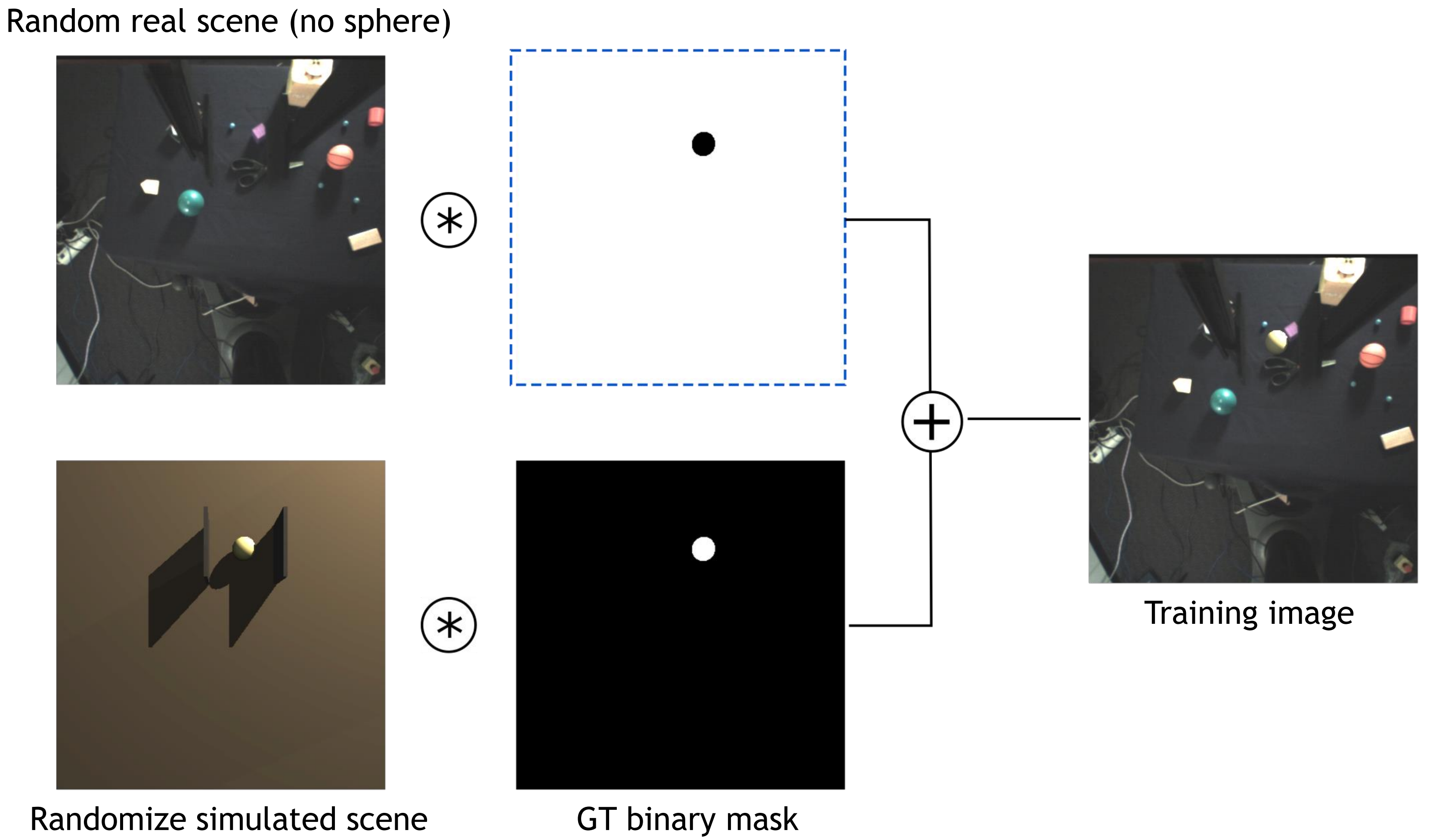


Tobin, Josh et al. “Domain randomization for transferring deep neural networks from simulation to the real world.” 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2017): 23-30.

