DETERMINISTIC IMPLEMENTATIONS FOR REPRODUCIBILITY IN DEEP REINFORCEMENT LEARNING

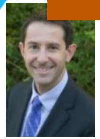
Prabhat Nagarajan[1], *Garrett Warnell*[2,3], *and Peter Stone*[2]

Preferred Networks, Tokyo, Japan

Department of Computer Science, The University of Texas at Austin

Computational and Information Sciences Directorate, US Army Research Laboratory

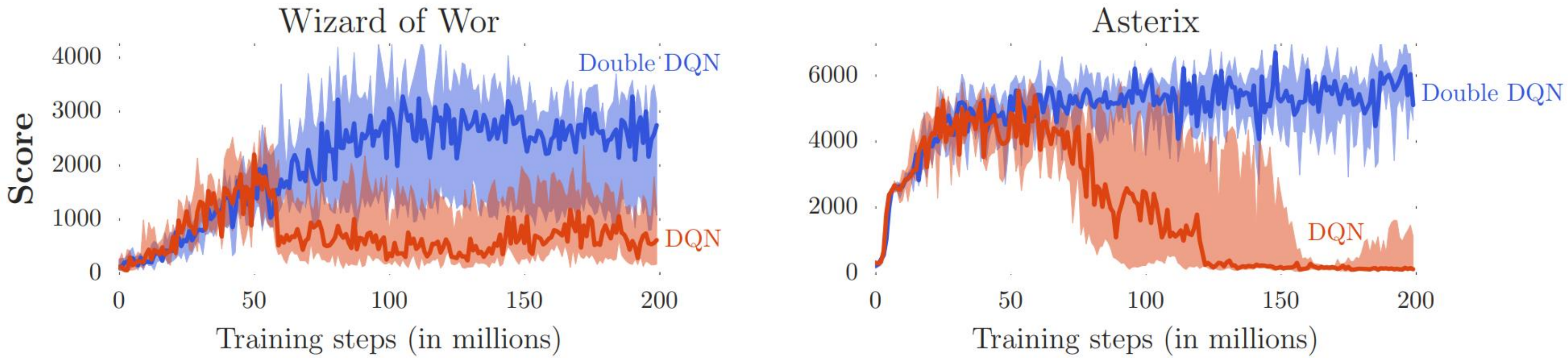
Garrett Warnell



Peter Stone

Reproducibility in Reinforcement Learning

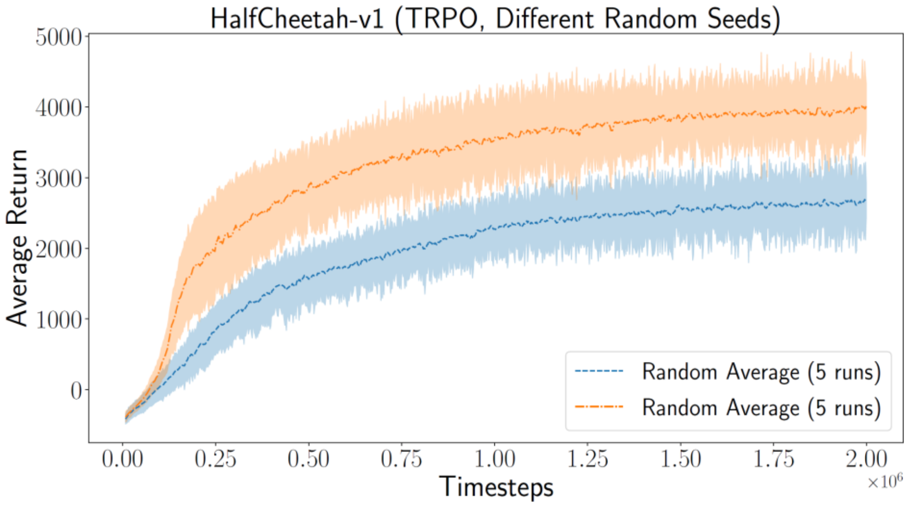
Instability ≠ Incorrect Implementation



Van Hasselt, Guez, Silver. AAAI. 2016

Reproducibility in Reinforcement Learning

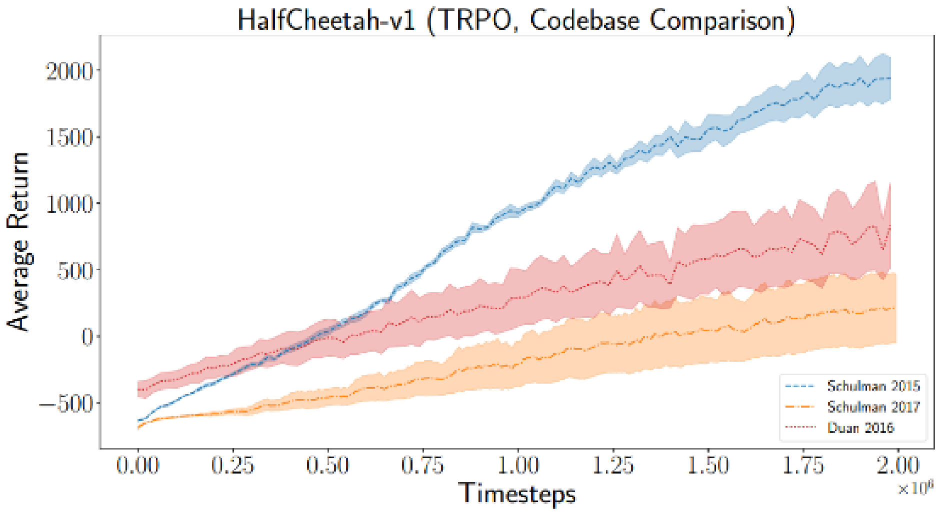
Intrinsic Variance



