The Need for Dynamic Deep Learning

Rich Sutton with
Shibhansh Dohare, Fernando Hernandez-Garcia, Qingfeng Lan,
Parash Rahman, and Rupam Mahmood

Keen Technologies
University of Alberta
Alberta Machine Intelligence Institute
Reinforcement Learning and Artificial Intelligence Lab
Openmind Research Institute

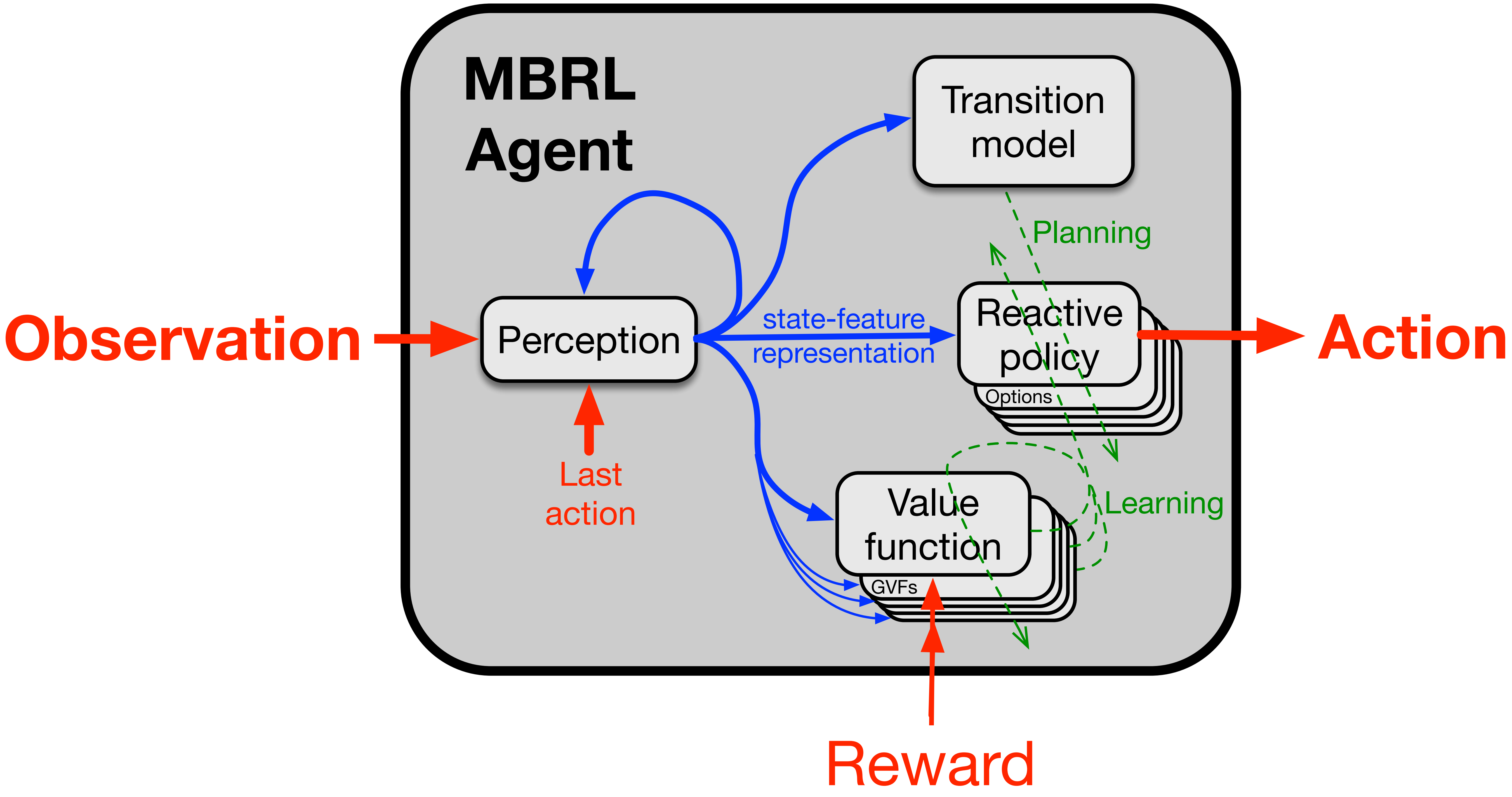
Artificial intelligence research is ambitious

- AI researchers seek to understand intelligence well enough to create beings of greater intelligence than current humans
- Reaching this profound intellectual milestone will enrich our economies and challenge our societal institutions
    - It will be unprecedented and transformational, but also a continuation of trends that are thousands of years old
- People have always created tools and been changed by them; it's what humans do
- The next big step is to understand ourselves
- This is a quest grand and glorious, and quintessentially human
- Of course, it is also totally hyped up

My perspective

- I seek to understand and create maximally intelligent agents
- An agent is defined to be intelligent to the extent that it is able to predict and control its input stream, particularly its reward
- The creation of super-intelligent agents, and super-intelligent augmented humans, will be an unalloyed good for the world (though the near future will be tough, as we are in a 4th turning)
- The path to intelligent agents runs through reinforcement learning (and not through LLMs, however amazing and useful those might be)
- The biggest current bottleneck to ambitious RL-based AI is inadequate deep learning methods

The standard agent architecture* of RL and the Alberta Plan**



The common agent comprises four components:

Perception produces the state representation used by all components

Reactive Policies quickly produce actions that achieve high rewards or features

Value Functions evaluates how well things are going, and changes the policy (learning)

Transition model predicts the consequences of alternate choices, and changes the policy (planning)

(**Subtasks***** are not explicitly shown)

* The Quest for a Common Model of the Intelligent Agent, by Sutton, RLDM 2022, arXiv.
** The Alberta Plan for AI Research, by Sutton, Bowling, & Pilarski, 2023, arXiv.
*** Reward Respecting Subtasks for Model-based Reinforcement Learning, by Sutton, Machado, Holland, Timbers, Tanner, White, AIj 2023.

Loss of Plasticity in Deep Continual Learning

Rich Sutton with
Shibhansh Dohare, Fernando Hernandez-Garcia, Qingfeng Lan,
Parash Rahman, and Rupam Mahmood

Keen Technologies
University of Alberta
Alberta Machine Intelligence Institute
Reinforcement Learning and Artificial Intelligence Lab
Openmind Research Institute

Main message:

Deep learning does not work for continual learning

- by “not work”, I mean that learning slows, eventually to a very low level (loss of plasticity)
- by “deep learning”, I mean the standard artificial-neural-network methods, specialized as they are for non-continual learning
  - without replay buffers (which themselves are an acknowledgement that DL doesn’t work)

Better learning algorithms, specialized for continual learning, are not hard to find.

But we have to start looking for them

Outline

- Demonstrations of Loss of Plasticity in deep learning
  and attempts to maintain it (w/L2 reg, Shrink & Perturb, Continual Backprop)

  - in convolutional networks on continual versions of ImageNet

  - in residual networks on continual versions CIFAR-100

  - in stationary and non-stationary reinforcement learning

- Some goals and ideas for a next generation of deep learning

A lesson in perseverance

- Paper submissions based on this work were rejected 5 times over 5 years
- But now it is published in *Nature*

References:

Dohare, S., Hernandez-Garcia, J.F., Rahman, P., Lan, Q., Sutton, R.S., Mahmood, A.R. (2024) Loss of plasticity in deep continual learning. *Nature 632*, pp. 768-774, August 22, 2024.

Dohare, S., Hernandez-Garcia, J.F., Rahman, P., Sutton, R.S., Mahmood, A.R. (2023) Maintaining plasticity in deep continual learning. ArXiv:2306.13812.

Dohare, S., Sutton, R., Mahmood, A.R. (2021). Continual backprop: Stochastic gradient descent with persistent randomness. ArXiv:2108.06325.

Plasticity = the ability to learn

Loss of Plasticity = loss of the ability to learn
= not being able to learn continually
= not continual learning

Maintaining Plasticity = maintaining the ability to learn

In AI, we should prioritize maintaining plasticity

Early indications of problems with deep continual learning

- Catastrophic Forgetting (French, 1999; McCloskey & Cohen, 1989)
- Loss of Plasticity in early neural networks in the psych literature (Ellis & Ralph, 2000; Zevin & Seidenberg, 2002; Bonin et al., 2004)
- The failure of warm-starting (Ash & Adams, 2020)
- Primacy Bias and resetting in Deep RL (Nikishin et al., 2022)
- Capacity Loss in RL (Lyle et al, 2022)

But no one has previously done a thorough demonstration of Loss of Plasticity in supervised learning

Loss of Plasticity in Supervised Learning

ImageNet

ImageNet

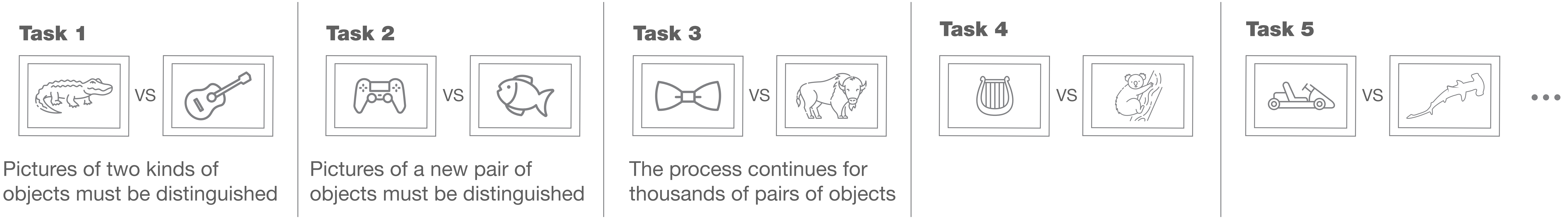
a classic deep-learning problem

ImageNet Dataset

- A database of millions of images labelled by nouns (classes)
- 1000 classes with 700 or more images
- Widely used in deep learning to classify images: image⇒class

