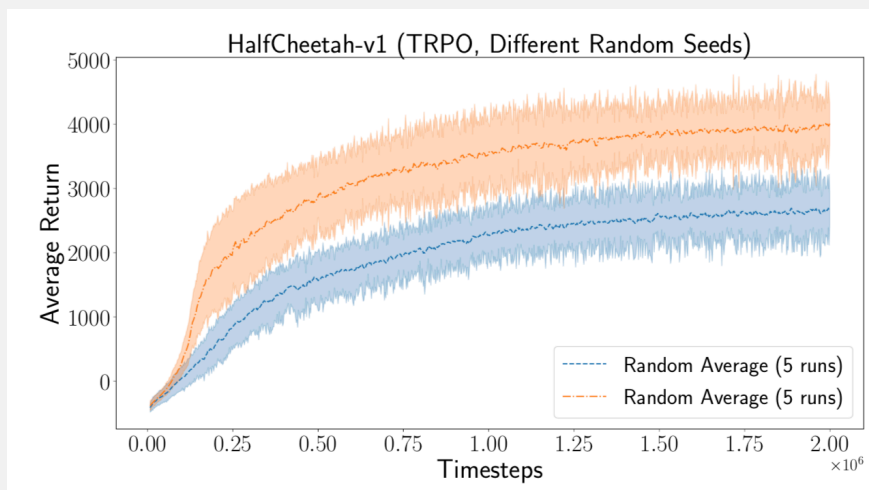


The Rotten Truth of Deep RL

The Rotten Truth of Deep RL

Deep RL can successfully solve tasks, but...

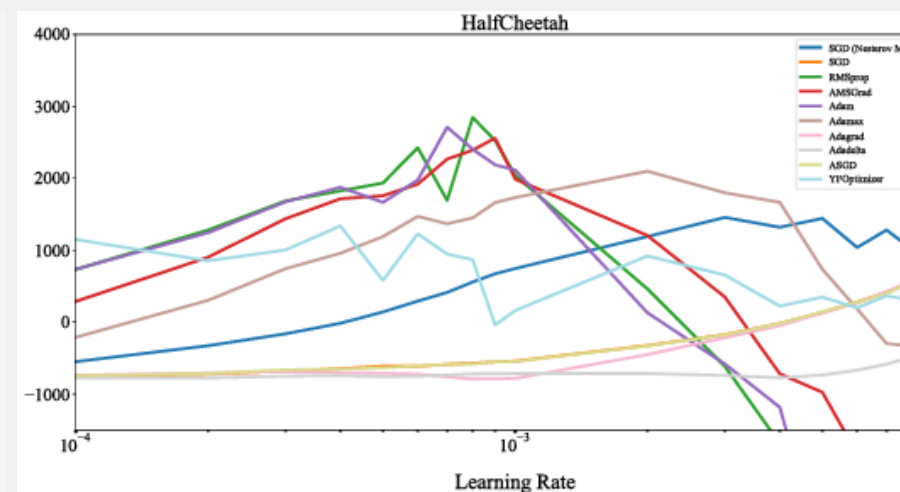
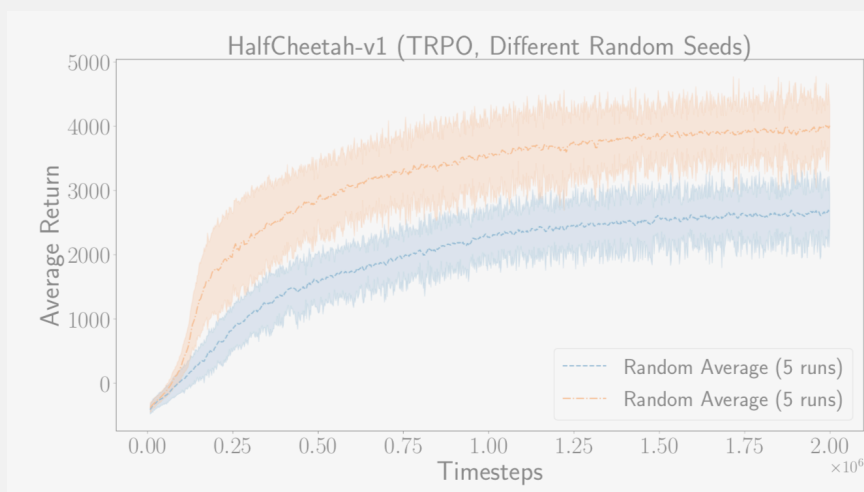
- ▶ Poor reliability over repeated runs



The Rotten Truth of Deep RL

Deep RL can successfully solve tasks, but...

- ▶ Poor reliability over repeated runs
- ▶ High sensitivity to hyperparameters

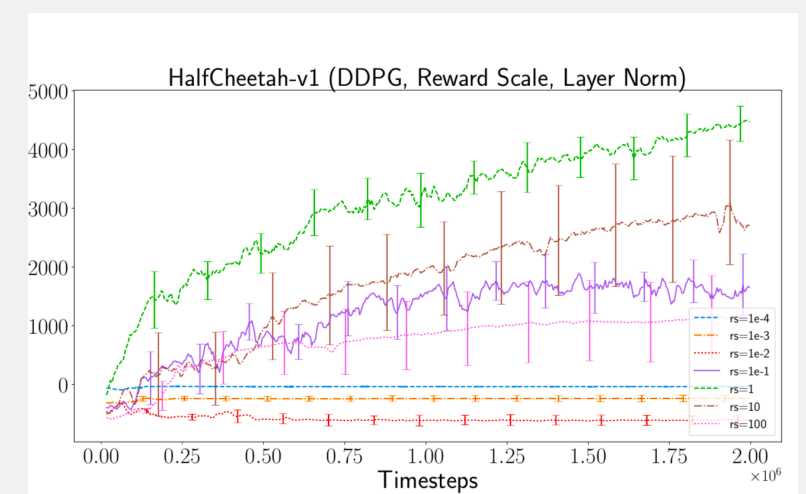
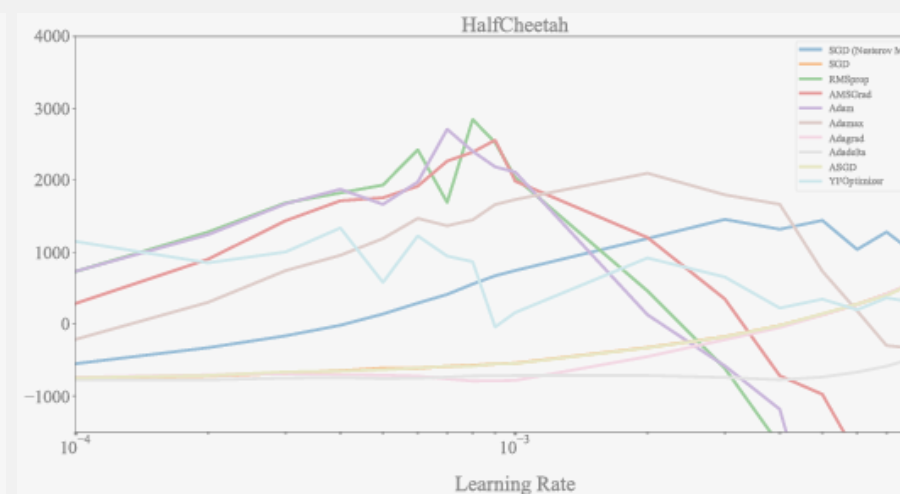
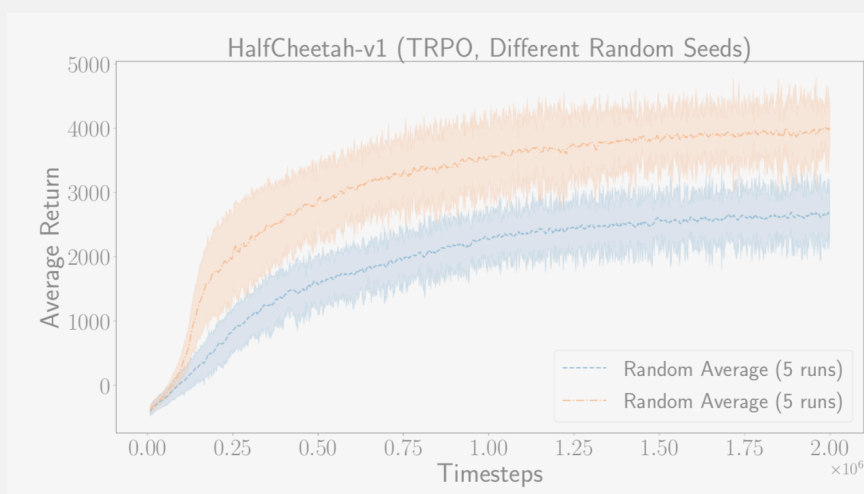


[Henderson et al, 2017a,b] [Lewis et al, 2018]

The Rotten Truth of Deep RL

Deep RL can successfully solve tasks, but...

- ▶ Poor reliability over repeated runs
- ▶ High sensitivity to hyperparameters
- ▶ Lack of robustness to environmental artifacts



[Henderson et al, 2017a,b] [Lewis et al, 2018]

The Rotten Truth of Deep RL

Deep RL can successfully solve tasks, but...

- ▶ Poor reliability over repeated runs
- ▶ High sensitivity to hyperparameters
- ▶ Lack of robustness to environmental artifacts

Notably, benchmarks don't reveal these issues

What's going on?

[Ilyas Engstrom Santurkar Tsipras Janoos Rudolph **M** 2018]

Implementation Obscures Deep RL Algorithms

openai / baselines

Watch 383 Star 5,061 Fork 1,455

Code Issues 137 Pull requests 71 Projects 0 Wiki Insights

OpenAI Baselines: high-quality implementations of reinforcement learning algorithms

- Differences between the policies. In ppo, the `klPolicy` setup requires an important modification and honestly, I don't understand why such code.

- Huge architectural differences. In ppo you get an implementation of the `Runner` class.

- Nontrivial changes to the paper, part 2. The code is sprinkled with small tricks. For example, apply a normalization to the advantage function, [here](#). This one could be my last. But I've only seen something like this in dueling q-learning. You subtract the mean.

epithedee commented 24 days ago

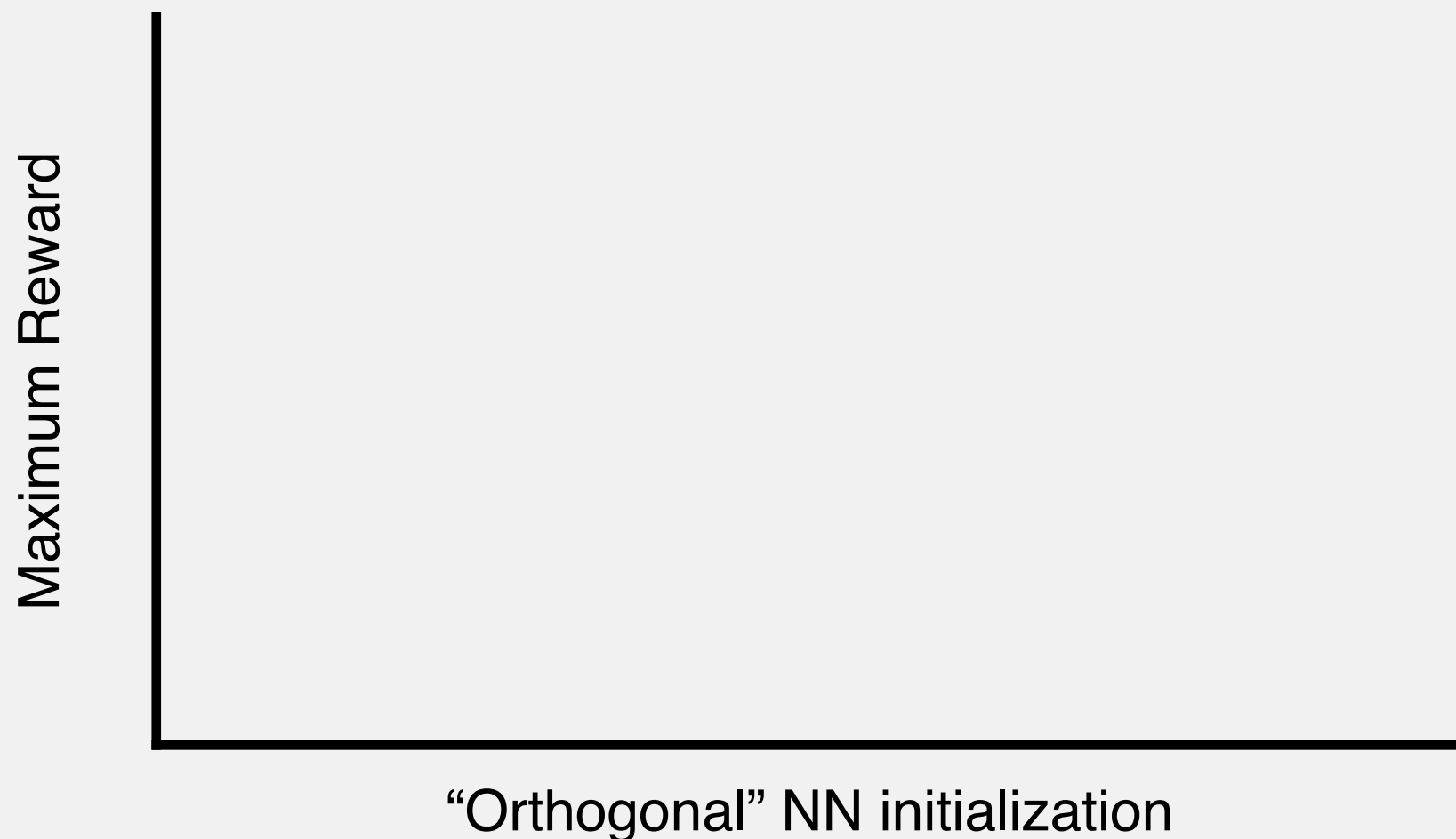
There is one thing between PPO1 and PPO2 that I don't understand
the model actually updates both the old and new set of parameters

