FUJITSU

shaping tomorrow with you

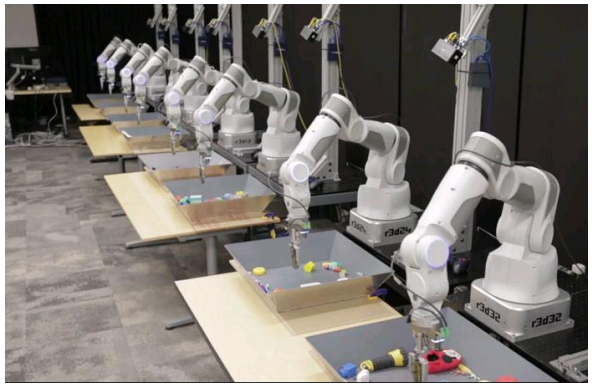
GUNREAL: GPU-accelerated UNsupervised REinforcement and Auxiliary Learning

Koichi Shirahata, Youri Coppens, Takuya Fukagai, Yasumoto Tomita, and Atsushi Ike
FUJITSU LABORATORIES LTD.
March 27, 2018

Deep Reinforcement Learning on GPUs

FUJITSU

- Deep Learning has led to tremendous advances in image, speech, and natural language recognitions
  - Deep learning benefits from GPU acceleration, by parallelizing the matrix computations of the deep neural network (DNN) units
- Deep Reinforcement Learning has been applied on decision-making tasks and control tasks such as robotics, games, and HVAC (heating, ventilation, and air conditioning)
- Frequent interactions between the environment and DNN makes Deep RL slow and unstable
  - Deep Q-Network (DQN) [Mnih et al. 2015] stabilized the DNN training on GPU by introducing an experience replay memory buffer and mini batches



State-of-the-art Deep RL Algorithms

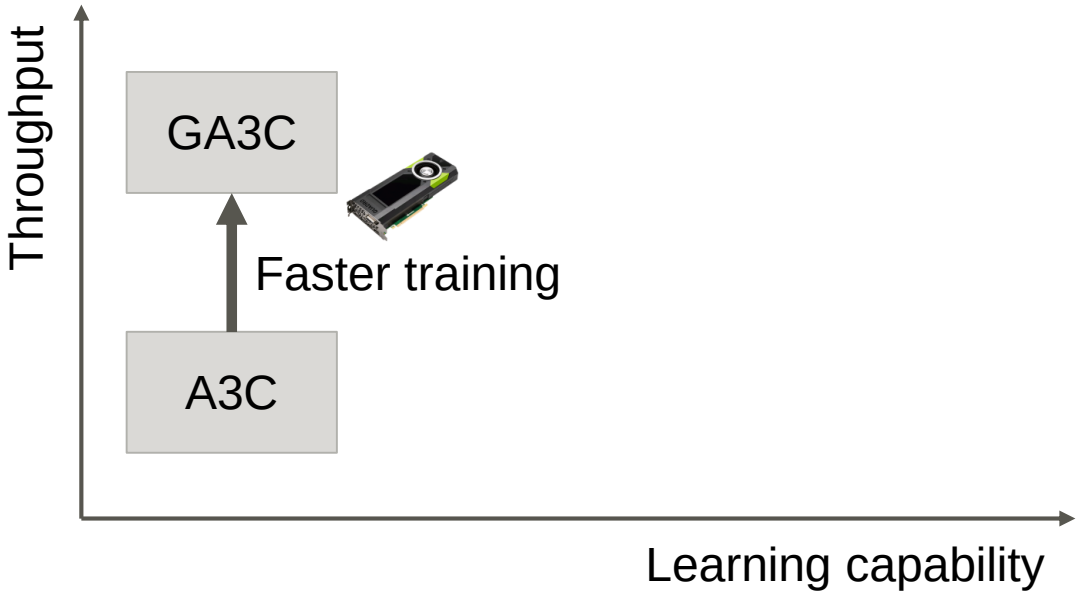
FUJITSU

- Asynchronous Advantage Actor-Critic (A3C) [Mnih et al. 2016] focused on concurrent simulations on single machines with only CPUs
  - Interactions become overhead when using GPU due to small batches

State-of-the-art Deep RL Algorithms

FUJITSU

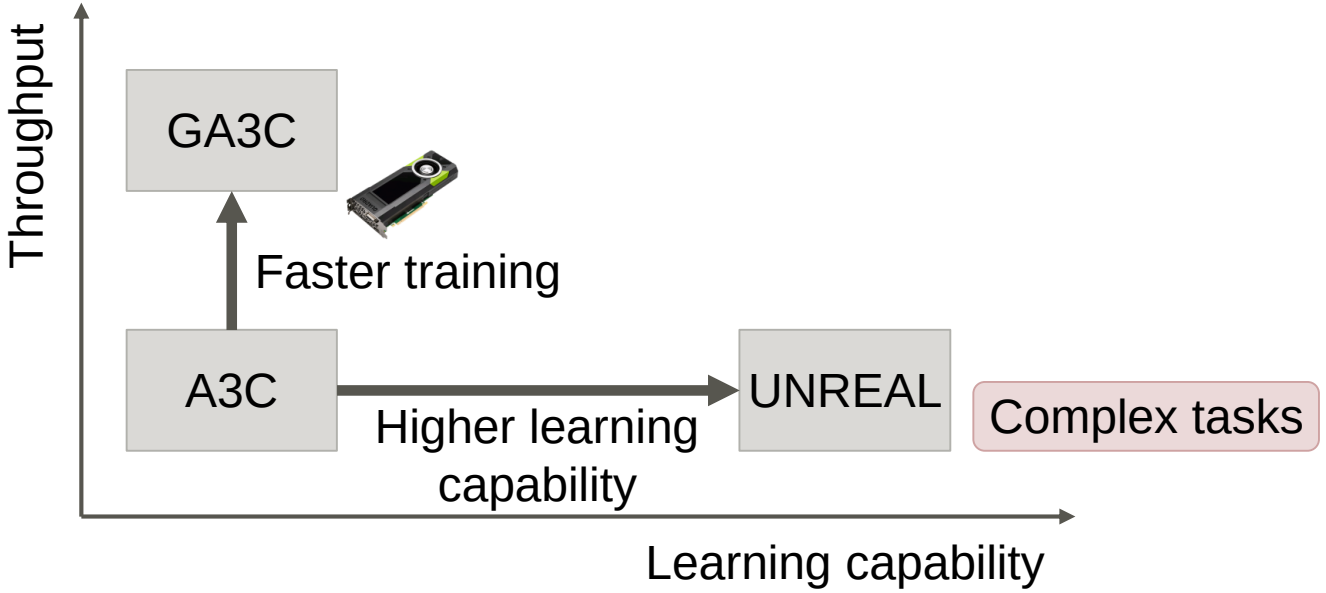
- Asynchronous Advantage Actor-Critic (A3C) [Mnih et al. 2016] focused on concurrent simulations on single machines with only CPUs
  - Interactions become overhead when using GPU due to small batches
- GA3C [Babaeizadeh et al. 2017] modified A3C to benefit better from GPU
  - Run concurrent simulations on the CPU, and sends larger batches to the GPU
  - Learning capability is not enough for complex tasks using 3D environments