The *Continual* ImageNet Problem

- The classical ImageNet problem was minimally changed to make it continual
- Each class was separated into 600 training examples and 100 test examples
- Classes were taken in pairs to produce a sequence of 500+ binary classification tasks
  - e.g., Class1 vs Class2 for 1200 training examples and 200 test examples, then Class3 vs Class4 for 1200 training examples and 200 test examples, etc



- Performance measure: %correct on test set (by argmax) at end of each task
- Averaged over 30 independent runs, varying class pairings, test sets

Network and Training Procedure (ImageNet)

- All 500+ binary classification tasks share the same network; both heads reset at task switch
- Standard neural network, though slightly narrow for ImageNet (bc. only 2 classes at a time)
  (3 convolution layers of 32/64/128 filters + 3 fully-interconnected layers of 128/128/2 units)
- For each task, 12 batches of 100 examples, 250 epochs (passes through the data)
- Weights initialized by the standard Kaiming distribution, only once, before the first task
- Backpropagation with momentum on the cross-entropy loss, ReLU activations
- Many variations on the network and hyper-parameters were tested
  to obtain good and representative performance on the first task

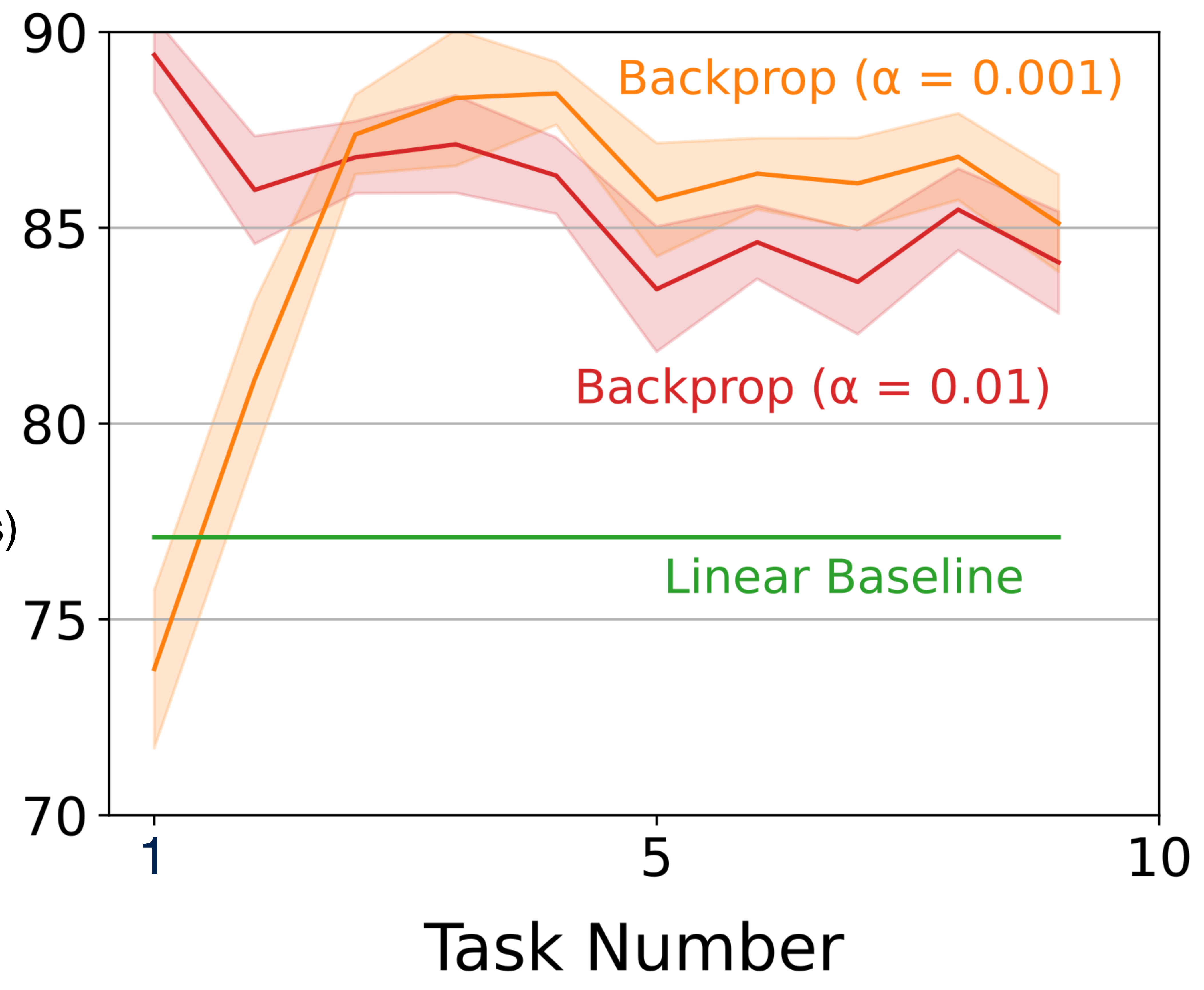
How will performance evolve over the sequence of tasks?

Will performance be better on the 1st task or the 2nd task? the 500th?

BackProp on Continual ImageNet (first 10 tasks)

%Correct
on Test Set

(at end of each task, averaged over 30 runs)

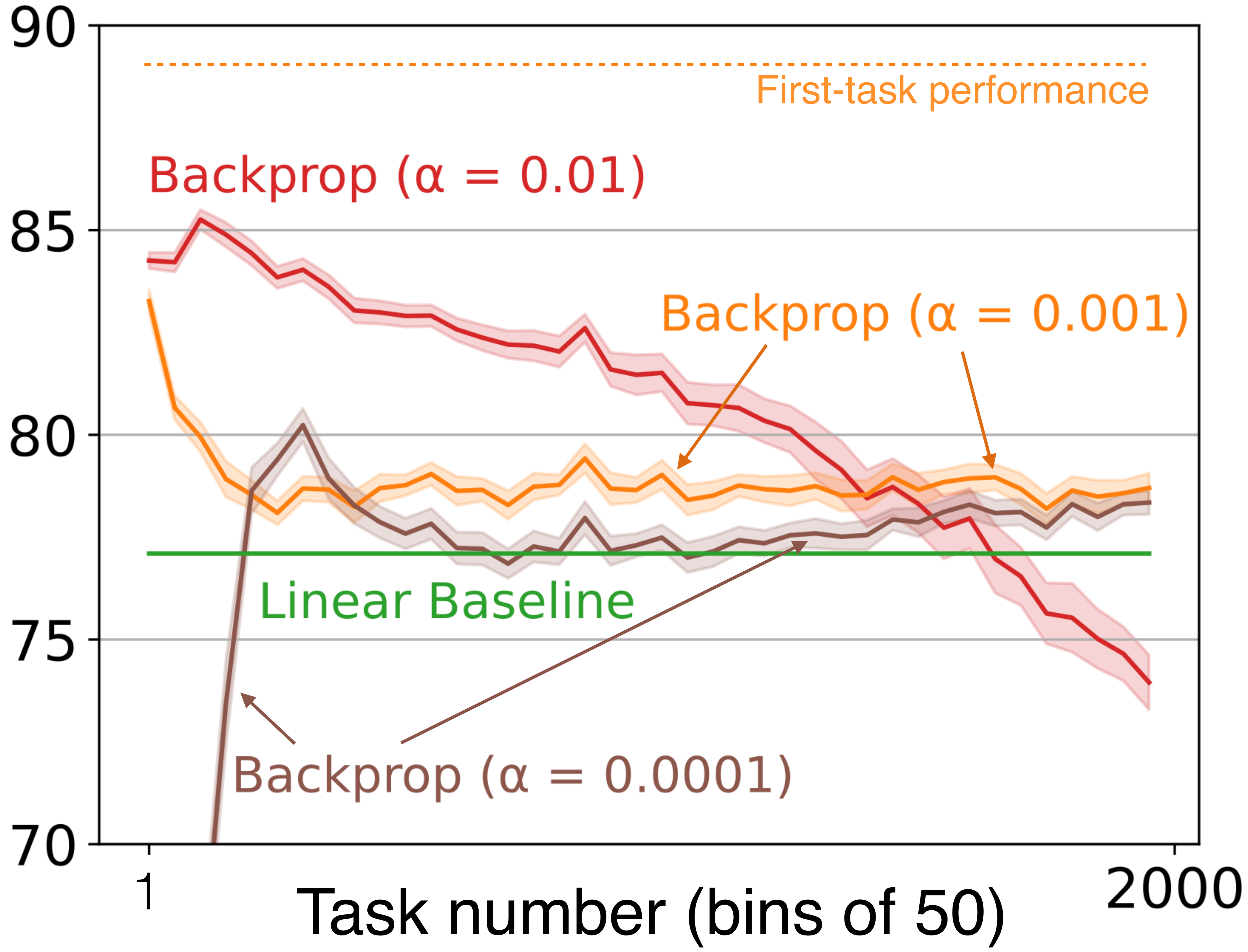


- Chance performance is 50%
- Best performance on first task is ≈89%
- Shaded region is one standard error
- Linear baseline is the performance of linear heads direct from pixels

Learning rate (plasticity) sometimes improves over early tasks, then…?