- Nontrivial changes to the paper. I read the openai blogpost and the ppo paper. In training of a clipped ratio of action probabilities, ppo does that, ppo2 also updates the value function. I would consider that a big difference between ppo and ppo2

Source: GitHub issues

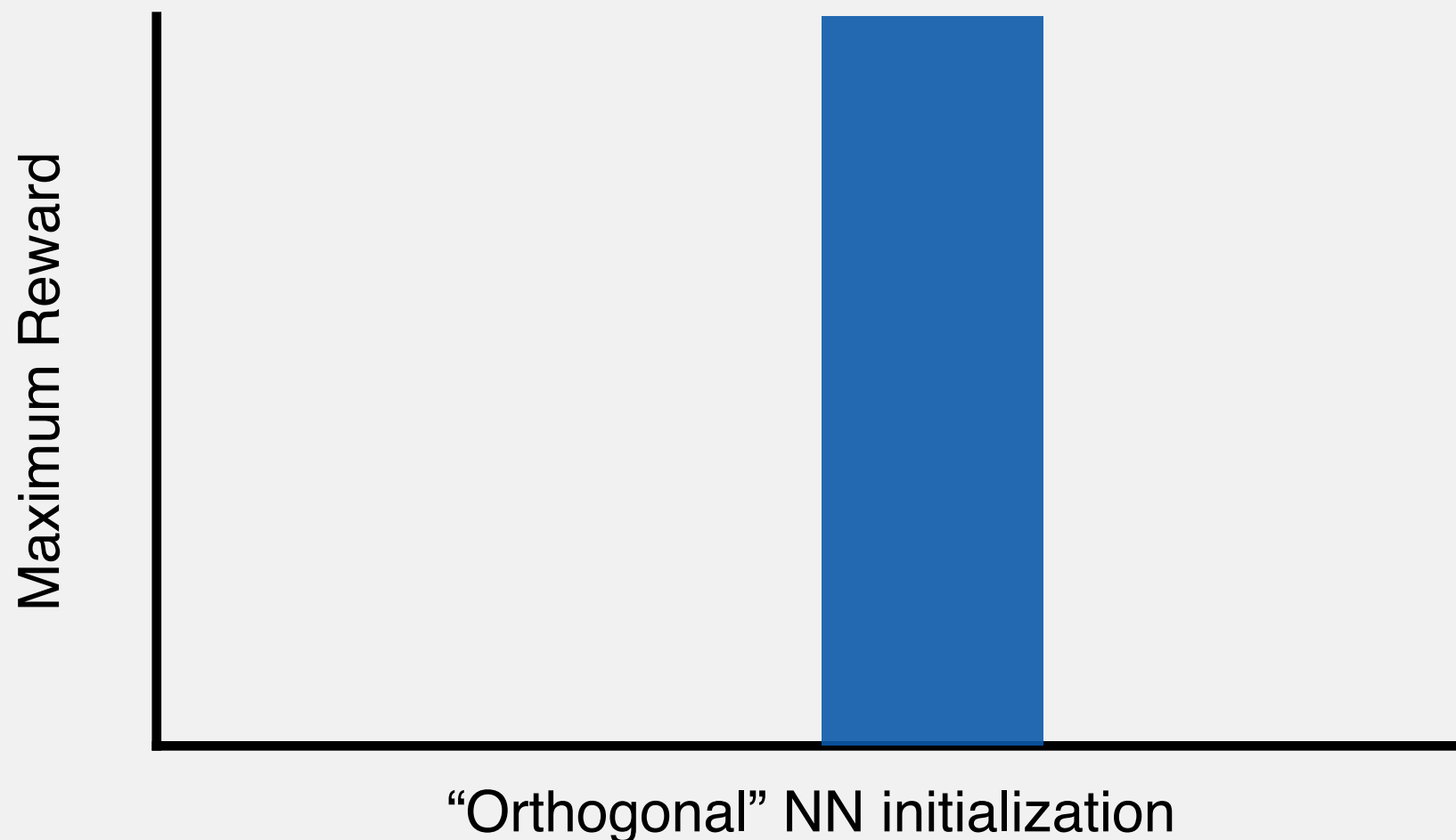
Implementation Obscures Deep RL Algorithms

■ Without Optimization ■ With Optimization



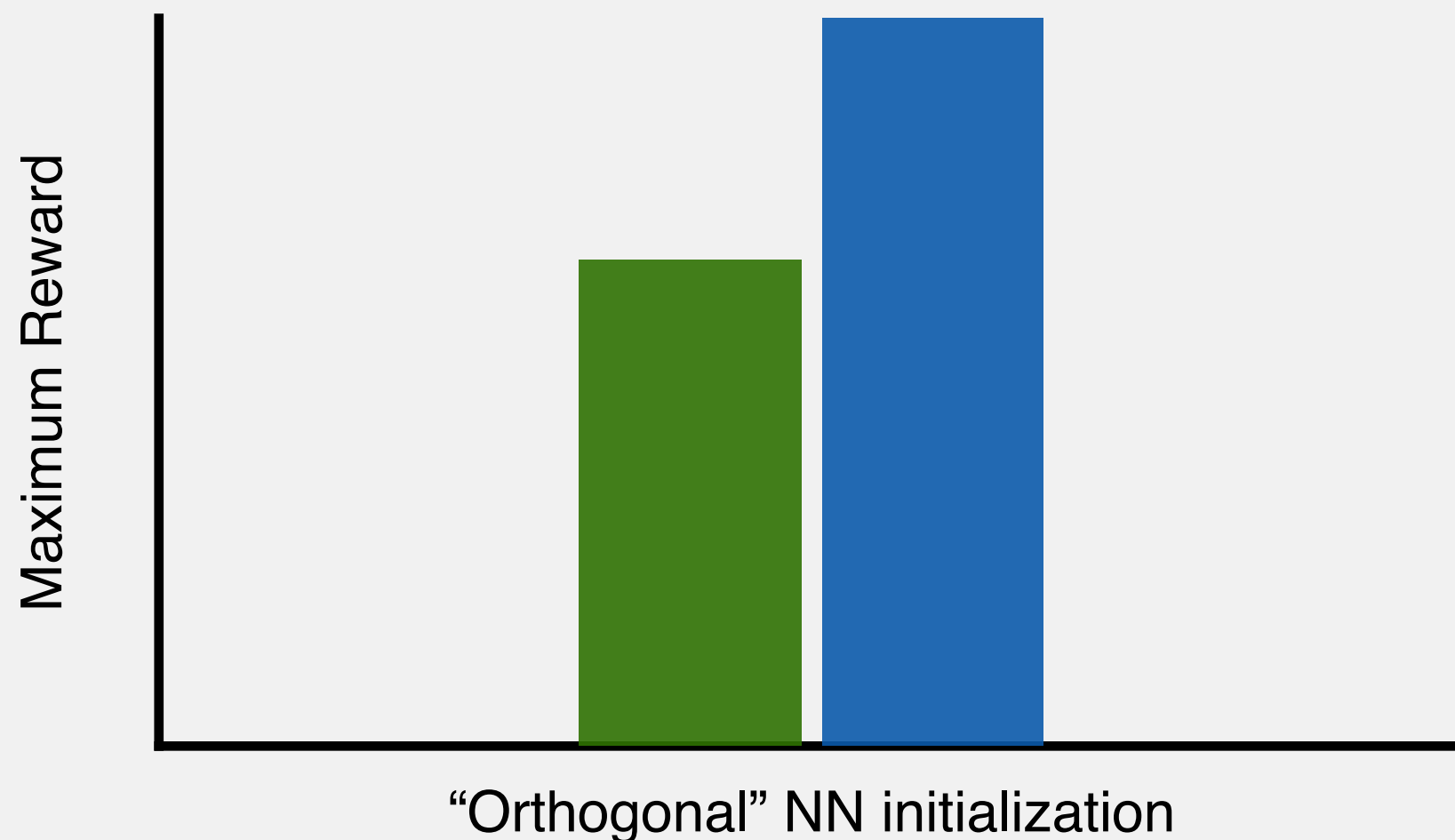
Implementation Obscures Deep RL Algorithms

■ Without Optimization ■ With Optimization

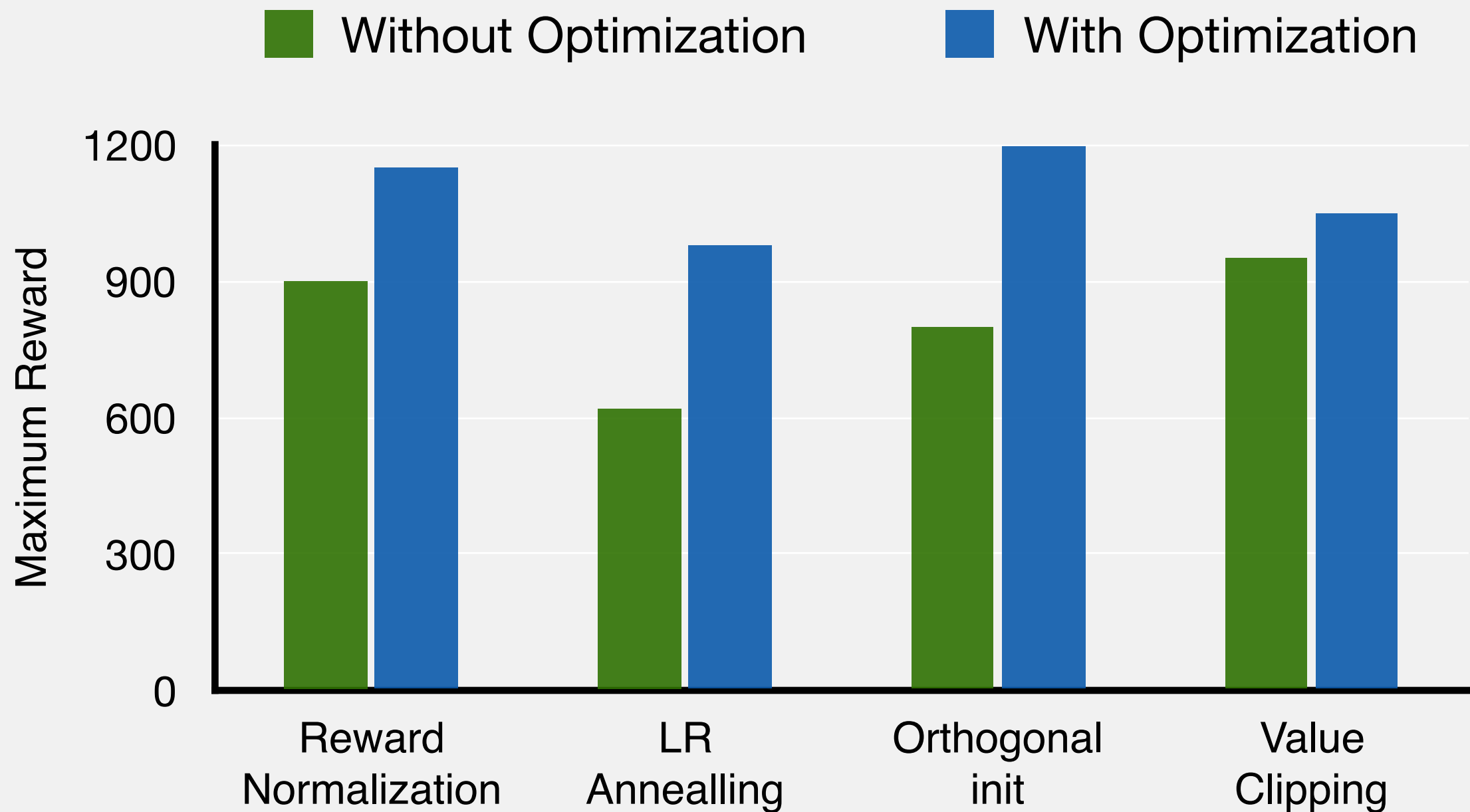


Implementation Obscures Deep RL Algorithms

■ Without Optimization ■ With Optimization



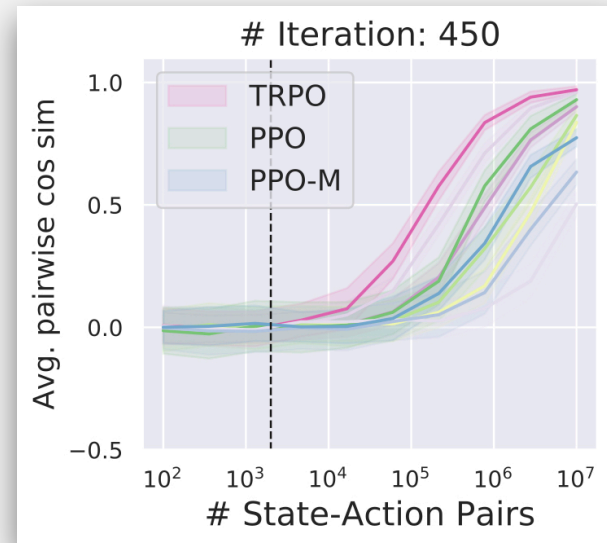
Implementation Obscures Deep RL Algorithms



