DeepMind

Policy improvement by planning with Gumbel

Ivo Danihelka, Arthur Guez, Julian Schrittwieser, David Silver

Motivation: Policy improvement in reinforcement learning

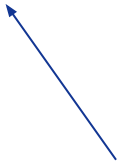
while the policy is changing:

    policy := **policy_improvement**(policy)

Motivation: Policy improvement in reinforcement learning

while the policy is changing:

policy := **policy_improvement**(policy)

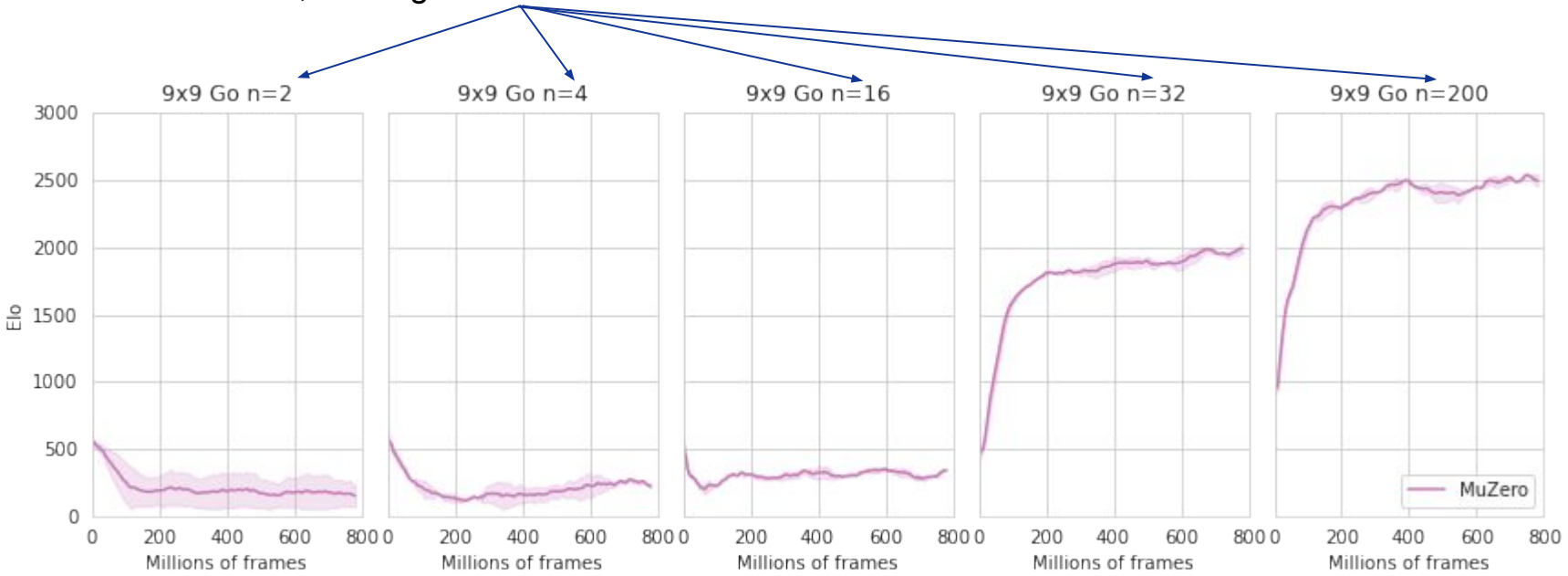


Which algorithm would produce the policy improvement?

We usually do not have well estimated Q-values for all actions.

AlphaZero and MuZero can fail to improve the policy network, if not visiting all actions at the root.

MuZero at 9x9 Go, training with n simulations.



(Evaluation uses 800 simulations.)

