Evolution Strategies as a Scalable Alternative to Reinforcement Learning

Tim Salimans, Jonathan Ho, Peter Chen, Ilya Sutskever

OpenAI

OpenAI

- OpenAI’s mission is to build safe AGI, and ensure AGI's benefits are as widely and evenly distributed as possible.

OpenAI

- We're a non-profit research company. Our full-time staff of 60 researchers and engineers is dedicated to working towards our mission regardless of the opportunities for selfish gain which arise along the way.

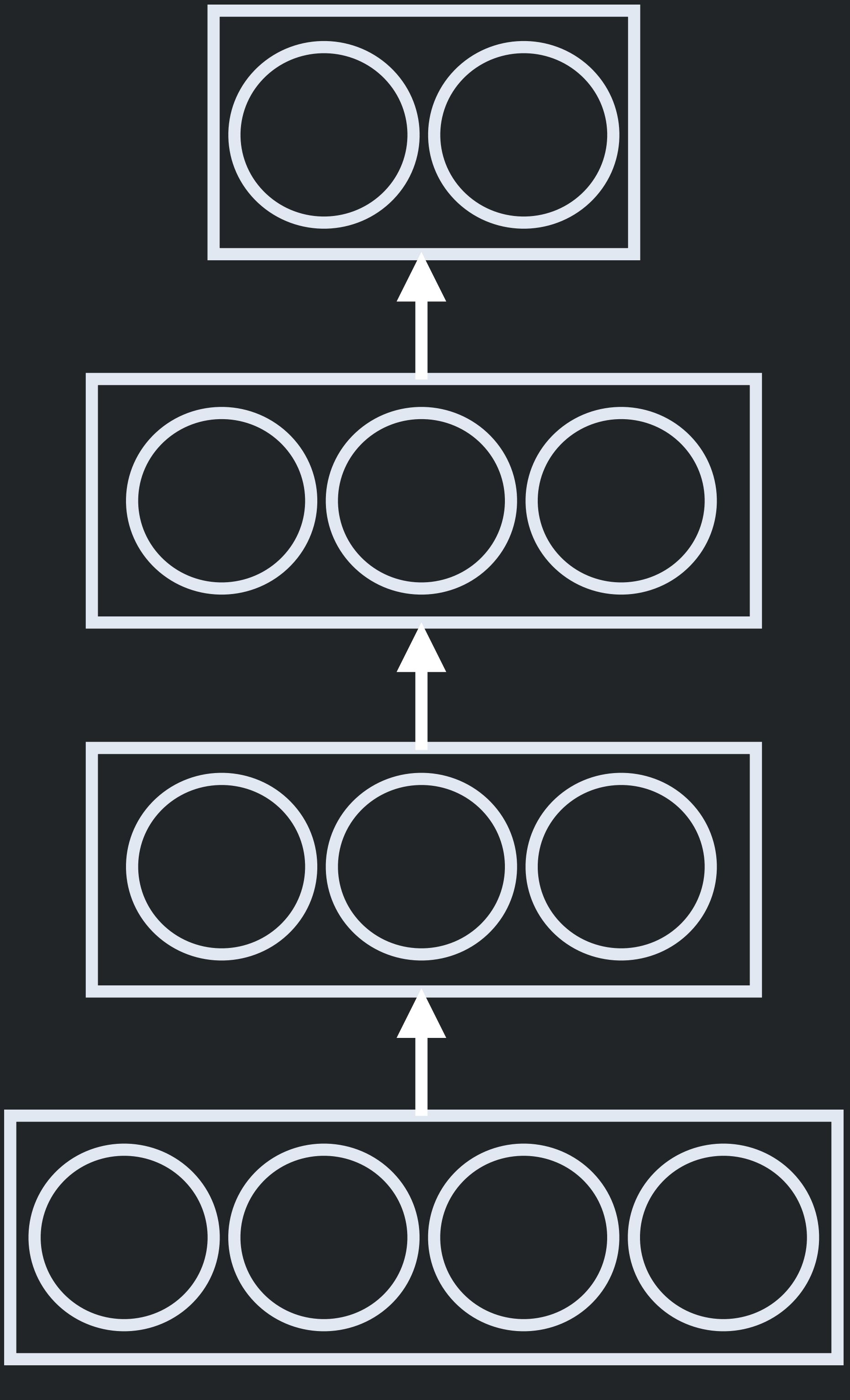
OpenAI

- We focus on long-term research, working on problems that require us to make fundamental advances in AI capabilities. By being at the forefront of the field, we can influence the conditions under which AGI is created. As Alan Kay said, "The best way to predict the future is to invent it."