Henderson et al. AAAI. 2018

Reproducibility in Reinforcement Learning

Different Implementation = Different Performance



Henderson et al. AAAI. 2018

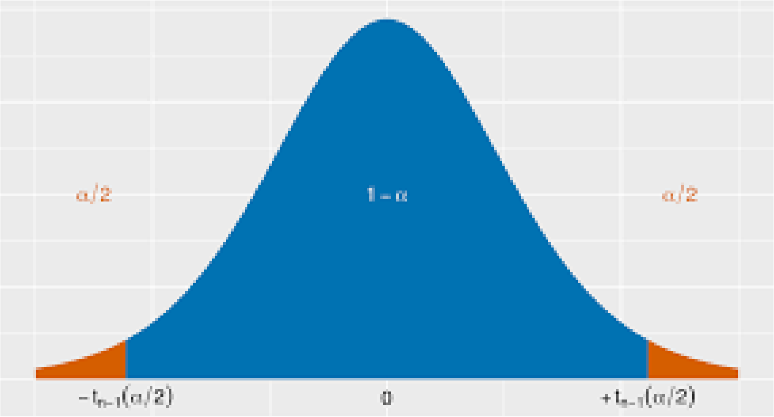
Other work

- Hausknecht & Stone. AAAI-2015 workshop on Learning for General Competency in Video Games. 2015
- Machado et al., JAIR. 2018
- Henderson, Romoff, & Pineau, EWRL. 2018.
- Islam, Henderson, Gomrokchi, and Precup. ICML RML. 2017

Preferred Networks

Reproducibility vs. Replicability

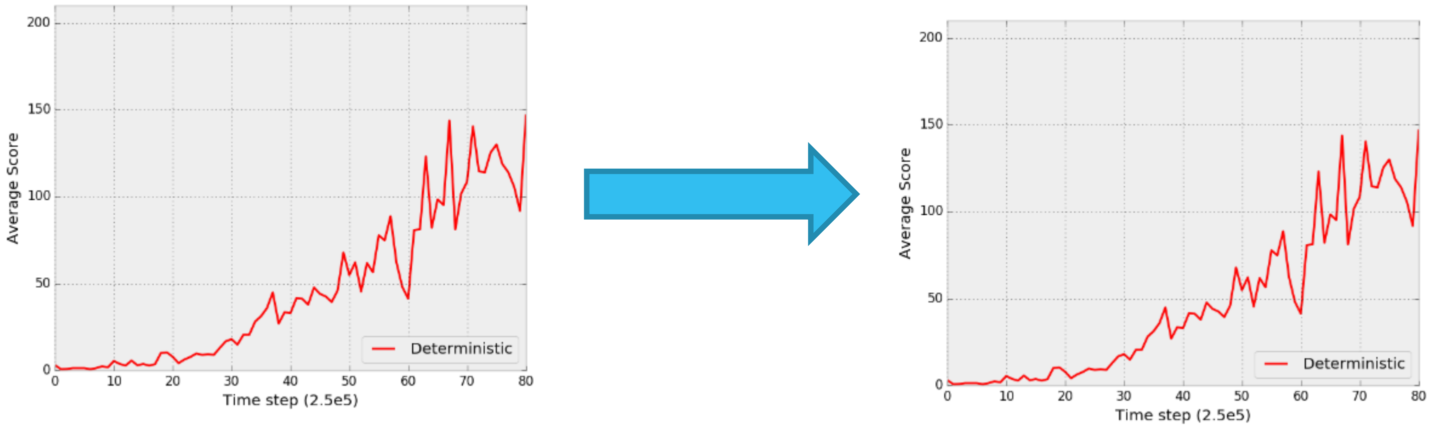
- **Reproducibility*** **-** the ability of an experiment to be repeated with minor differences from the original experiment, while achieving the same *qualitative* results.



