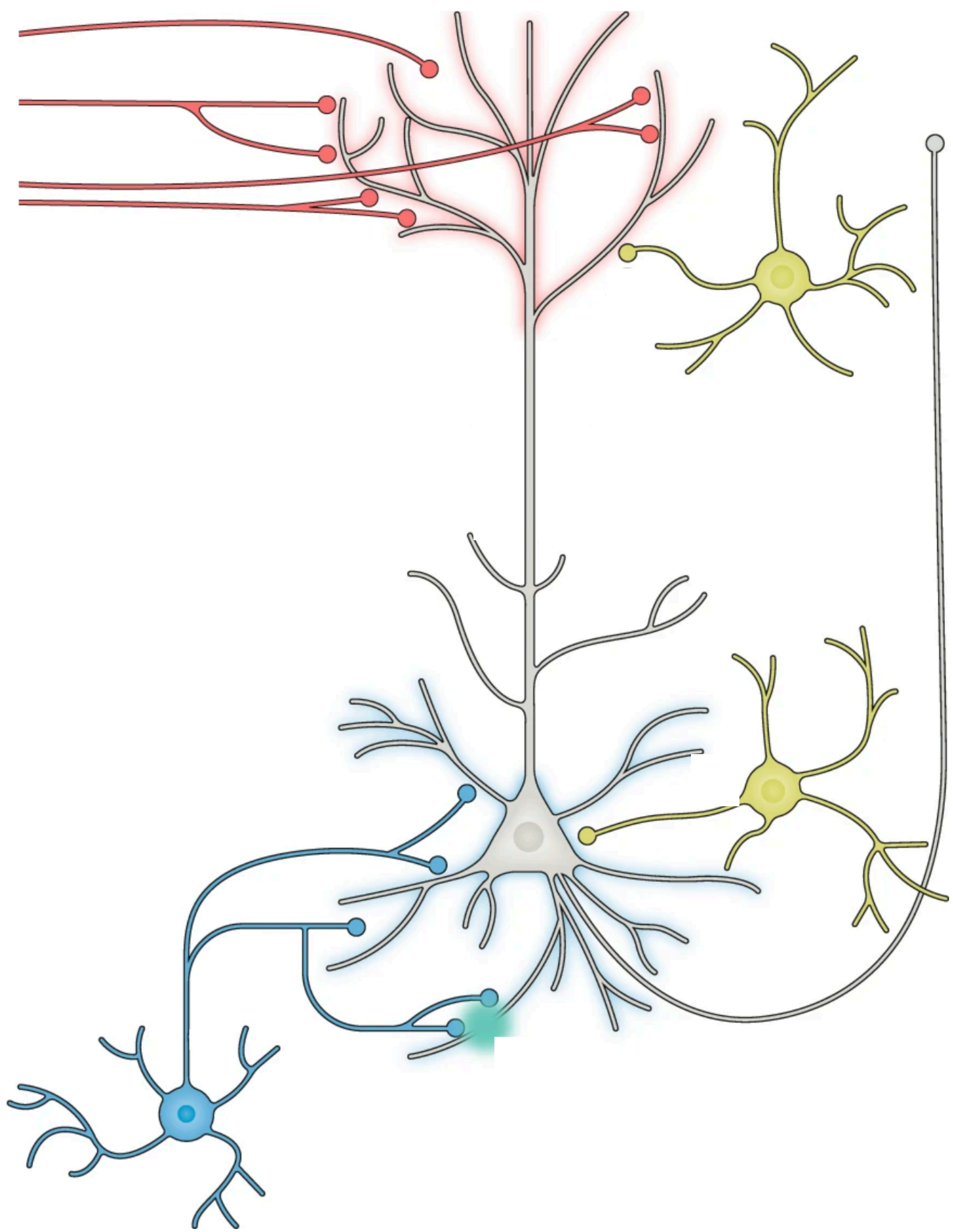
Identifying Learning Rules From Neural Network Observables