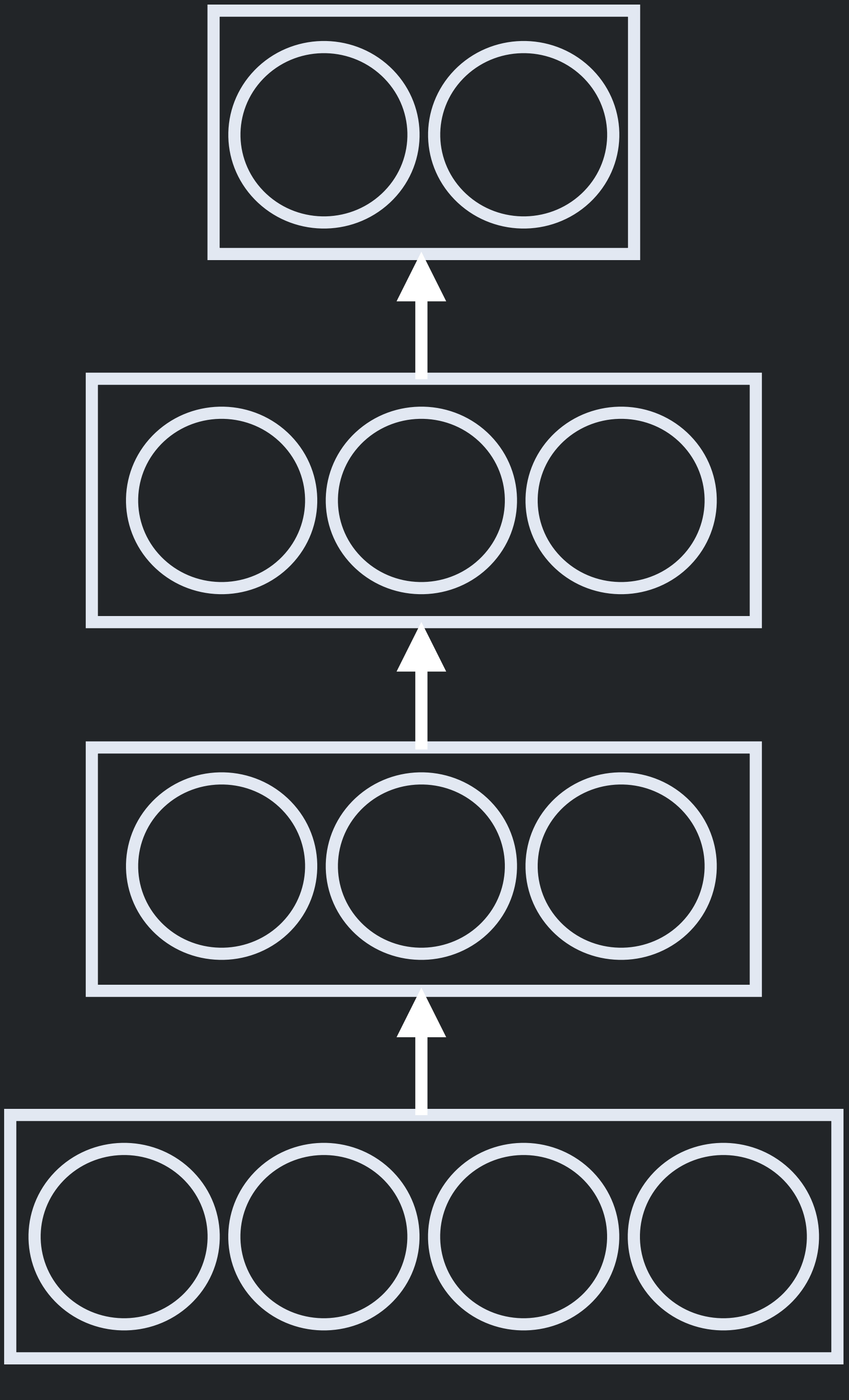
OpenAI

Learning algorithms that work

- Supervised back propagation



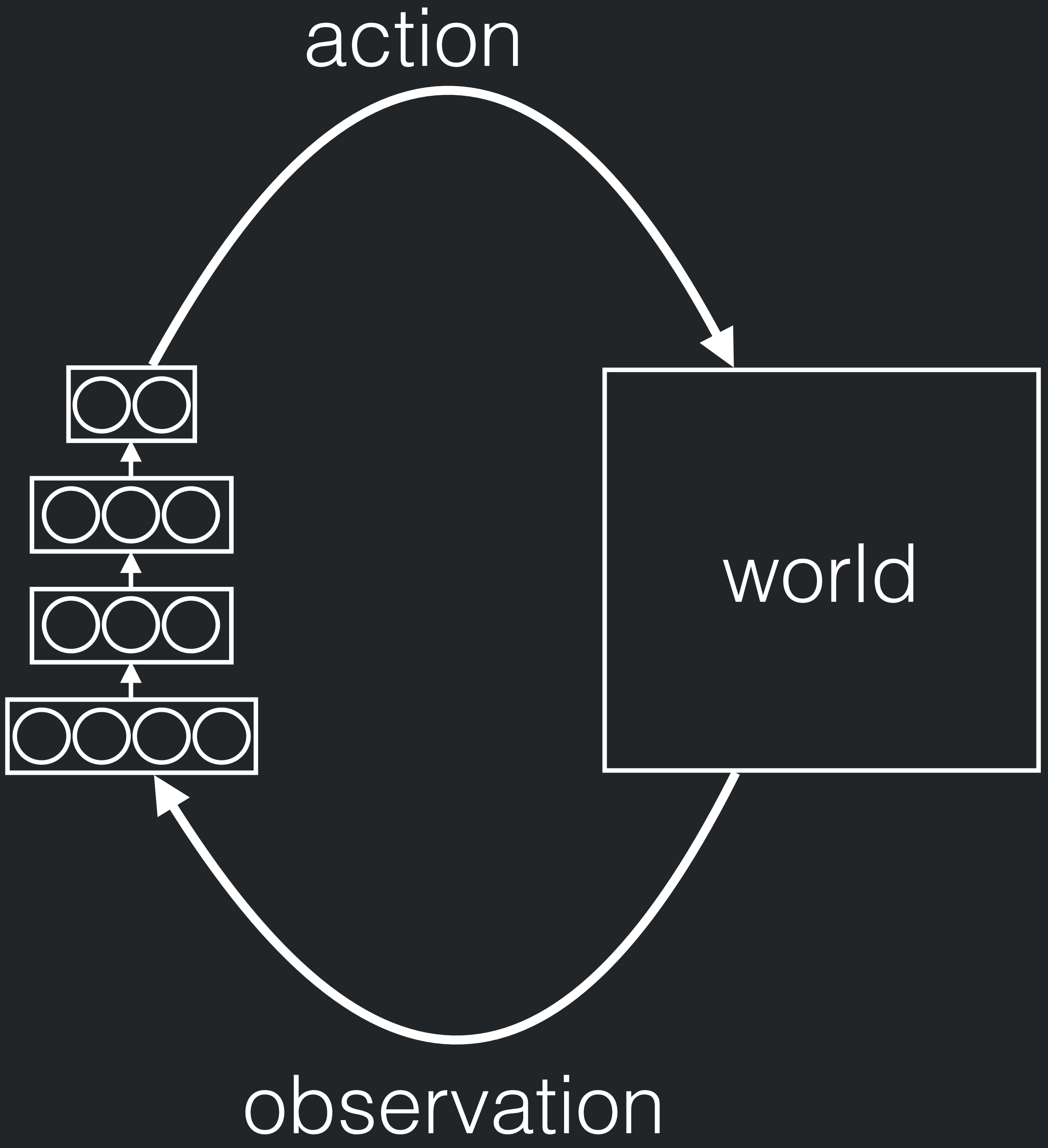
Learning algorithms that work



- It actually works!

Reinforcement Learning?

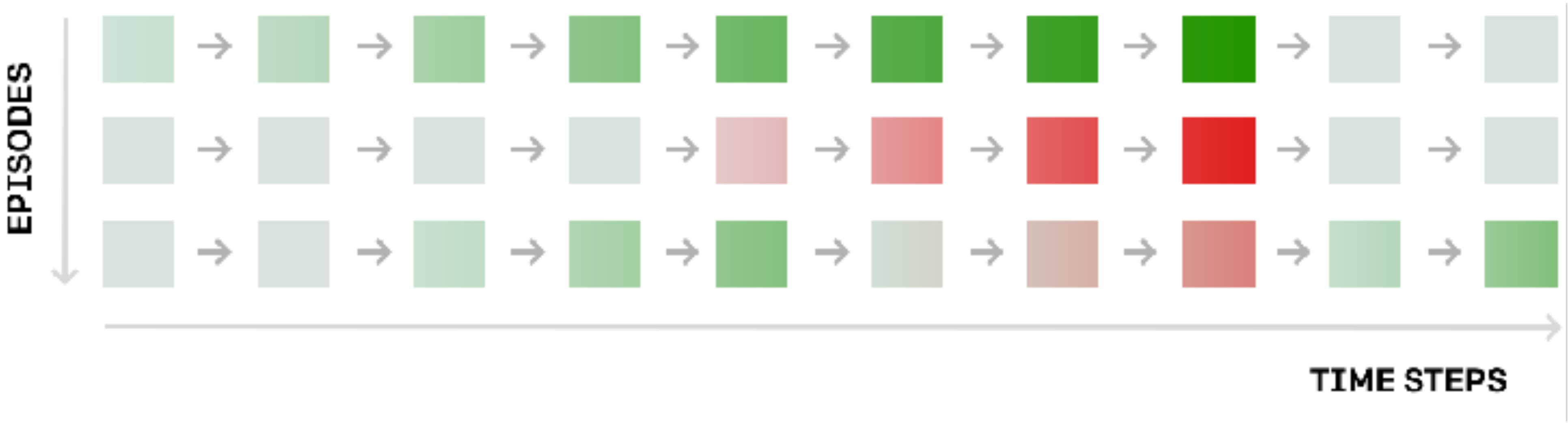
- Reinforcement learning is the right problem
- Do we have excellent RL algorithms?