BackProp on Continual ImageNet (2000 tasks)

%Correct
on Test Set
(at end of each task,
averaged over 30 runs)

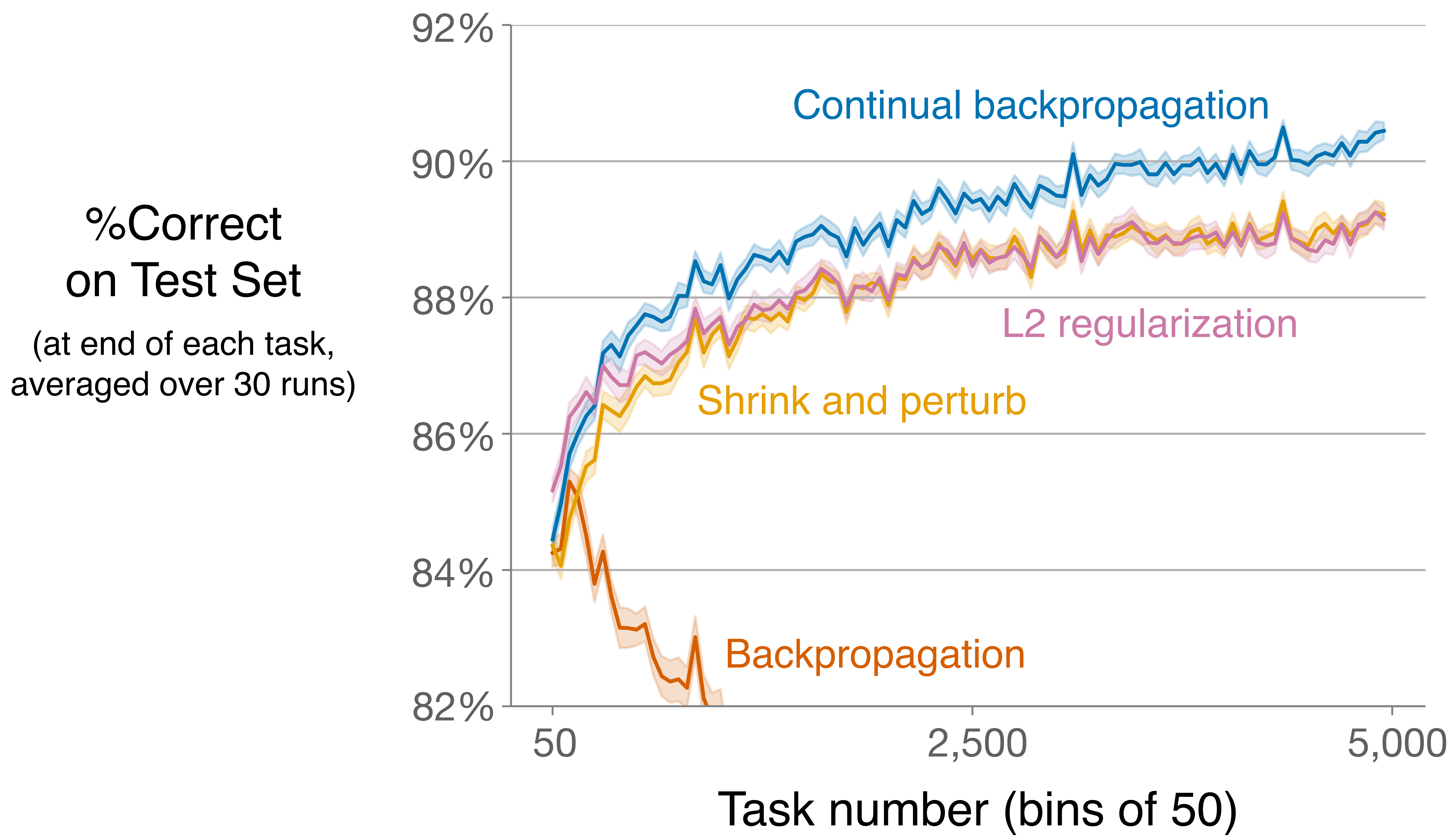


- This data is representative, the details depend on the details:
  - #epochs
  - step-sizes
  - network sizes
- Each line takes ≈24 hours to compute
- Most other variations of BackProp (Adam, Dropout, Batch norm) are worse

For good hyper-parameters, plasticity decreases across tasks,
nearing the poor performance level of a one-layer (linear) network, or worse

BackProp shows “Catastrophic” Loss of Plasticity

There are better Algorithms on Continual ImageNet



- Dropout, Adam, other activations are even worse than vanilly backprop
- L2 regularization adds a penalty for large weights
- Shrink and Perturb is L2 reg. plus random variation of all weights
- Continual Backpropagation continually re-initializes a small fraction of units
    - otherwise its just like BackProp

Continual Backpropagation:

Stochastic Gradient Descent with Selective Reinitialization

- Just like backprop, except continually re-initialize a small fraction of the units (*re-initialization rate*)

- And it is better to re-initialize selectively,
for example, dormant units or some other notion of *utility*

- The idea of selective random re-initialization was introduced by Mahmood and Sutton (2012); they called it *generate and test*

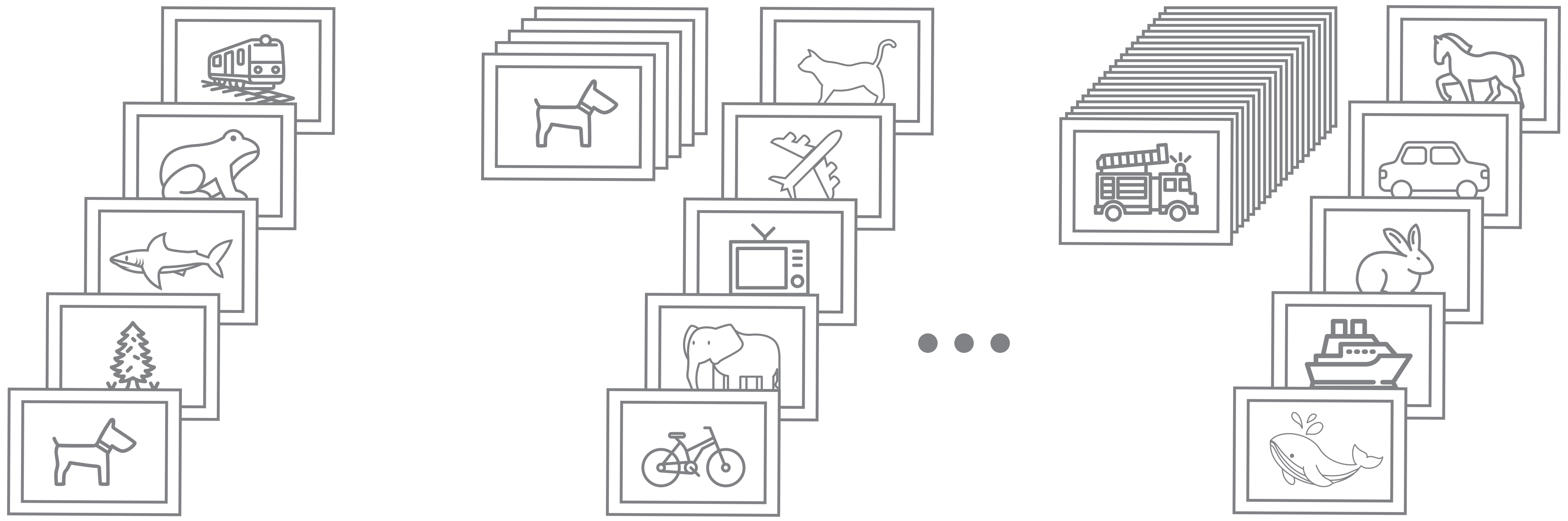
- Continual Backprop extends the idea to general multi-layer networks

Loss of Plasticity in Residual Networks

Cifar-100

Loss of Plasticity in class-incremental CIFAR-100 with 18-layer Residual Networks