* *Definitions inspired by Drummond, 2009.*

Reproducibility vs. Replicability

- **Replicability* -** the ability of an experiment to be repeated exactly, producing the same *quantitative* results.



* *Definitions inspired by Drummond, 2009.*

Replicable Experiment

Deterministic Implementation

a computer program that, when run under some fixed experimental conditions, will always produce identical outputs for a given input

Fixed Experimental Conditions

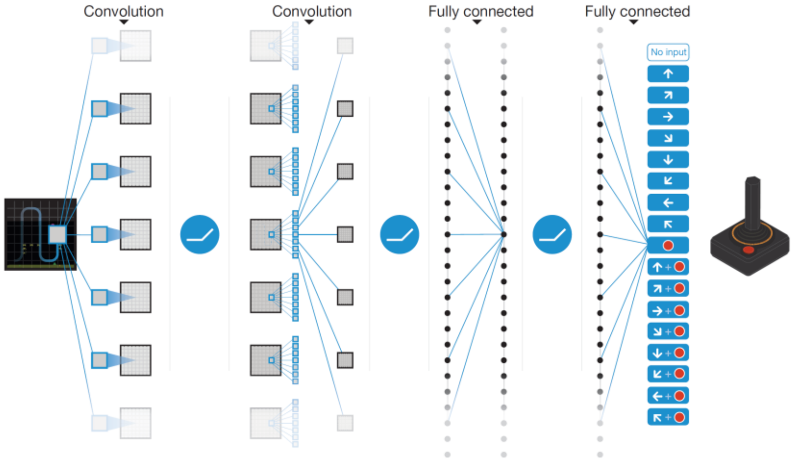
OS, Software Versions, GPU, Deep learning library, Dependencies

Motivation for Deterministic Implementations

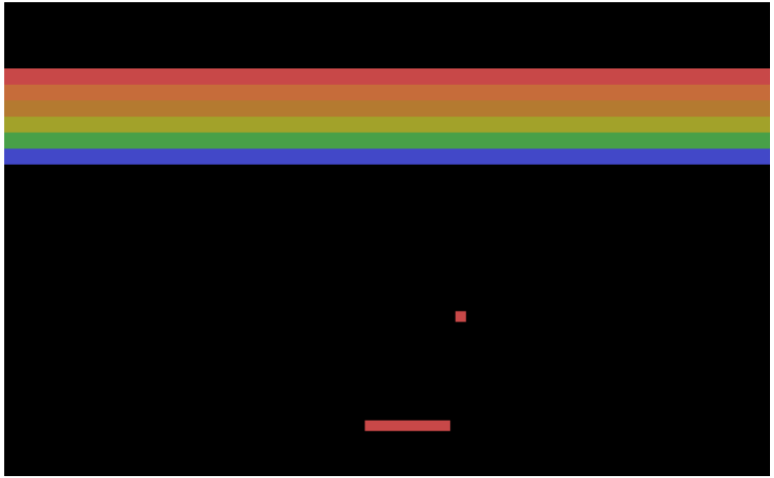
- Randomness and Implementation details
- Develop cleaner computational testing environments
- What benefits to we hope to achieve?
  - Debugging & Verification
  - Algorithm Comparisons
  - Ablation Studies
  - Can learn more about implementation details