Back to First Principles

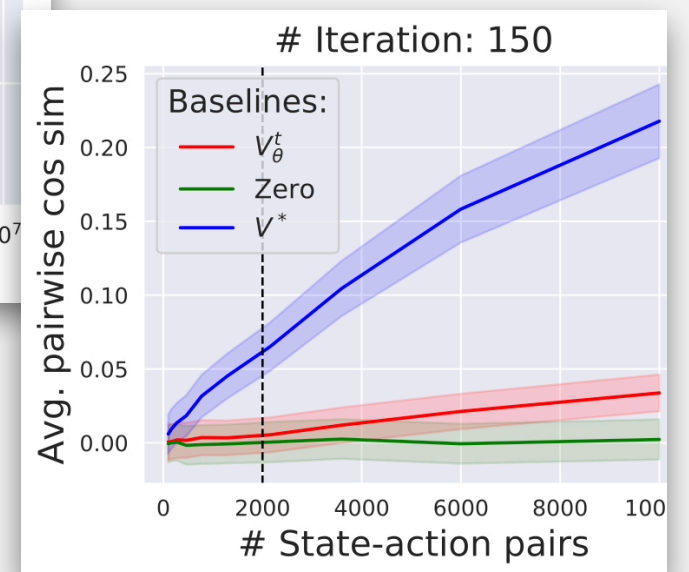
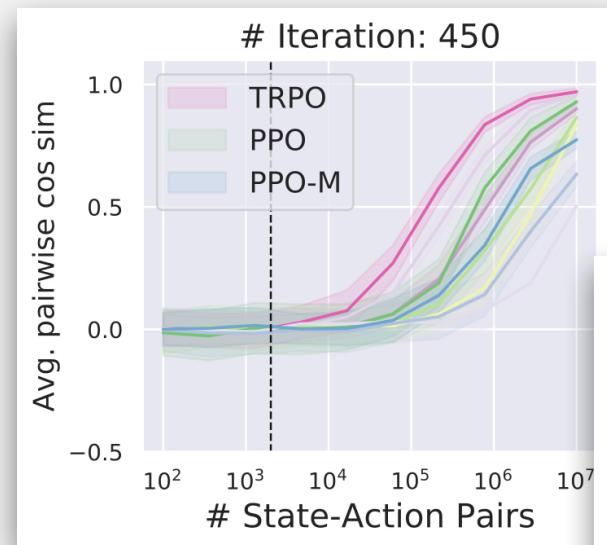
Back to First Principles

- ▶ Gradient Estimates



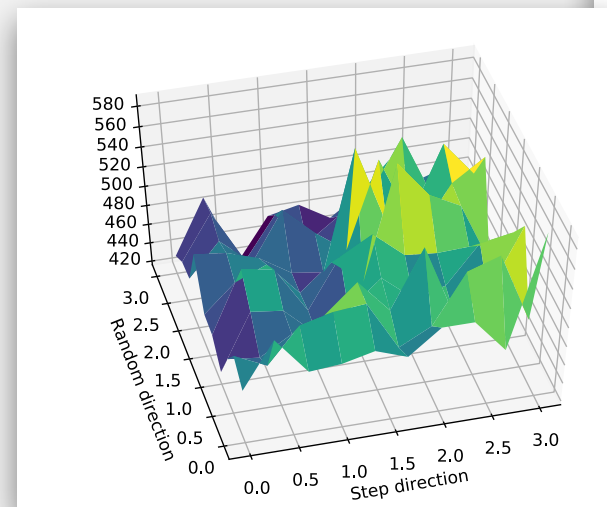
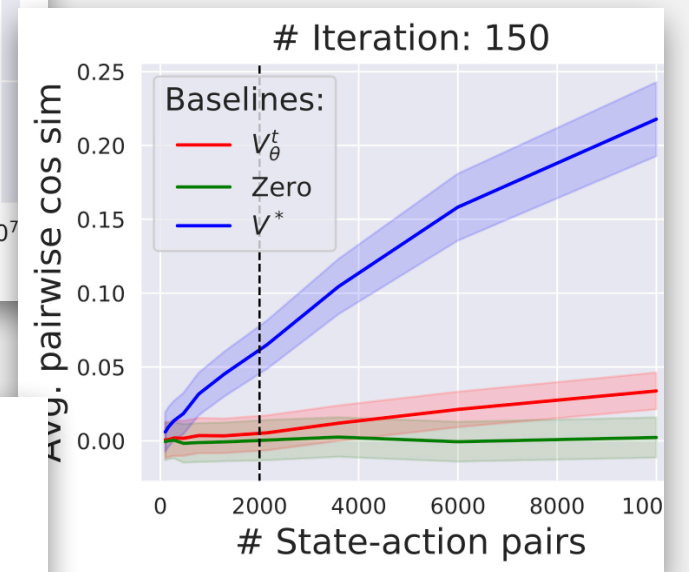
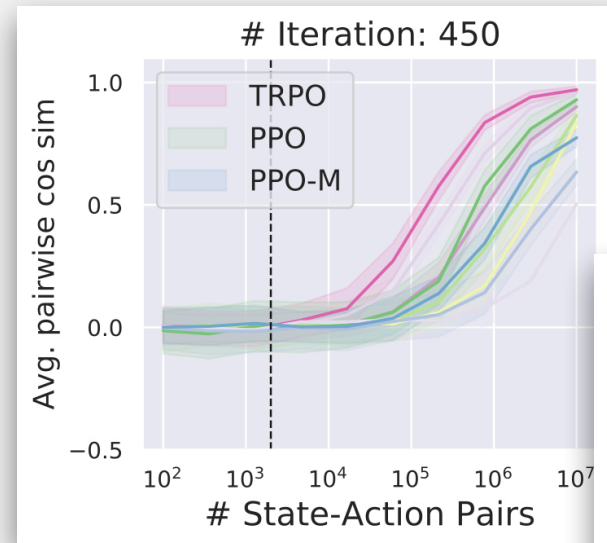
Back to First Principles

- ▶ Gradient Estimates
- ▶ Value Prediction



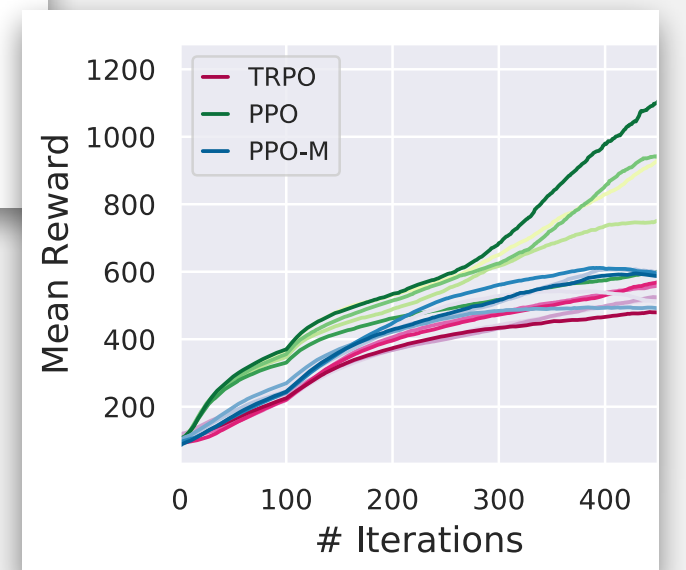
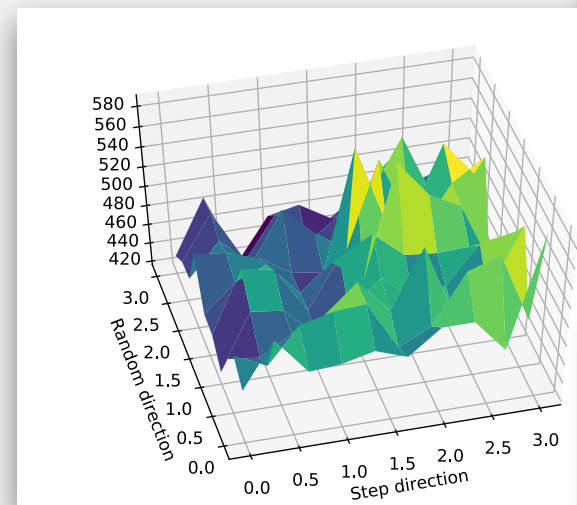
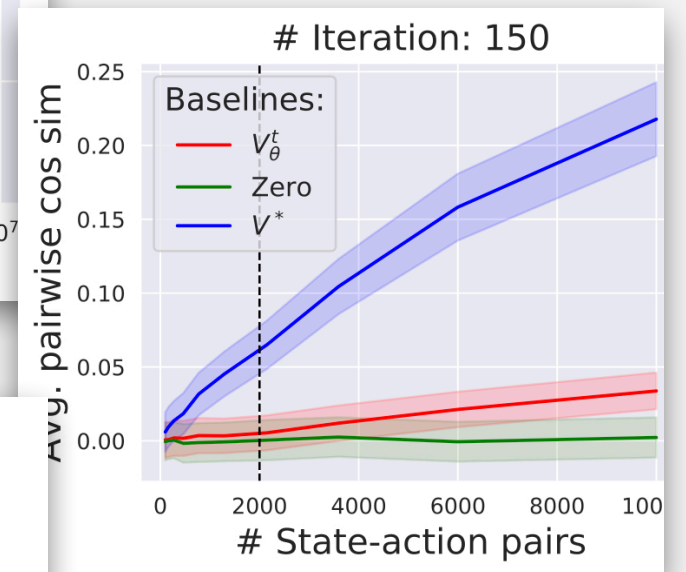
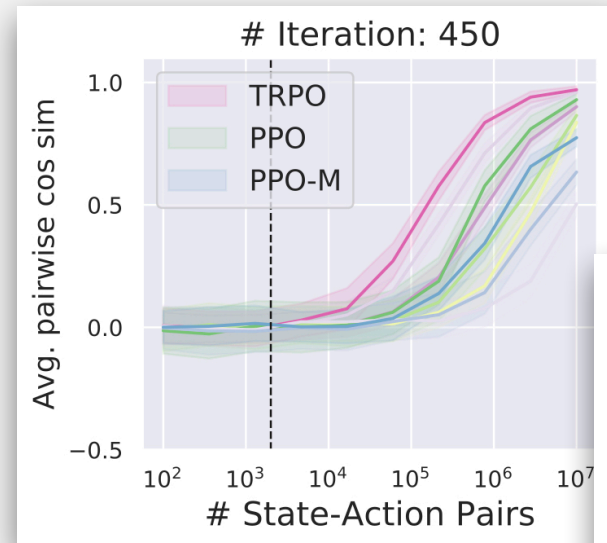
Back to First Principles