Problem

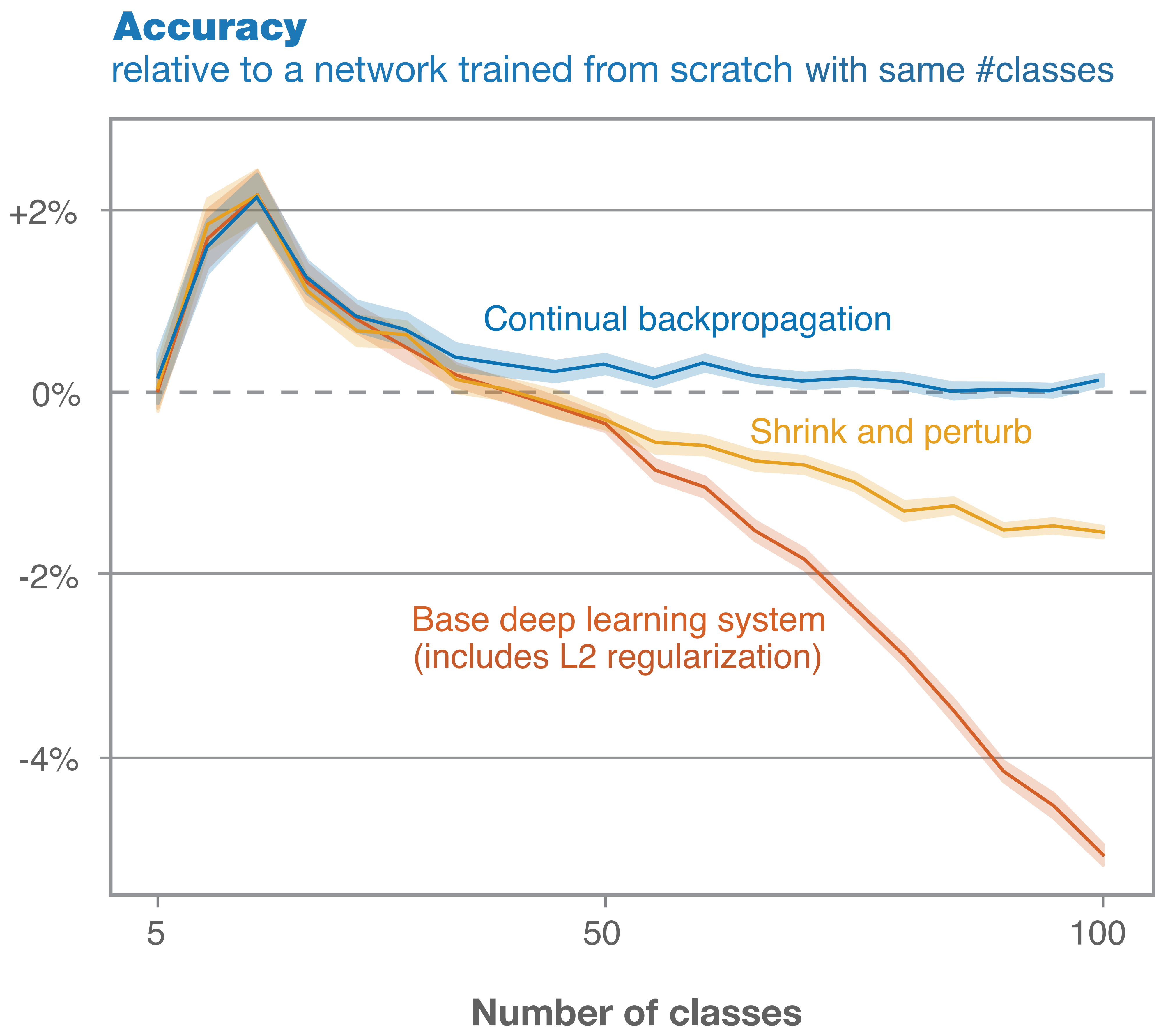


At the start, pictures from five classes have to be distinguished

five more classes are added, and pictures from all ten have to be distinguished

The process continues until finally all 100 classes have to be distinguished

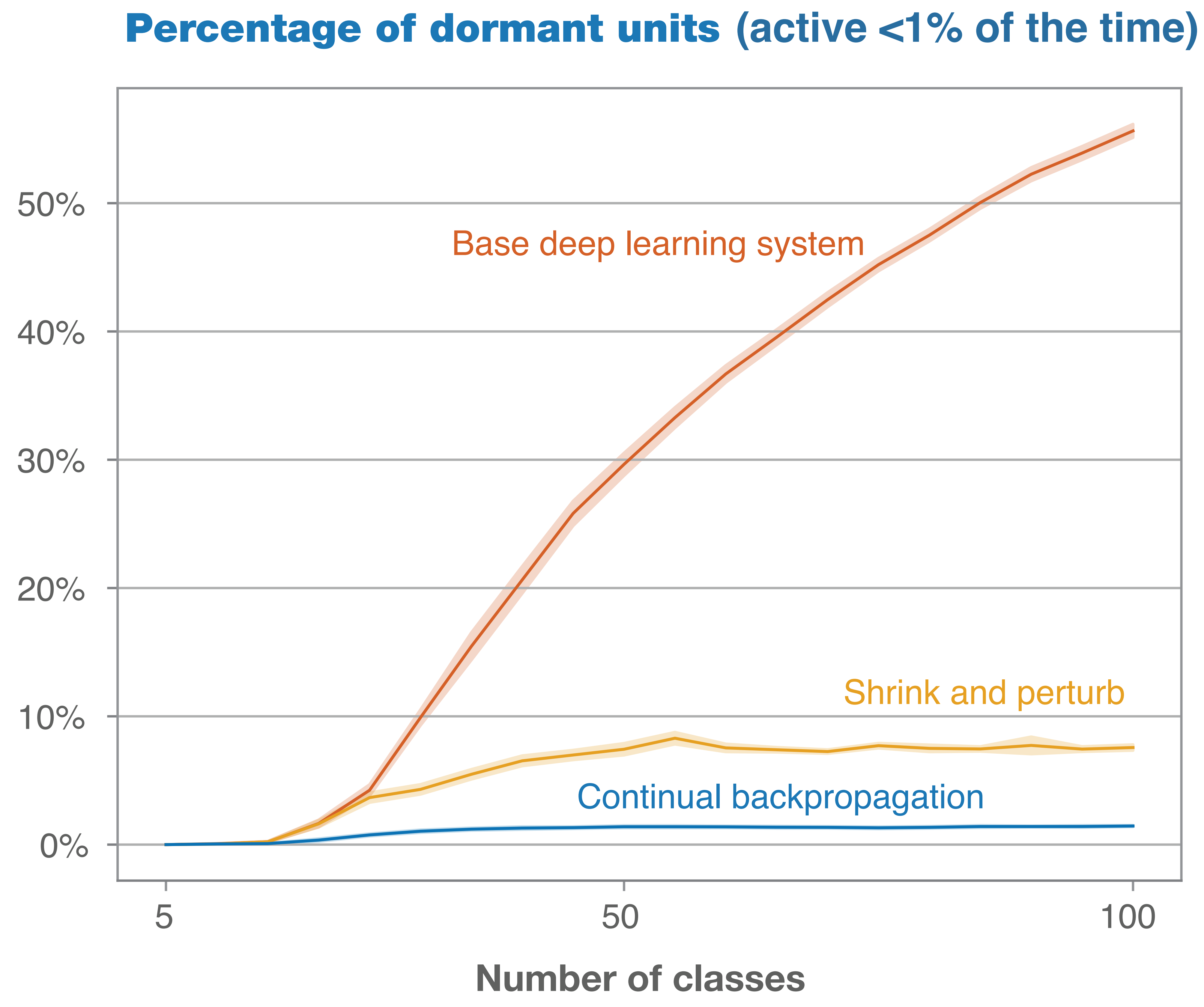
Results

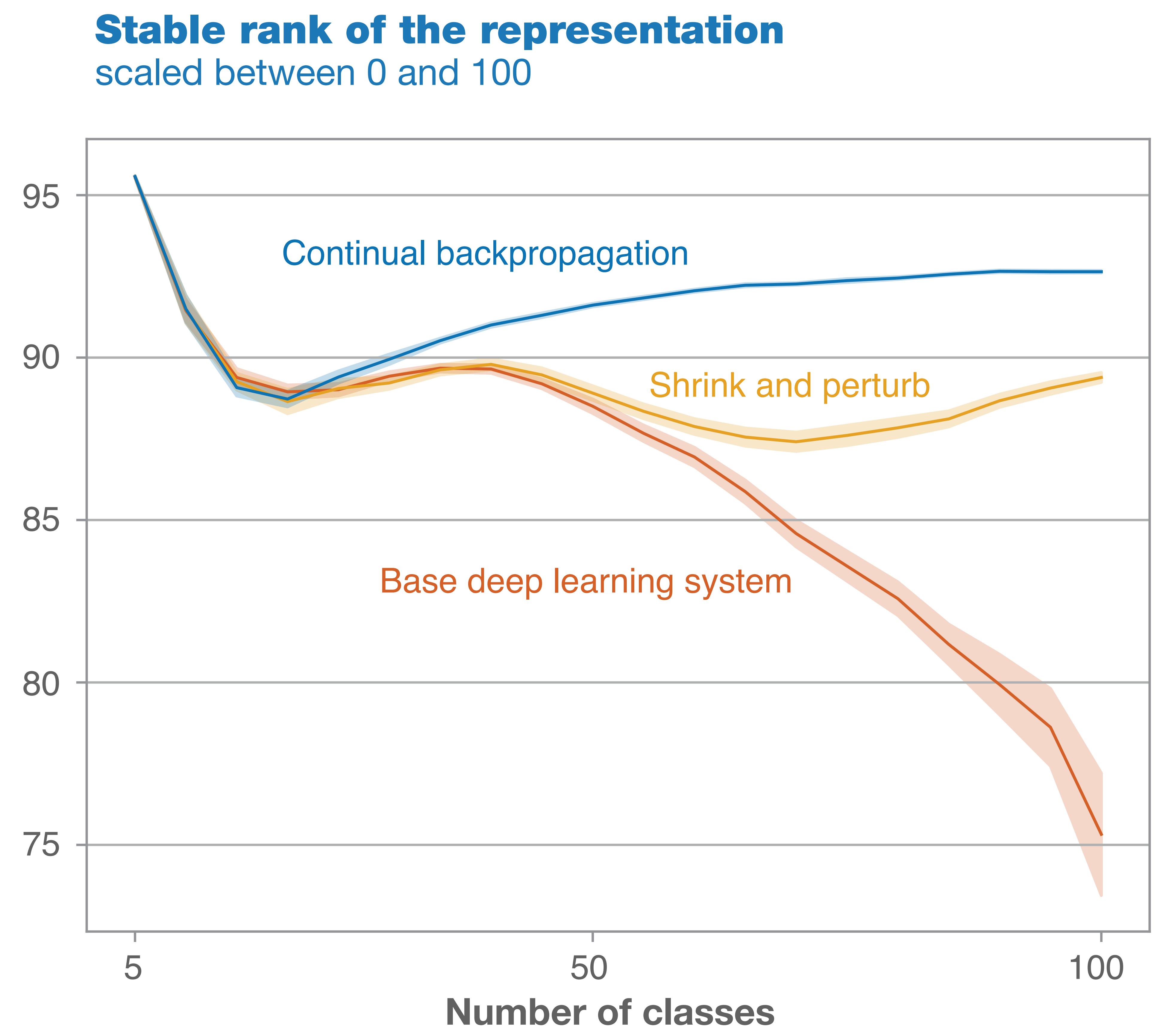


Continual BackProp > Shrink & Perturb > L2 > Backprop

Loss of Plasticity in class-incremental CIFAR-100 with 18-layer Residual Networks