State-of-the-art Deep RL Algorithms

FUJITSU

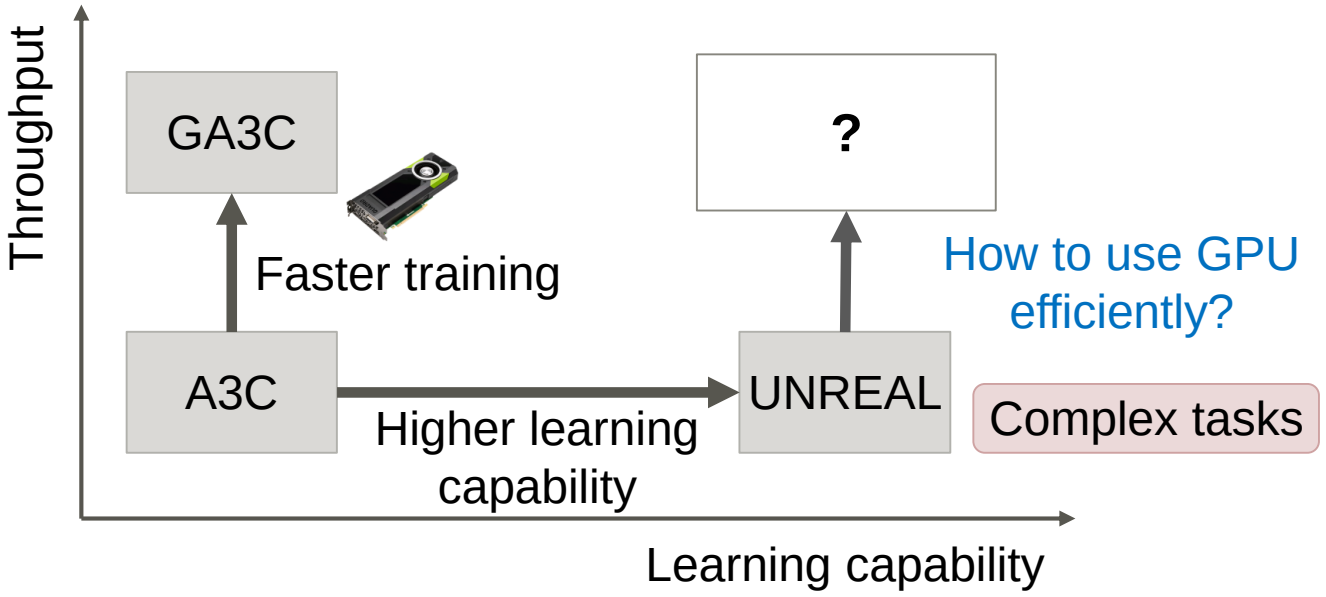
- Asynchronous Advantage Actor-Critic (A3C) [Mnih et al. 2016] focused on concurrent simulations on single machines with only CPUs
  - Interactions become overhead when using GPU due to small batches
- GA3C [Babaeizadeh et al. 2017] modified A3C to benefit better from GPU
  - Run concurrent simulations on the CPU, and sends larger batches to the GPU
  - Learning capability is not enough for complex tasks using 3D environments
- UNREAL [Jaderberg et al. 2017] extended A3C with auxiliary tasks
  - Training the auxiliary tasks guides a part of the main DNN model



Problems on Fast and Efficient Deep RL

FUJITSU

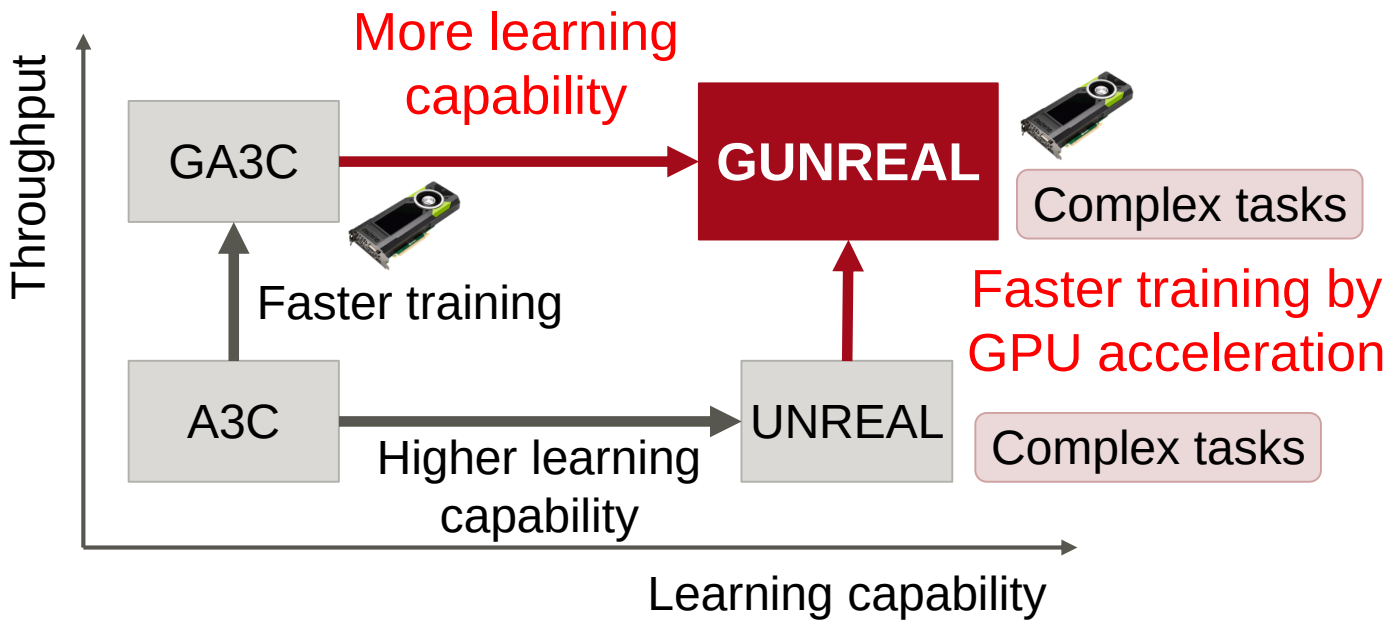
- Ideally, we would like to have an algorithm that can run fast (GPU efficiency) and has also high learning capability (ability to learn complex environments)
- How to use GPU efficiently for the auxiliary tasks in UNREAL?
  - Frequent interactions between the environment and DNN for both main task and auxiliary tasks
  - Realizing both high training speed and learning capability at the same time