- ▶ Gradient Estimates
- ▶ Value Prediction
- ▶ Loss Landscape



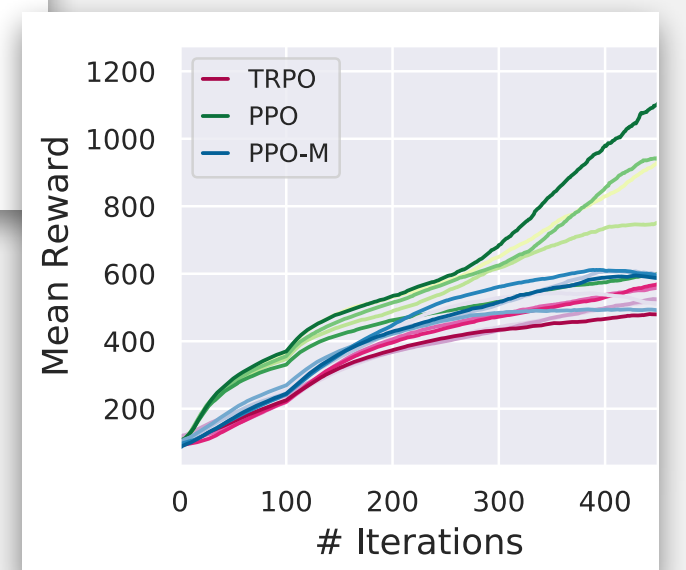
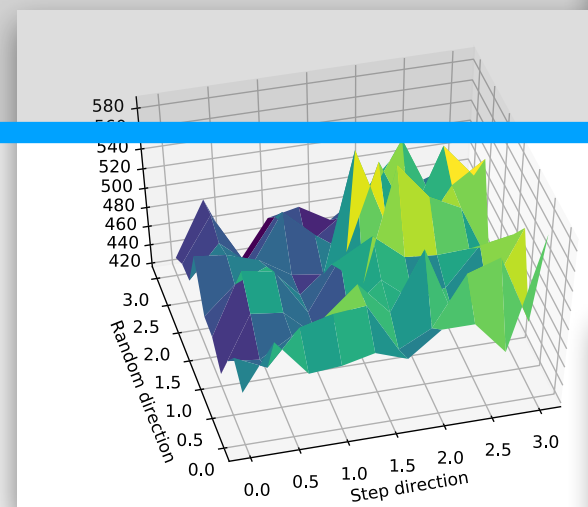
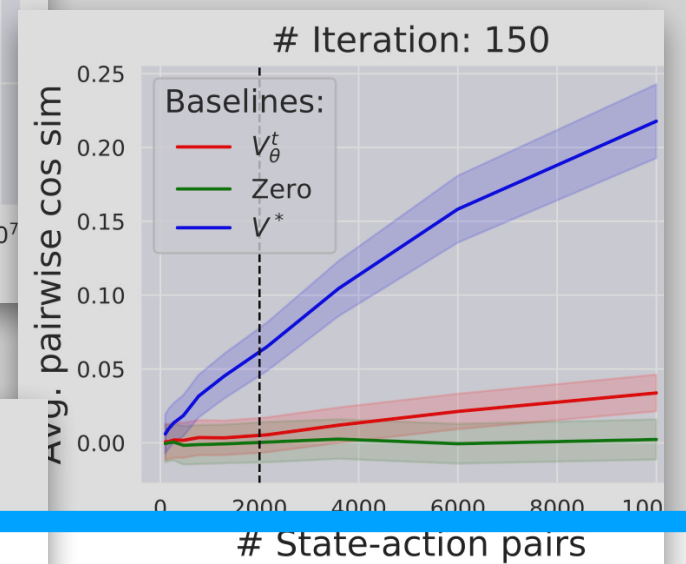
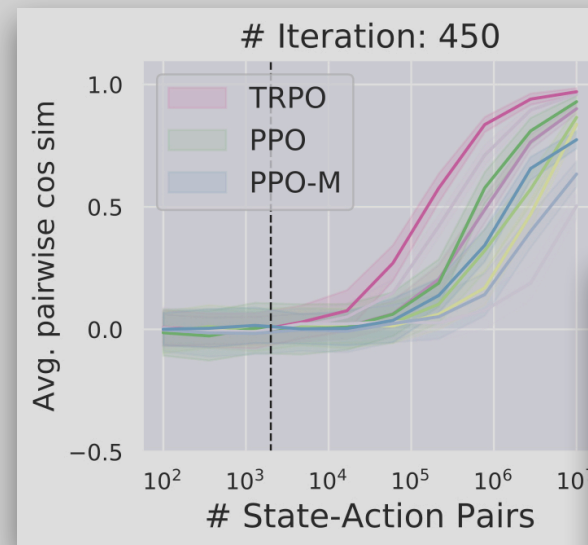
Back to First Principles

- ▶ Gradient Estimates
- ▶ Value Prediction
- ▶ Loss Landscape
- ▶ Trust Region



Back to First Principles

- ▶ Gradient Estimates
- ▶ Value Prediction
- ▶ Loss Landscape
- ▶ Trust Region



Gradient Estimation

Key assumption of policy gradient framework:

$$\mathbb{E}_{X \sim P}[X] \approx \frac{1}{N} \sum_{x_i \sim P} x_i$$

Gradient Estimation

Key assumption of policy gradient framework:

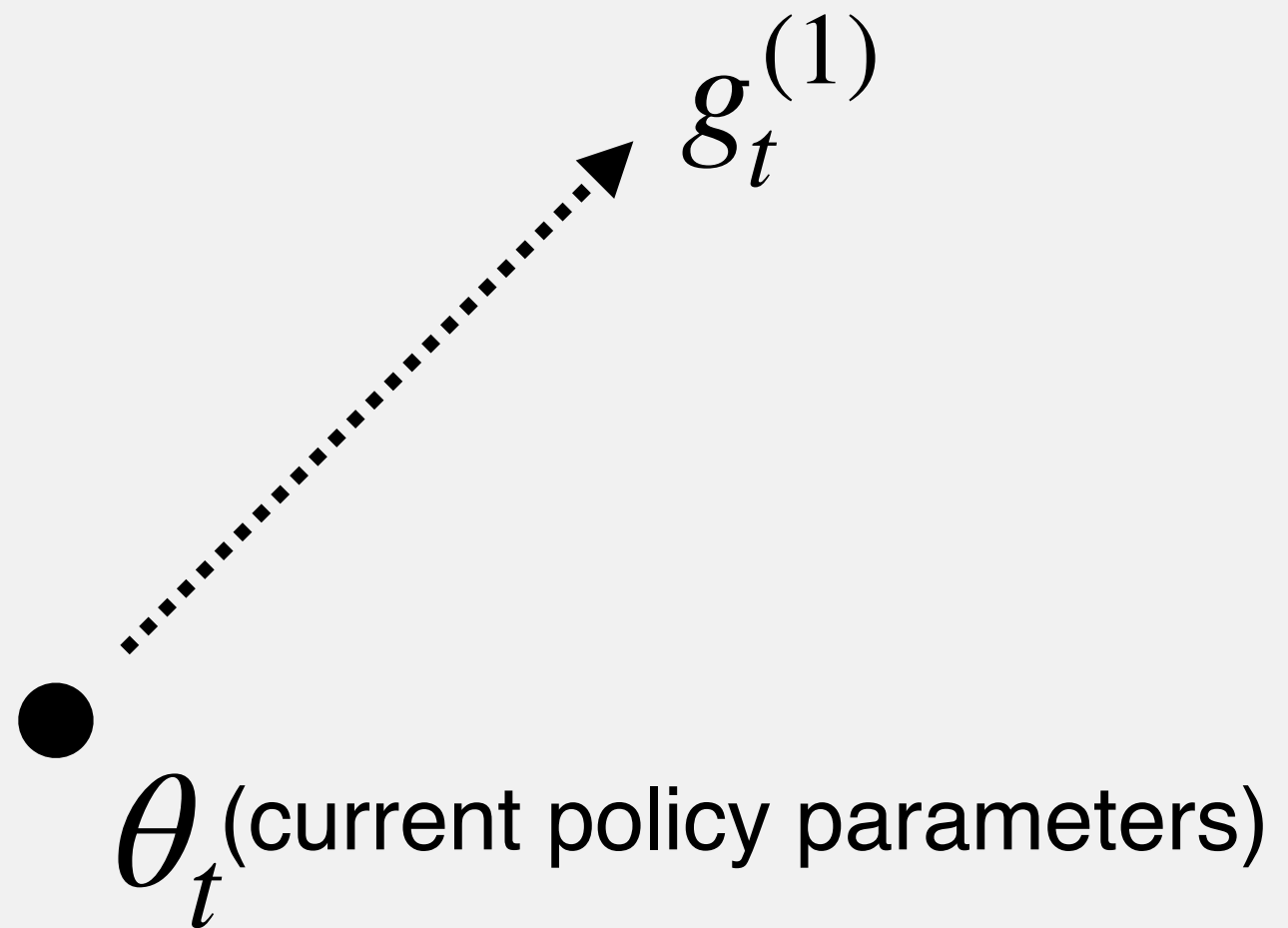
$$\mathbb{E}_{X \sim P}[X] \approx \frac{1}{N} \sum_{x_i \sim P} x_i$$

How well does this work?

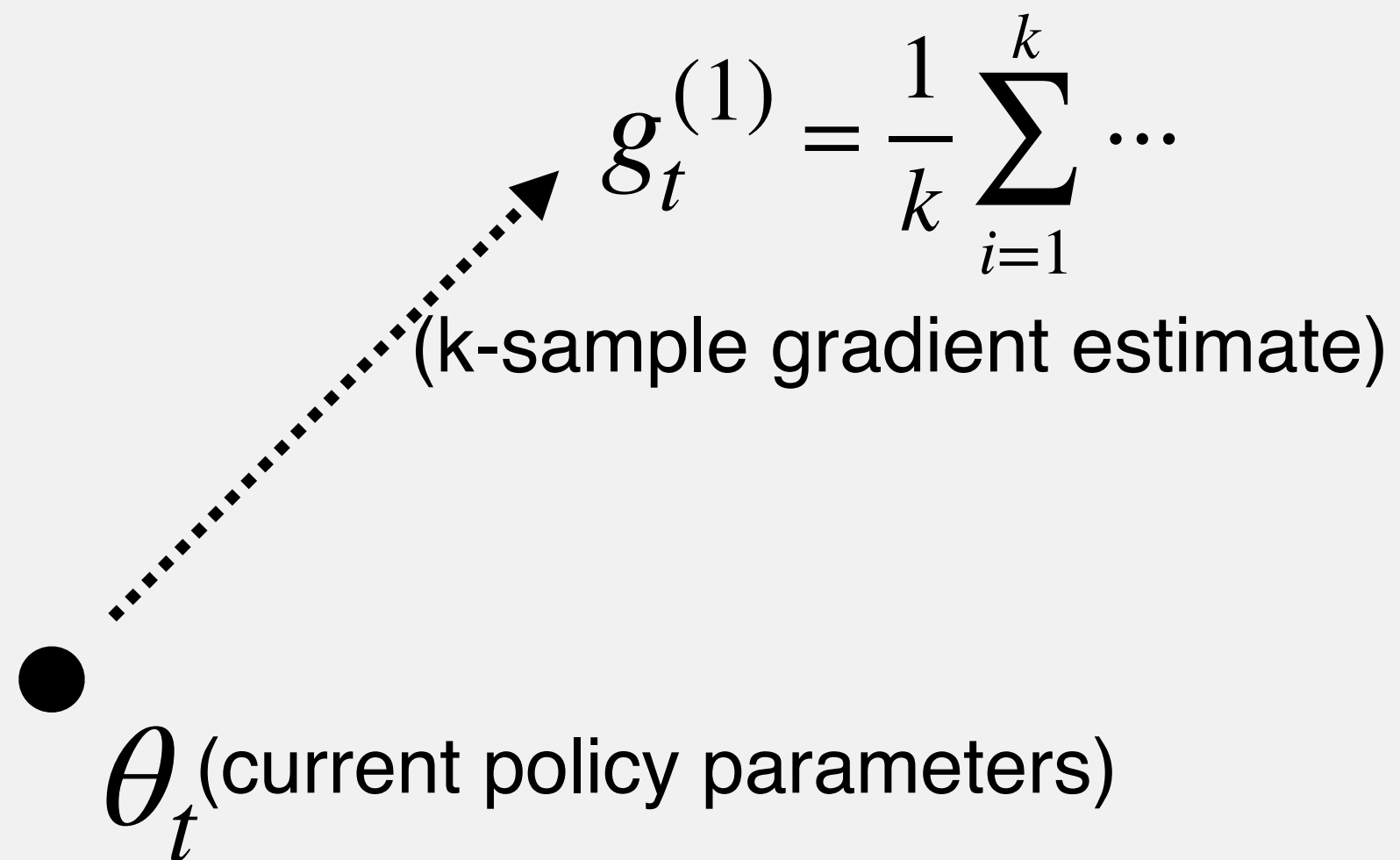
Gradient Estimation

- θ_t (current policy parameters)