CONTROLLER AND VISUAL DOMAIN TRANSFER

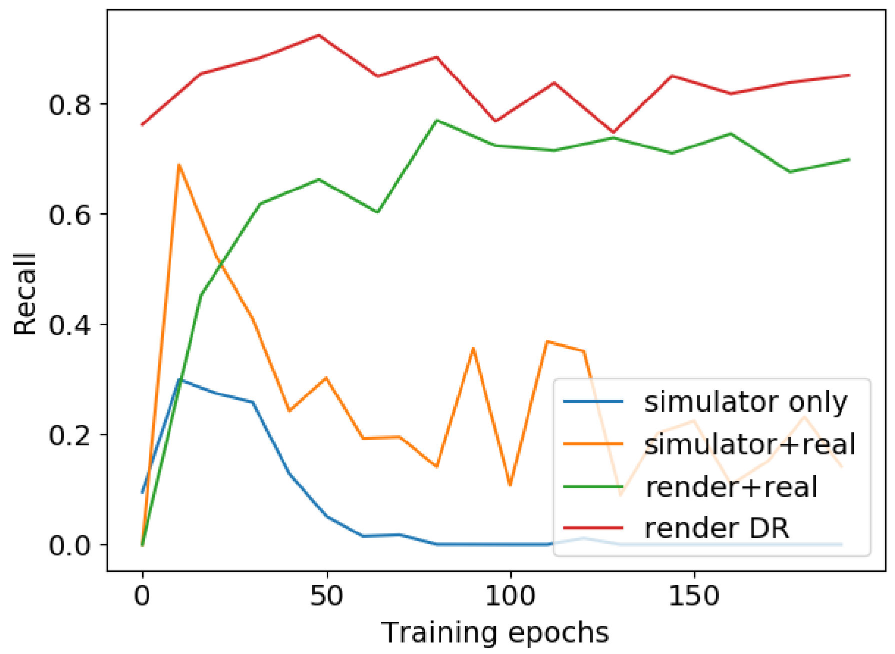


SIM-REAL COMPOSITION

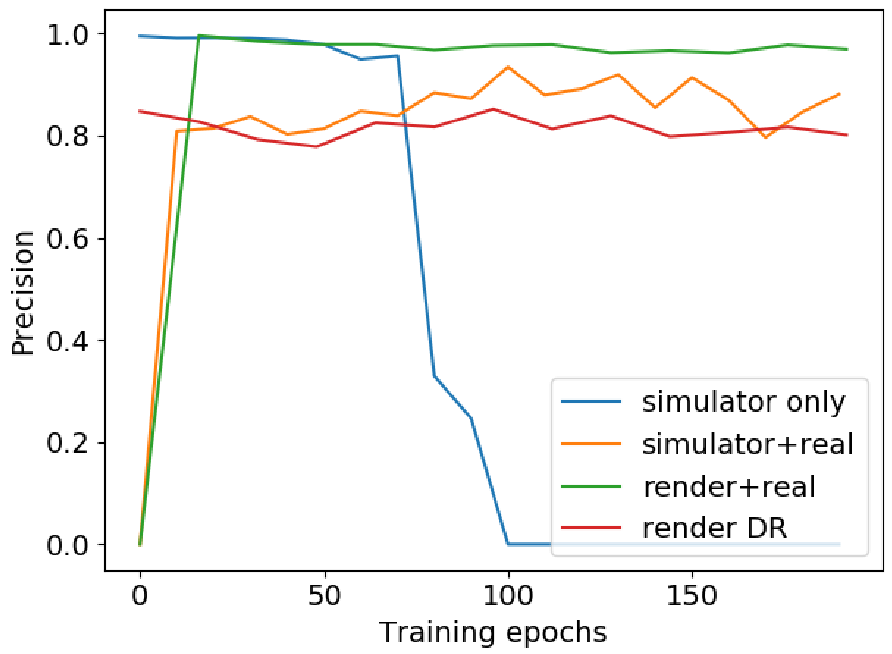