GUNREAL: GPU-accelerated Unsupervised REinforcement and Auxiliary Learning

FUJITSU

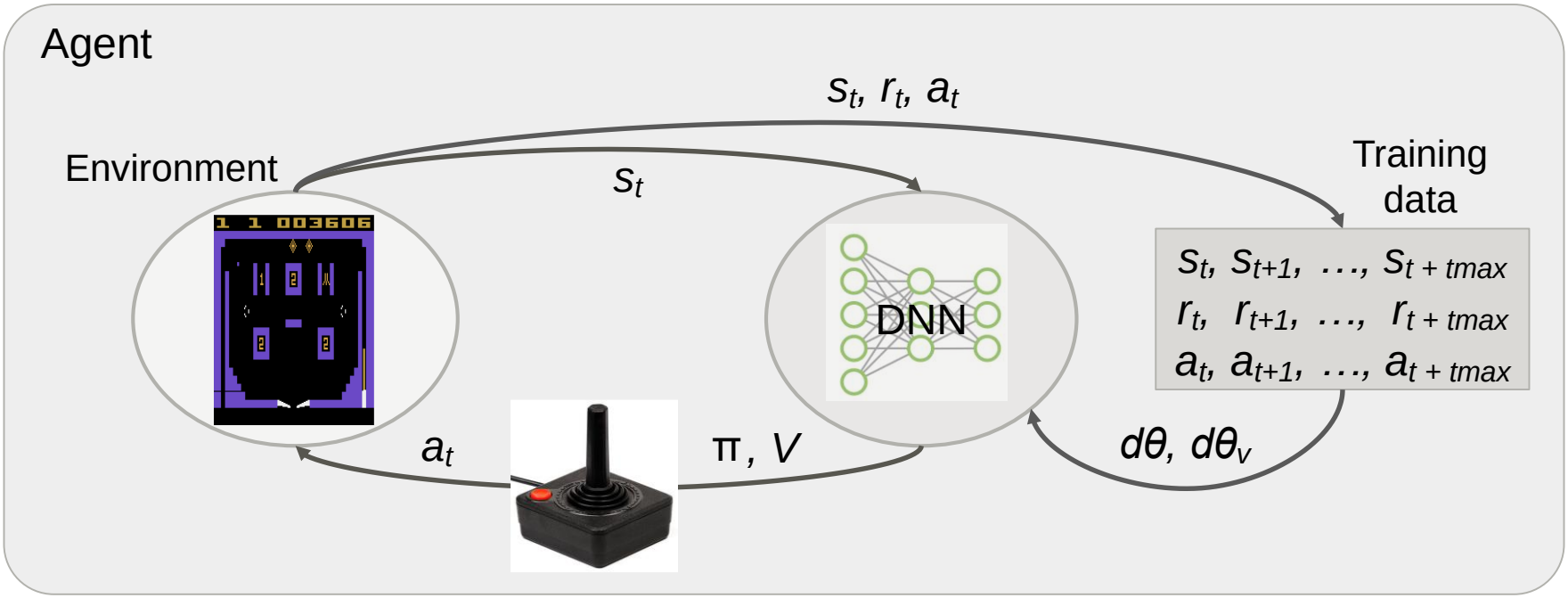
- Proposal: GUNREAL (GPU-accelerated UNREAL)
  - We extend the GA3C algorithm to learn auxiliary tasks
  - We improve the throughput on the GPU by adding a preprocessing phase and changing the data format
- Results
  - GUNREAL learned 4.0x faster than UNREAL
  - GUNREAL learned a 3D environment 73% more efficiently than GA3C



Deep Reinforcement Learning

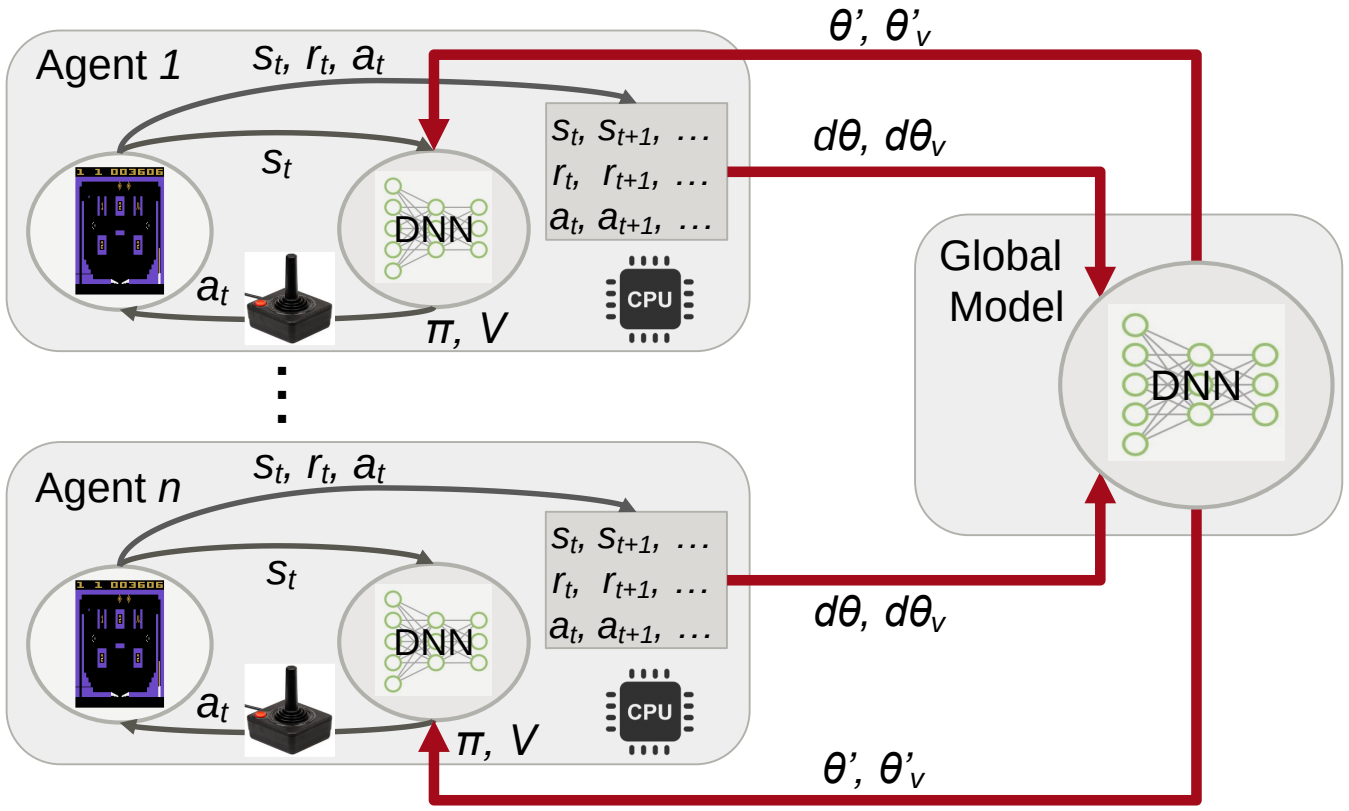
FUJITSU

- Reinforcement Learning (RL): teaching agents control tasks through iterations with the environment
- Deep RL = Using DNNs (DL) to learn the agents control tasks (RL)
- Training data is generated through the interactions with the environment
  - Difficult to parallelize the DNN computations using mini batches



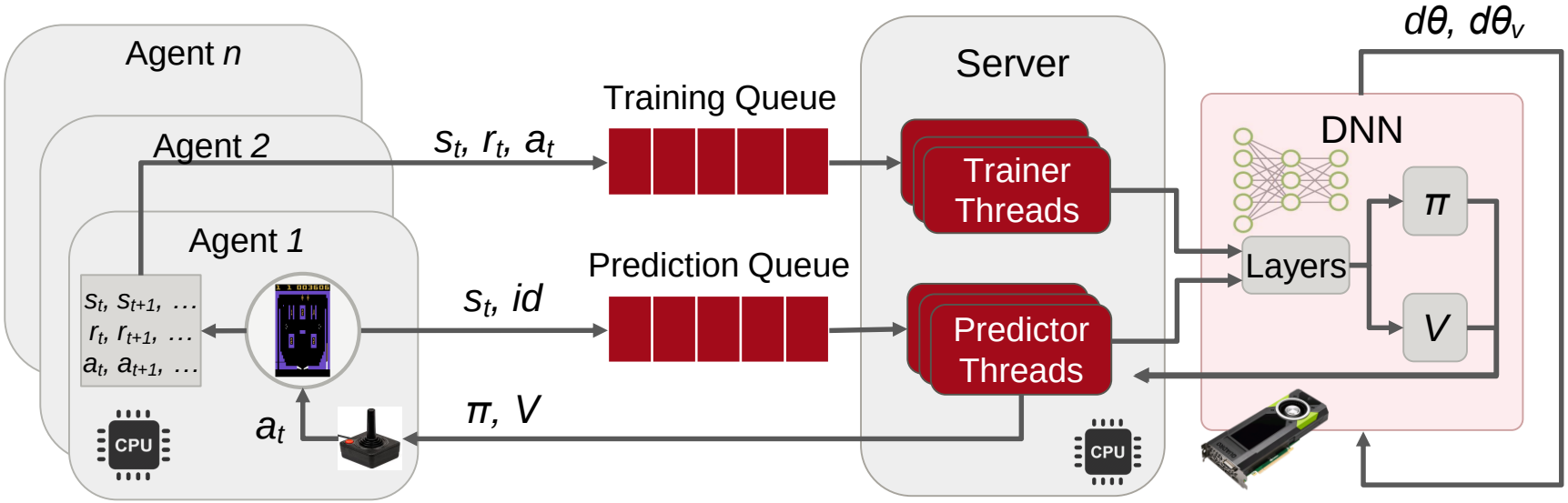
A3C

- **Asynchronous Advantage Actor-Critic** by DeepMind [Mnih et al. 2016]
- Multiple agents simulate in parallel to train one global DNN
- Targets single machines with CPUs
- Interactions become overhead when using GPU due to small batches