Aran Nayebi, Sanjana Srivastava, Surya Ganguli, Daniel L.K. Yamins

NeurIPS 2020 Spotlight

Stanford Neuro AI Lab
Stanford Neuroscience and Artificial Intelligence Lab

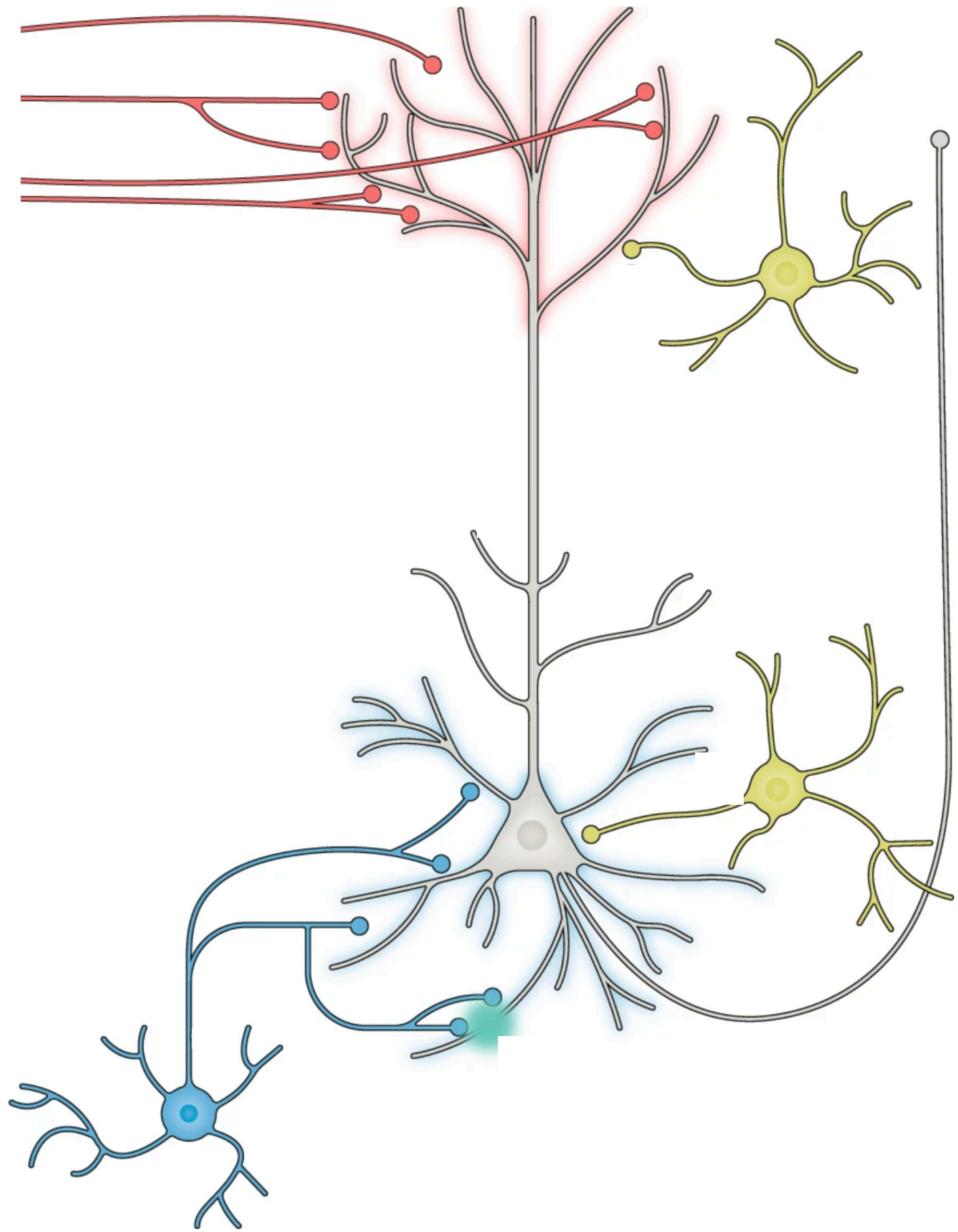
Problem Set-Up



Lillicrap et al.
Nat. Rev. Neurosci. *(2020)*

Problem Set-Up

Post-synaptic activities?

