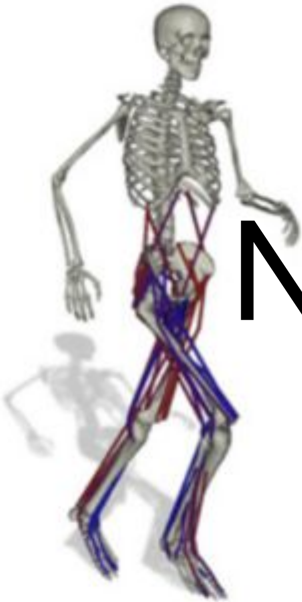
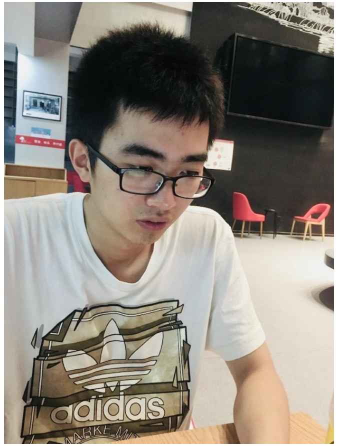


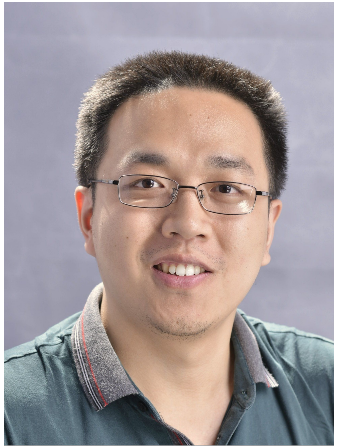
NIPS ’17 Learning to Run

hzwer@pku.edu.cn

zsc@megvii.com

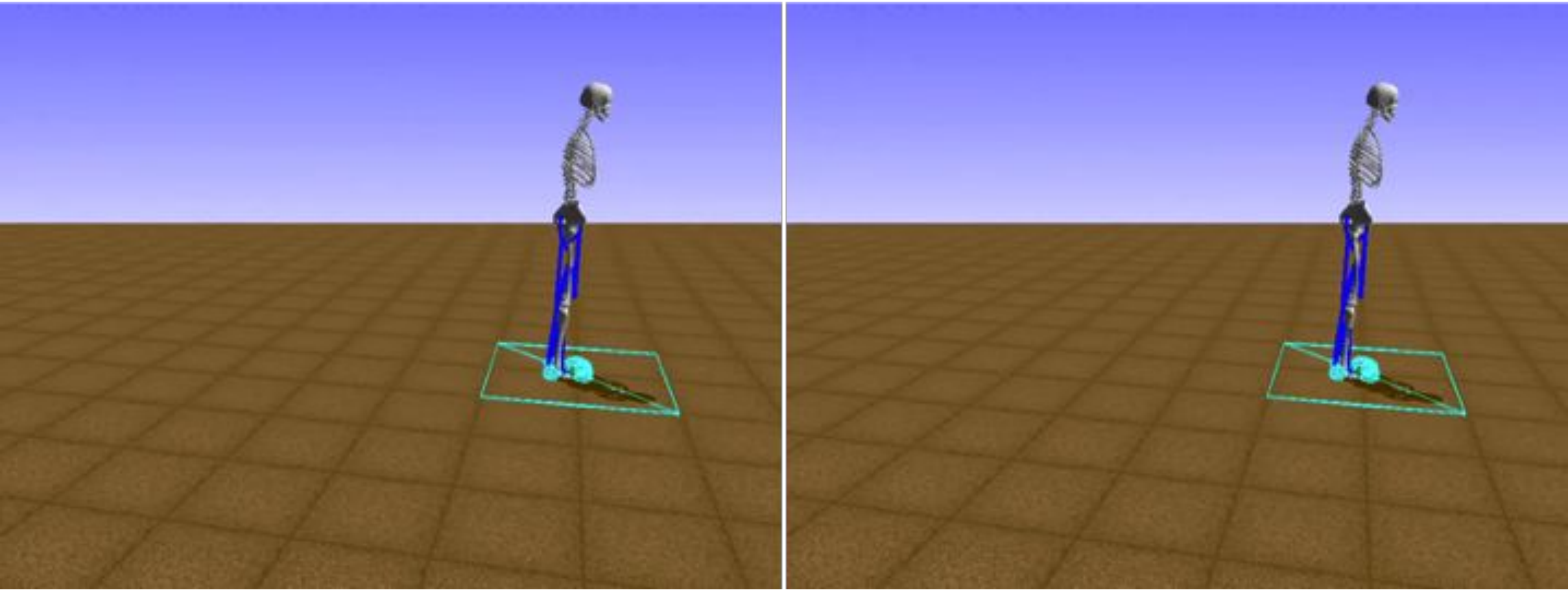


黄哲威
Zhewei
Huang



周舒畅
Shuchang
Zhou

Demo



Results

Round 1

#obstacles = 3

01.	USTC-IMCL	44.6184074052	47
02.	Megvii-hzwer	43.9677458037	57

Round 2

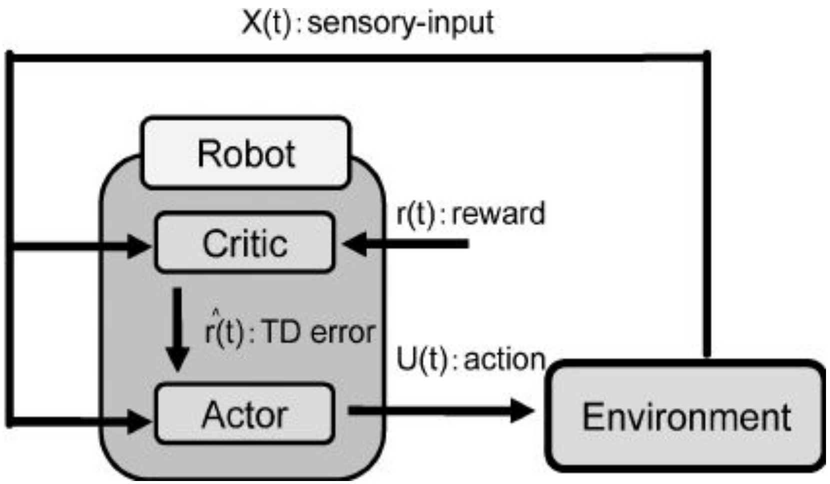
#obstacles = 10,