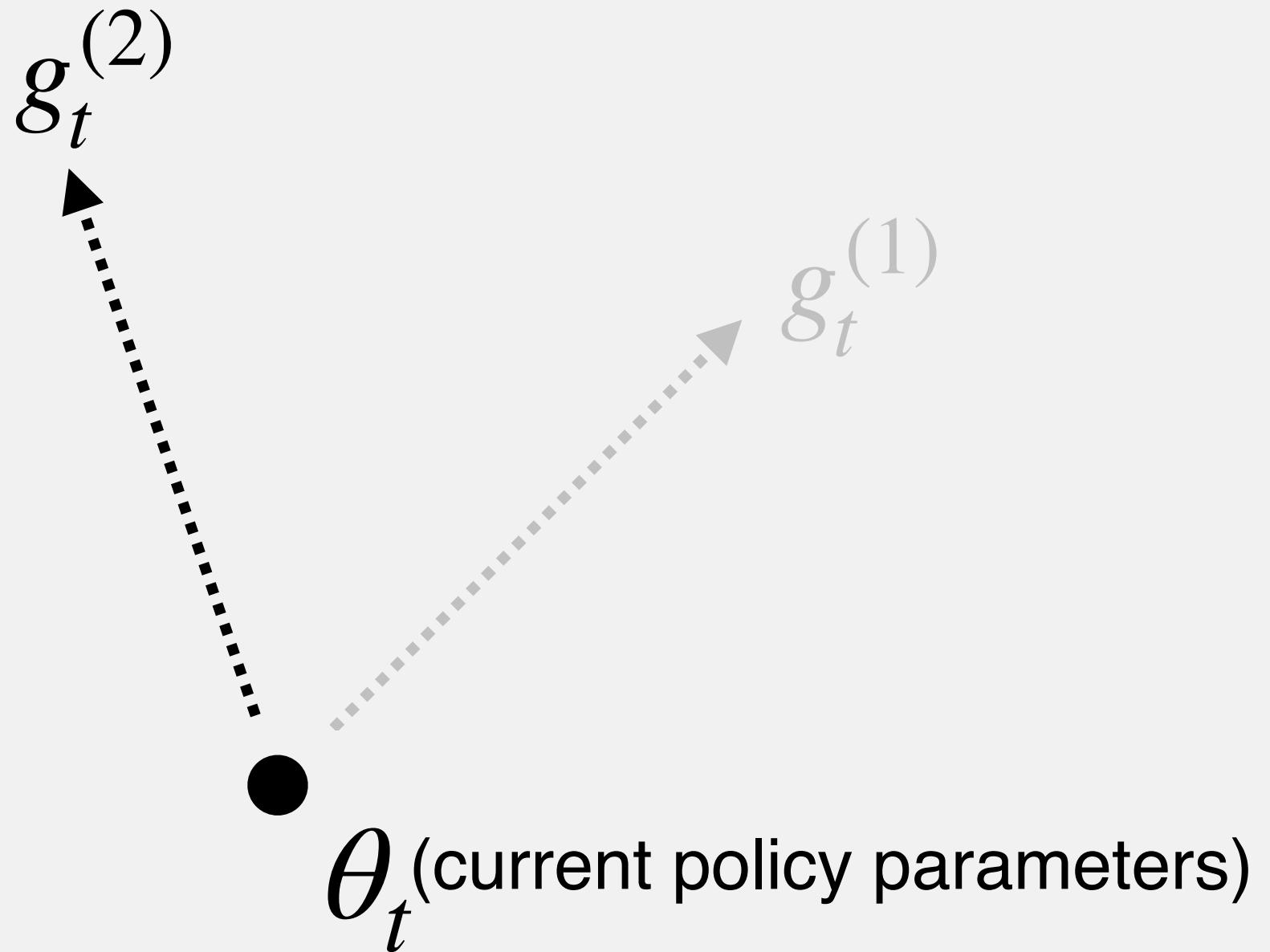
Gradient Estimation



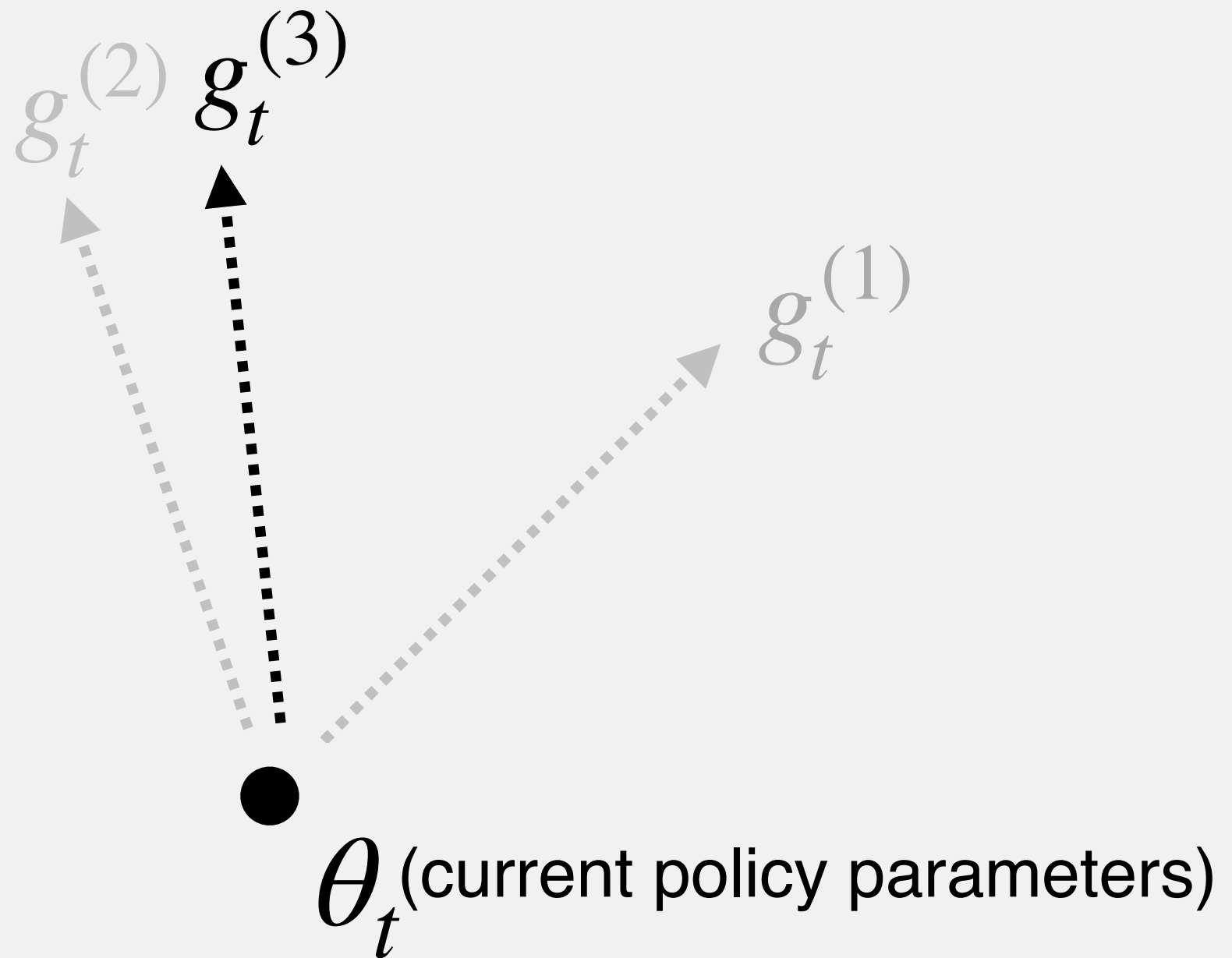
Gradient Estimation



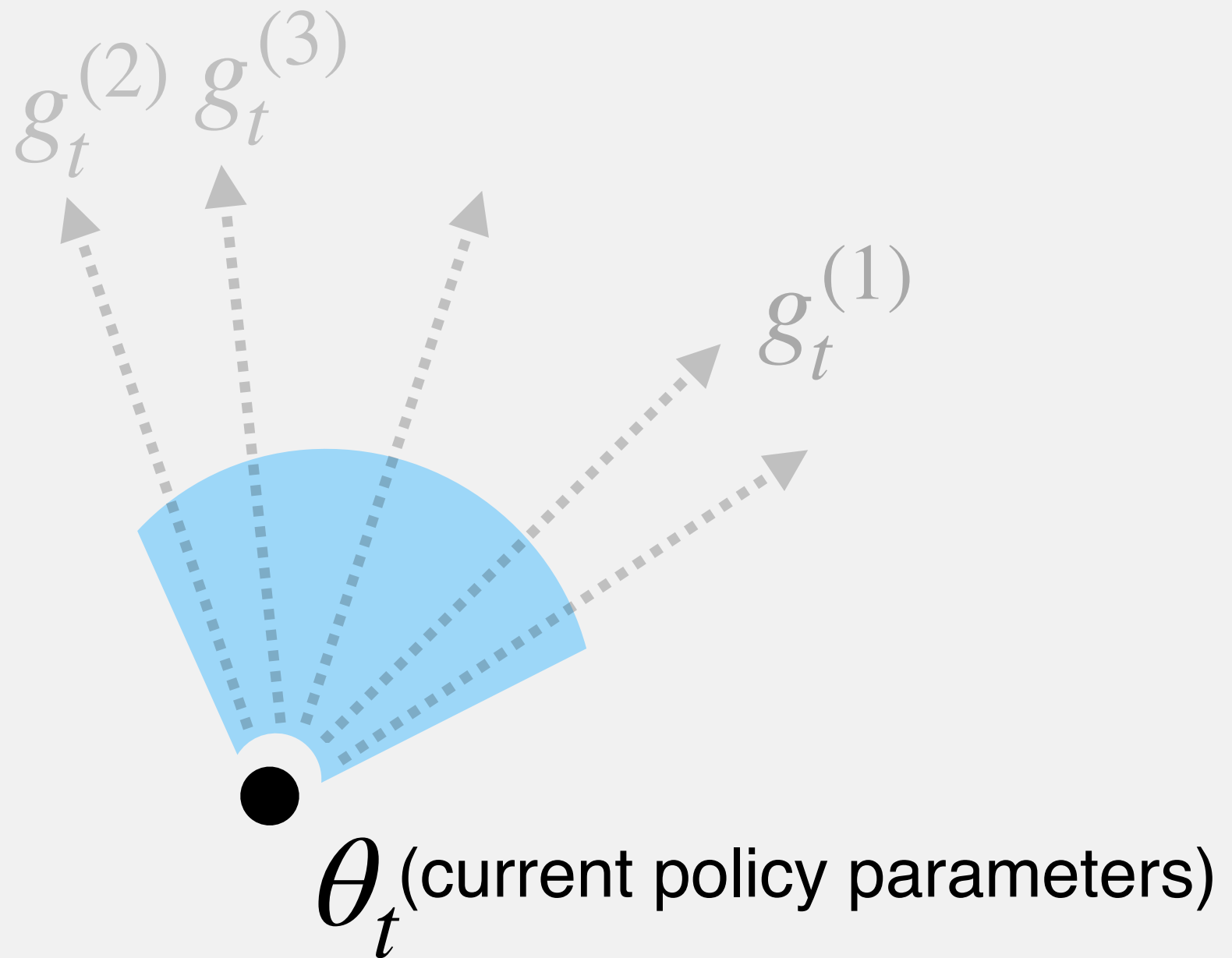
Gradient Estimation



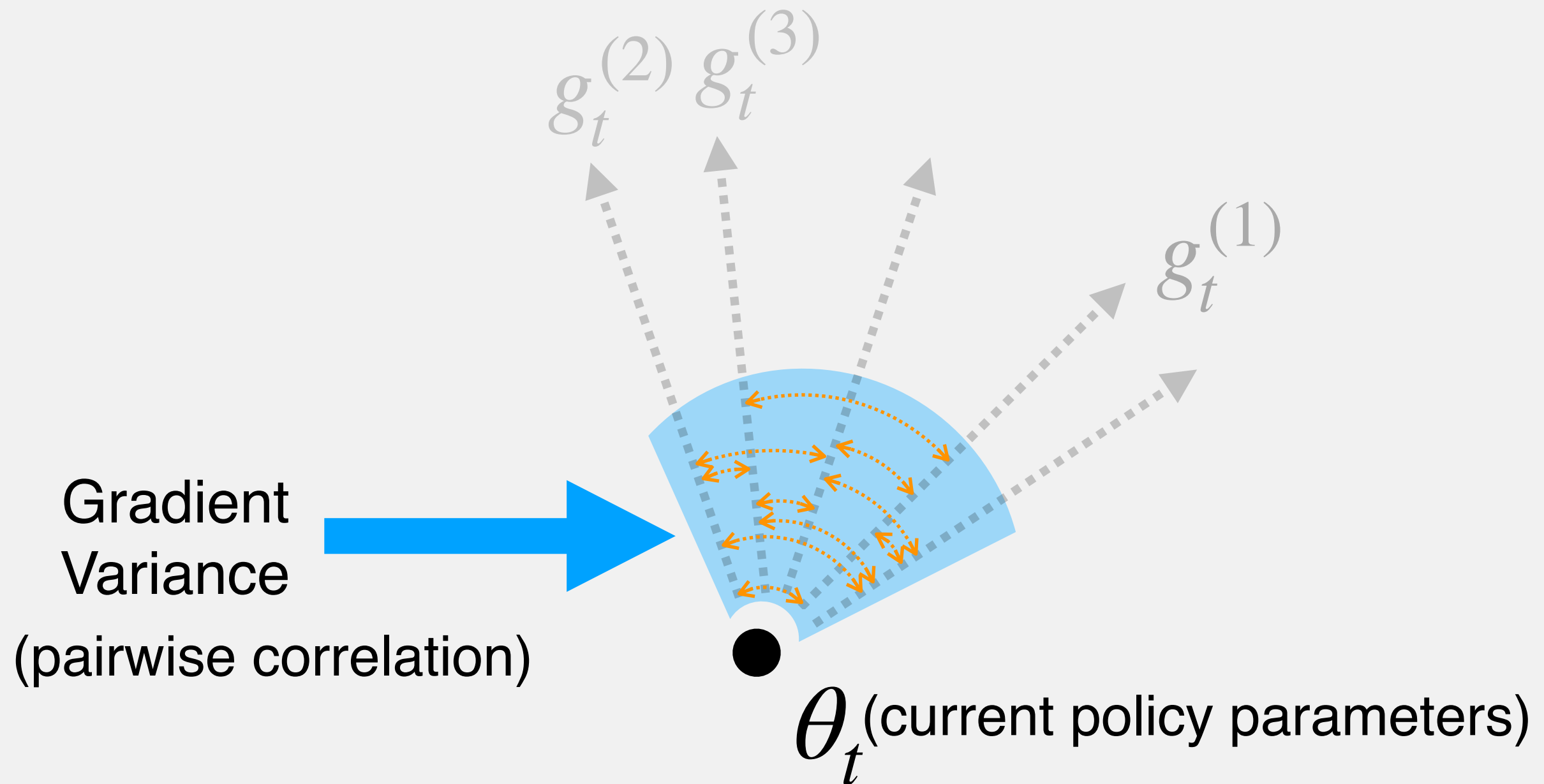
Gradient Estimation



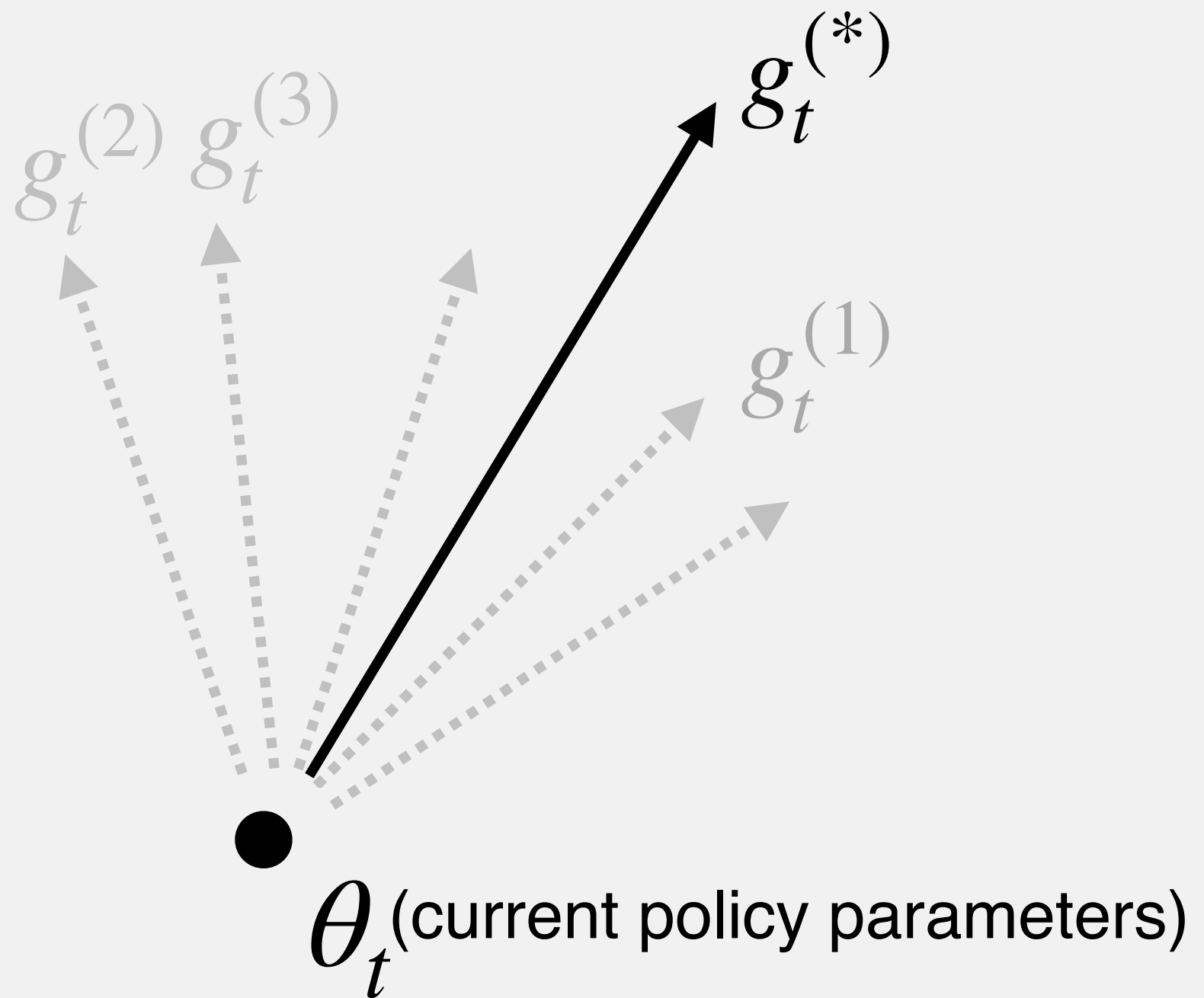
Gradient Estimation



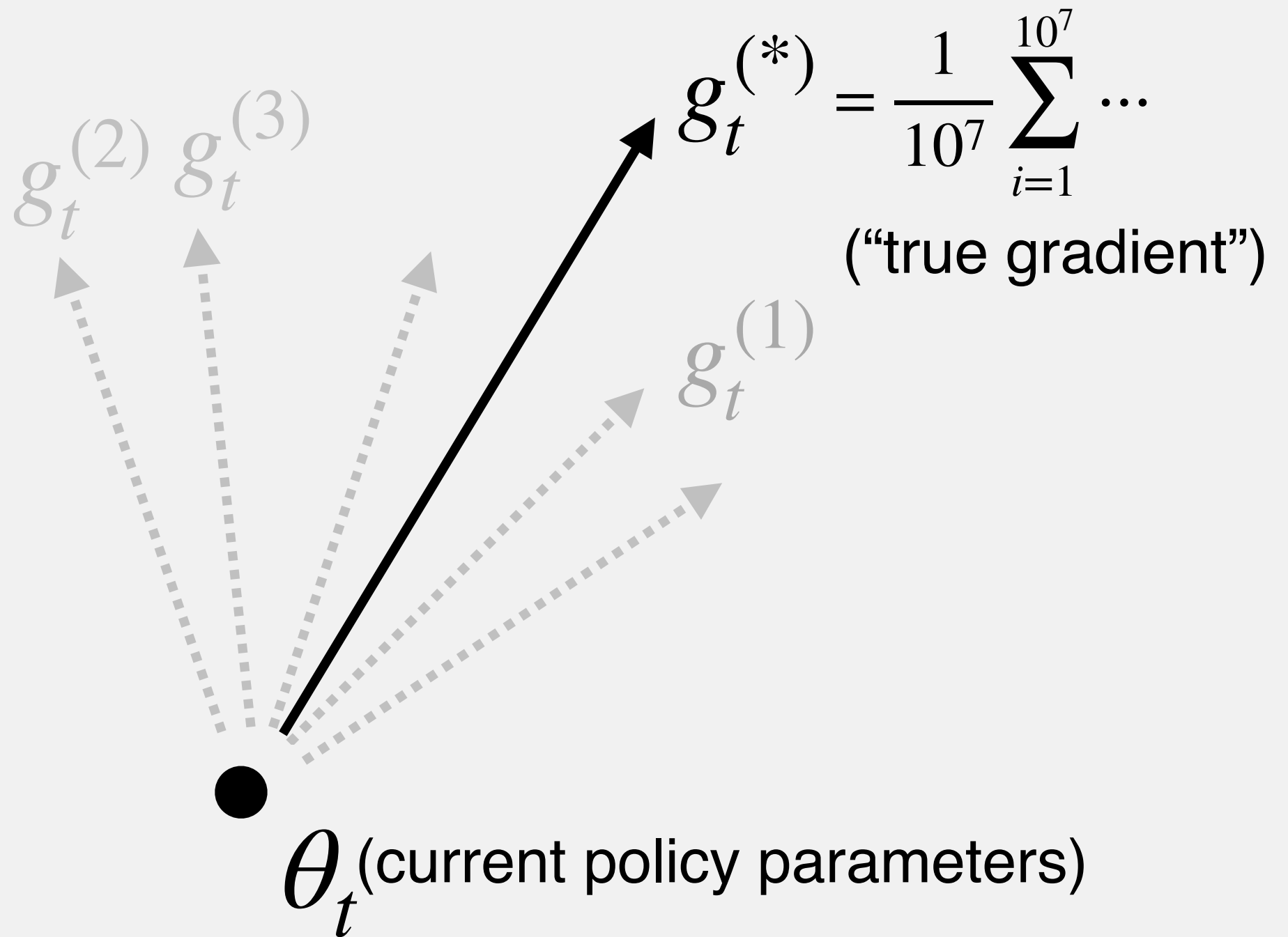
Gradient Estimation



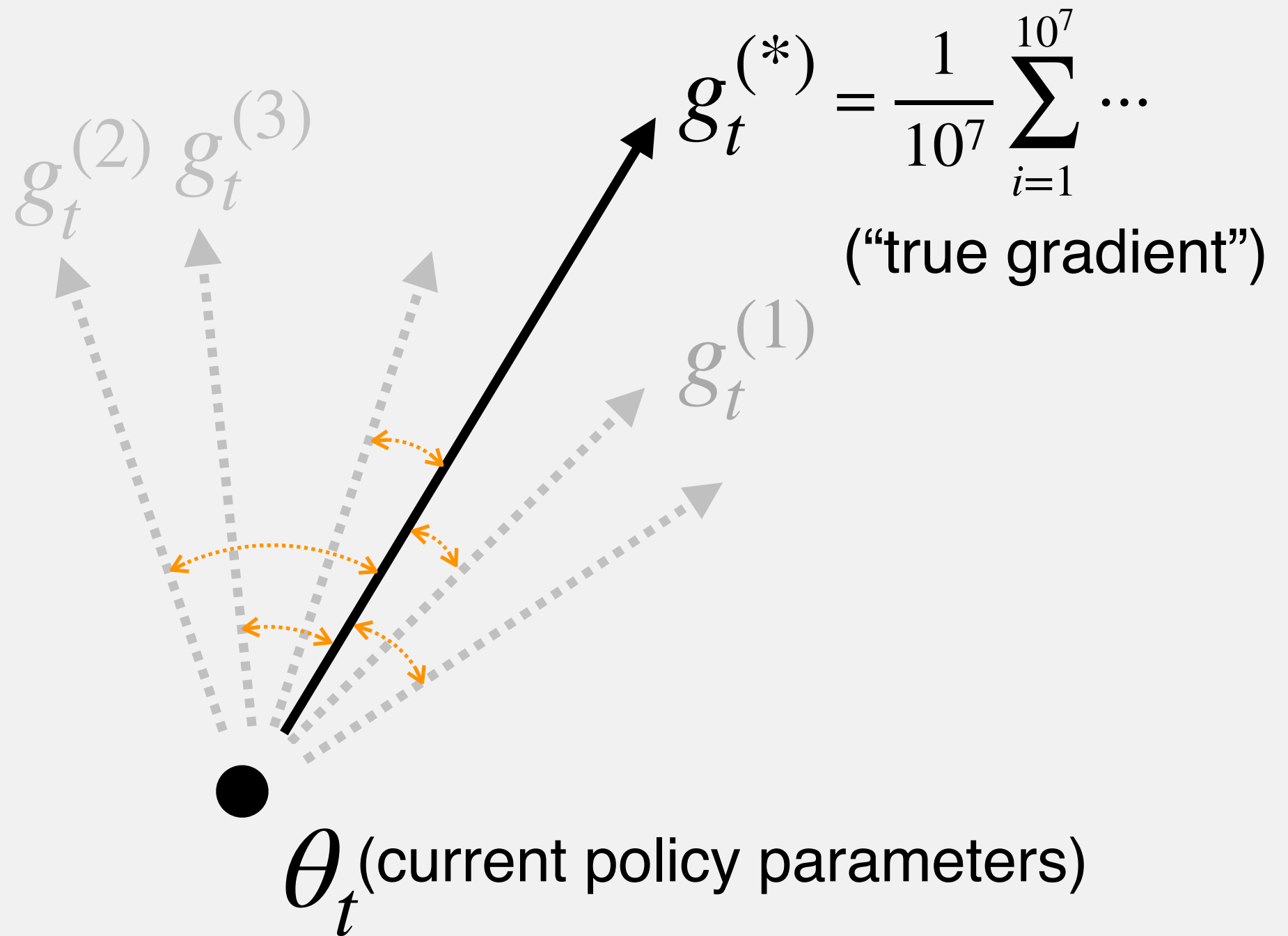
Gradient Estimation



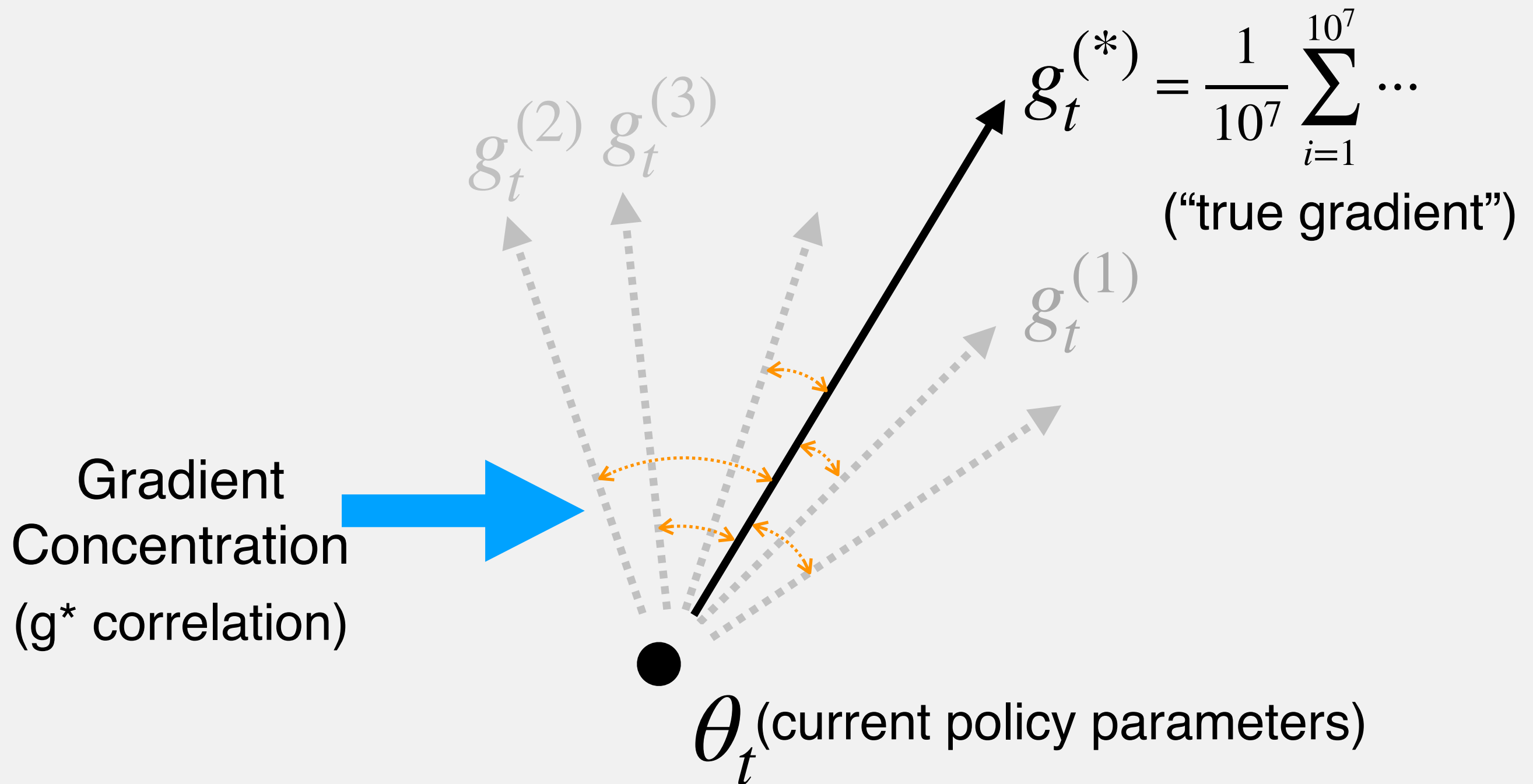
Gradient Estimation



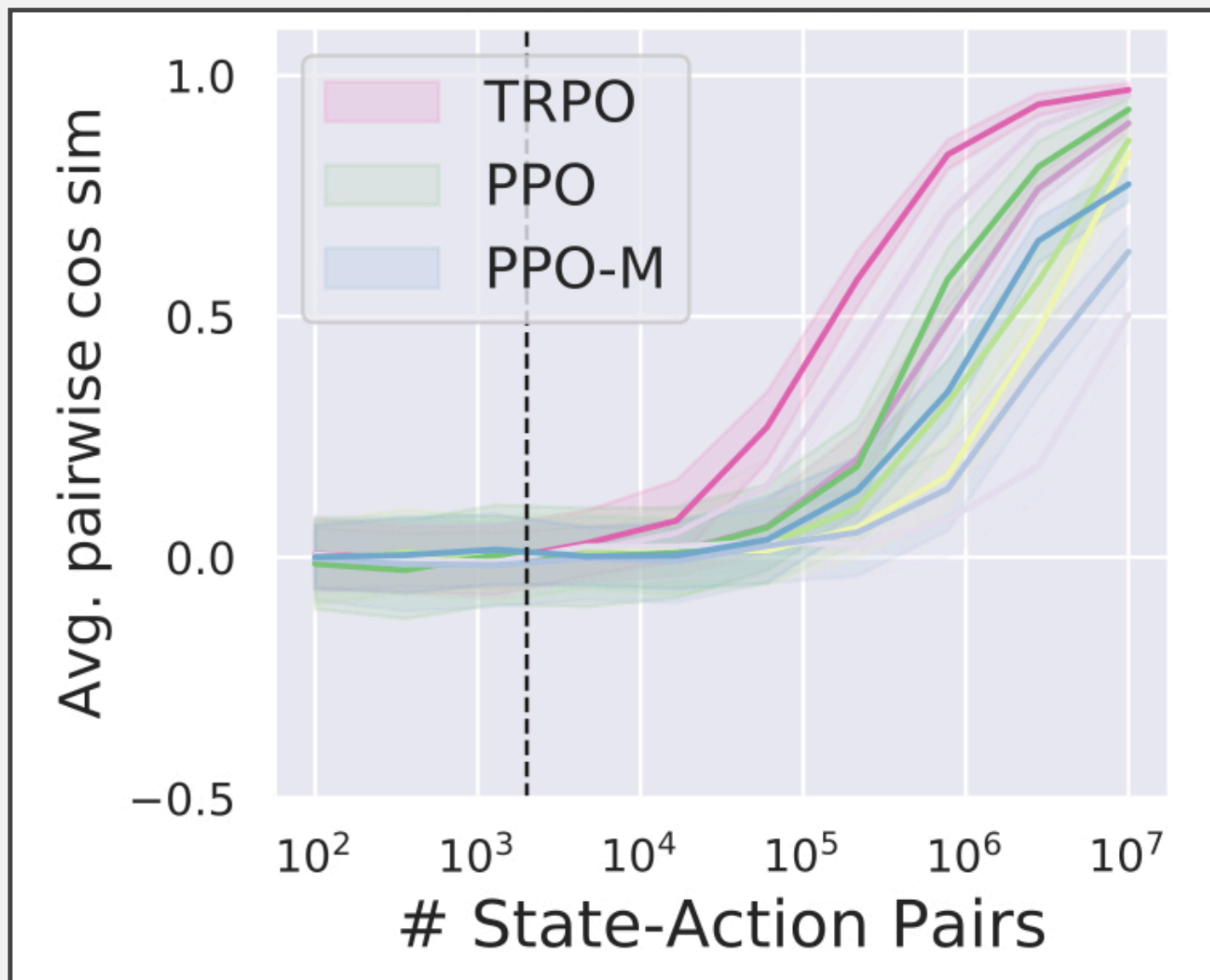
Gradient Estimation



Gradient Estimation