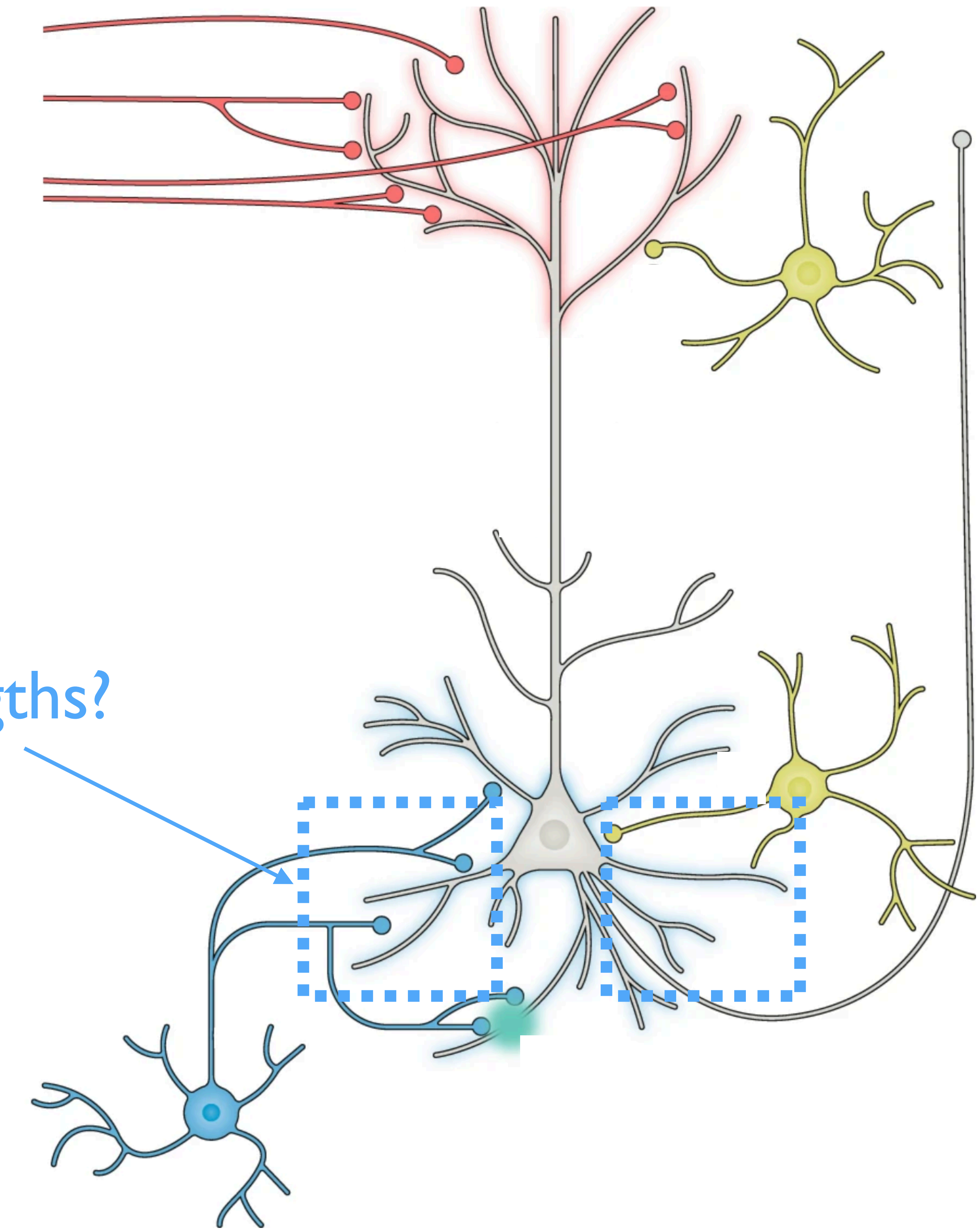


Lillicrap et al.
Nat. Rev. Neurosci. (2020)

Problem Set-Up

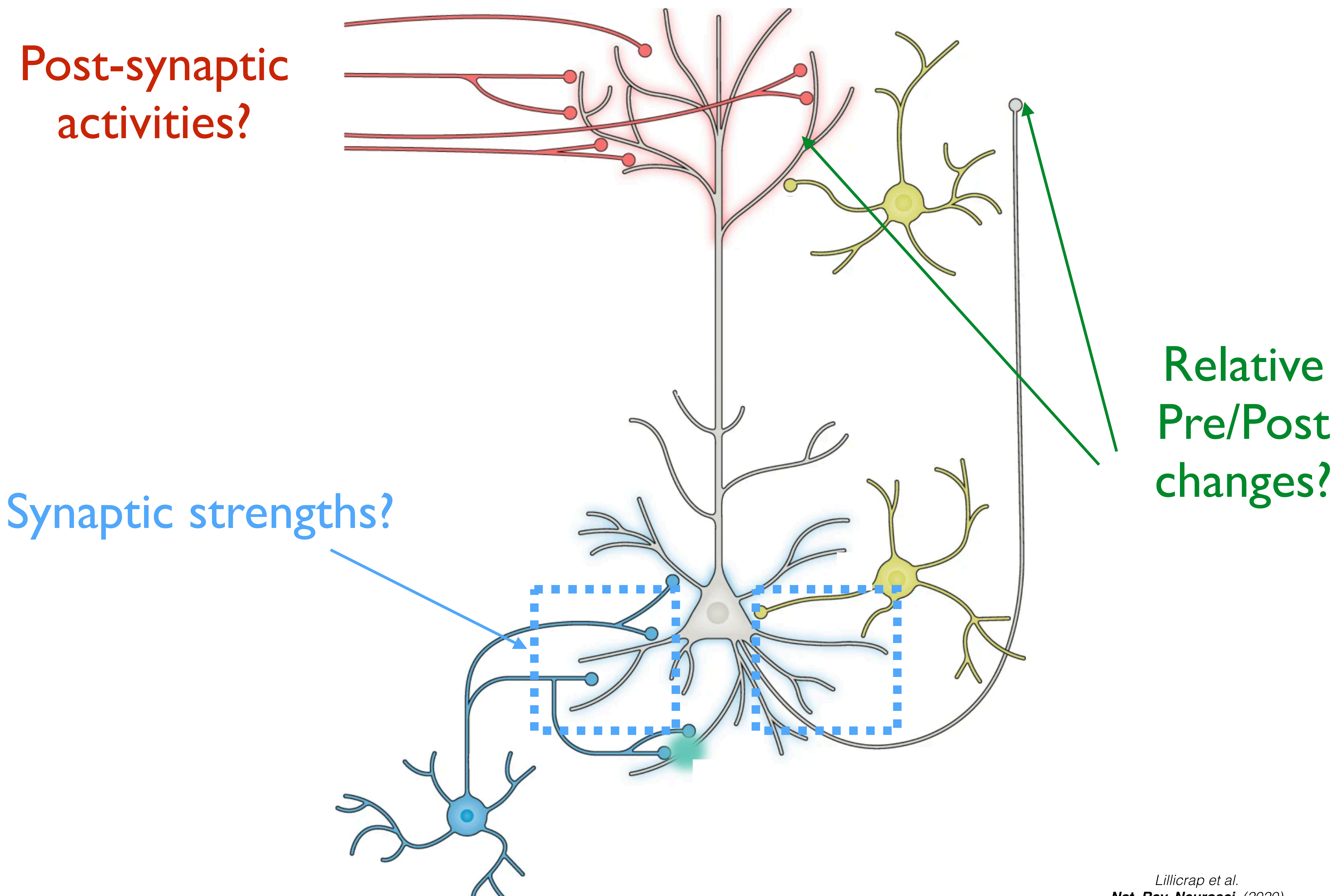
Post-synaptic activities?