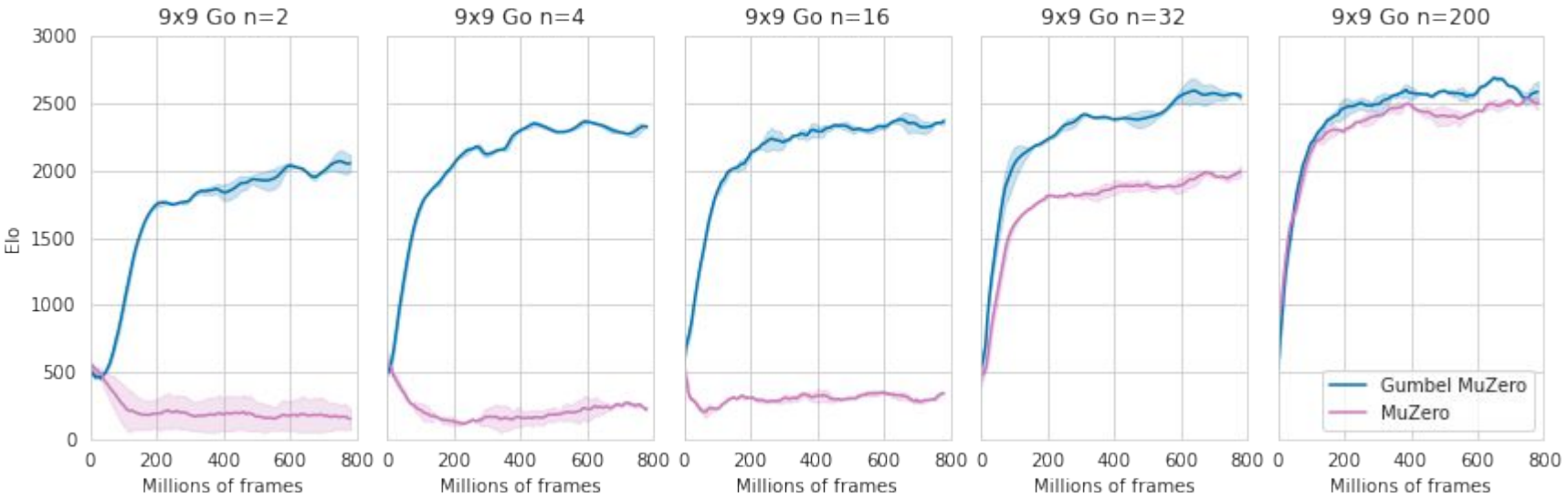
Designing a policy improvement

The basic algorithm:

1. Use the policy network to sample **n** actions without replacement.
2. From the sampled actions, select $\arg\max_a q(a)$

=> New principled algorithm: Gumbel MuZero

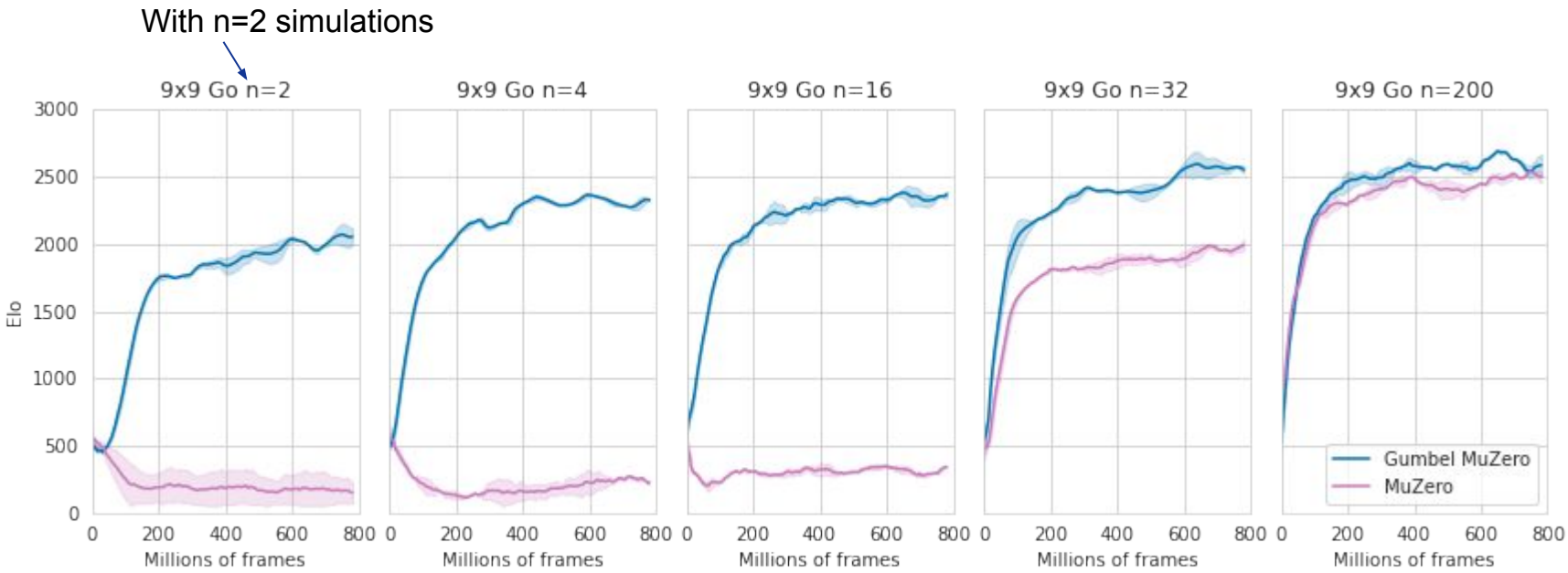
MuZero's Monte Carlo Tree search estimates the Q-values of the sampled actions.



(Evaluation uses 800 simulations.)

=> New principled algorithm: Gumbel MuZero

MuZero's Monte Carlo Tree search estimates the Q-values of the sampled actions.



(Evaluation uses 800 simulations.)

More in the paper

- Sampling without replacement combined with regularized policy optimization.
- Exploration at the root of the search tree.
- Results on 19x19 Go, chess, and Atari.

Code: github.com/deepmind/mctx