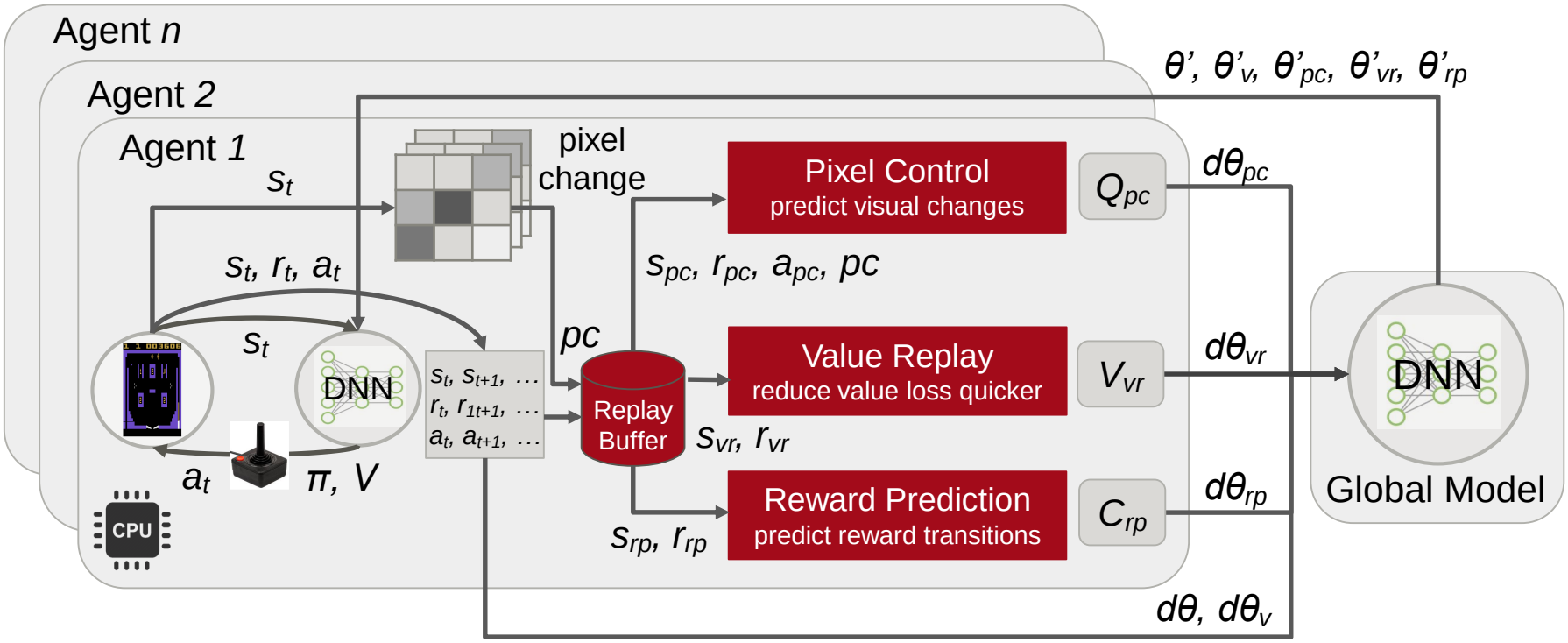
GA3C

- Developed by NVIDIA Research Labs [Babaeizadeh et al. 2017]
- Only one global DNN on the GPU
  - No local DNNs = no synchronization
- Dynamic producer-consumer architecture to utilize GPU better
  - Introduce queues for prediction and training
- Learning capability is not enough for complex tasks (e.g. 3D environments)



UNREAL

- **UN**supervised **RE**inforcement and **A**uxiliary **L**earning by DeepMind [Jaderberg et al. 2017]
- A3C + three auxiliary tasks
  - Pixel Control (PC), Value Replay (VR), Reward Prediction (RP)
  - Higher learning capability to learn more complex tasks
- More training time per step to process the auxiliary tasks