VISION MODULE RESULTS

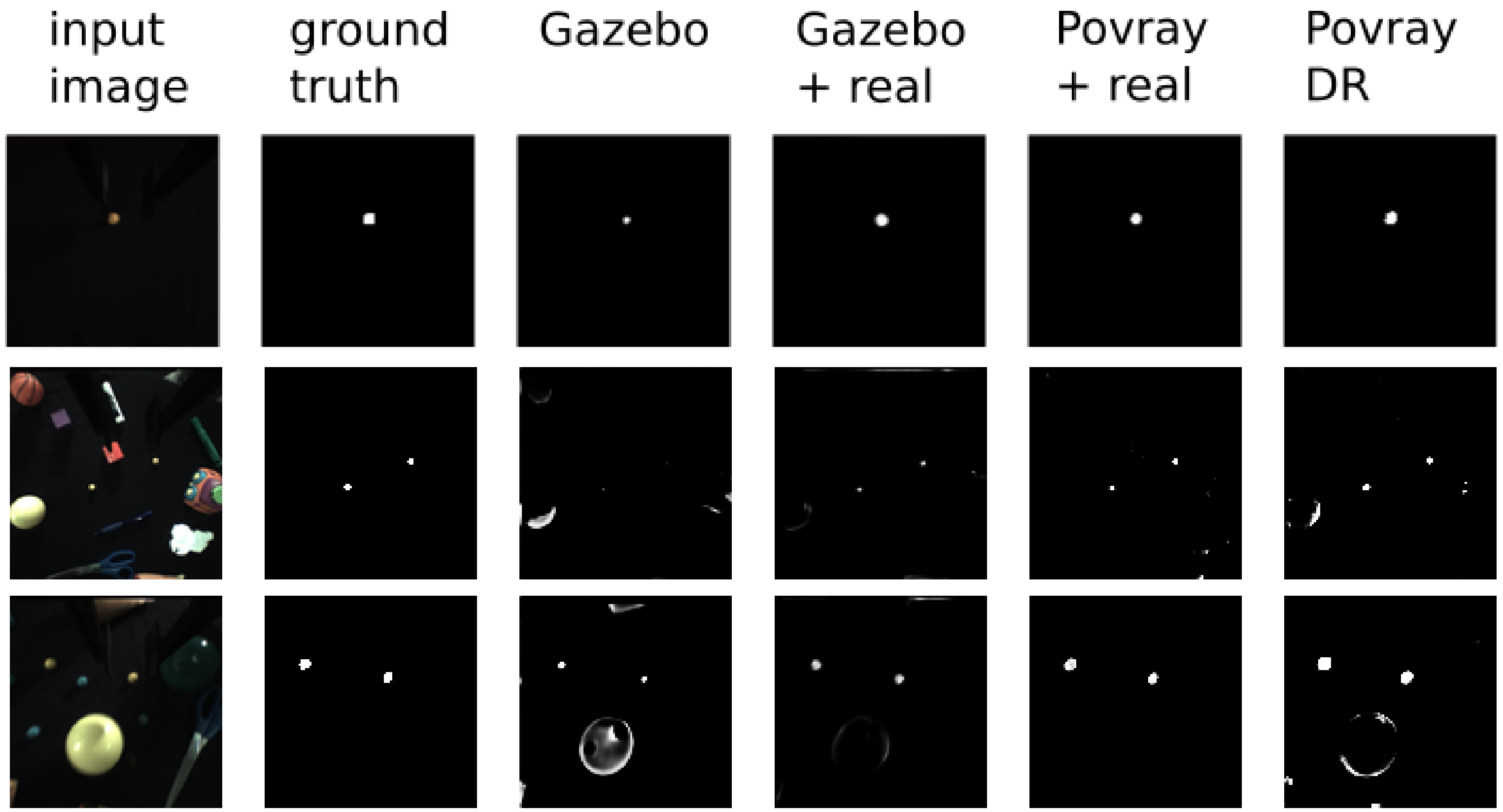
TP / (TP + FN)
% of sphere pixels that are captured