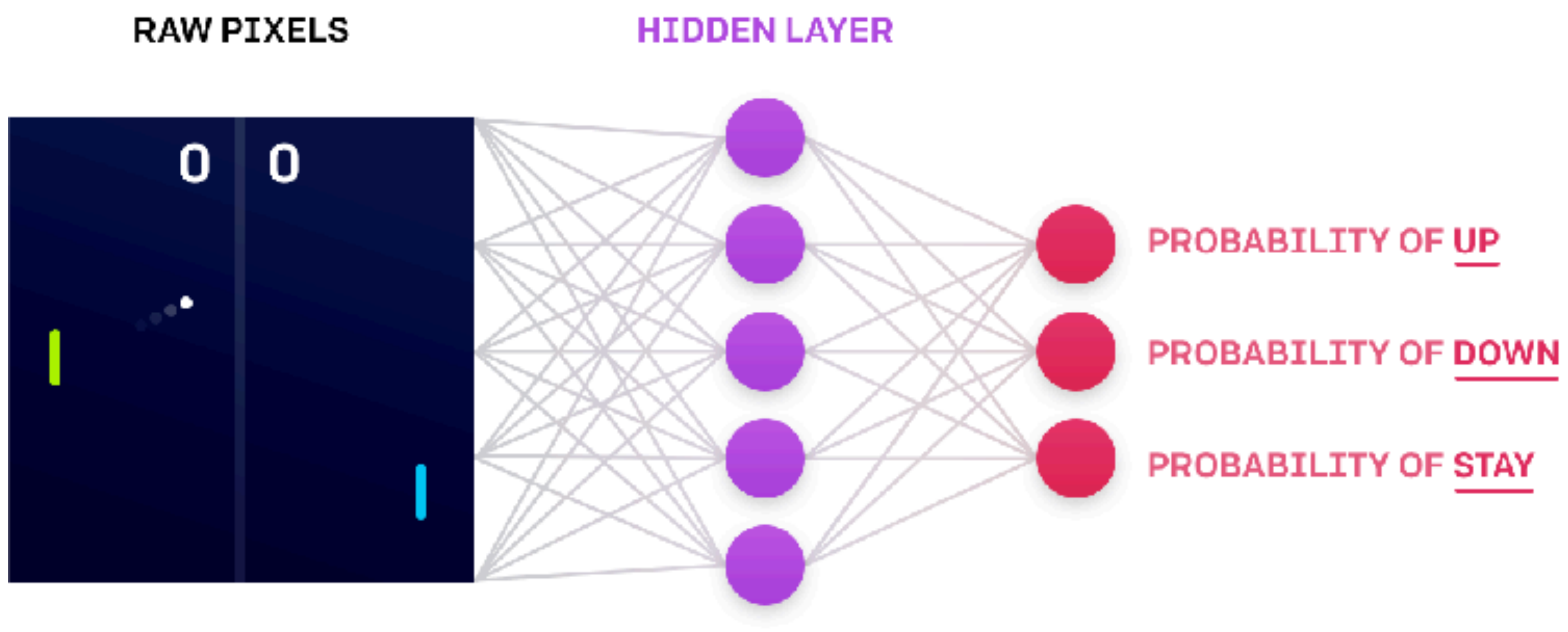
Reinforcement Learning?

- Reinforcement Learning is the right problem

- RL algorithms can train agents on games and robots, which is really exciting



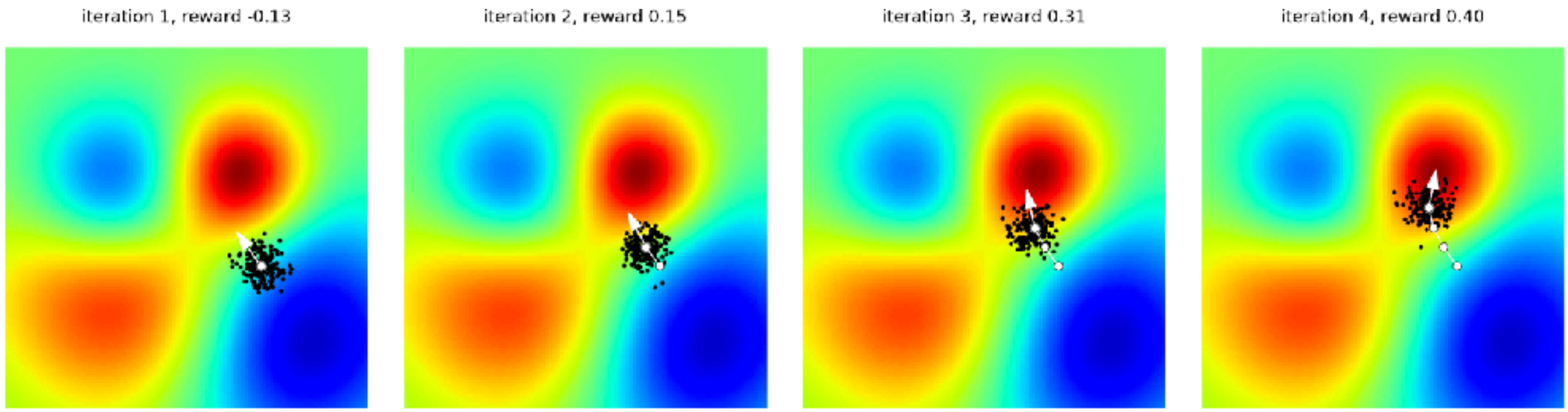
Why is RL exciting?



- Super good RL algorithm brings us closer to AGI
- There exist RL reasonably general RL algorithms

Evolution Strategies

- Simplest algorithm imaginable:
  - Add noise to the parameters
  - If the result improves, keep the change
  - Repeat



Evolution Strategies

- Neural networks have millions of parameters
- There’s no chance for this kind of random hillclimbing to succeed

Surprise!

- Evolution Strategies is competitive with today’s RL algorithms on standard benchmarks

Evolution Strategies

- Evolution Strategies (randomized finite differences, simultaneous perturbation algorithm, …) is a very old, very well-studio algorithm

- Key contributions:

  - Show that the algorithm is competitive with today's existing RL algorithms

  - Show that it parallelizes *extremely well*

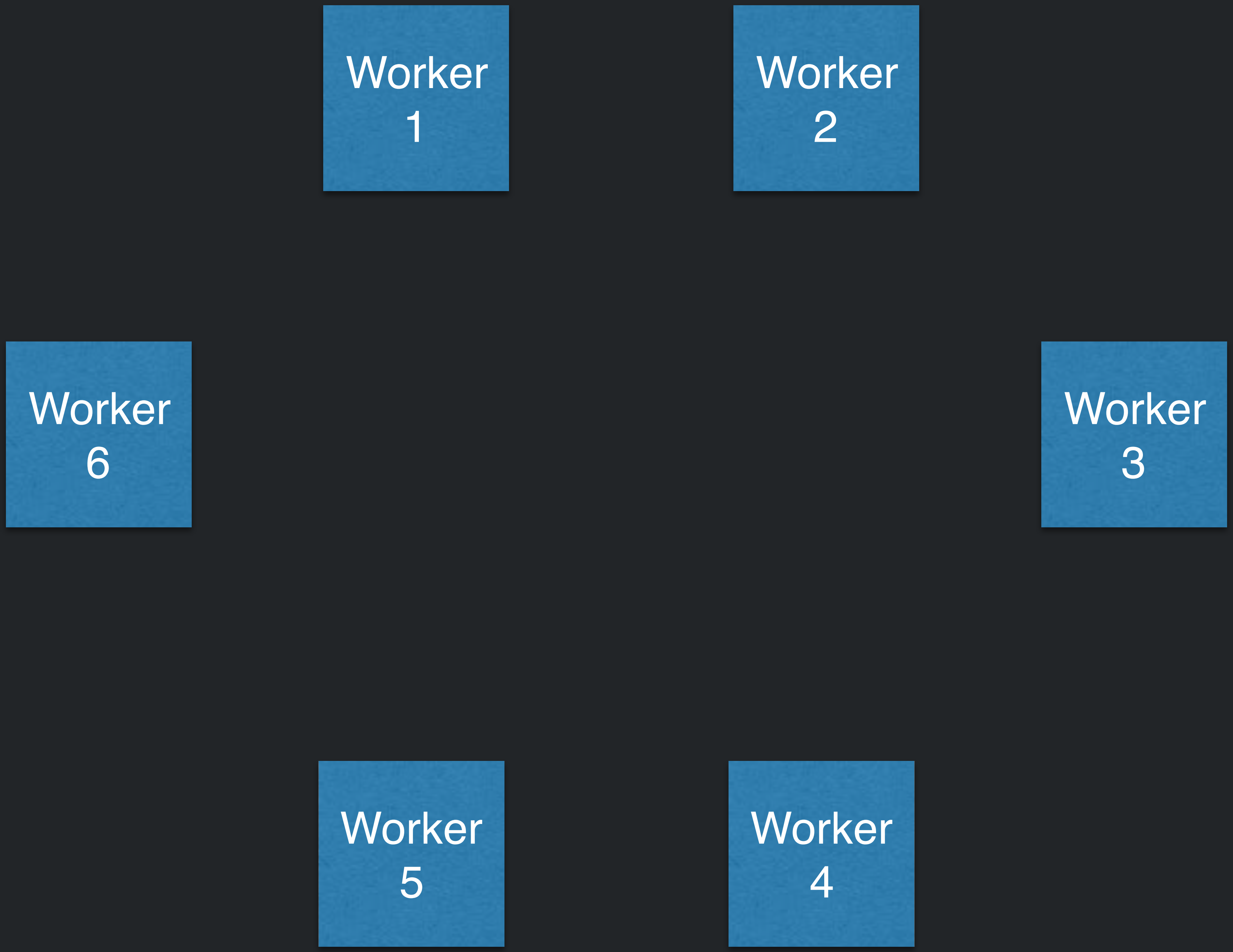
Parallelization

- ES parallelizes extremely well

Parallelization

- You have a bunch of workers
- They all try on different random noise
- Then they report how good the random noise was
- But they *don't need to communicate the noise vector*
- Because they know each other's seeds