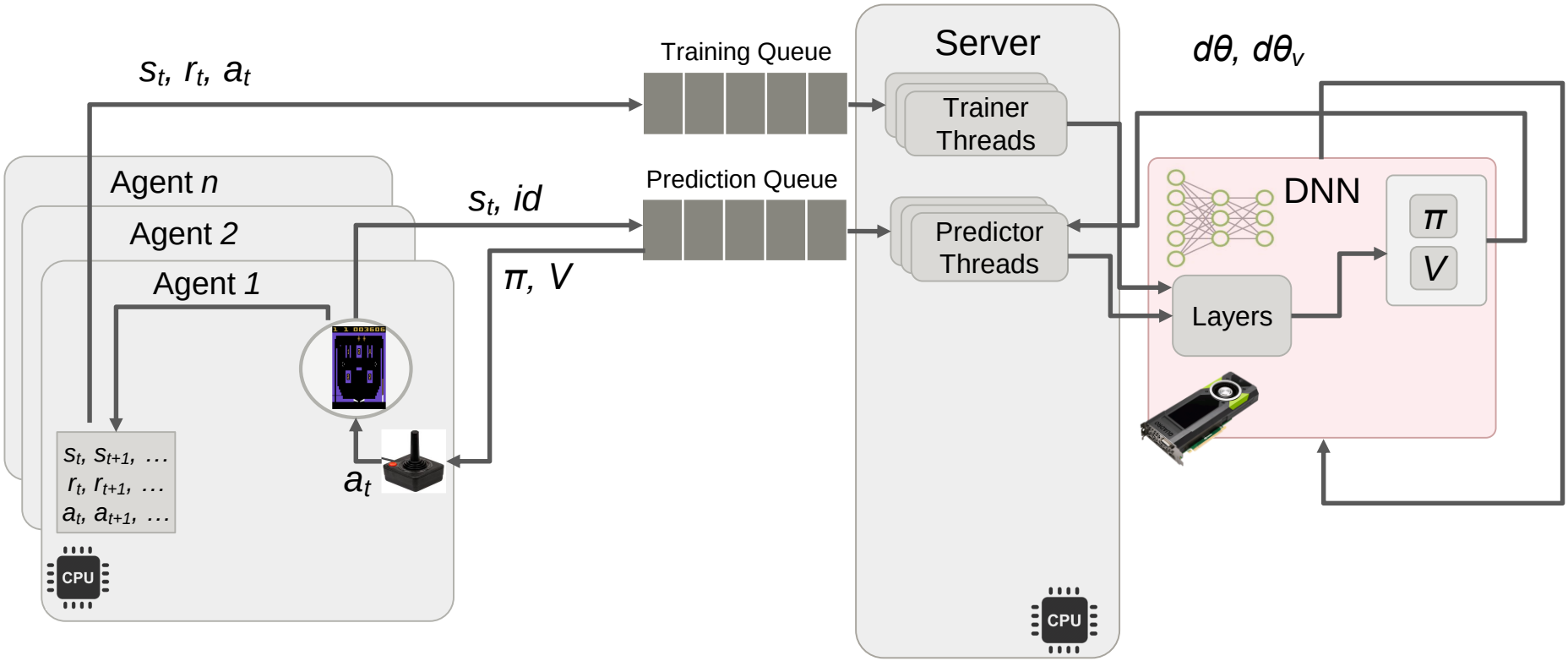


FUJITSU

GA3C + Auxiliary Tasks = GUNREAL

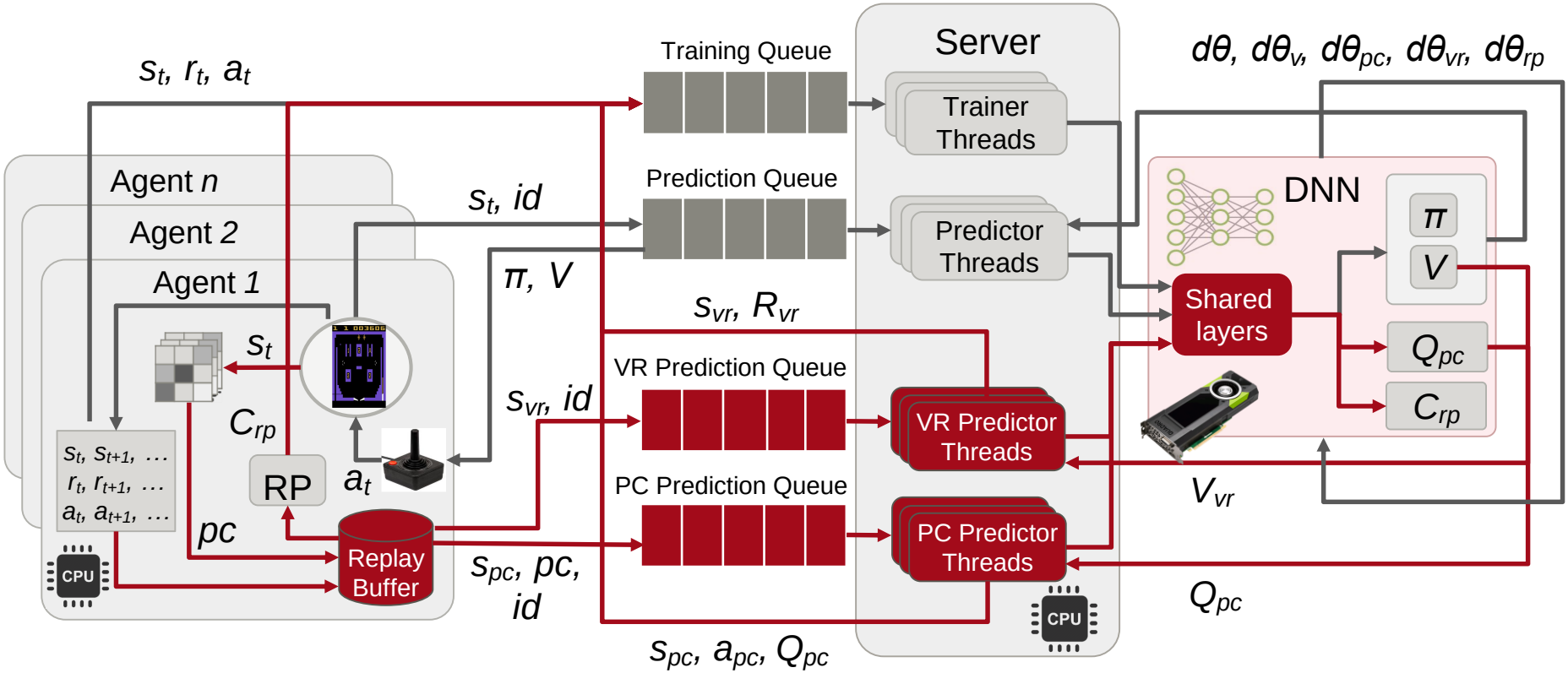
GUNREAL

- GA3C's prediction architecture for main A3C

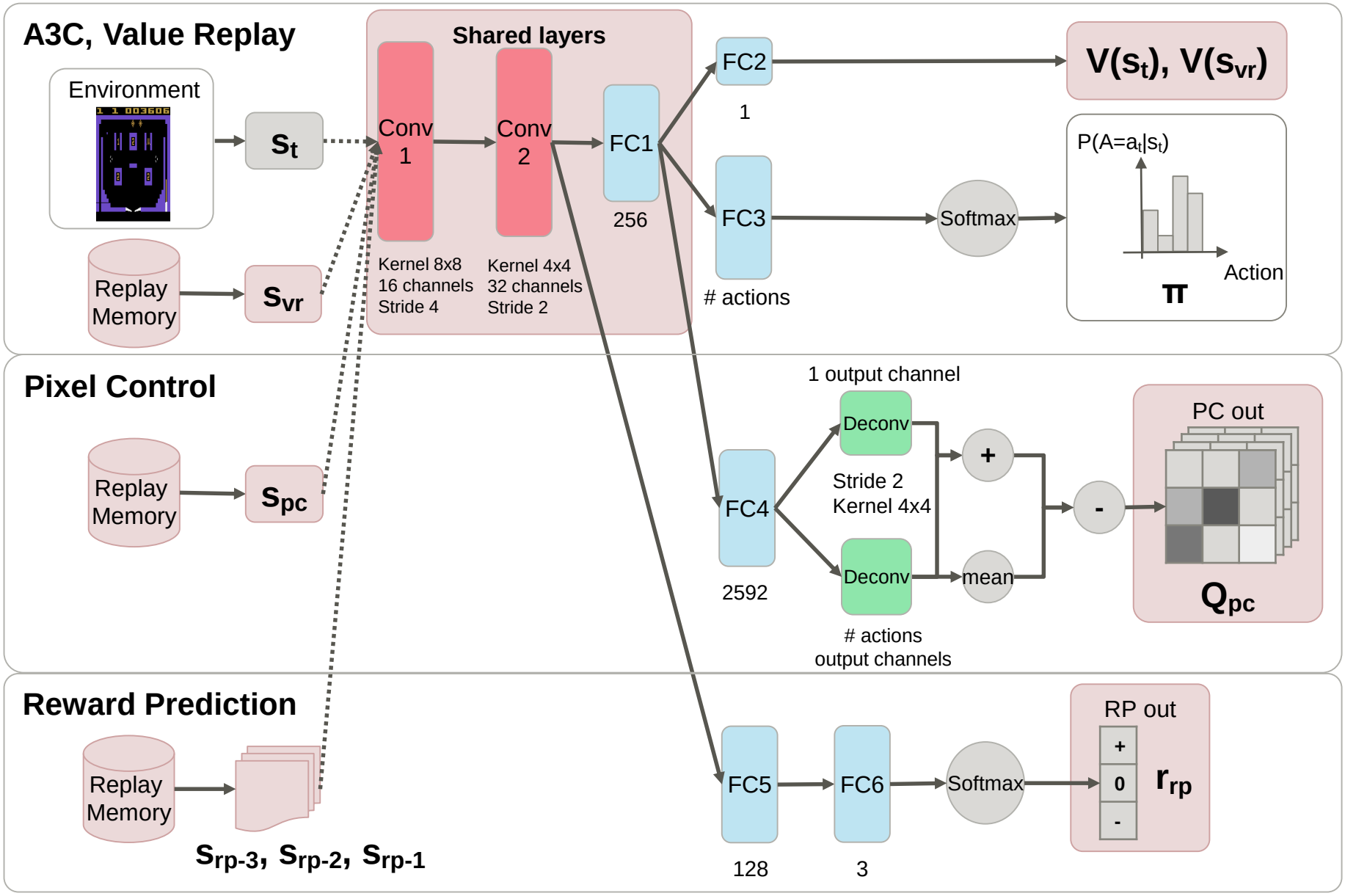


GUNREAL

- GA3C's prediction architecture for main A3C
- Two extra prediction lines for VR and PC training batch generation
  - More training data sent to the trainers
- Extending the DNN