A closer look at the results





Many units go dormant unless random variation is continually injected

Causing a collapse of representational diversity

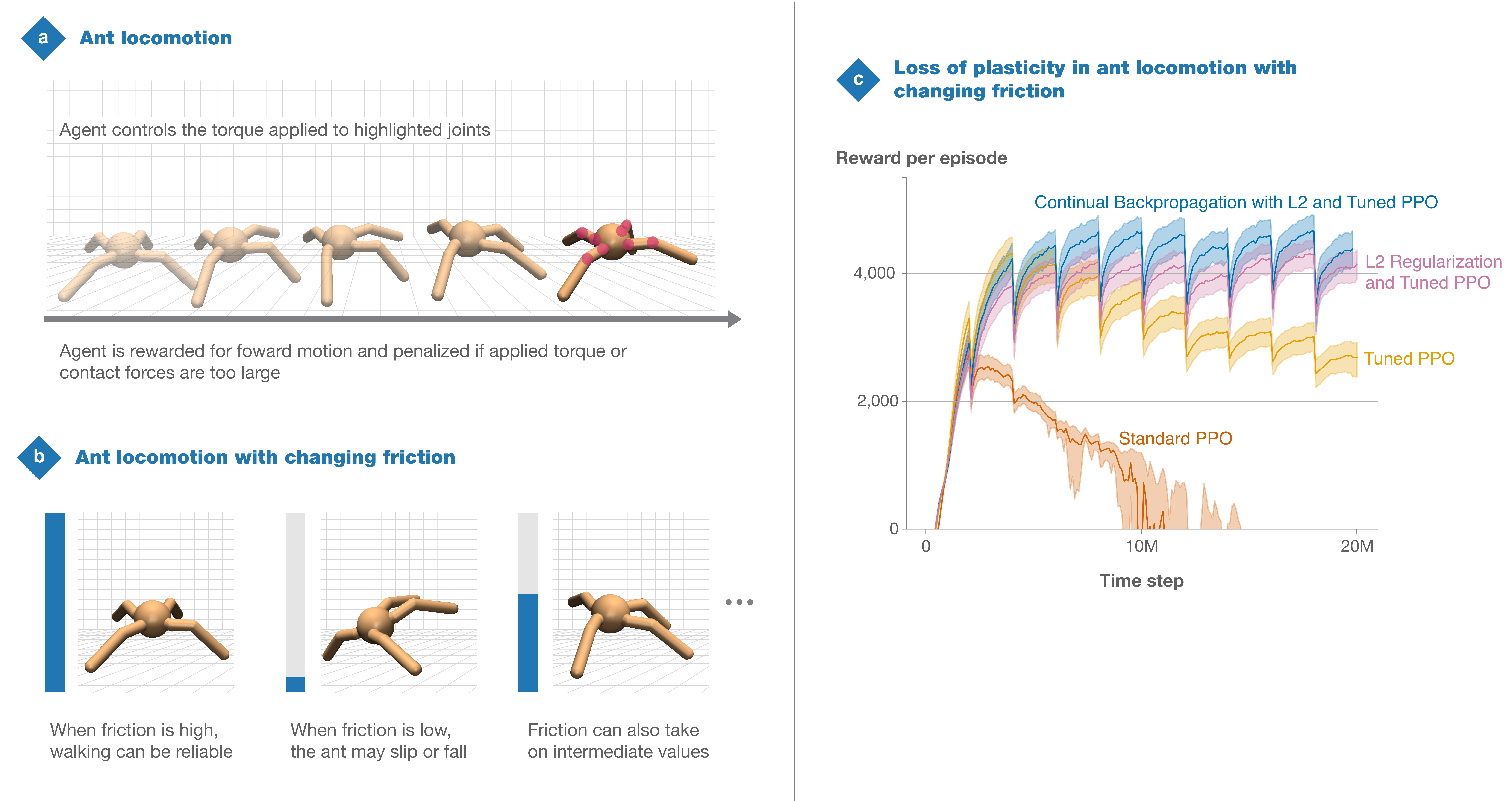
This pattern of results is very robust

- In ImageNet, in CIFAR-100, in MNIST, in idealized problems, and in RL, across network architectures, activation functions, and hyper-parameters:
  - Deep supervised learning loses plasticity dramatically under continued training
  - L2 regularization reduces the loss of plasticity, but just a little
  - Shrink & Perturb (L2 + weight randomizing) often helps further
  - Continual BackProp (BackProp + re-initialization of un-used units) does the best at maintaining plasticity
    - and has little sensitivity to its hyper-parameter (re-initialization rate)

Loss of Plasticity in Reinforcement Learning

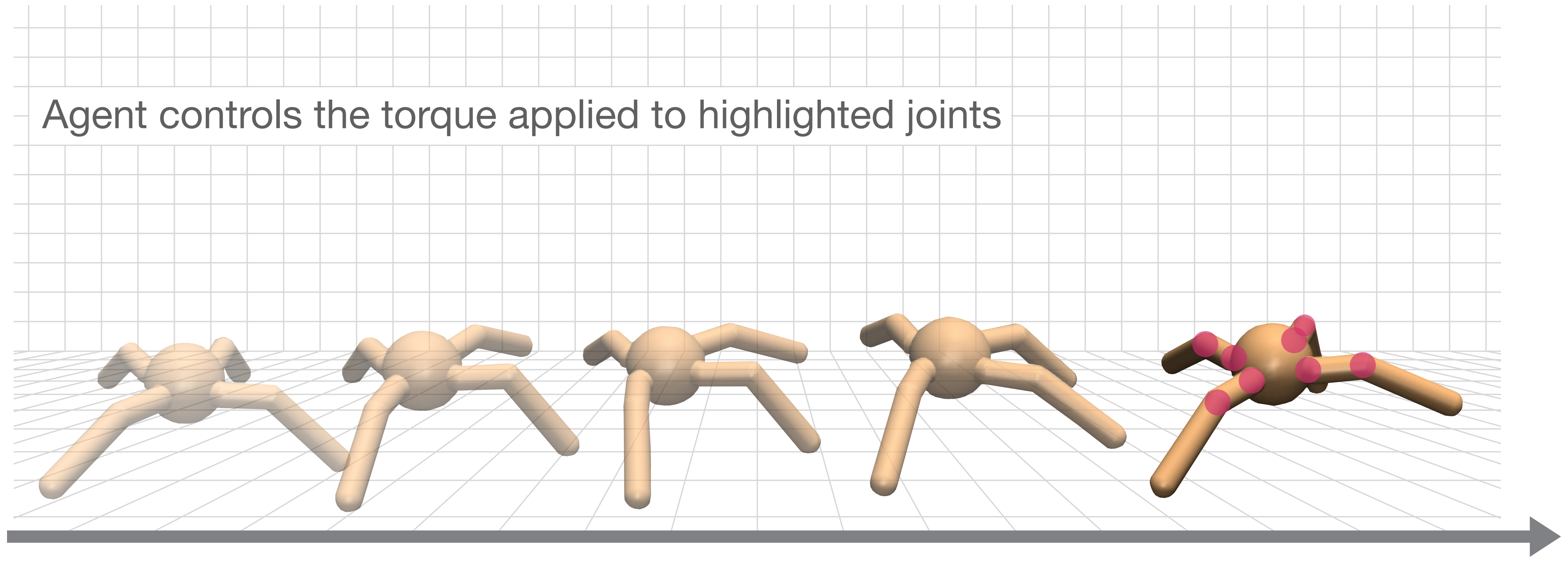
Ant Locomotion

Loss of Plasticity in Nonstationary Reinforcement Learning



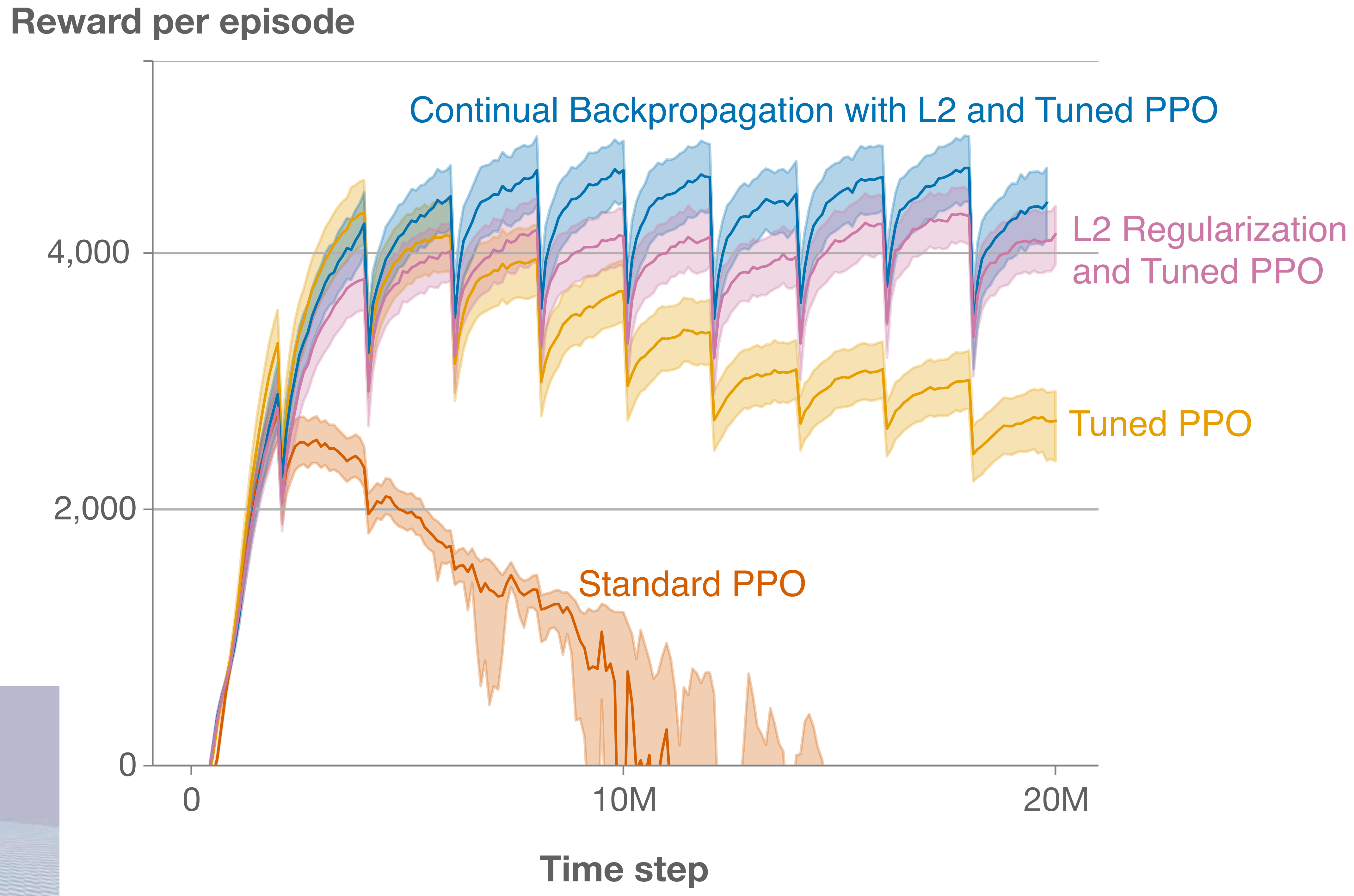
Loss of Plasticity in Nonstationary Reinforcement Learning