Distributed Deep Learning



Worker 1

Worker 2

Worker 6

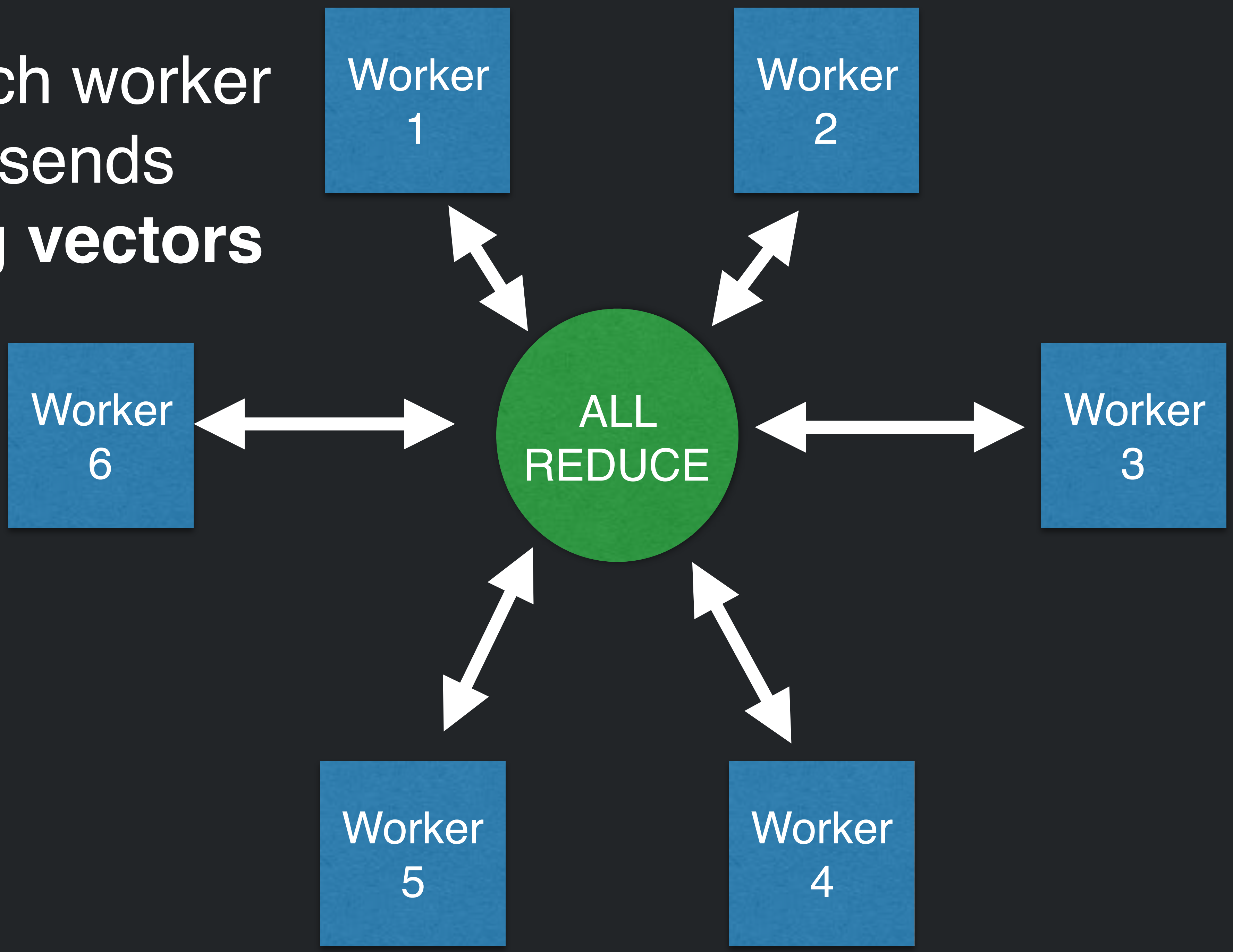
Worker 3

Worker 5

Worker 4

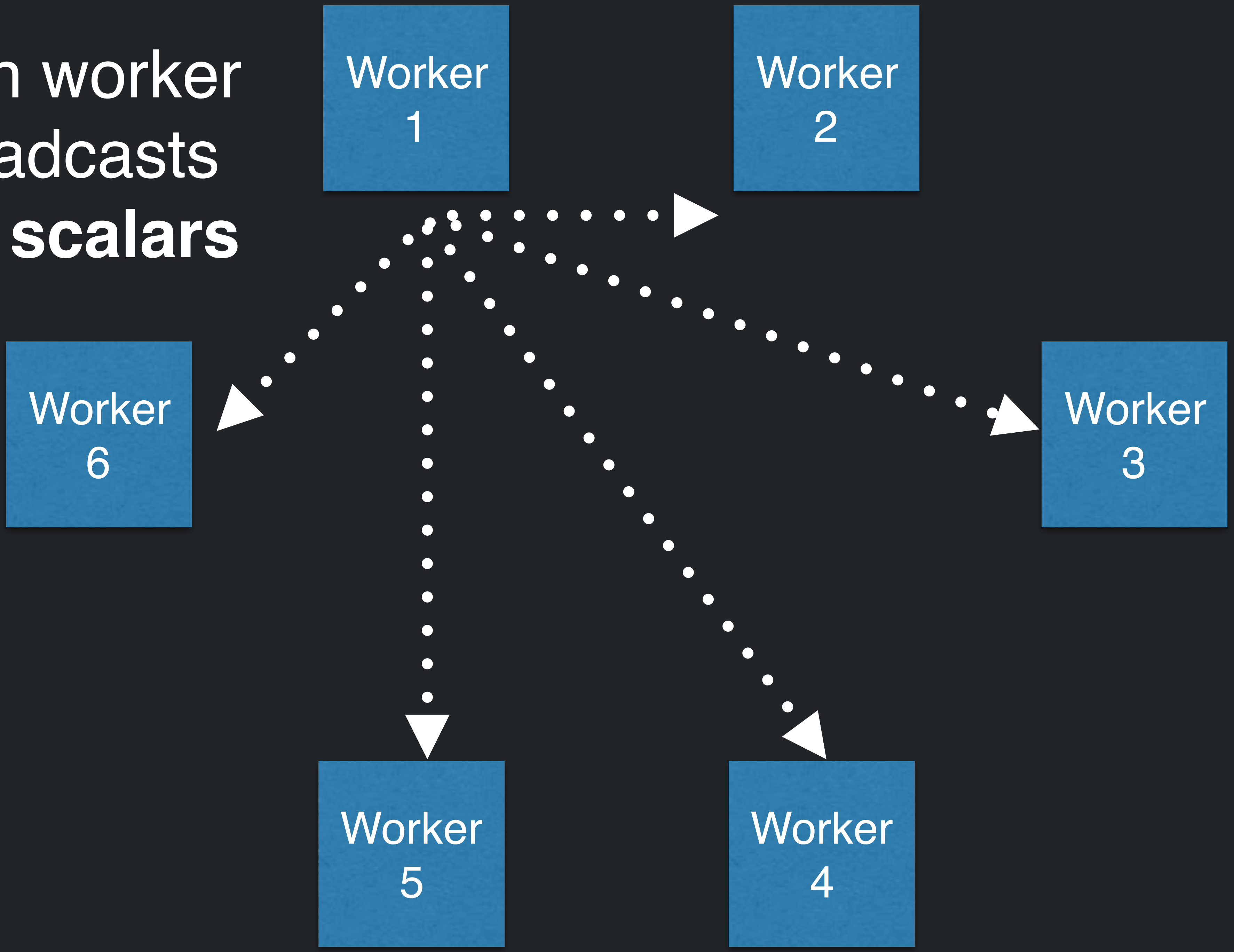
Distributed Deep Learning

Each worker sends **big vectors**



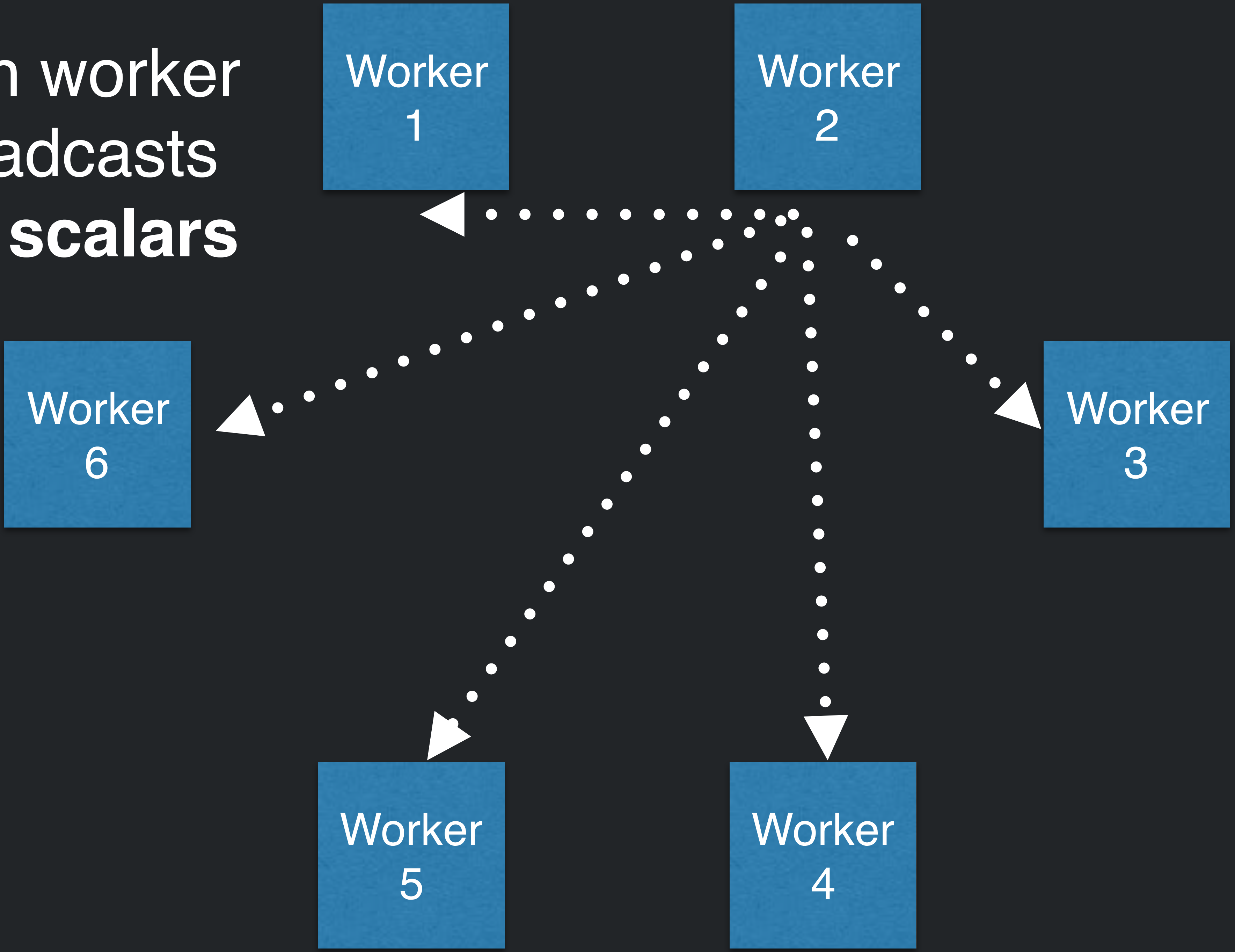
Distributed Evolution Strategies

Each worker broadcasts **tiny scalars**