Deterministic Implementations

- **Goal**: Develop cleaner computational testing environments

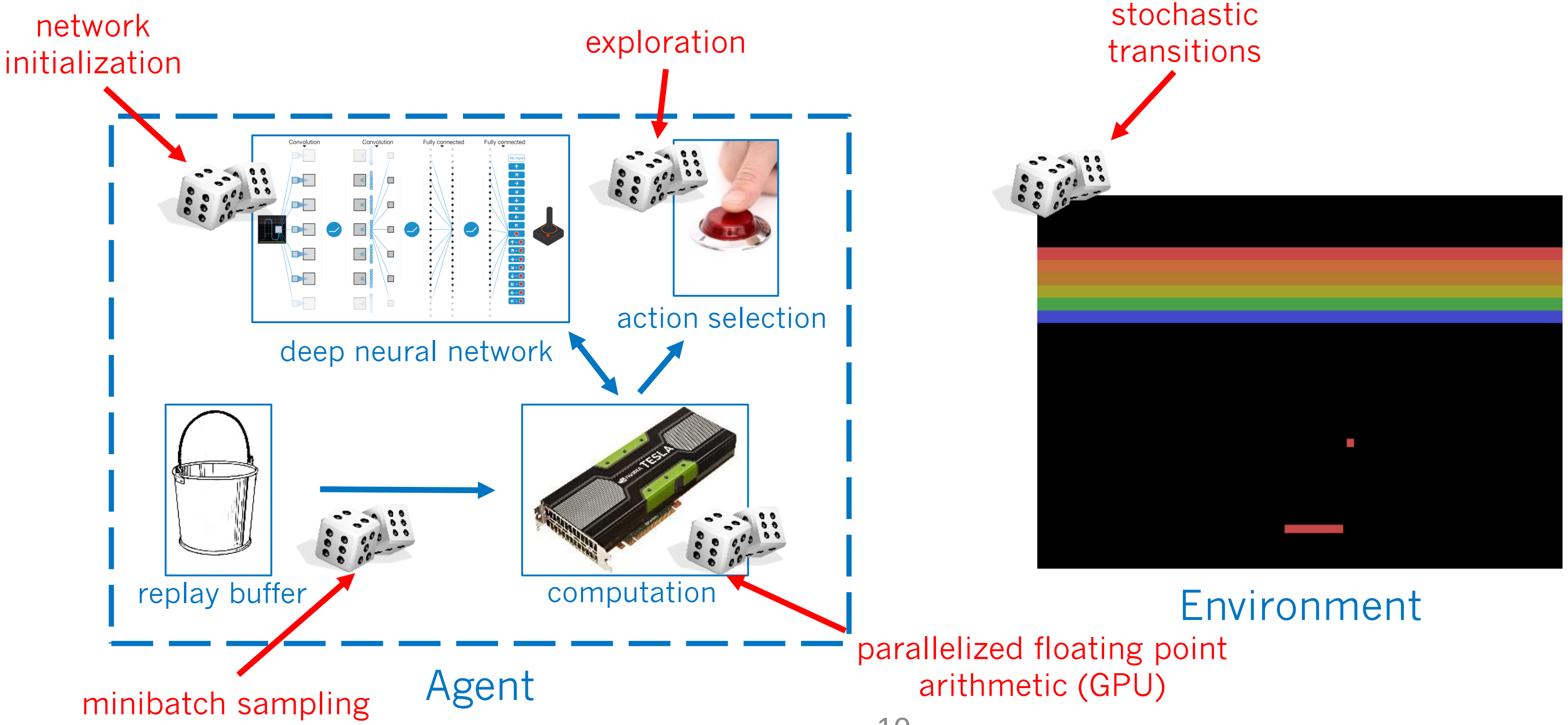


Mnih et al. Nature Letters. 2015.