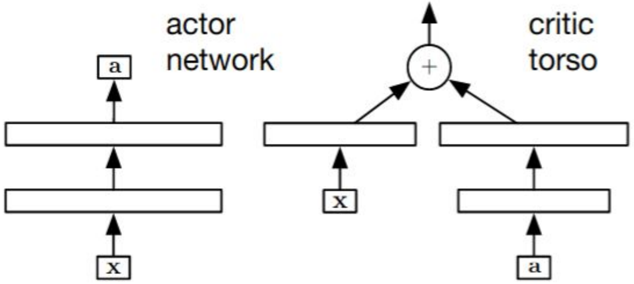
easier to fall down.

01.	NNAISENSE	45.9655247366	5
02.	Megvii-hzwer	41.6620914769	2

https://www.crowdai.org/challenges/nips-2017-learning-to-run/leaderboards

Background: Actor-Critic & DDPG





Actor-Critic:
Critic learns the reward and instructs the actor.

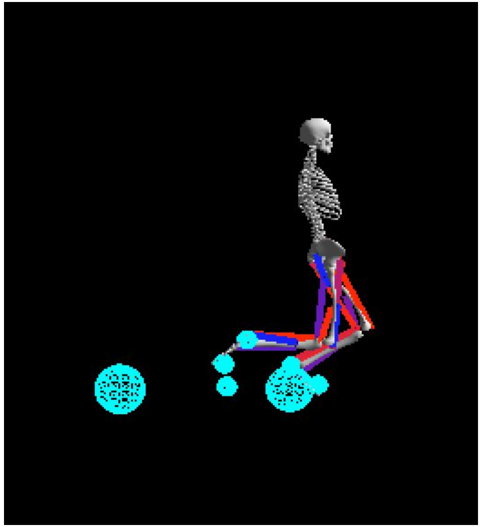
DDPG:
Deterministic policy eliminates the expectation and allows off-policy training.