Synaptic strengths?



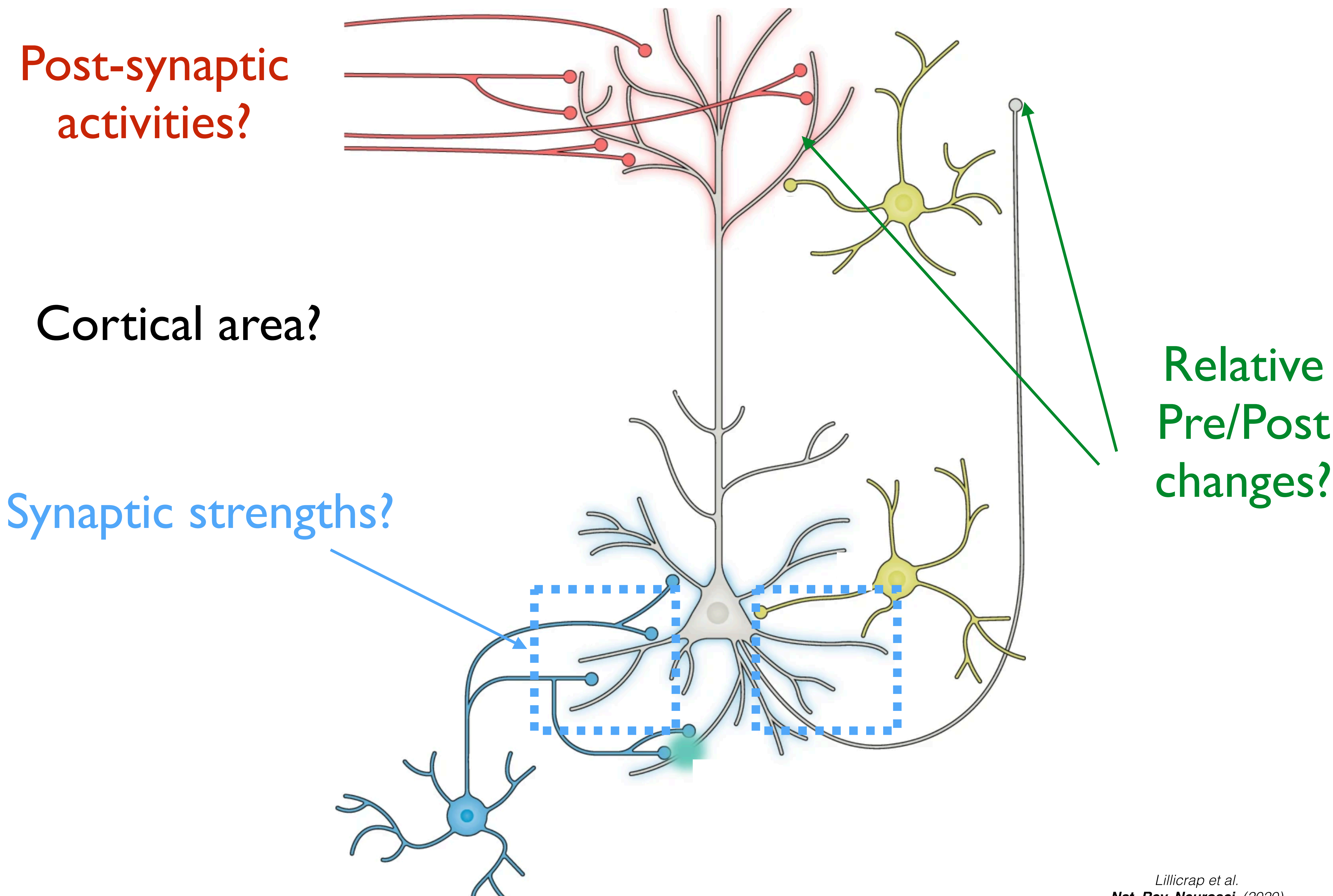
Lillicrap et al.
Nat. Rev. Neurosci. *(2020)*

Problem Set-Up



Lillicrap et al.
Nat. Rev. Neurosci. (2020)