TP / (TP + FP)
% of sphere pixels that are real sphere pixels



VISION MODULE RESULTS



CONCLUSION

90% success in grasping a tiny sphere

Modular network for increased interpretability and flexible training

Imitation learning (DAGGER) for data-efficient controller learning

Sim-real composition as a form of randomization for visual transfer

RESOURCES

THEORY

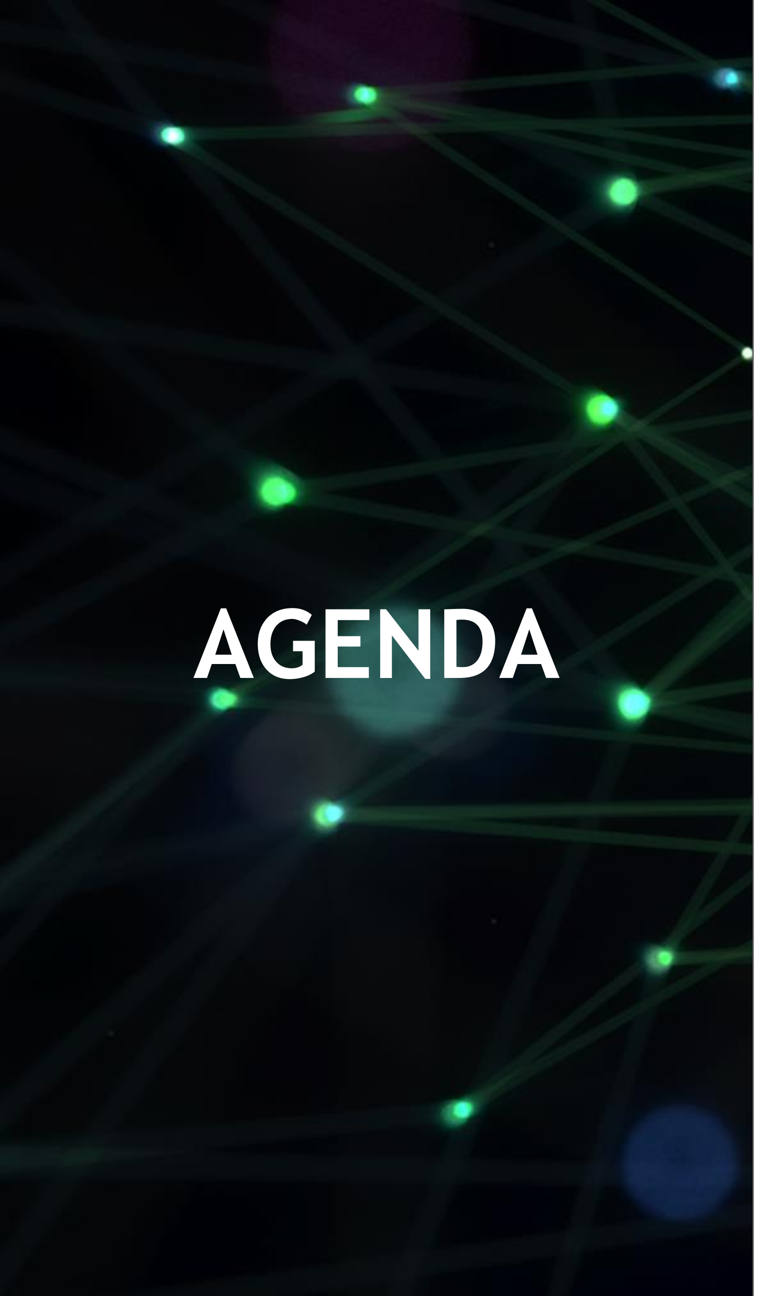
Mengyuan Yan, Iuri Frosio, Stephen Tyree, Jan Kautz, Sim-to-Real Transfer of Accurate Grasping with Eye-In-Hand Observations and Continuous Control, Neural Information Processing Systems (NIPS) 2017 Workshop on Acting and Interacting in the Real World: Challenges in Robot Learning, https://arxiv.org/abs/1712.03303.