$$Q^{\pi}(s_t, a_t) = \mathbb{E}_{r_t, s_{t+1} \sim E}\left[r(s_t, a_t) + \gamma \boxed{\mathbb{E}_{a_{t+1} \sim \pi}}\left[Q^{\pi}(s_{t+1}, a_{t+1})\right]\right]$$

$$Q^{\mu}(s_t, a_t) = \mathbb{E}_{r_t, s_{t+1} \sim E}\left[r(s_t, a_t) + \gamma Q^{\mu}(s_{t+1}, \mu(s_{t+1}))\right]$$

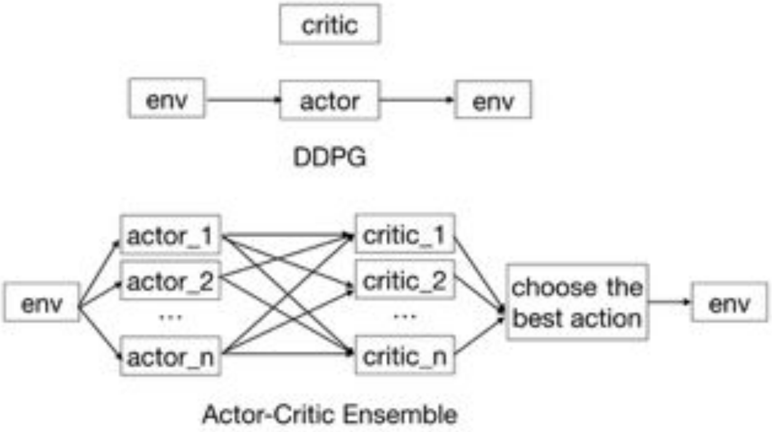
Dooming Actions

- Actions having fatal consequences
- E.g.: Legs of the skeleton tripped by obstacles
    - limbs swinging
    - non-recoverable by actions
- Critics in fact ***know*** which are dooming actions
    - can give low scores
    - but DDPG don't have a mechanism to recover
- Solution: **Actor-Critic Ensemble (ACE)**



Inference with Actor/Critic Ensemble

- Round2 challenge: more obstacles make it much easier to fall down
  - Single actor may not recover from bad state
  - Sometimes the critic has given several consecutive low scores, but the actor turns a deaf ear and proceed to fail!
  - Having multiple actors allows more chances of recovery
- Multiple critics also improves robustness
- Actor/Critic Ensemble reduces falling from 25% to <5%

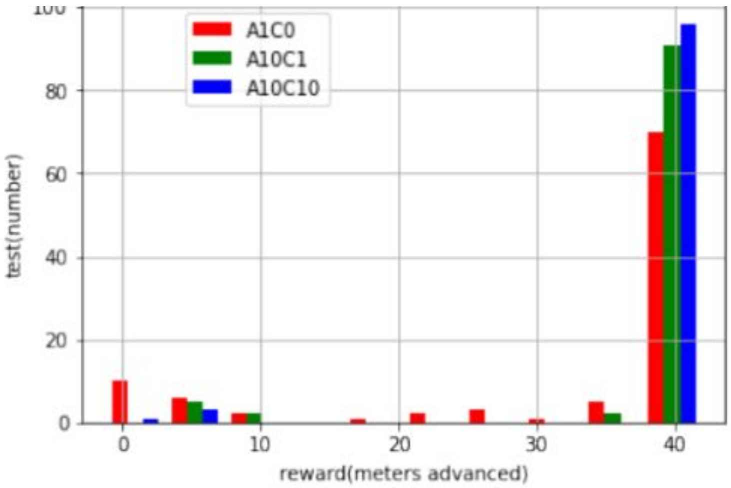


Inference with Actor/Critic Ensemble

Table 2: Performance of ACE

Experiment	# Test	# Actor	# Critic	Average reward	Max reward	# Fall off
A1C0	100	1	0	32.0789	41.4203	25
A10C1	100	10	1	37.7578	41.4445	7
A10C10	100	10	10	39.2579	41.9507	4

AXCY stands for X number of actors and Y number of critics



Training with Actor/Critic Ensemble

Train with Actor Ensemble

- All actors can be updated at every step (even if its action is not used)

$$i_{t+1} = \arg\max_j Q(s_{t+1}, \mu_j(s_{t+1}))$$

$$Q(s_t, a_t) = r(s_t, a_t) + \gamma Q(s_{t+1}, \mu_{i_{t+1}}(s_{t+1}))$$

Train with Critic Ensemble

- Just like Ensemble method in classification

No significant gain yet.