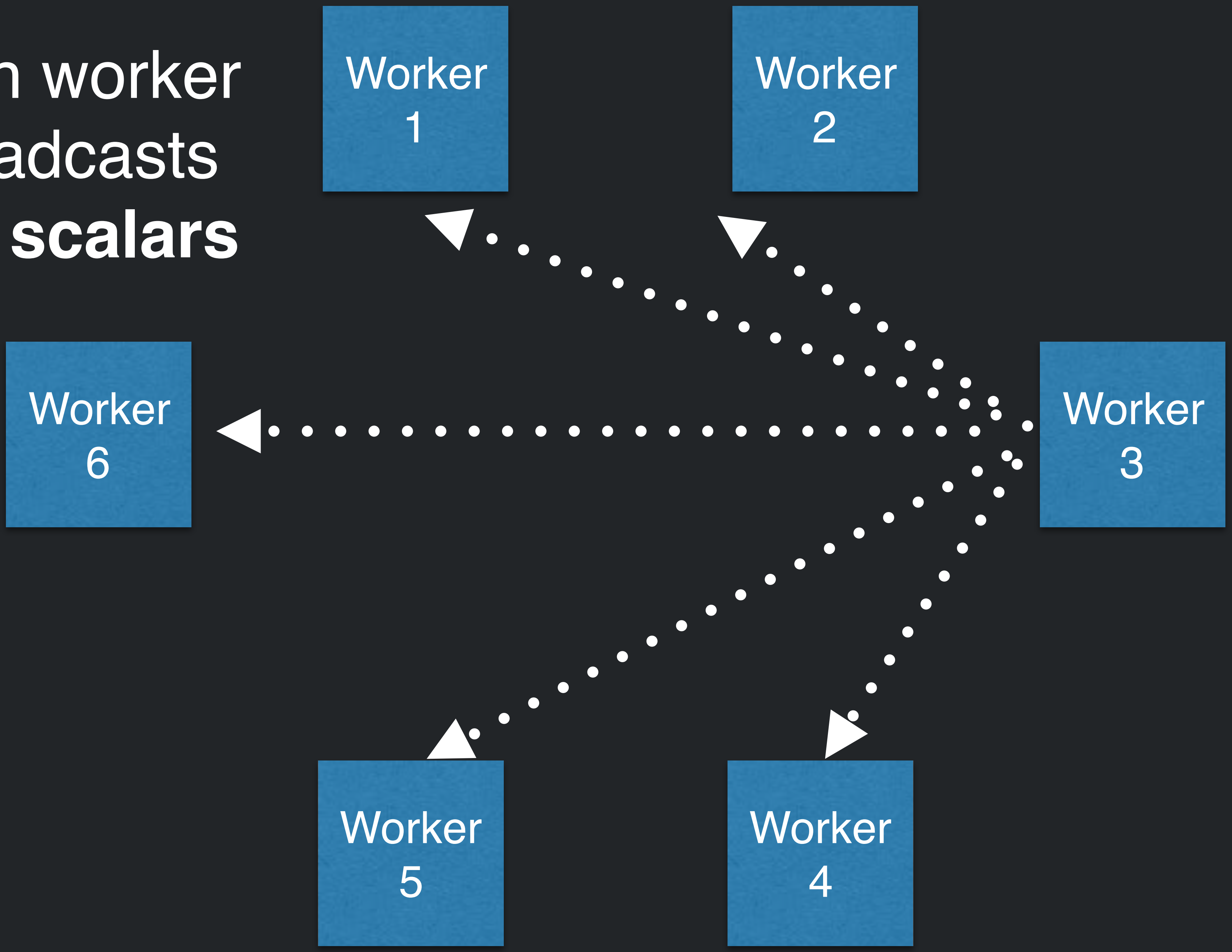
Distributed Evolution Strategies

Each worker broadcasts **tiny scalars**



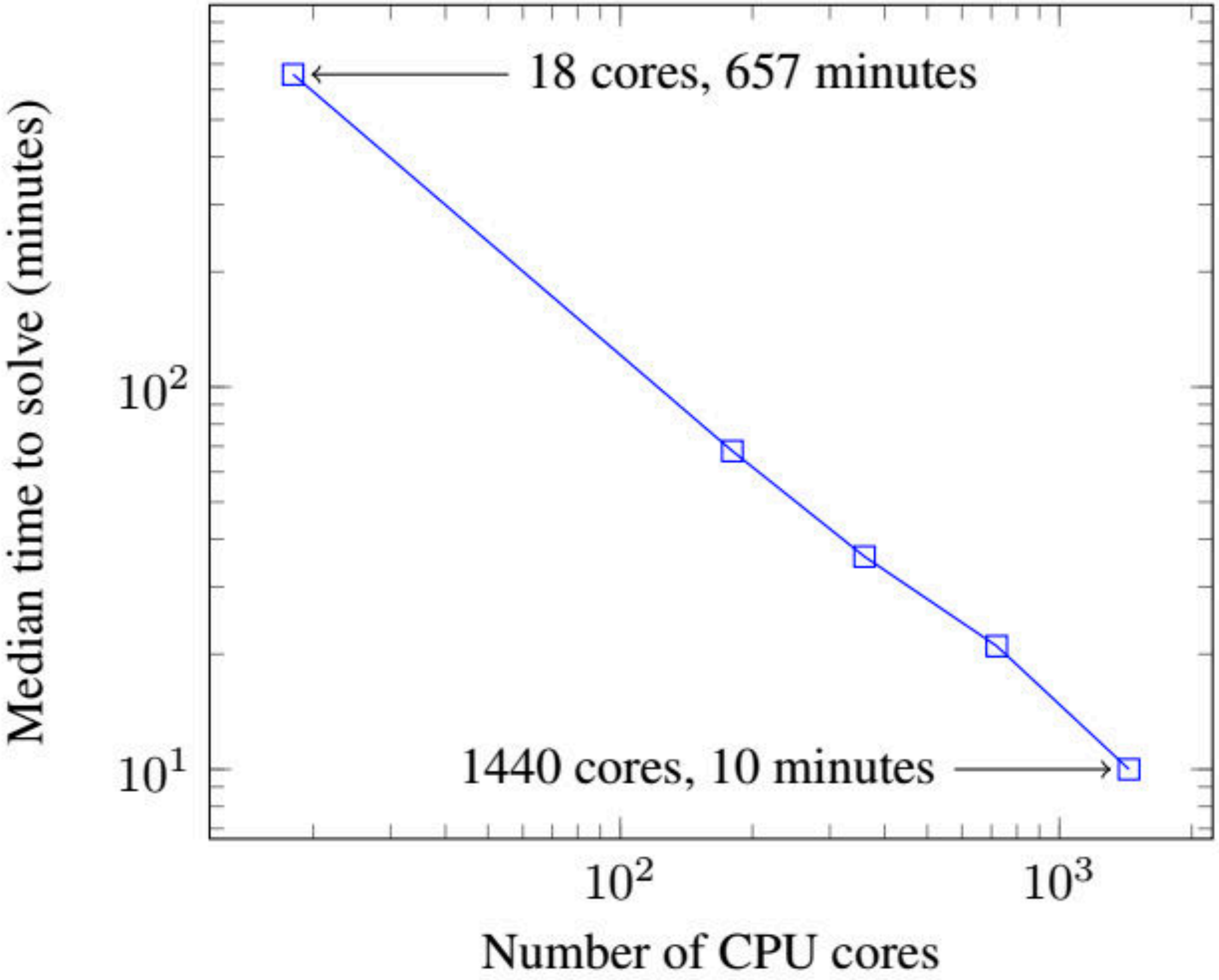
Distributed Evolution Strategies

Each worker broadcasts **tiny scalars**



Distributed Evolution Strategies

- Quantitative results on the Humanoid Mujoco task:



Competing RL algorithms

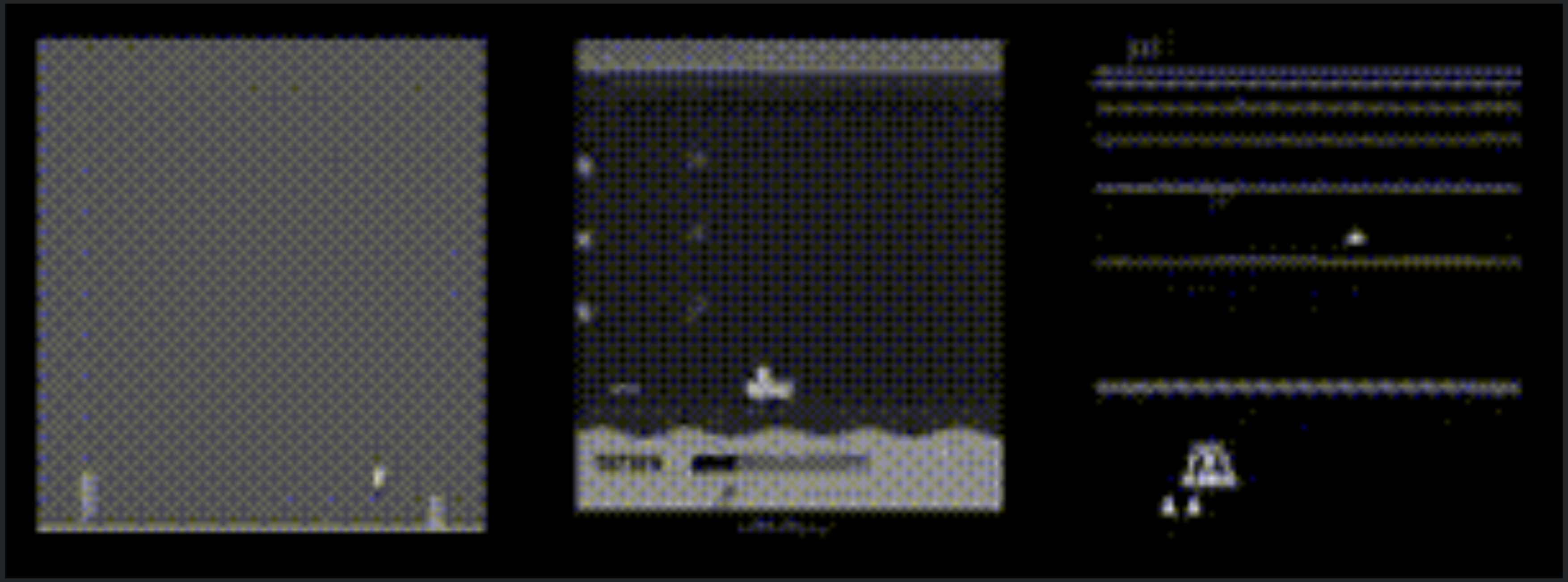
- A3C [Mnih et al., 2016]
- Trust Region Policy Optimization (TRPO) [Schulman et al., 2016]

Details of Results

- We can match one-day A3C on Atari games on average (better on 50%, worse on 50% of games) in **1 hour** of our distributed implementation with 720 cores

- We need 3x-10x more data

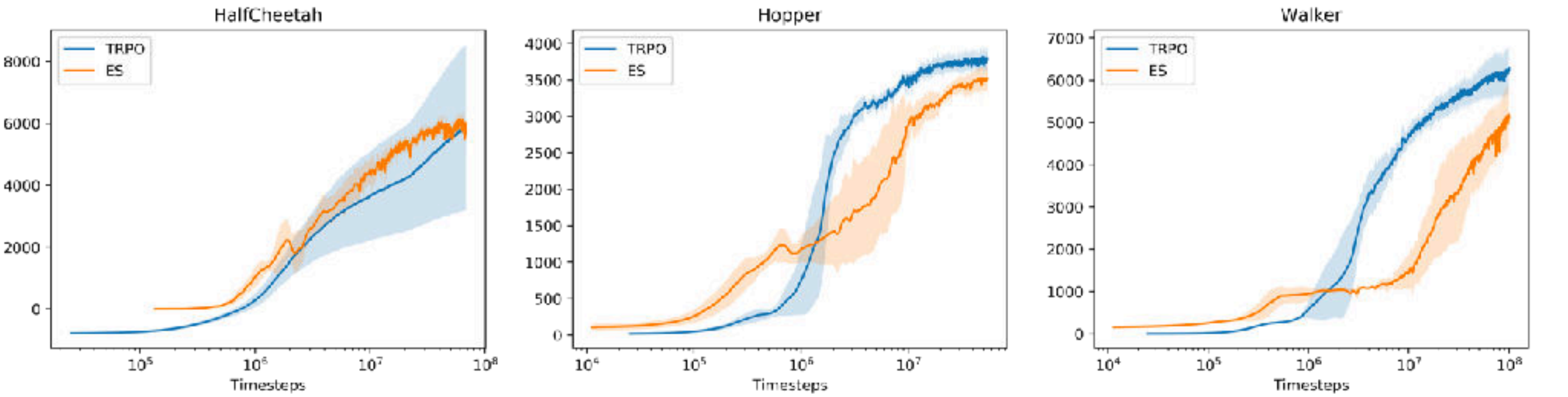
- No backward pass, no need to store activations in memory