Problem Set-Up

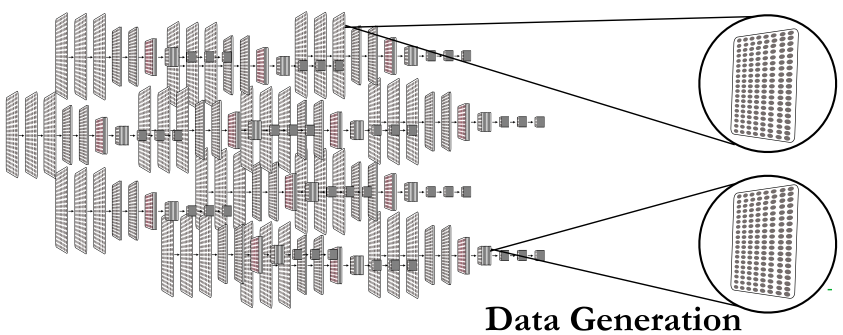


Lillicrap et al.
Nat. Rev. Neurosci. *(2020)*

“Virtual Experimental” Approach

What would you need to measure
to reliably distinguish *classes* of learning rules?

"Virtual Experimental" Approach



Data Generation
10 architectures, 4 tasks, 12 hyperparameter settings, 4 learning rules

Generating a large-scale dataset

Learning Rules

SGD+Momentum (SGDM)

Adam

Information Alignment (IA)

Feedback Alignment (FA)

Tasks

ImageNet (supervised)

SimCLR (self-supervised)

Word-Speaker-Noise (supervised)

CIFAR-10 (supervised)

Architectures

ResNet-34v2

ResNet-34

ResNet-18v2

ResNet-18

AlexNet

AlexNet-LRN

KNet4

KNet4-LRN

KNet5

KNet5-LRN

Hyperparameters

Batch size (128, 256, 512)

Model seed (None, 0)

Dataset seed (None, 0)

Generating a large-scale dataset

Learning Rules

SGD+Momentum (SGDM)

Adam

Information Alignment (IA)

Feedback Alignment (FA)

Tasks

ImageNet (supervised)

SimCLR (self-supervised)

Word-Speaker-Noise (supervised)

CIFAR-10 (supervised)

Architectures

ResNet-34v2

ResNet-34

ResNet-18v2

ResNet-18

AlexNet

AlexNet-LRN

KNet4

KNet4-LRN

KNet5

KNet5-LRN

Hyperparameters

Batch size (128, 256, 512)

Model seed (None, 0)

Dataset seed (None, 0)

Generating a large-scale dataset

Learning Rules

SGD+Momentum (SGDM)

Adam

Information Alignment (IA)

Feedback Alignment (FA)

Tasks

ImageNet (supervised)

SimCLR (self-supervised)

Word-Speaker-Noise (supervised)

CIFAR-10 (supervised)

Architectures

ResNet-34v2

ResNet-34

ResNet-18v2

ResNet-18

AlexNet

AlexNet-LRN

KNet4

KNet4-LRN

KNet5

KNet5-LRN

Hyperparameters

Batch size (128, 256, 512)

Model seed (None, 0)

Dataset seed (None, 0)

Generating a large-scale dataset

Learning Rules

SGD+Momentum (SGDM)

Adam

Information Alignment (IA)

Feedback Alignment (FA)

Tasks

ImageNet (supervised)

SimCLR (self-supervised)

Word-Speaker-Noise (supervised)

CIFAR-10 (supervised)

Architectures

ResNet-34v2

ResNet-34

ResNet-18v2

ResNet-18

AlexNet

AlexNet-LRN

KNet4

KNet4-LRN

KNet5

KNet5-LRN

Hyperparameters

Batch size (128, 256, 512)

Model seed (None, 0)

Dataset seed (None, 0)

Generating a large-scale dataset

Learning Rules

SGD+Momentum (SGDM)

Adam

Information Alignment (IA)

Feedback Alignment (FA)

Tasks

ImageNet (supervised)

SimCLR (self-supervised)

Word-Speaker-Noise (supervised)

CIFAR-10 (supervised)

Architectures

ResNet-34v2

ResNet-34

ResNet-18v2

ResNet-18

AlexNet

AlexNet-LRN

KNet4

KNet4-LRN

KNet5

KNet5-LRN

Hyperparameters

Batch size (128, 256, 512)

Model seed (None, 0)

Dataset seed (None, 0)

Generating a large-scale dataset

Learning Rules

SGD+Momentum (SGDM)

Adam

Information Alignment (IA)

Feedback Alignment (FA)

Tasks

ImageNet (supervised)

SimCLR (self-supervised)

Word-Speaker-Noise (supervised)

CIFAR-10 (supervised)

Architectures

ResNet-34v2

ResNet-34

ResNet-18v2

ResNet-18

AlexNet

AlexNet-LRN

KNet4

KNet4-LRN

KNet5

KNet5-LRN

Hyperparameters

Batch size (128, 256, 512)

Model seed (None, 0)

Dataset seed (None, 0)