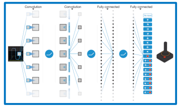
Atari Breakout

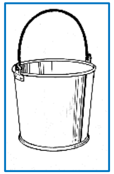
DQN – Sources of Nondeterminism



Our Deterministic Implementation

- **GPU** – `torch.backends.cudnn.deterministic = True`
  - GPU operations are deterministic
- **Network initialization** – PyTorch enables fixed initialization
  - Identical network initialization across program runs.
- **Minibatch Sampling** – seeded random number generator
- **Exploration** – seeded random number generator
- *Note: Deep learning library can matter!*





Preferred Networks

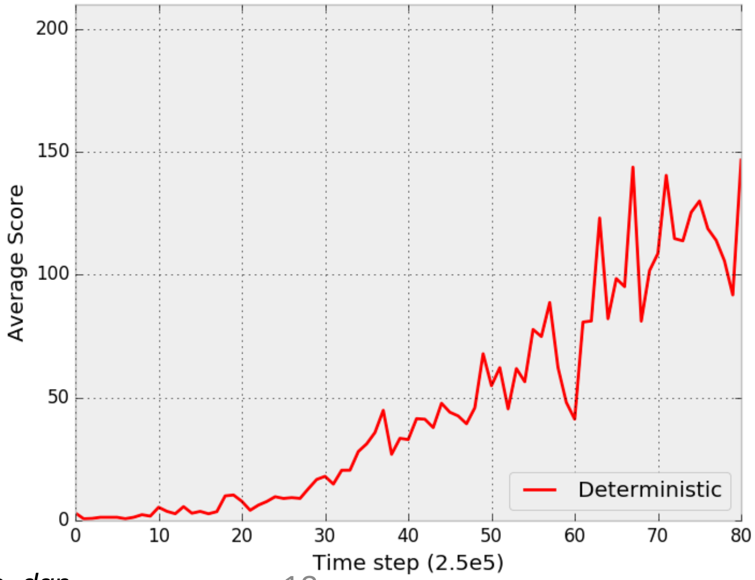
Result