Vanishing Gradient Challenge

N-step DDPG

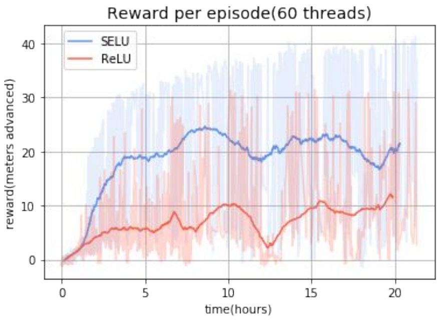
$$Q^{\mu}(s_t, a_t) = \mathbb{E}_{r_t, s_{t+1} \sim E}\left[r(s_t, a_t) + \gamma Q^{\mu}(s_{t+1}, \mu(s_{t+1}))\right]$$

$$(\mathcal{T}_{\pi}^{N} Q)(\mathbf{x}_0, \mathbf{a}_0) = r(\mathbf{x}_0, \mathbf{a}_0) + \mathbb{E}\left[\sum_{n=1}^{N-1} \gamma^n r(\mathbf{x}_n, \mathbf{a}_n) + \gamma^N Q(\mathbf{x}_N, \pi(\mathbf{x}_N)) \,\middle|\, \mathbf{x}_0, \mathbf{a}_0\right]$$

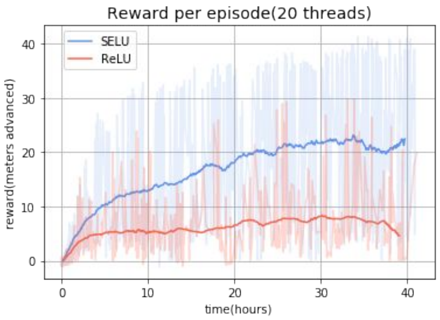
N-step Simulation

Simulate at 4x FPS, or equivalently, use the same action for 4-steps and do TD learning on 4-steps.

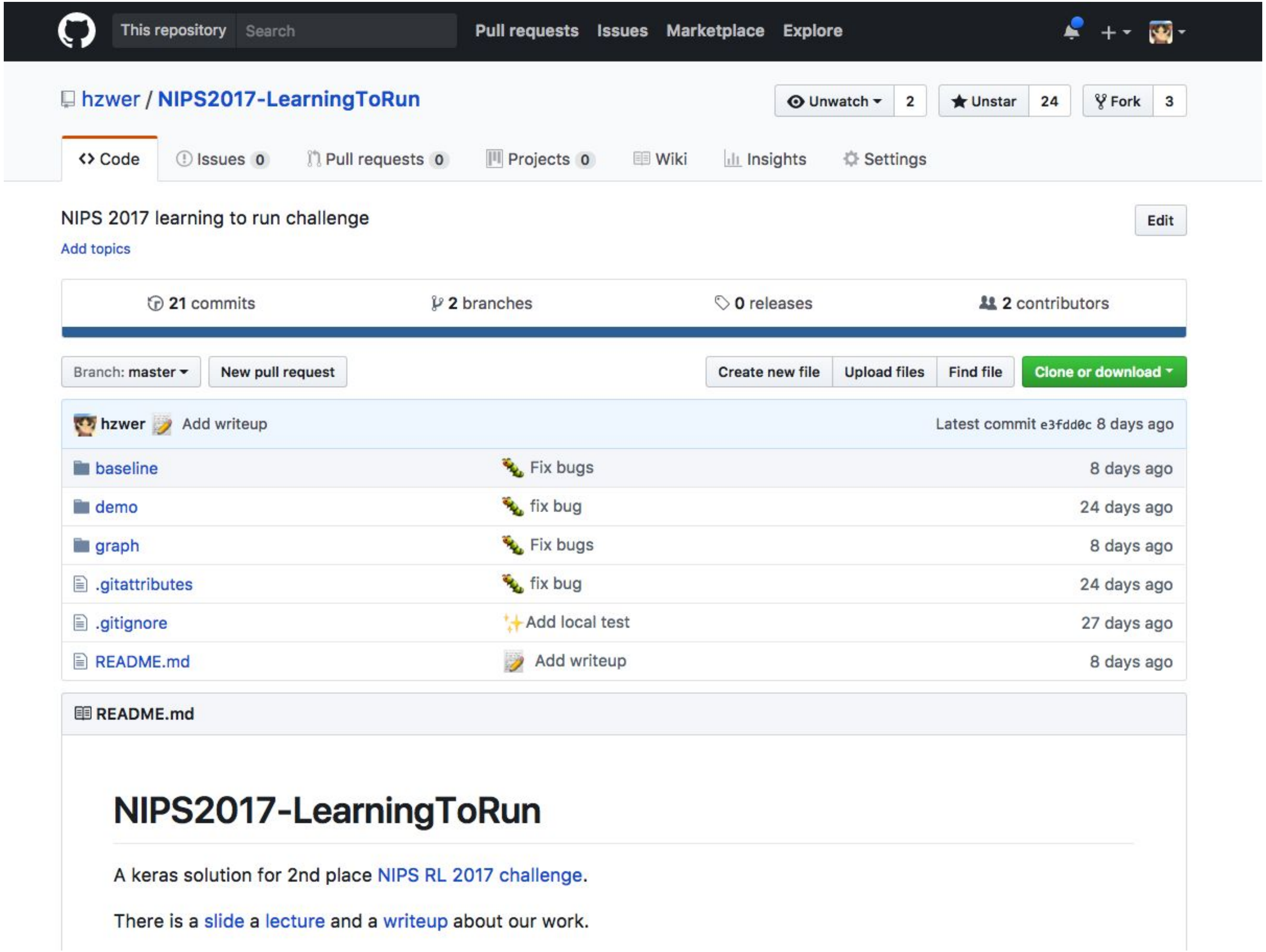
Smooth activation: SELU is unexpectedly good