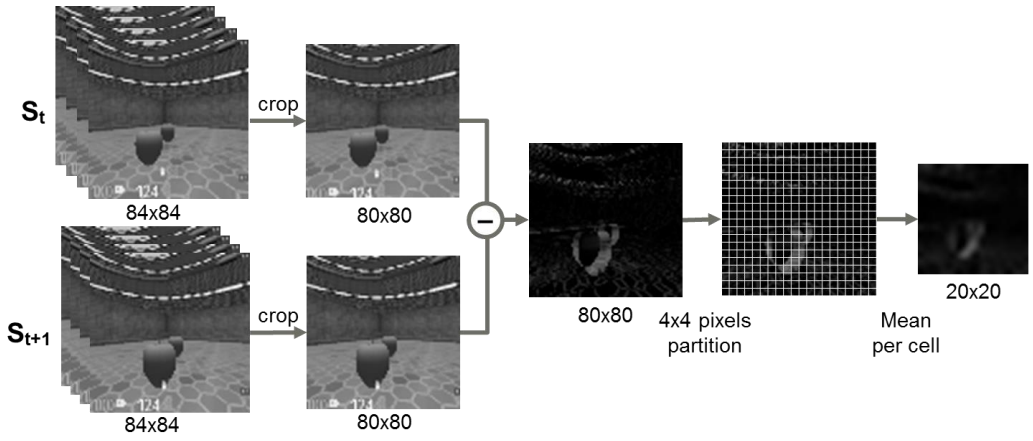
GUNREAL's DNN Structure



Preprocessing the environment observations

- (G)A3C has a preprocessing phase, collecting four sequential gray-scale frames
  - Instead, UNREAL uses single RGB frames and no preprocessing
- We modified the Pixel Control task for GUNREAL, since the preprocessing affects the calculation process for the PC task
- The preprocessed observations result in quicker learning for GUNREAL

Pixel Control task for GUNREAL



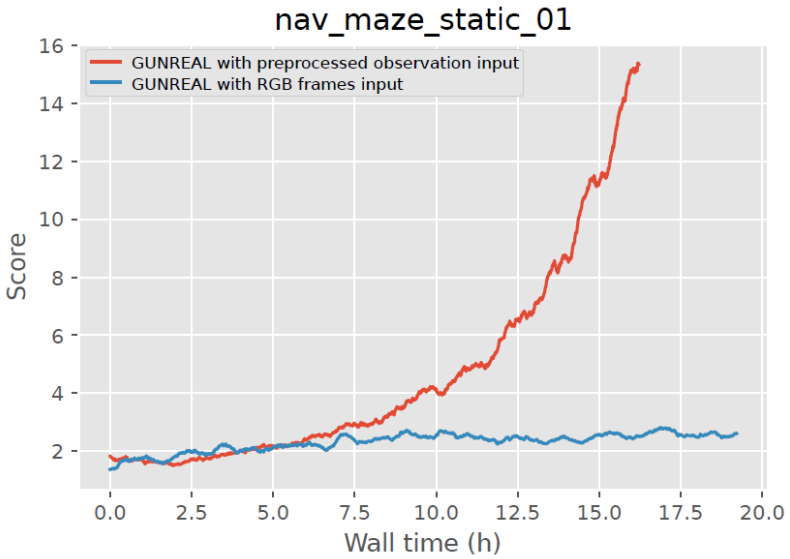
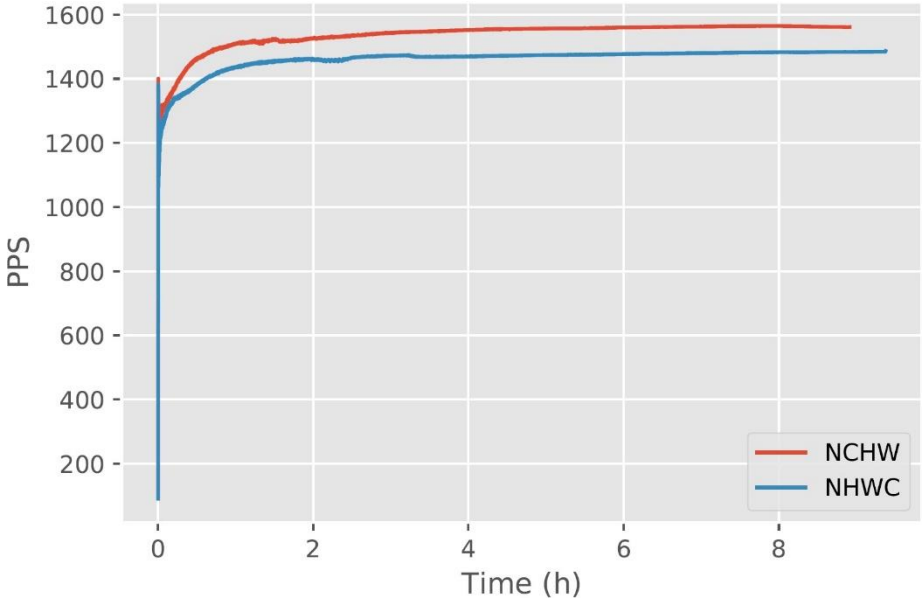


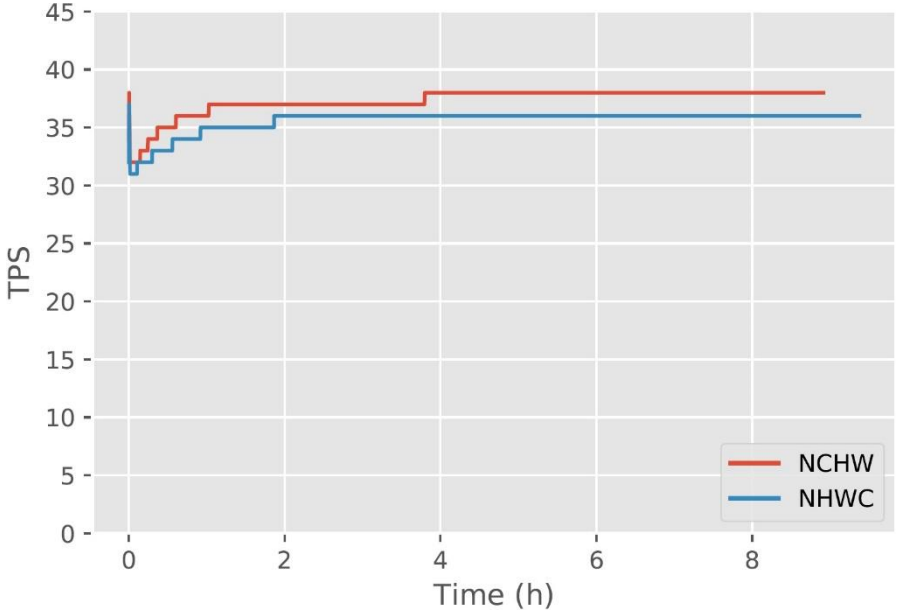
Image data format

- Two formats exist to process image batches
  - NHWC: CPU optimal (default in TensorFlow)
  - NCHW: cuDNN is optimized to process these
    - N: the number of images, C: channel size, H: image height, W: image width
- Getting more throughput by using NCHW batches in GUNREAL
  - 5% better in predictions per second (PPS) and trainings per second (TPS)

PPS (predictions per second)



TPS (trainings per second)



Experiments

FUJITSU

- Implementations
  - GUNREAL: we implemented using TensorFlow
  - GA3C: publicly available OSS from the authors using TensorFlow[1]
  - UNREAL: open-source replication using TensorFlow[2]
    - UNREAL has been enabled with GPU usage for DNN computations
- Environments
  - OpenAI Gym: Atari environment (a 2D game simulator)
  - DeepMind Lab: a 3D game simulator