- *Deterministic implementation[1] achievable!*

Run 1

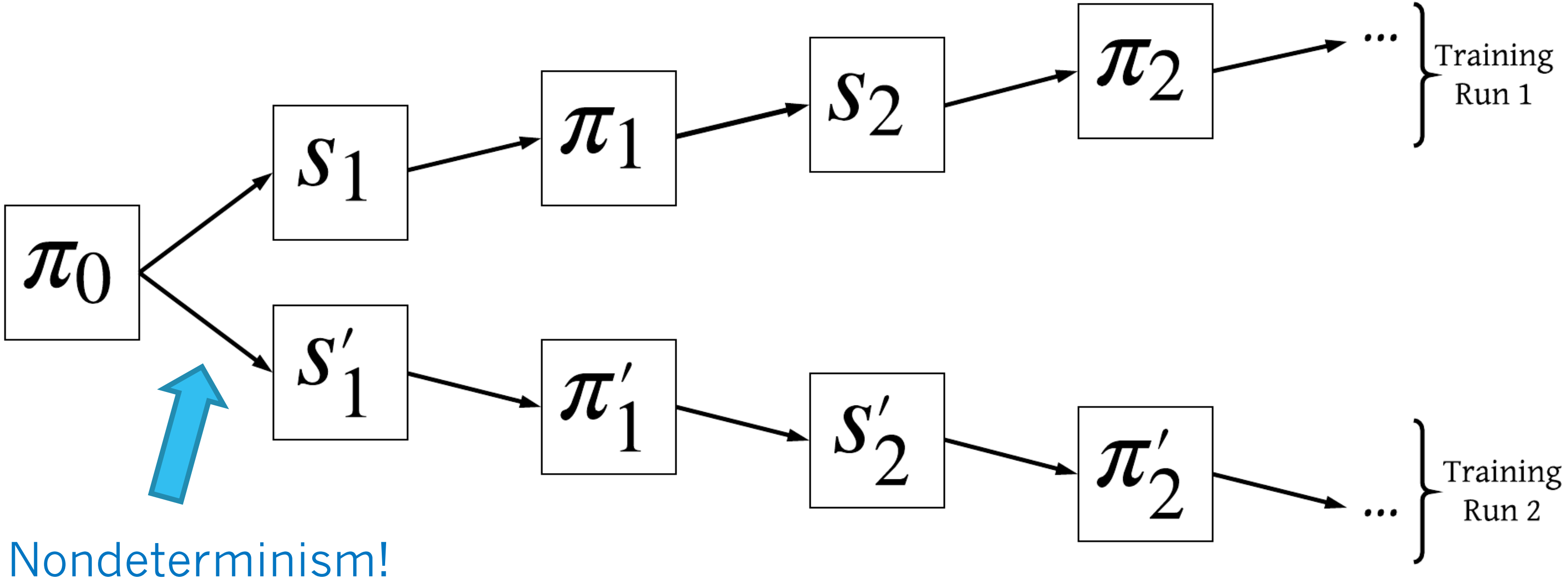
+ Run 2

+ Runs 3, 4, 5



[1] *https://github.com/prabhatnagarajan/repro_dqn*

Cascading Effect



Sensitivity Analysis

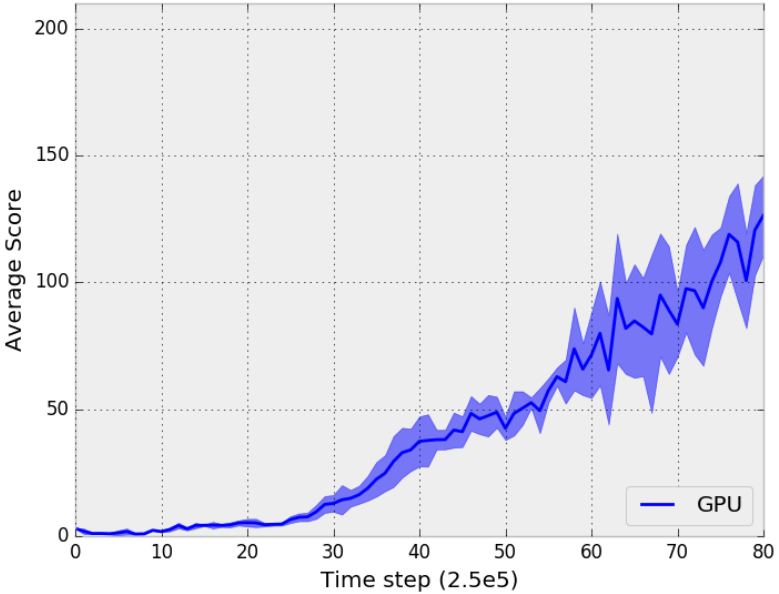
*What kind of experimental **variance** is induced by each source of nondeterminism in deep reinforcement learning?*

Experimental setup:

- Ablation on sources of nondeterminism
- Permit a single source of nondeterminism to affect training (GPU, Network Init, Minibatch, Exploration)
- 5 DQN training runs per source of nondeterminism
- Measure variance of the achieved scores