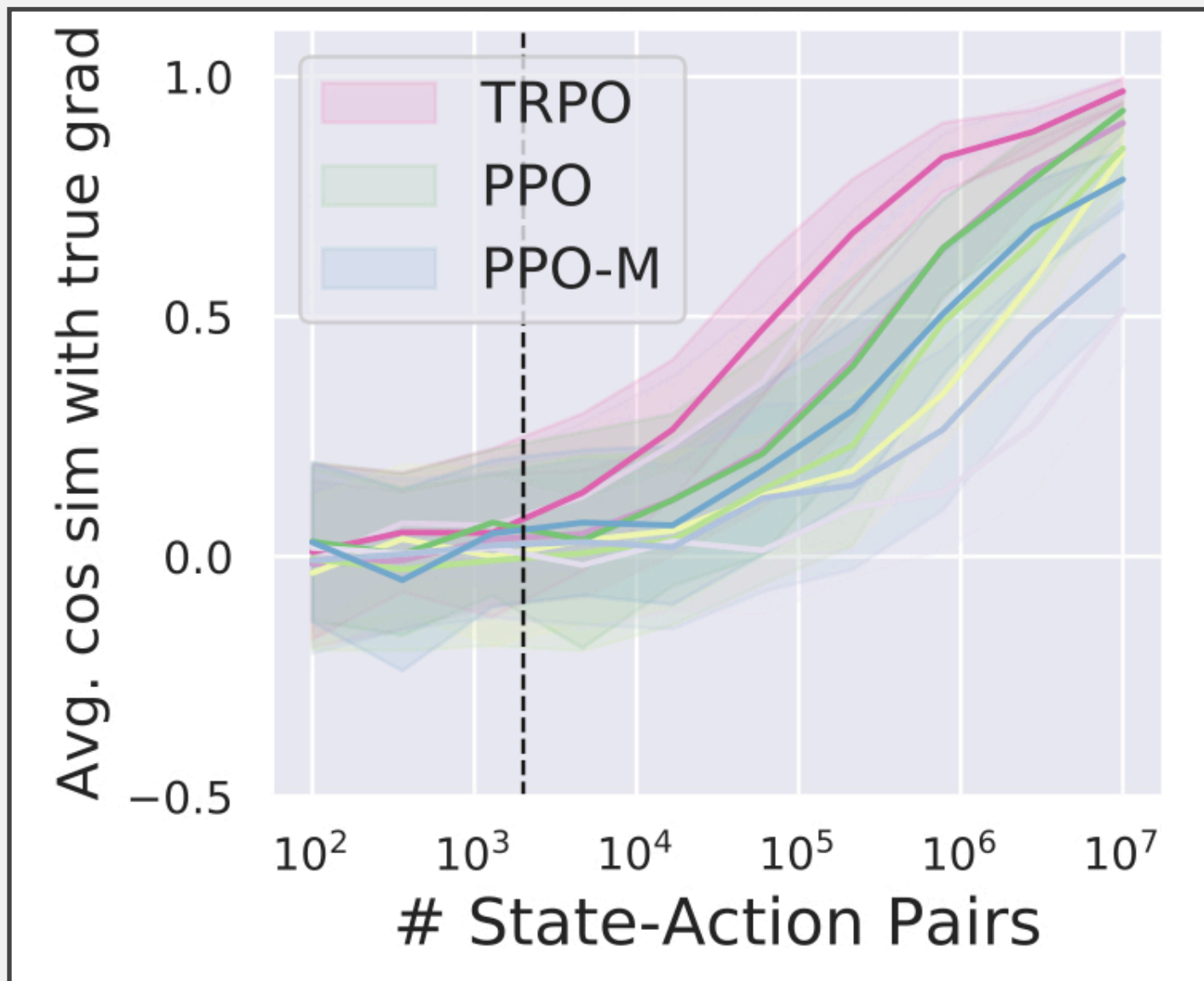


Gradient Variance



- ▶ Black line: relevant sample regime
- ▶ Gradients are less concentrated than they could be
- ▶ Less correlated for “harder” tasks, later iterations

Gradient Concentration



- ▶ Black line: relevant sample regime
- ▶ Gradients are less concentrated than they could be
- ▶ Less correlated for “harder” tasks, later iterations

Gradient Estimation

- ▶ No good understanding of training dynamics
 - ▶ How does variance influence optimization?
 - ▶ Can we use insights from stochastic opt?
- ▶ Missing a link from reliability to sample size