Generating a large-scale dataset

Learning Rules

SGD+Momentum (SGDM)

Adam

Information Alignment (IA)

Feedback Alignment (FA)

Tasks

ImageNet (supervised)

SimCLR (self-supervised)

Word-Speaker-Noise (supervised)

CIFAR-10 (supervised)

Architectures

ResNet-34v2

ResNet-34

ResNet-18v2

ResNet-18

AlexNet

AlexNet-LRN

KNet4

KNet4-LRN

KNet5

KNet5-LRN

Hyperparameters

Batch size (128, 256, 512)

Model seed (None, 0)

Dataset seed (None, 0)

Generating a large-scale dataset

Learning Rules

SGD+Momentum (SGDM)

Adam

Information Alignment (IA)

Feedback Alignment (FA)

Tasks

ImageNet (supervised)

SimCLR (self-supervised)

Word-Speaker-Noise (supervised)

CIFAR-10 (supervised)

Architectures

ResNet-34v2

ResNet-34

ResNet-18v2

ResNet-18

AlexNet

AlexNet-LRN

KNet4

KNet4-LRN

KNet5

KNet5-LRN

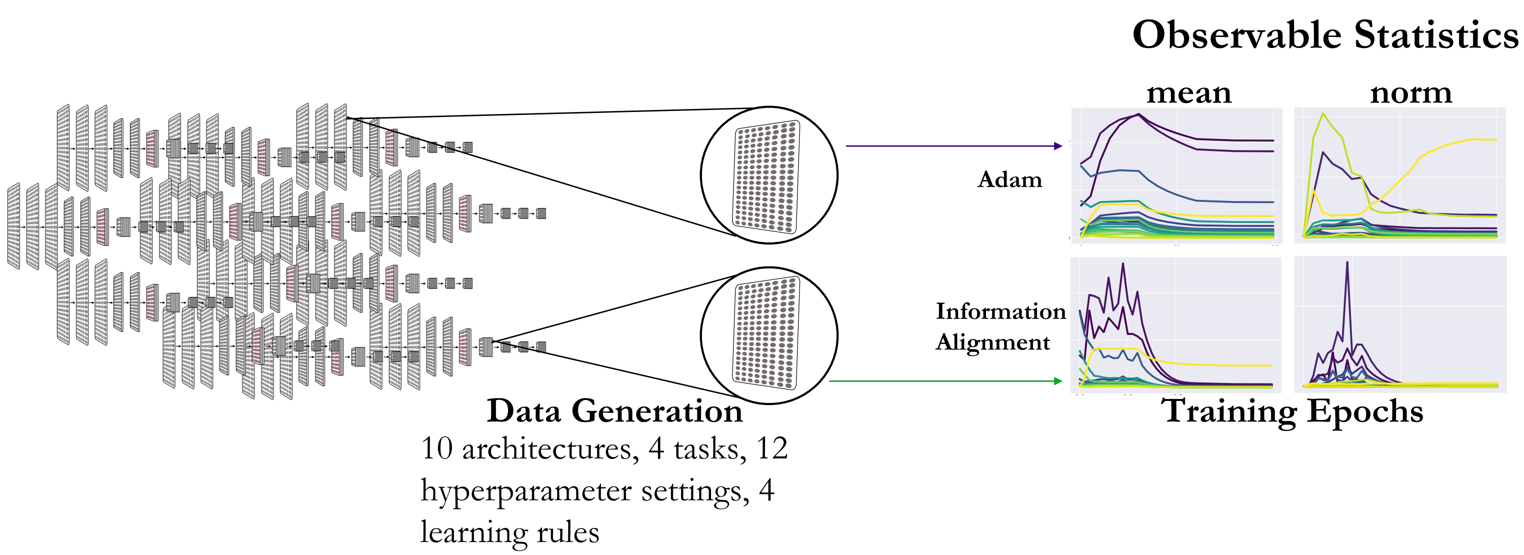
Hyperparameters

Batch size (128, 256, 512)

Model seed (None, 0)

Dataset seed (None, 0)