DeepMind Lab

Environment	OpenAI Gym 0.8.0	DeepMind Lab
Processor	Intel Xeon E5-1650v3 (12 threads, 3.50 GHz)	Intel Xeon E3-1245v5 (8 threads, 3.50 GHz)
GPU (NVIDIA)	GeForce GTX Titan X	Quadro M4000
Python Interpreter	CPython 3.6.1	CPython 2.7.13
CUDA Version	CUDA 8 + cuDNN 6	CUDA 8 + cuDNN 5.1
Framework	TensorFlow 1.2	

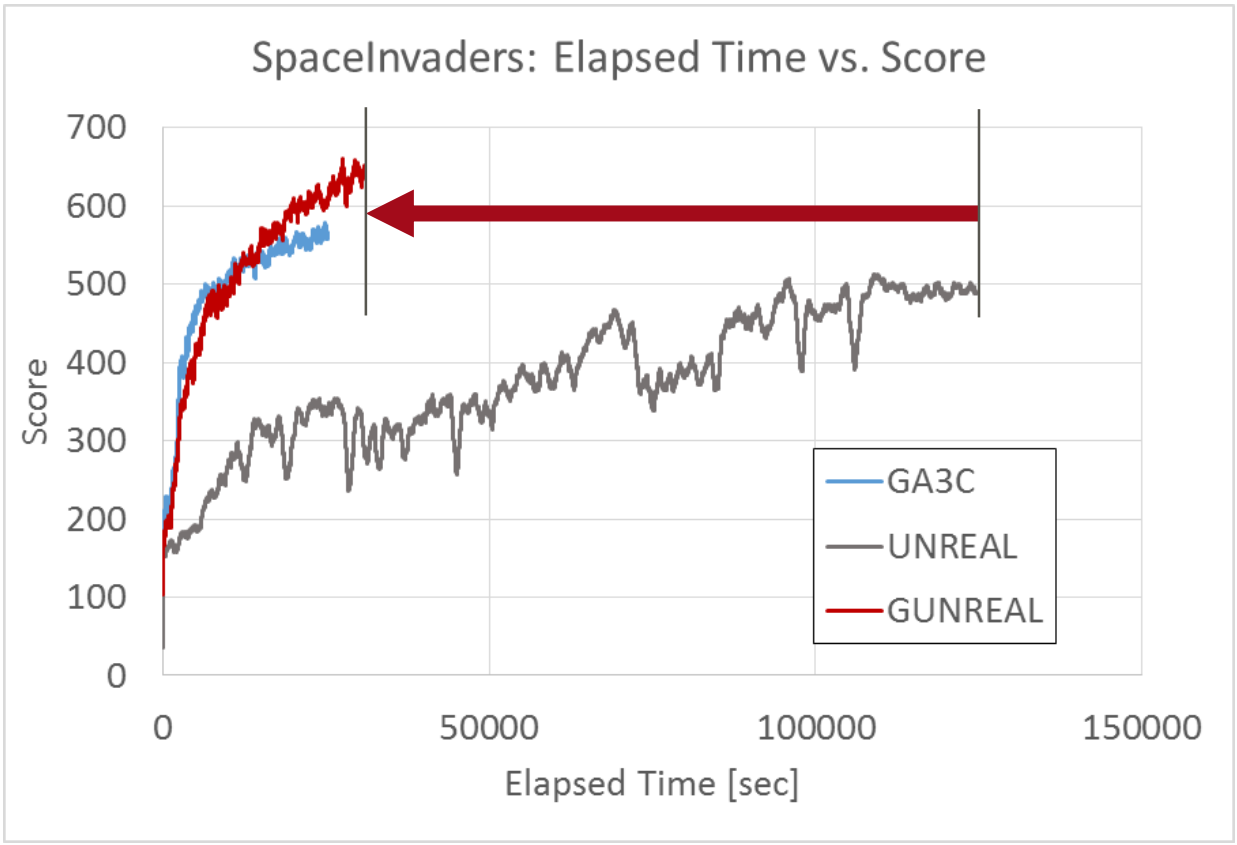
[1]https://github.com/NVlabs/GA3C, [2]https://github.com/miyosuda/unreal

Results: OpenAI Gym (2D simulator)

FUJITSU

- Space Invaders (Elapsed time vs. Score)
- GUNREAL learns faster than UNREAL in real-time

4.0x faster than UNREAL

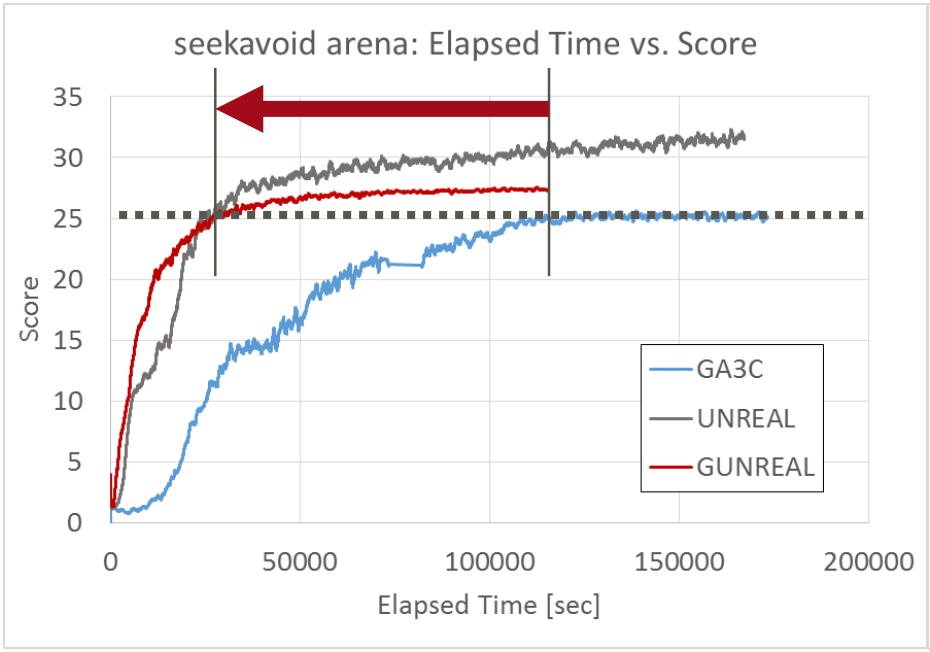


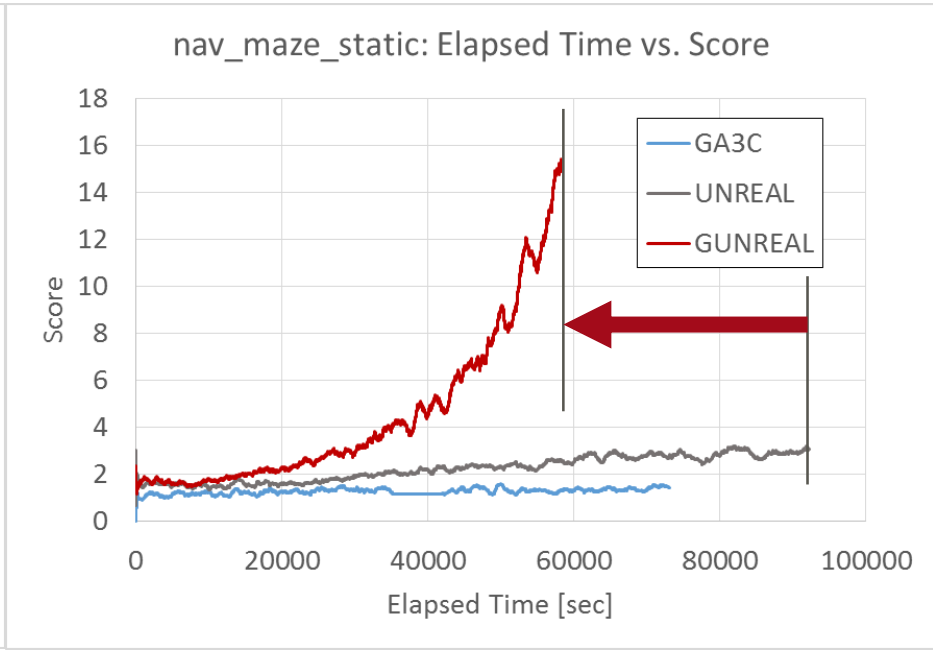
Results: DeepMind Lab (3D simulator)