Value Prediction

Value Prediction

Policy gradient is a sum weighted by returns

Value Prediction

Policy gradient is a sum weighted by returns

Concentration is hindered by high variance

Value Prediction

Policy gradient is a sum weighted by returns

Concentration is hindered by high variance

Observation: If we can estimate the value of a state, can significantly lower variance

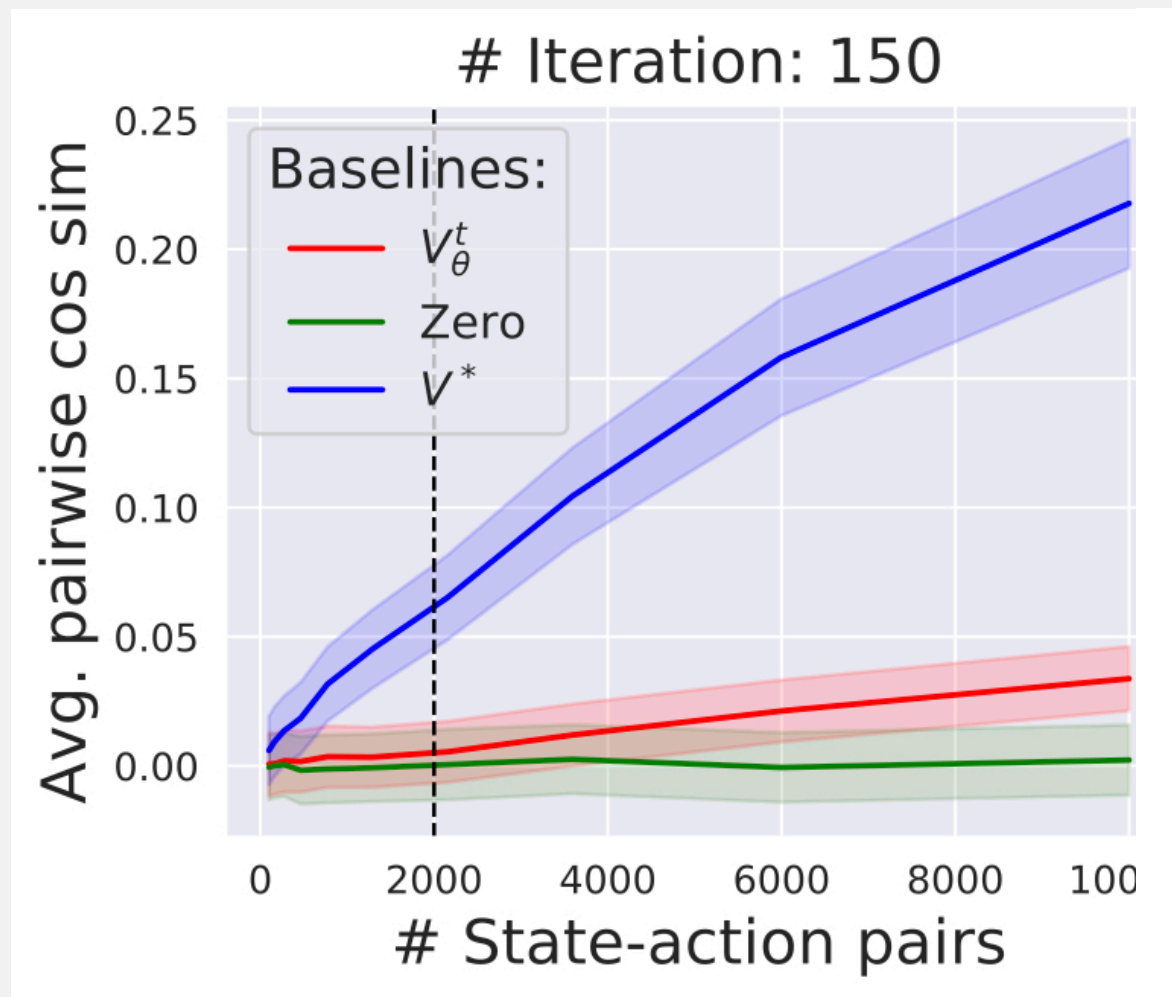
Value Prediction

Variance reduction needs good value estimates

In Deep RL, values come from a neural network

To what degree do we actually reduce variance?