a Ant locomotion

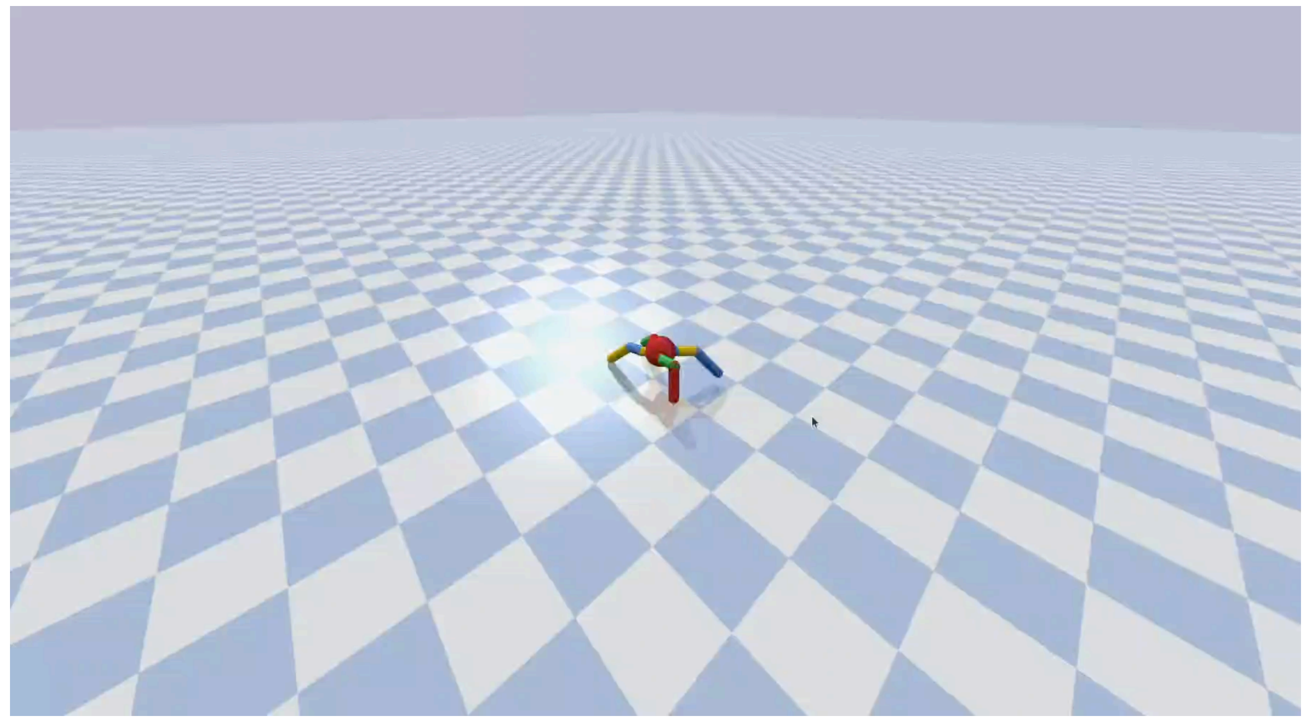


Agent is rewarded for foward motion and penalized if applied torque or contact forces are too large

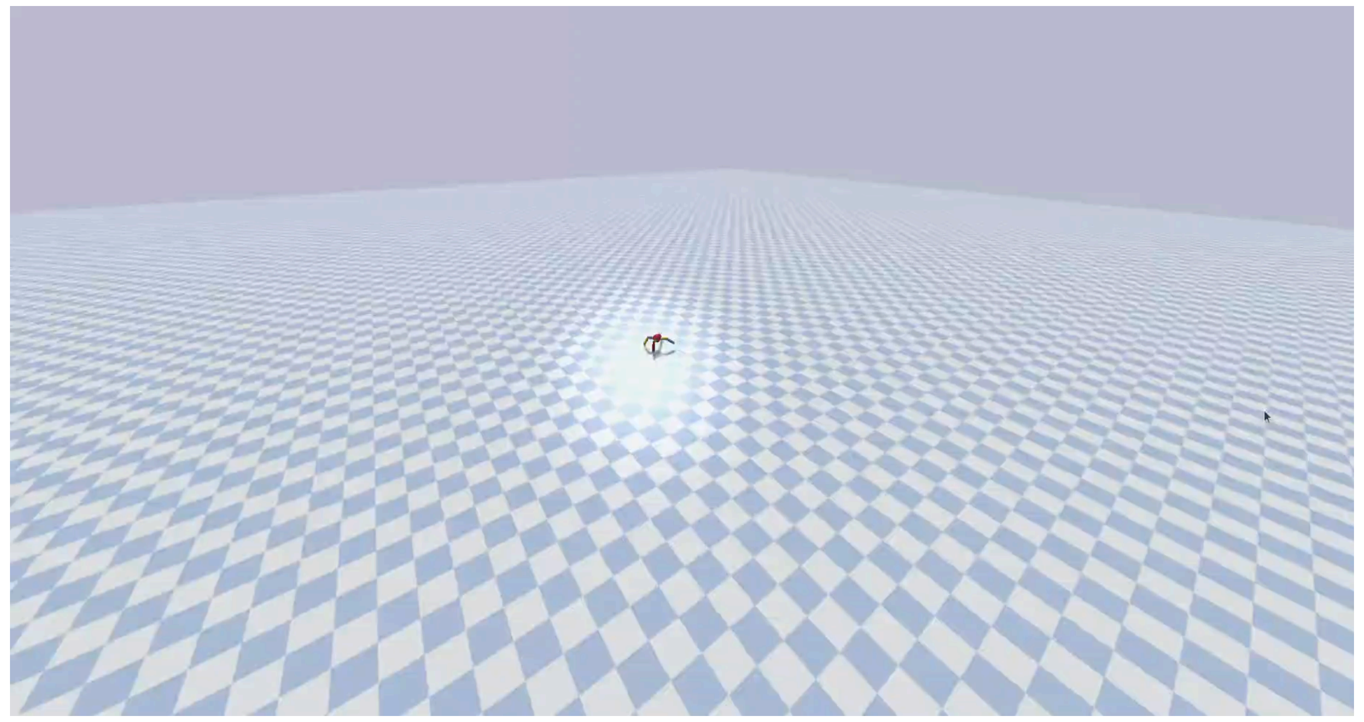
c Loss of plasticity in ant locomotion with changing friction