simulation with 60 processes



simulation with 20 processes



Many thanks to people who helped us and the Brain++@Megvii team for support.

TODO

- The framework from Qin Yongliang is great, but can be improved
  - Was using pickle for Replay Memory (later changes to HDF5)
  - Pyro4 encountered timeouts for successful runs
  - Used Keras/Tensorflow
    - Keras was hard to hack
- The simulation speed of OpenSim may be improved
  - Can use lower-precision simulation

Thanks!

Code: https://github.com/hzwer/NIPS2017-LearningToRun

References

Another competitor's write-up:
https://medium.com/@stelmaszczykadam/our-nips-2017-learning-to-run-approach-b80a295d3bb5

http://blog.otoro.net/2017/11/12/evolving-stable-strategies/
BipedalWalkerHardcore-v2

Backup after this slide

Challenges

- Slow simulation
  - Some steps take minutes to simulate
  - Answer: uses ~1000 cores
- Strange setups
  - Reward = horizontal movement - ligament penalties
  - Pelvis cannot be lower than 0.65
  - Collision detection is fishy

N-step DDPG

DDPG

$$Q^{\mu}(s_t, a_t) = \mathbb{E}_{r_t, s_{t+1} \sim E}\left[r(s_t, a_t) + \gamma Q^{\mu}(s_{t+1}, \mu(s_{t+1}))\right]$$

N-Step: not as effective as multi-frame simulation.

$$(\mathcal{T}_{\pi}^{N} Q)(\mathbf{x}_0, \mathbf{a}_0) = r(\mathbf{x}_0, \mathbf{a}_0) + \mathbb{E}\Big[\sum_{n=1}^{N-1} \gamma^n r(\mathbf{x}_n, \mathbf{a}_n) + \gamma^N Q(\mathbf{x}_N, \pi(\mathbf{x}_N)) \Big| \mathbf{x}_0, \mathbf{a}_0\Big]$$

Important: make Q zero when the episode ends, otherwise Q becomes ill-defined.

Methods

- Manual features
  - acceleration + velocity
  - roughly three time frames
- Tricks
  - Simulate the game at 4x speed when training
    - Reduce number of steps to 250
- Distributed training environment
  - RPC framework built on Pyro4 + multiprocessing (credit Qin Yongliang)
    - Bad code, but works