Value Prediction



True value function

Agent's value function

No value function

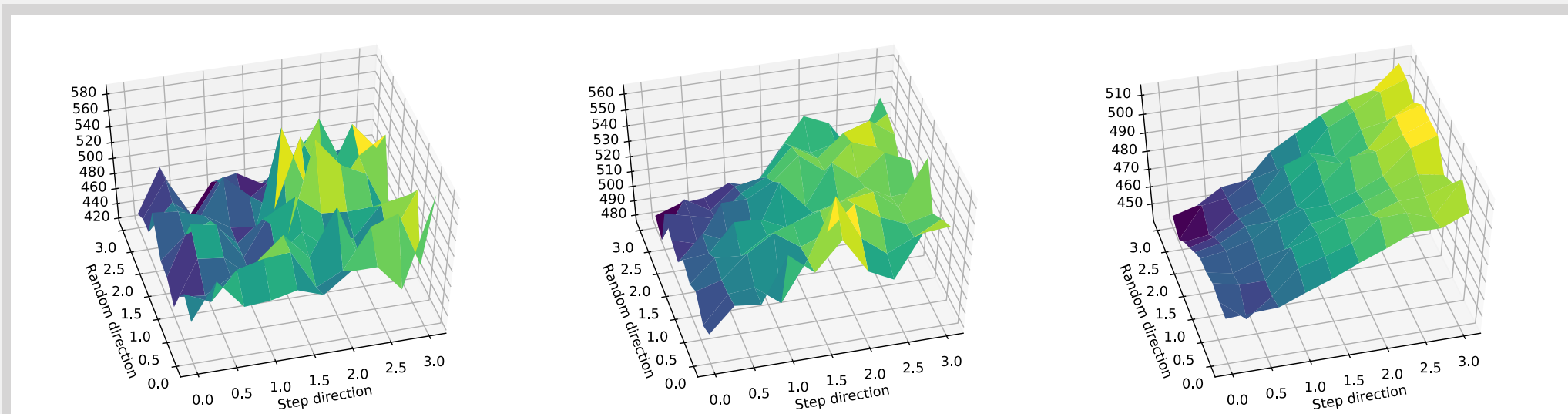
The agent's value network helps in variance reduction, but not nearly as much as the true value

Value Prediction

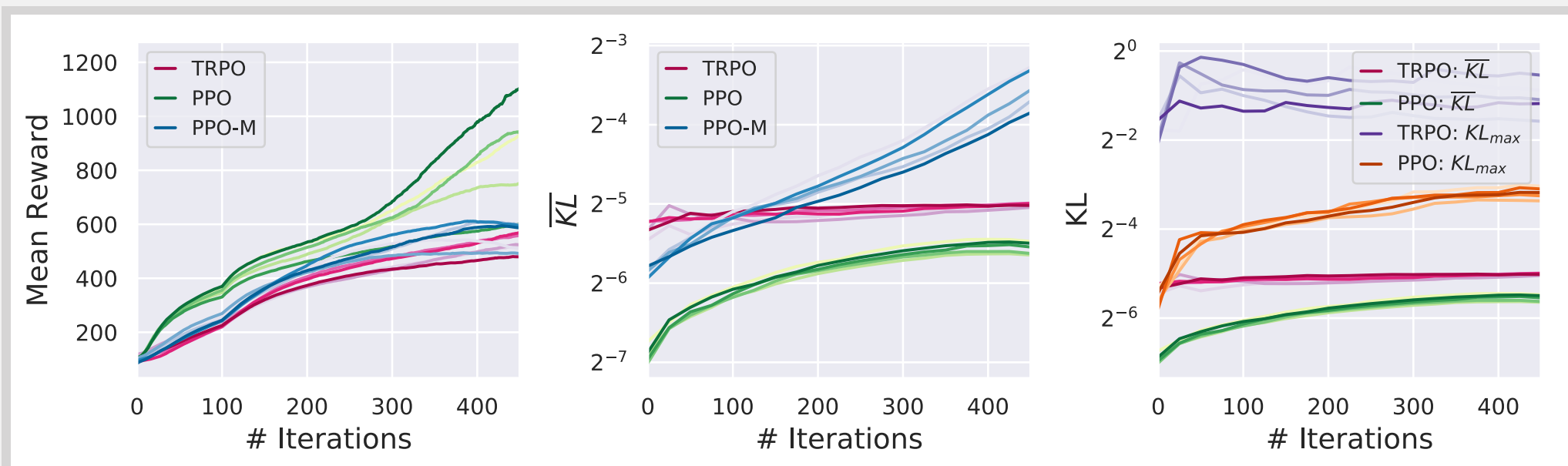
- ▶ Might look small, but using a value network makes big difference
- ▶ How would using the true value affect training?
- ▶ Can we get better value estimates (info barrier)

More analysis (from the paper)

Similar conclusions from:



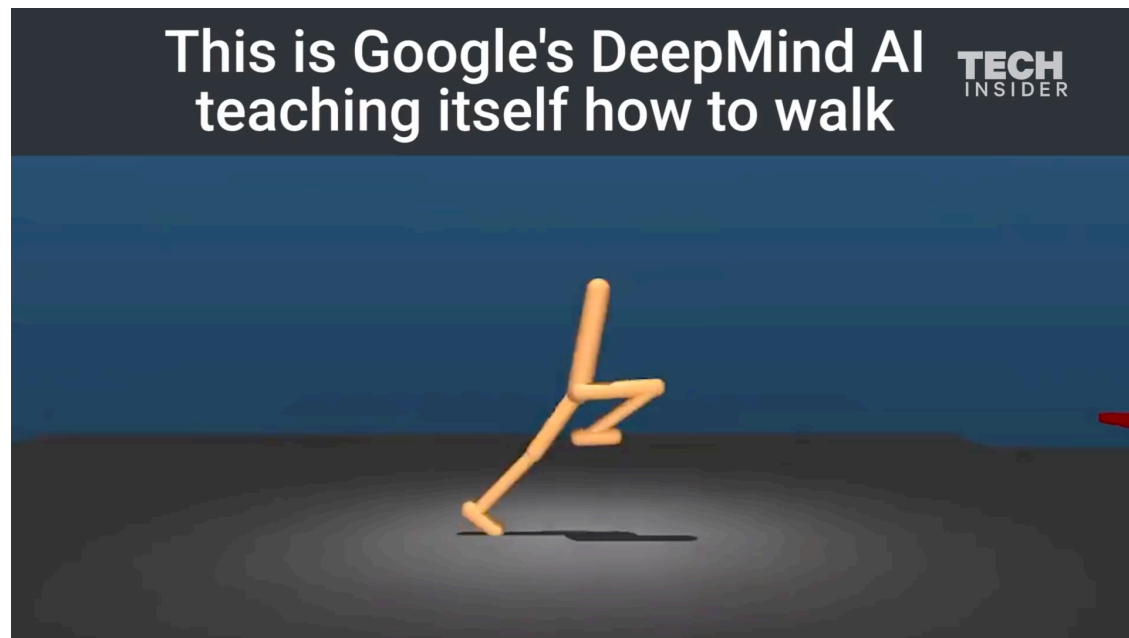
Optimization landscape is often noisy/misleading



Enforcement of “trust regions” has theoretical and practical caveats

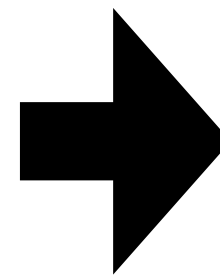
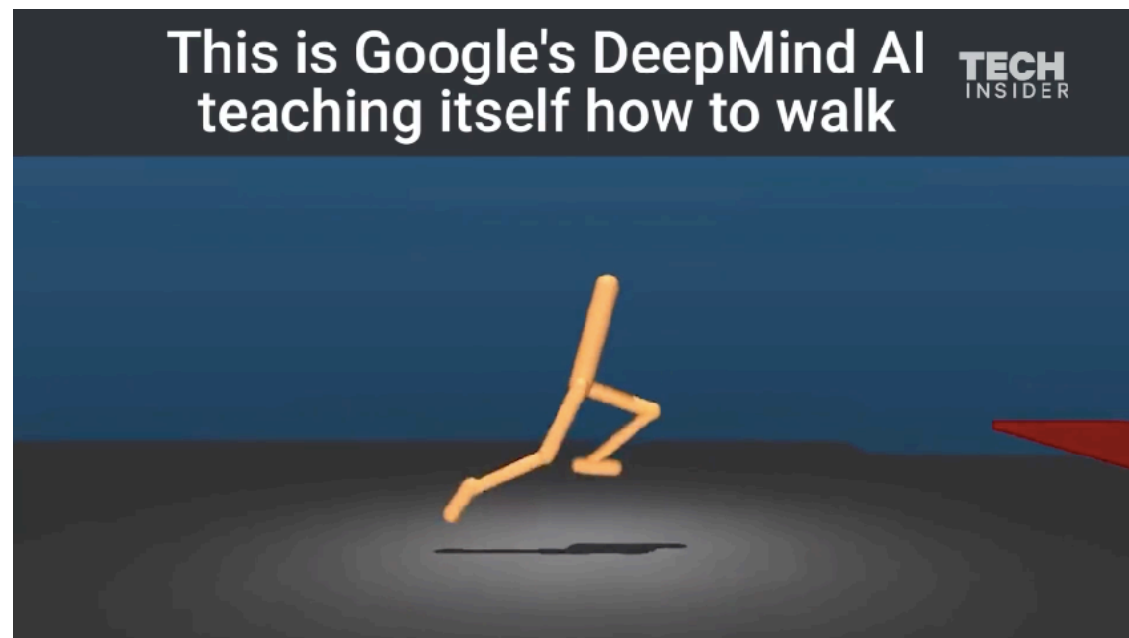
Does AI translate from simulation to reality?

Simulation

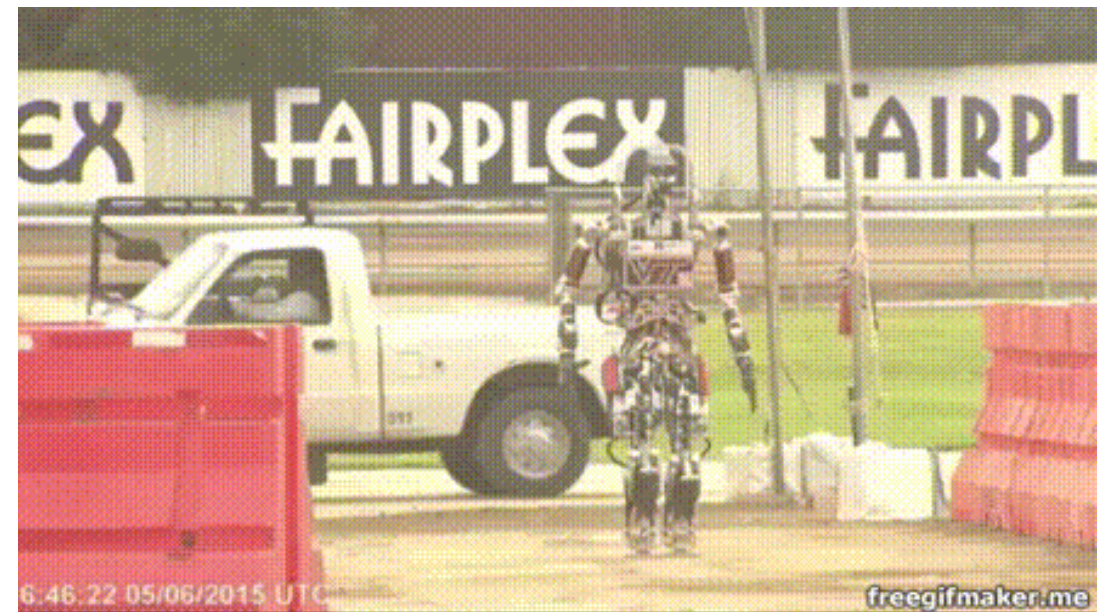


Does AI translate from simulation to reality?

Simulation



Reality



Also: Are we even optimizing the right thing?

Takeaways

How do we proceed?

- ▶ Reconciling RL with our conceptual framework
 - ▶ How predictive are theoretical principles in practice?
 - ▶ What is the right way to model the RL setting?
- ▶ Rethinking primitives for modern settings
 - ▶ How do we deal with high dimensionality?
 - ▶ Delayed rewards?
- ▶ Better evaluation for RL systems
 - ▶ Benchmarks don't capture reliability, safety, or robustness of RL agents