"Virtual Experimental" Approach



Defining observable statistics

Weights

Proxy for synaptic strengths

Activations

Proxy for post-synaptic activities

Layer-wise Activity Changes

Proxy for relative change between pre- and post-synaptic activations

Defining observable statistics

Weights

Activations

Layer-wise Activity Changes

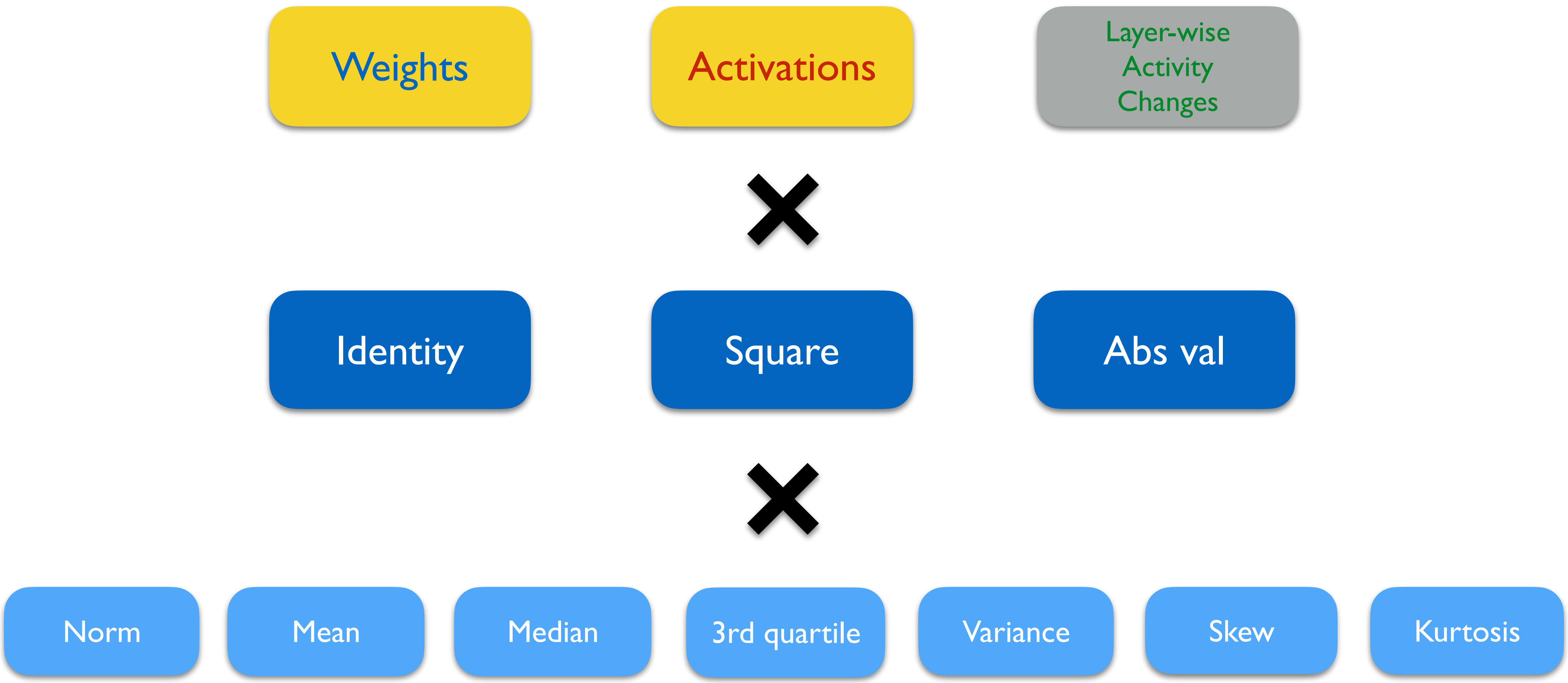
×

Identity

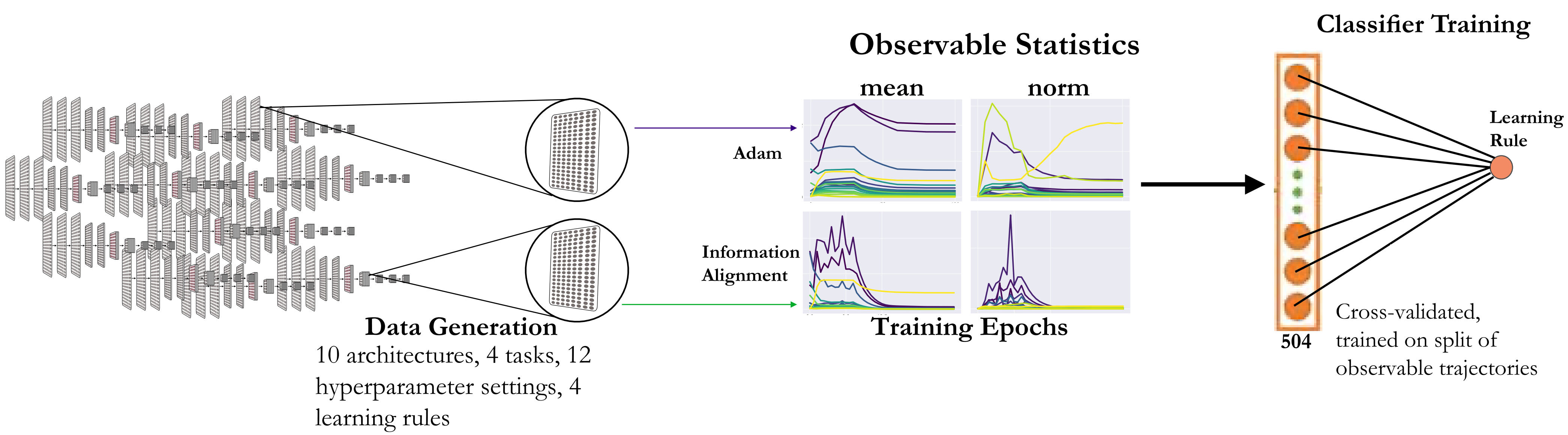
Square