CODING

DIY ☺

Gazebo: http://gazebosim.org/

ISAAC: https://www.nvidia.com/en-us/deep-learning-ai/industries/robotics/



AGENDA

FOUR LESSONS

- ✓ Balancing computational resources at system level
- ✓ Move simulation to GPU, create new algorithms
- ✓ Imitation learning & sample efficiency
- ✓ Domain transfer

ACKNOWLEDGEMENTS

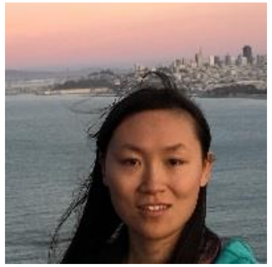
Stephen Tyree
NVIDIA

Jan Kautz
NVIDIA

Mohammad Babaizadeh
NVIDIA / UIUC

Jason Clemons
NVIDIA

Jeannette Bohg
Stanford


Mengyuan Yan
NVIDIA / Stanford

Steven Dalton
NVIDIA

Jarod Hoberock
NVIDIA

LINKS: REFERENCES

NVIDIA Research homepage: https://www.nvidia.com/en-us/research/

Mohammad Babaeizadeh, Iuri Frosio, Stephen Tyree, Jason Clemons, Jan Kautz, Reinforcement Learning through Asynchronous Advantage Actor-Critic on a GPU, ICLR 2017, https://openreview.net/forum?id=r1VGvBcxl.

GA3C on github: https://github.com/NVlabs/GA3C

S. Dalton, I. Frosio, J. Hoberock, J. Clemons, CULE: GPU ACCELERATED RL, GTC 2018, http://on-demand.gputechconf.com/gtc/2018/presentation/s8440-cule-a-companion-library-for-accelerated-rl-training.pdf.

CuLE on github (released soon): https://github.com/NVlabs/.

Mengyuan Yan, Iuri Frosio, Stephen Tyree, Jan Kautz, Sim-to-Real Transfer of Accurate Grasping with Eye-In-Hand Observations and Continuous Control, Neural Information Processing Systems (NIPS) 2017 Workshop on Acting and Interacting in the Real World: Challenges in Robot Learning, https://arxiv.org/abs/1712.03303.

LINKS: ISAAC

https://nvidianews.nvidia.com/news/nvidia-isaac-launches-new-era-of-autonomous-machines

Jetson Xavier:

… At the heart of NVIDIA Isaac is Jetson™ Xavier™, the world's first computer designed specifically for robotics…

Isaac Robotics Software:

Isaac SDK – a collection of APIs and tools to develop robotics algorithm software and runtime framework with fully accelerated libraries.

Isaac IMX – Isaac Intelligent Machine Acceleration applications, a collection of NVIDIA-developed robotics algorithm software.

Isaac Sim – a highly realistic virtual simulation environment for developers to train autonomous machines and perform hardware-in-the-loop testing with Jetson Xavier.

LINKS: INTERNSHIP

https://www.nvidia.com/en-us/research/internships/

Apply here: http://www.nvidia.com/object/universityrecruiting-internships.html

Or more easily, send me your CV to ifrosio@nvidia.com!

Typical length: 3/4 months in Silicon Valley

Typical outcome: 1 publication, 1 patent

Typical team: 1 intern, 1 mentor (senior research scientist), several co-mentors

LINKS: GRADUATE FELLOWSHIP

http://research.nvidia.com/graduate-fellowships

… $50,000 per award to Ph.D. students who are researching topics that will lead to major advances in the graphics and high-performance computing industries, and are investigating innovative ways of leveraging the power of the GPU. **NVIDIA particularly invites submissions from students pushing the envelope in artificial intelligence, deep neural networks, autonomous vehicles, and related fields…**

… Students must have already completed their first year of Ph.D. level studies (at the time of application)…

… Students must have majors in Computer Science, Computer Engineering, System Architecture, Electrical Engineering, or a related area…

LINKS: “FREE” GPUS & TEACHING MATERIAL

“Free” GPUs for researchers (not really free, but it is easy to get one):
https://developer.nvidia.com/academic_gpu_seeding

Free teaching material and GPU cloud resources through the Deep Learning Institute (Deep Learning, Accelerated Computing, & Robotics): https://developer.nvidia.com/teaching-kits