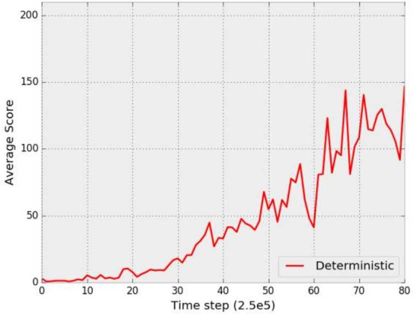
GPU Results

Computation

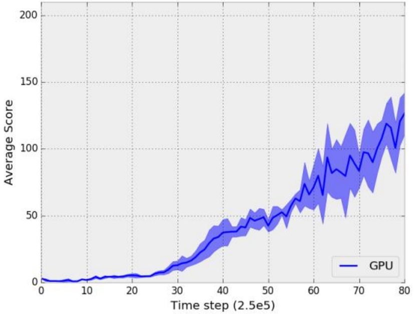


Growing variance…

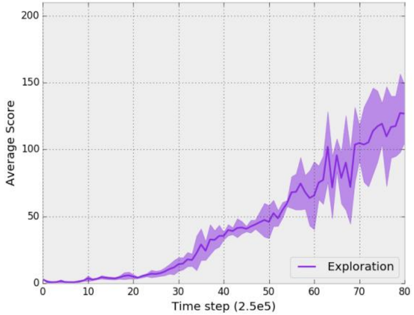
Results



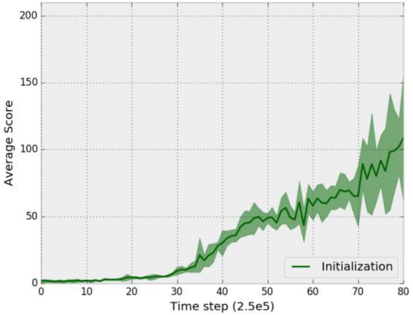
(a) Deterministic



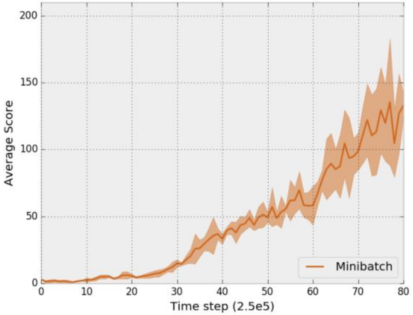
(b) GPU



(c) Exploration



(d) Initialization



(e) Minibatch

Results

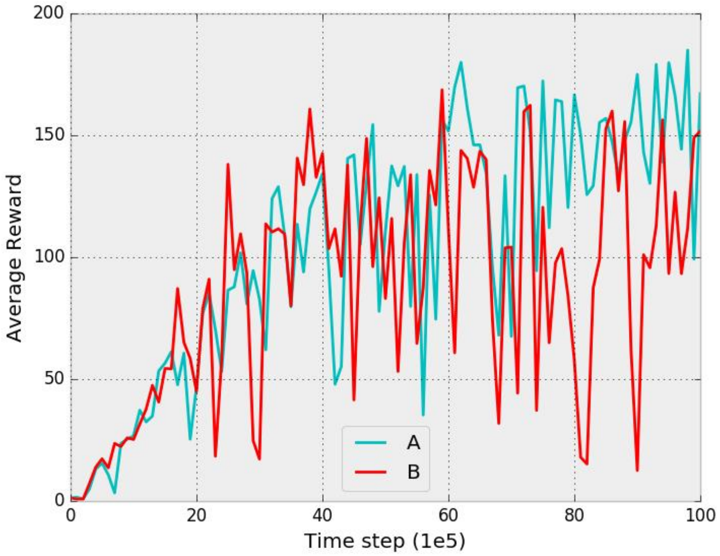
Metric	Deterministic	GPU	Exploration	Initialization	Minibatch
Average Score	146.7	141.9	148.6	131.2	153.38
Standard Deviation	0.0	8.8	17.0	31.0	32.96
Relative Standard Deviation	0.0%	6.22 %	11.42%	23.61%	21.49%

Preferred Networks

Limitations - ...on different hardware

Deterministic implementation ≠ Replicable

Same deterministic implementation on different machines!



Preferred Networks

Limitations

- Limited benefit beyond simulation
  - Robotics, demonstrations, human feedback
- Potential slower training times by making GPU deterministic
  - We did not observe this.
- Deterministic implementations can be time-consuming (depending on library)
- Replicability requires additional work

Future Work

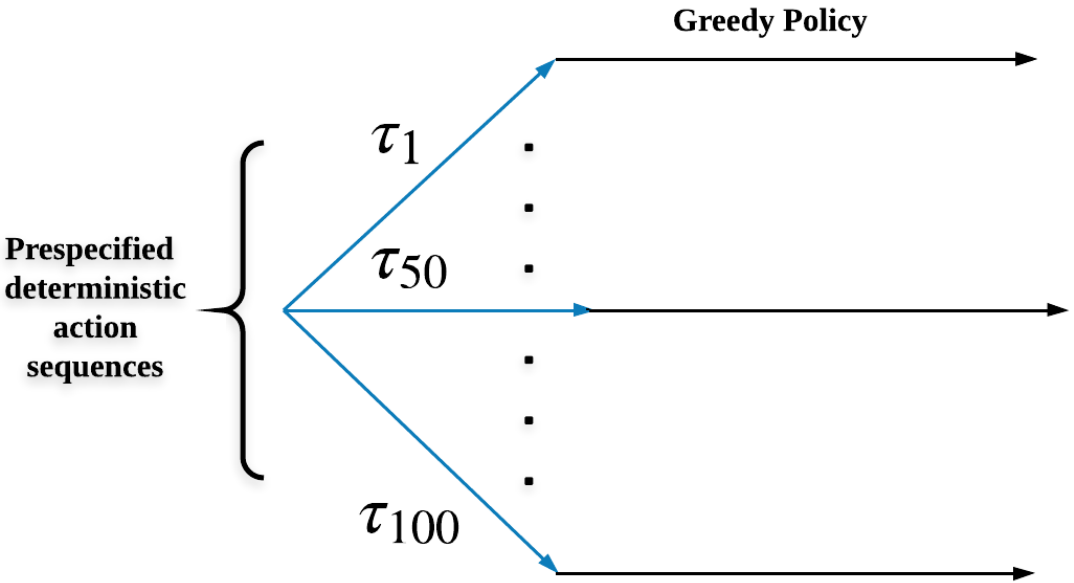
- Extend analysis to a larger suite of algorithms
  - E.g. Pong vs. Asterix
- Investigation of individual implementation details
  - Ablation study
- How can we improve statistical power by combining paired testing and deterministic implementations?