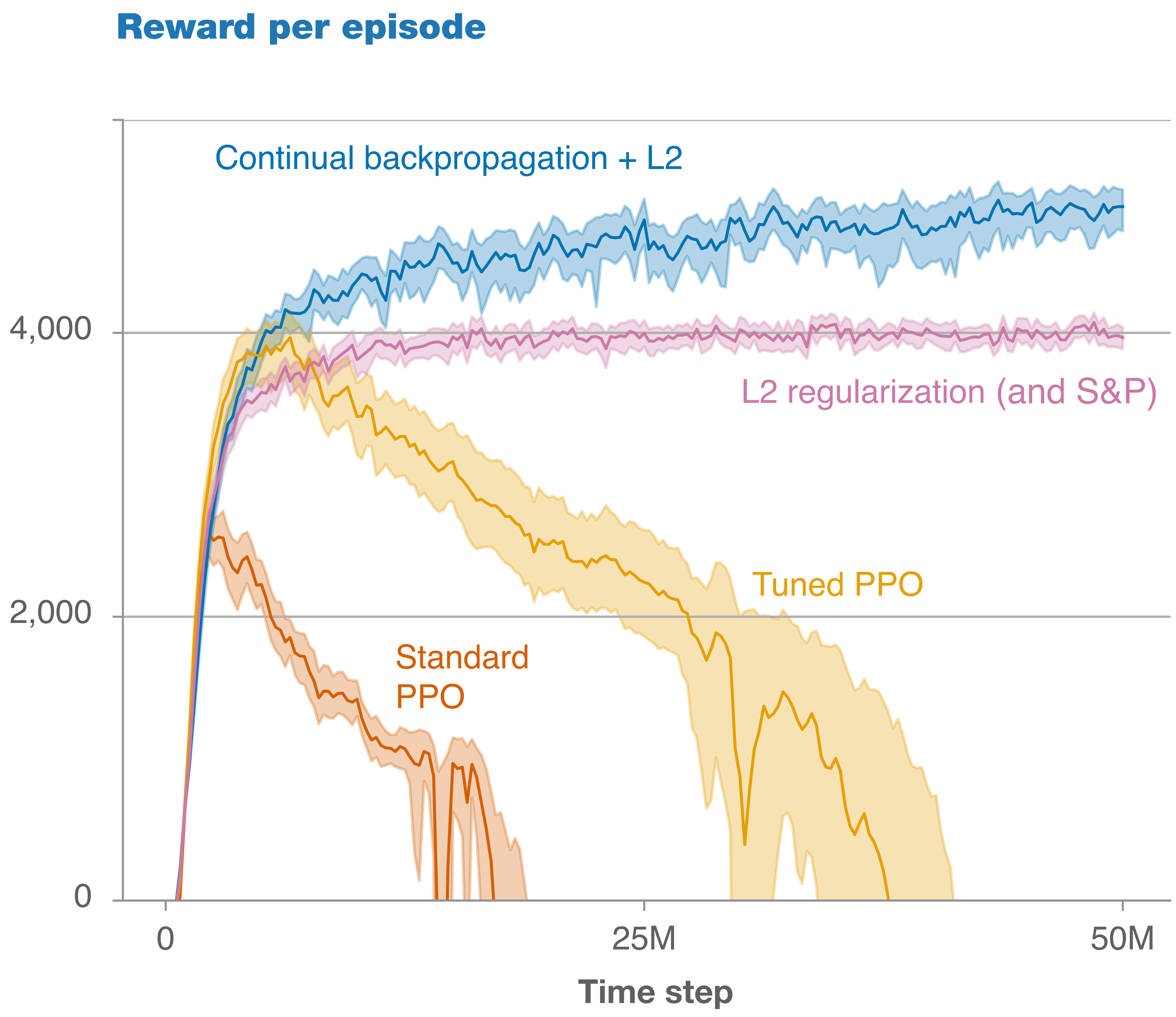
PPO



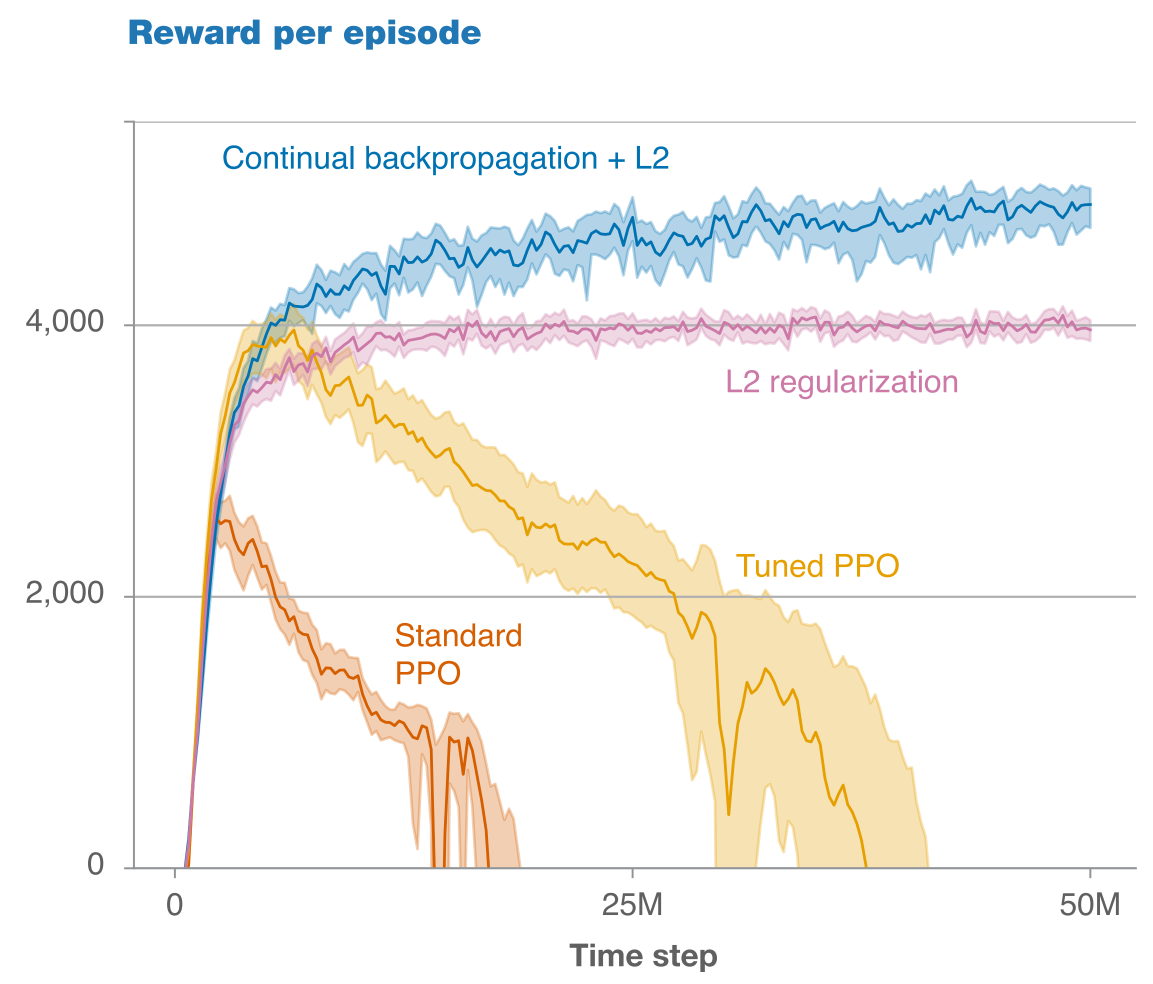
Continual PPO



Loss of Plasticity in Stationary Ant Locomotion

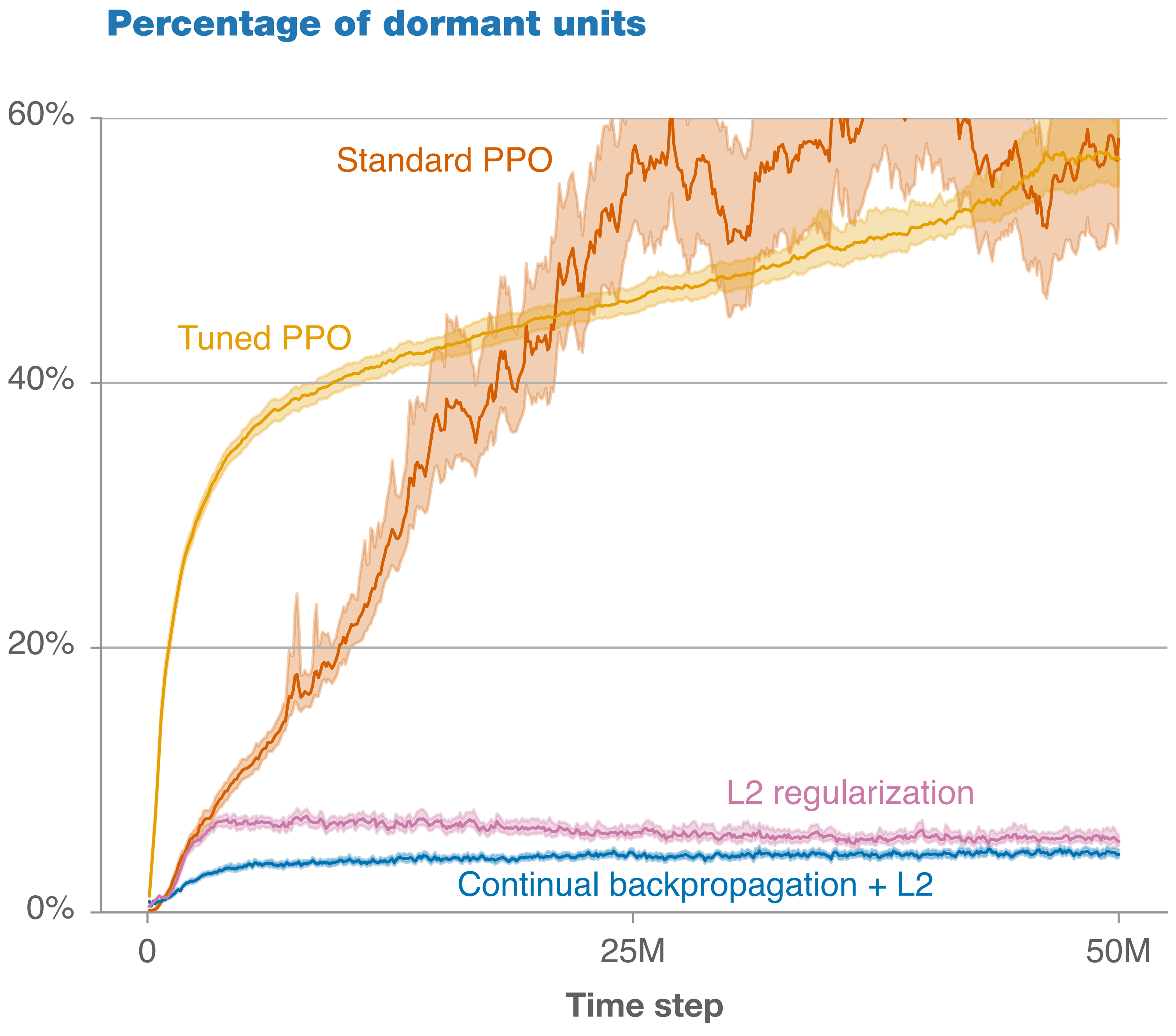


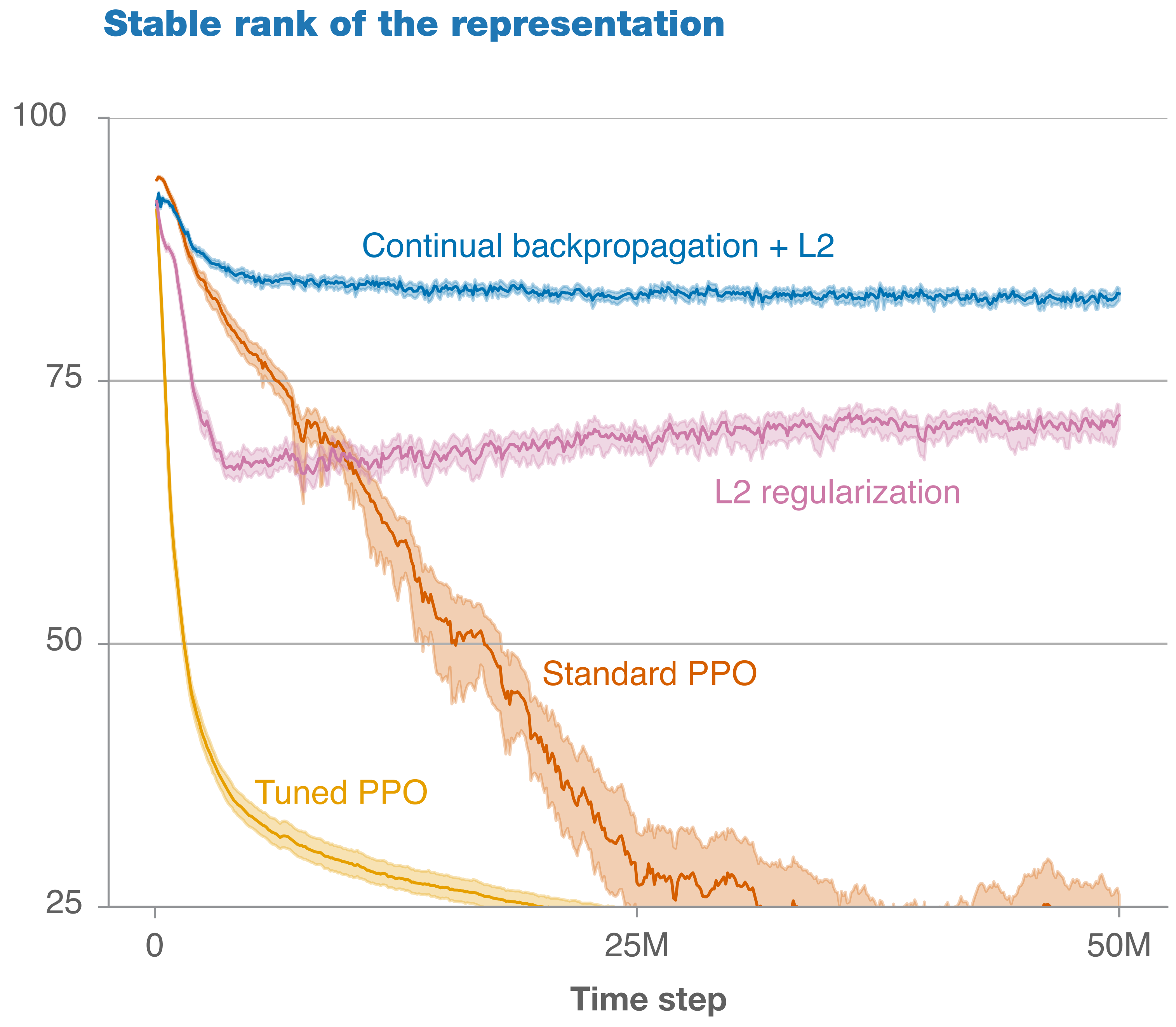
- The same pattern of results
- RL itself involves nonstationarity and continual learning

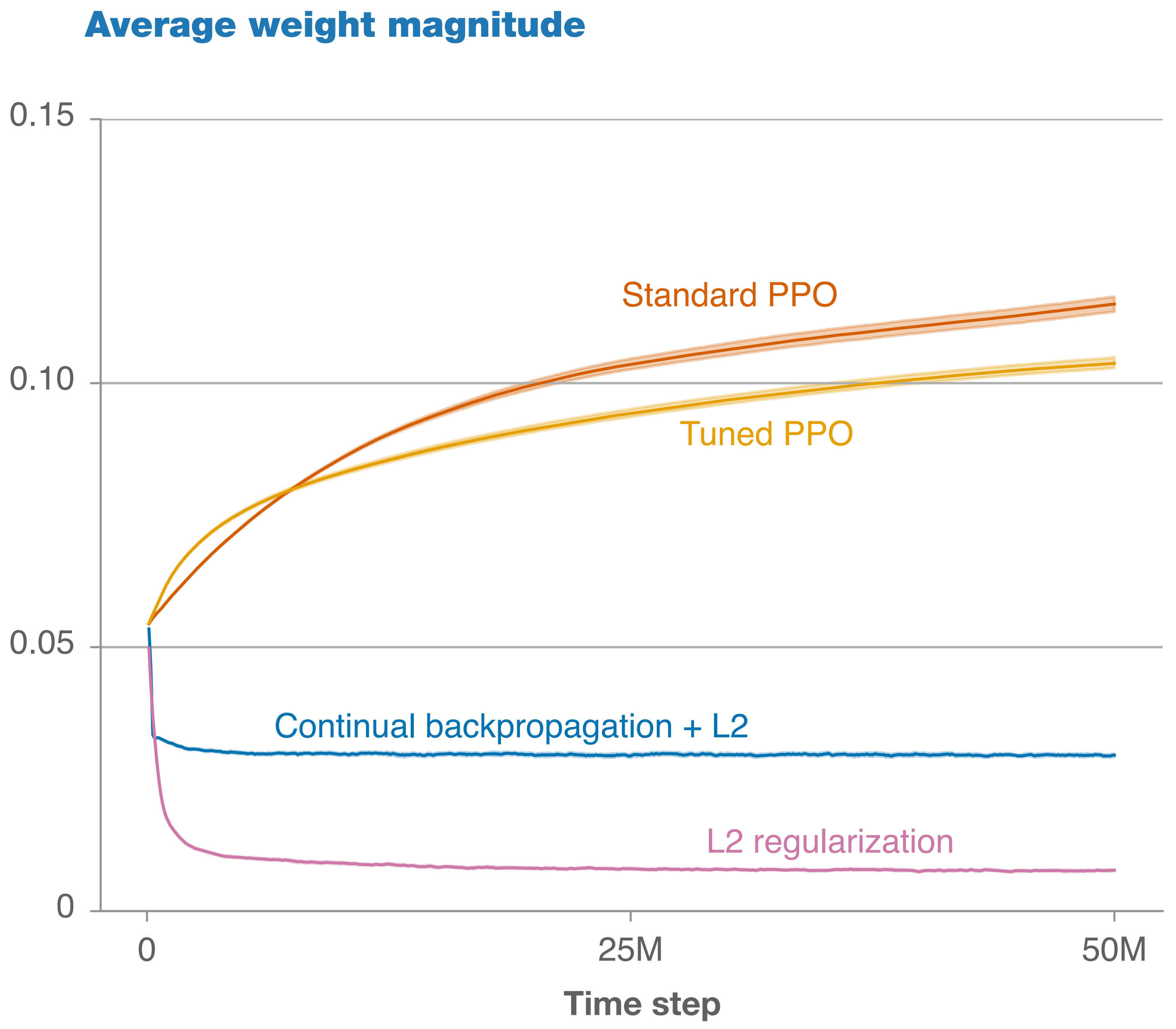


Loss of Plasticity in Stationary Ant Locomotion

A closer look at the results







Conclusions

- Deep-learning networks are optimized for one-time learning, and in a sense they totally fail for continual learning
- Simple changes, like Continual Backprop, can make them effective for continual learning
- Continual Backprop ranks units by their utility to the network, and preserves the most useful
    - The specific way ranking is done in Continual Backprop is probably not the last word; for example, new ideas are needed to extend to recurrent networks
- There is an exciting world ahead of deep-learning networks that can learn continually
- Deep continual learning opens up new possibilities in RL (which is inherently continual due to policy iteration) and model-based RL (with multiple interacting learning components)

Outline

- Demonstrations of Loss of Plasticity in deep continual learning and attempts to maintain it (w/L2 reg, Shrink & Perturb, Continual Backprop)
  - in convolution networks on continual versions of ImageNet
  - in residual networks on continual versions CIFAR-100
  - in stationary and non-stationary reinforcement learning
- Some goals and ideas for a new streaming version of deep learning

We need new deep learning methods that

- Can learn continually without loss of plasticity
- Can learn non-linear functions of great depth and complexity
- Are maximally efficient in data and computational resources
  - Computational complexity O(#weights) per example
  - Streaming (each example processed once, no replay buffer)
- Can meta-learn to generalize better