MuJoCo results

- ES needs more data, but it achieves nearly the same result

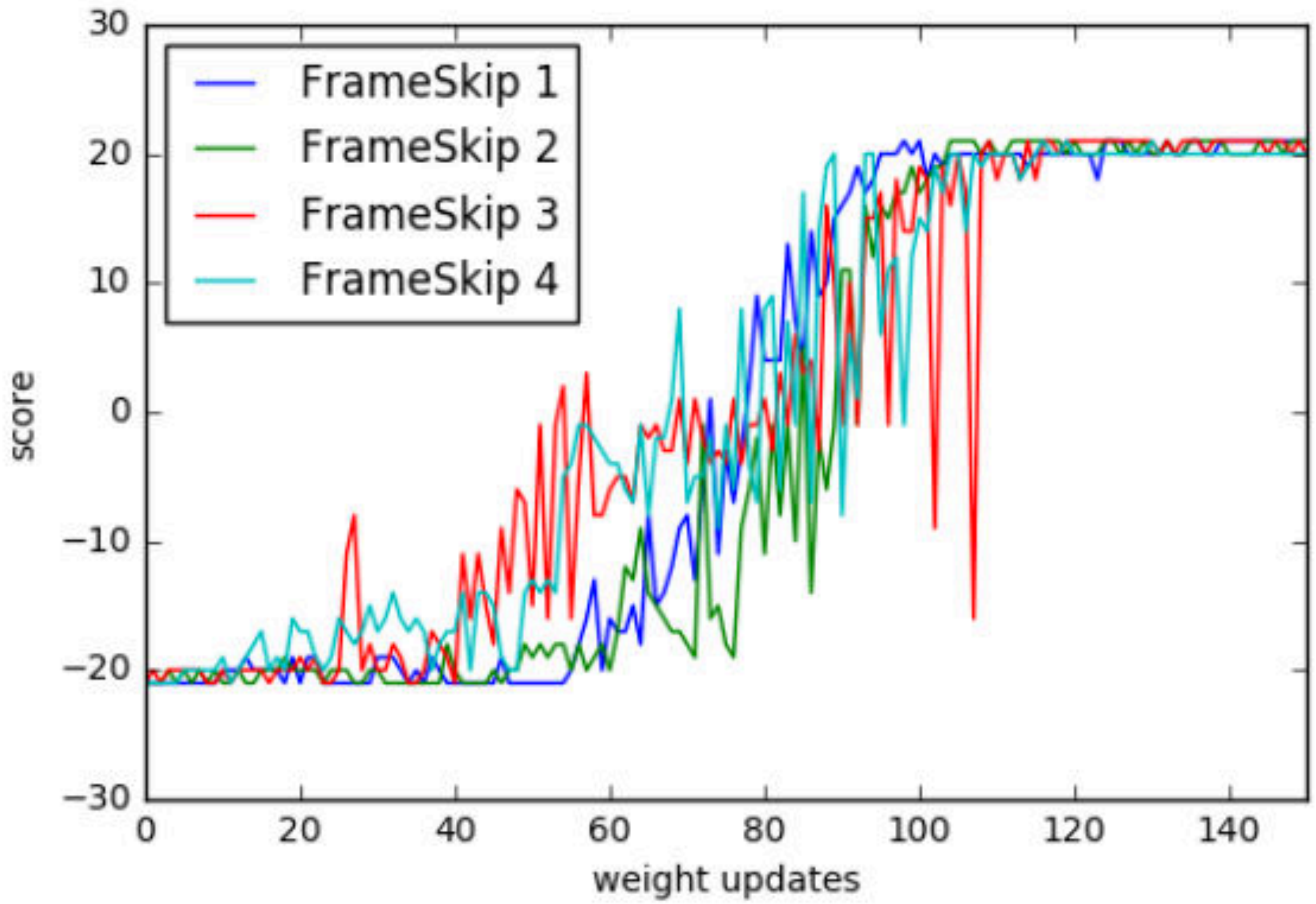
- If we use 1440 cores, we need **10 minutes** to solve the humanoid task, which takes 1 day with TRPO on a single machine



Long Horizons

- Long horizons are hard for RL
- RL is sensitive to action frequency
- Higher frequency of actions makes the RL problem more difficult
- Not so for Evolution Strategies

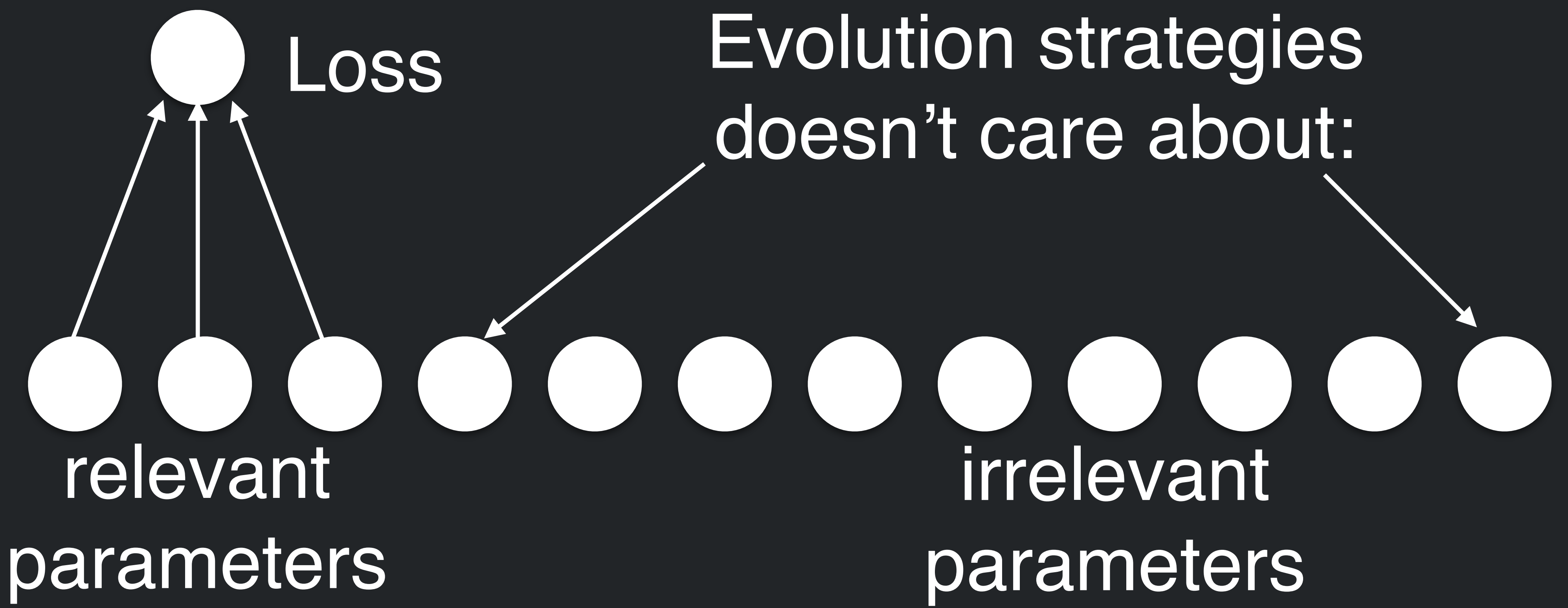
Long Horizons



What’s going on?

- Fact: the speed of Evolution Strategies depends on the intrinsic dimensionality of the problem, not on the actual dimensionality

Intrinsic Dimensionality



- Evolution strategies *automatically discards* the irrelevant dimensions — even when they live on a complicated subspace!

Intrinsic Dimensionality

- One explanation for how hill-climbing can succeed in a million-dimensional space!

Evolution Strategies and Supervised Learning

- RL is slow
- RL gradients are *very* noisy
- This fact helps Evolution Strategies to be competitive with RL

Supervised Learning

- Work in progress, but: often 1000x slower
- Can solve MNIST overnight on a single GPU
- Solving CIFAR-10 takes days

Backprop vs Evolution Strategies

- Evolution strategies does not use backprop
- So scale of initialization, vanishing gradients, etc, are not important?

Backprop vs Evolution Strategies

- Counterintuitive result: *every* trick that helps backprop, also helps evolution strategies
  - scale of random init, batch norm, ResNet…
- Why? Because evolution strategies tries to estimate *the gradient*!
  - If the gradient is vanishing, we won’t get much by estimating it!

Conclusion

- Evolution strategies works
- Do not shy away from policy gradient on high dimensional spaces: it too is likely to work very well