Summary

- Deterministic Implementations are achievable
  - Given a deterministic simulator, training data, demonstrations, etc.
- Deterministic implementations are insufficient for achieving perfect replicability
  - Need other experimental conditions to be fixed
- Variance in performance of DQN grows as training progresses.

prabhat@preferred.jp
prabhatnagarajan.com

Evaluation Protocol

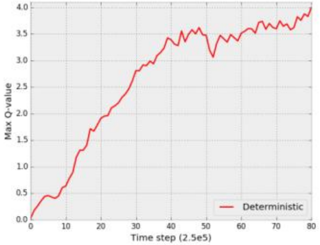


Supervised Learning vs. Deep RL

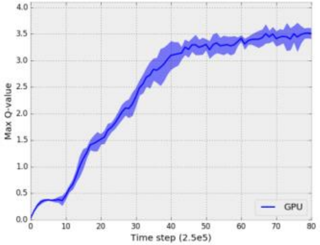
- Stationary vs. Nonstationary

Supervised	RL
GPU	GPU
Network Initialization	Network Initialization
Minibatch Sampling	Minibatch Sampling
	Transition function
	Policy

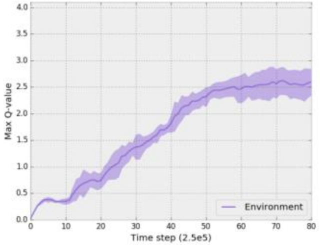
Q-Value Results



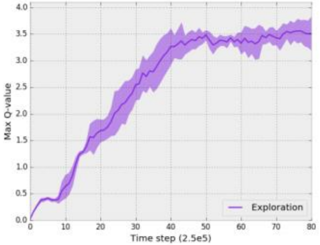
(a) Deterministic



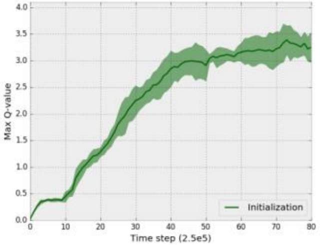
(b) GPU



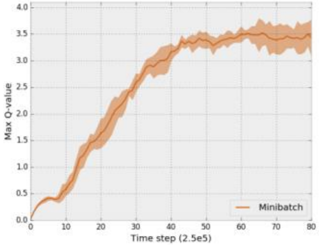
(c) Environment



(d) Exploration



(e) Initialization



(f) Minibatch

Q-Value Results

Metric	Det	GPU	Env	Exp	Init	Mini
Average Maximum Q-value (*Best*)	4.00	3.45	2.49	3.53	3.29	3.44
Standard Deviation (*Best*)	0.0	0.081	0.269	0.305	0.204	0.36
Relative Standard Deviation (*Best*)	0.0%	2.34%	10.83%	8.63%	6.19%	10.5%
Average Maximum Q-value (*Final*)	4.00	3.51	2.60%	3.51	3.25	3.43
Standard Deviation (*Final*)	0.0	0.096	0.245	0.315	0.284	0.34
Relative Standard Deviation (*Final*)	0.0%	2.73%	9.45%	8.98%	8.73%	9.97%

Full Tabular Results

Metric	**Deterministic**	**GPU**	**Environment**	**Exploration**	**Initialization**	**Minibatch**
Average Score (*Best*)	146.7	141.9	33.6	148.6	131.2	153.38
Standard Deviation (*Best*)	0.0	8.8	8.7	17.0	31.0	32.96
Relative Standard Deviation (*Best*)	0.0%	6.22 %	25.96%	11.42%	23.61%	21.49%
Average Score (*Final*)	146.7	126.5	29.0	126.9	108.6	132.84
Standard Deviation (*Final*)	0.0	15.7	10.9	21.4	47.4	8.89
Relative Standard Deviation (*Final*)	0.0%	12.41%	37.65%	16.85%	43.61%	6.69%