- Seek Avoid Arena / Nav Maze (Elapsed time vs. Score)
- GUNREAL clearly beats GA3C on more complex tasks

4.27x faster than GA3C (73% less steps)

1.6x faster than UNREAL

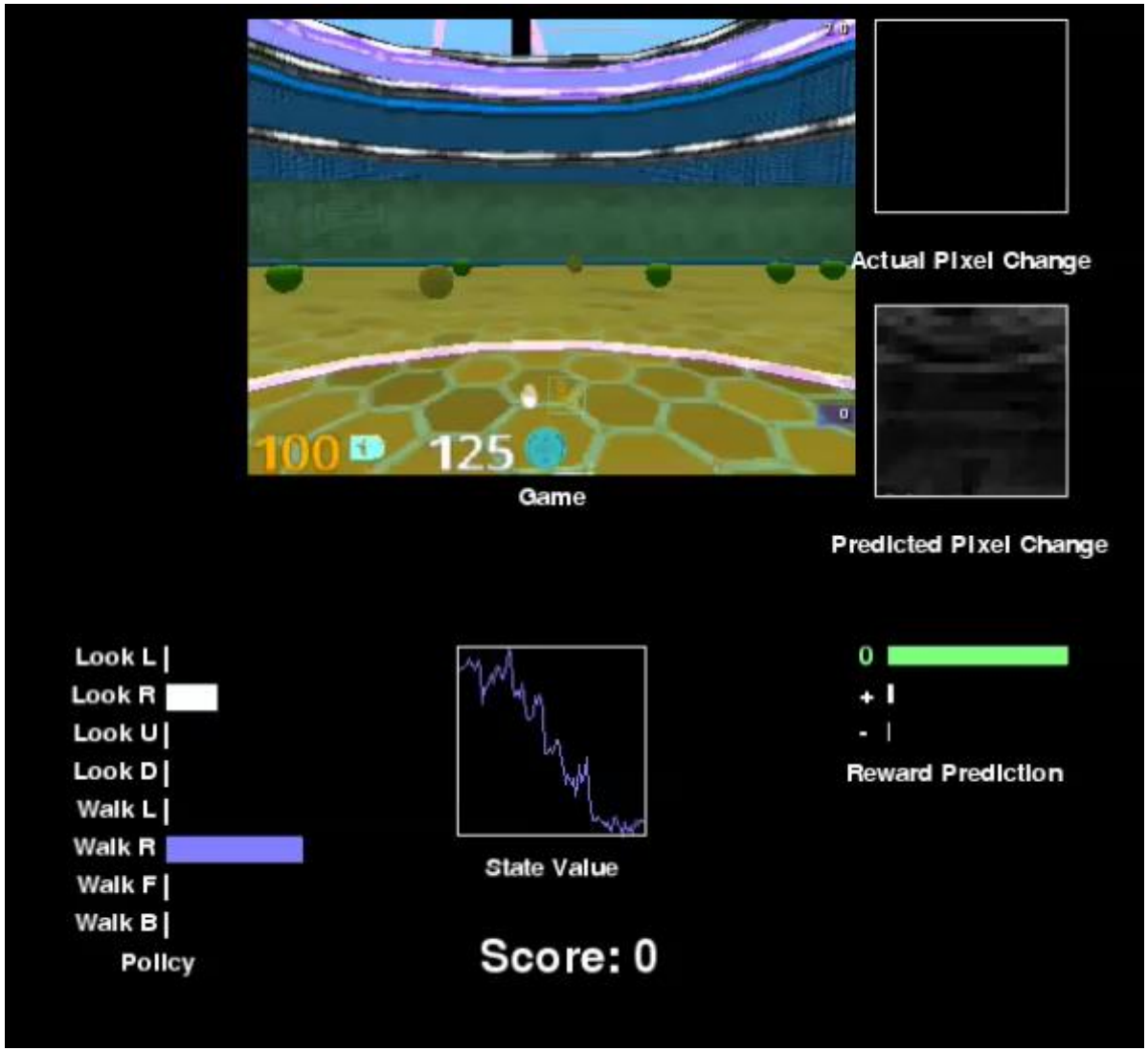




Demo: Seek Avoid Arena (DeepMind Lab)

FUJITSU

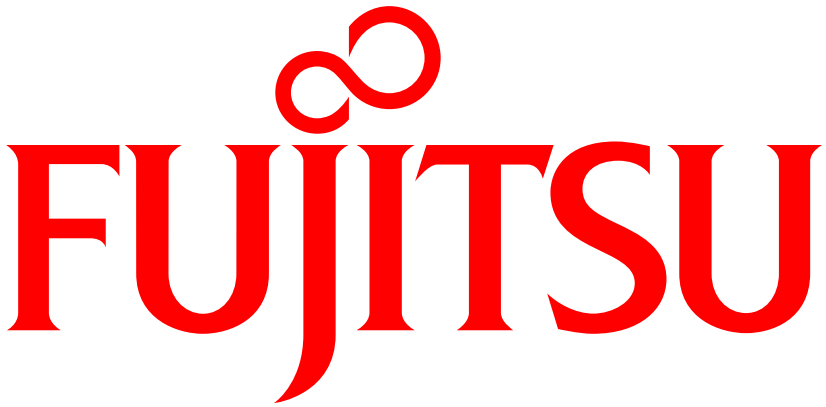
- Playing movie after training finished



Conclusions and Future Work

FUJITSU

- Conclusions
  - We developed GUNREAL, a fast Deep RL algorithm with high learning capability
    - An architecture that benefits from GPU acceleration by extending GA3C with UNREAL's auxiliary tasks
    - We evaluate the necessity of a preprocessing phase and improve the throughput on the GPU by changing the data format
  - GUNREAL learned 4.0x faster than UNREAL by better GPU acceleration
  - GUNREAL learned a 3D environment 73% more efficiently than GA3C
- Future work
  - Integration of an LSTM into GUNREAL's model
  - Assess the applicability of GUNREAL in real-world applications
  - Scalability study on cluster computing environment



shaping tomorrow with you

Related Work

FUJITSU

- GPU acceleration
  - **DQN** [Mnih et al. 2015] used GPU to train a DNN with a single agent, 12 to 14 days on Atari games
    - **Prioritized experience replay** [Schaul et al. 2016]
    - **Double Q-learning** [Hasselt et al. 2016]
    - **Dueling network architecture** [Wang et al. 2016]
  - **GA3C** [Babaeizadeh et al., 2017] improved multi-agent training, within a day on Atari games
  - **PAAC** [Clemente et al. 2017] is a framework for synchronously parallelized deep RL
- Cluster acceleration
  - **DistBelief** [Dean et al. 2012] has been used to train a DNN using tens of thousands of CPU cores
  - **Gorila** [Nair et al. 2015], a successor of DistBelief, distributed DQN across a cluster, outperformed standard DQN after training for 4 days