Ideally, streaming deep learning should adapt at 3 levels

1. Adapting the weights w_t (SGD w/randomness)

2. Adapting the step sizes α_t (building on IDBD*)

3. Adapting the connections between units (who connects to who)

* Sutton, R.S., "Adapting Bias by Gradient Descent: An Incremental Version of Delta-Bar-Delta," ICML 1992.

The first and most important idea:

Distinguish the part of the network that has <u>already been learned</u> (the ‘backbone’) from the rest of the network (the ‘fringe’)

Preserve and protect the backbone; let the fringe explore

Some first principles questions

- Are artificial neural networks enough? (as a structure)
- Is it enough to study supervised learning (rather than, e.g., RL)?
- Is stochastic gradient descent enough?
- Must some form of search/randomness be added?

Re-examining stochastic gradient descent (SGD)

$$w_{t+1} = w_t + \alpha_t \circ \nabla_w Error_t$$

- This is a vector equation
- α_t could be a scalar or a matrix. But for us it must be a vector
  - It might be better to call this algorithm "derivative descent"
- α_t cannot change rapidly from example to example
- Changing α_t slowly is OK and powerful: structural credit assignment, meta-learning, representation learning, learning to generalize well

Status:

Streaming deep learning is under active development

- The network version of the step-size optimization algorithm exists
  - for a very general case (paper under review*)
  - for a practical, approximate case (implemented in Lisp)
- We have cool new ideas for adding and adapting new units
  - for the linear case (paper at RLC2024**)
  - demonstrated live in talks (e.g., at Upperbound, Amii TeaTime)
- I am not quite done designing and testing the full DDL algorithm

* Sharifnassab, A., Sutton, R.S., "MetaOptimize: A Framework for Optimizing Step Sizes and Other Meta-parameters"
** Javed, K., Sharifnassab, A., Sutton, R.S., "SwiftTD: A Fast and Robust Algorithm for Temporal Difference Learning," RLC 2024.

Thank you for your attention