Abs val

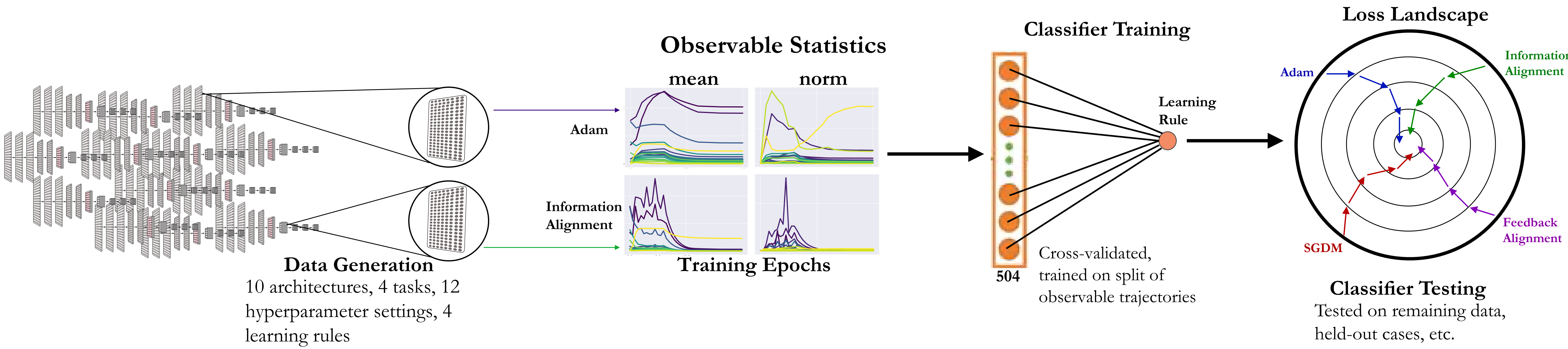
Defining observable statistics



"Virtual Experimental" Approach



"Virtual Experimental" Approach



Is this Problem Even Tractable?

Visualizing observables on ImageNet

Visualizing observables on ImageNet

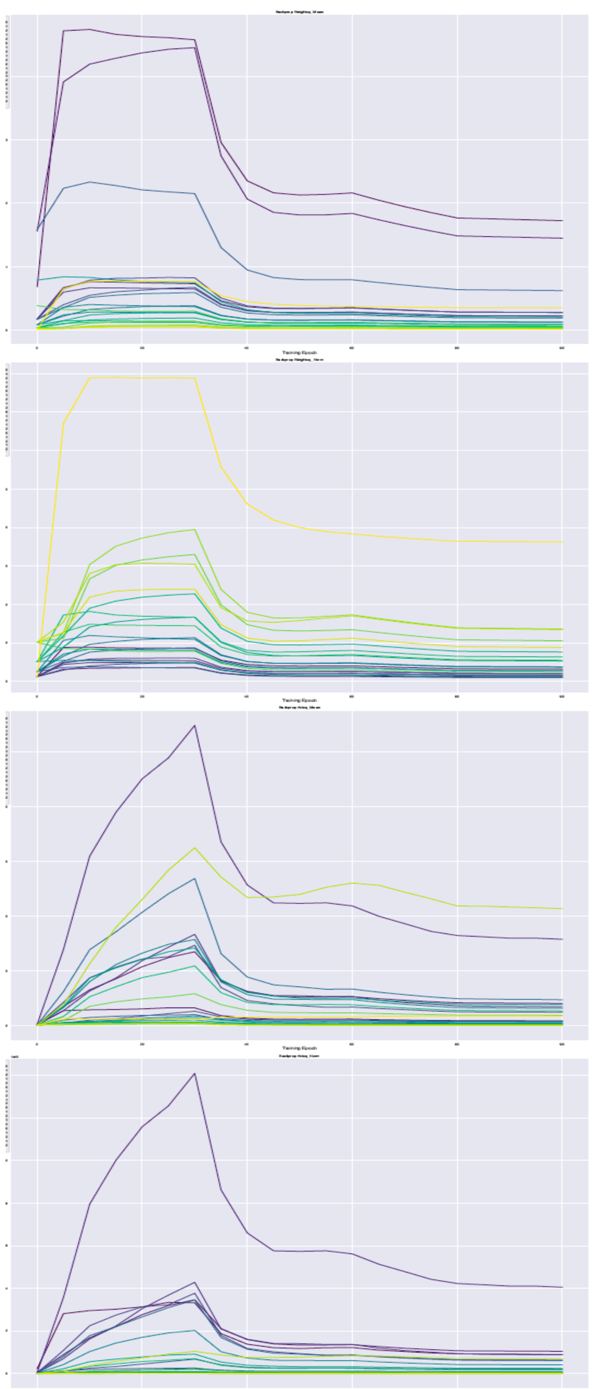
SGDM

weight-square-mean

weight-square-norm

activation-square-mean

activation-square-norm



Visualizing observables on ImageNet

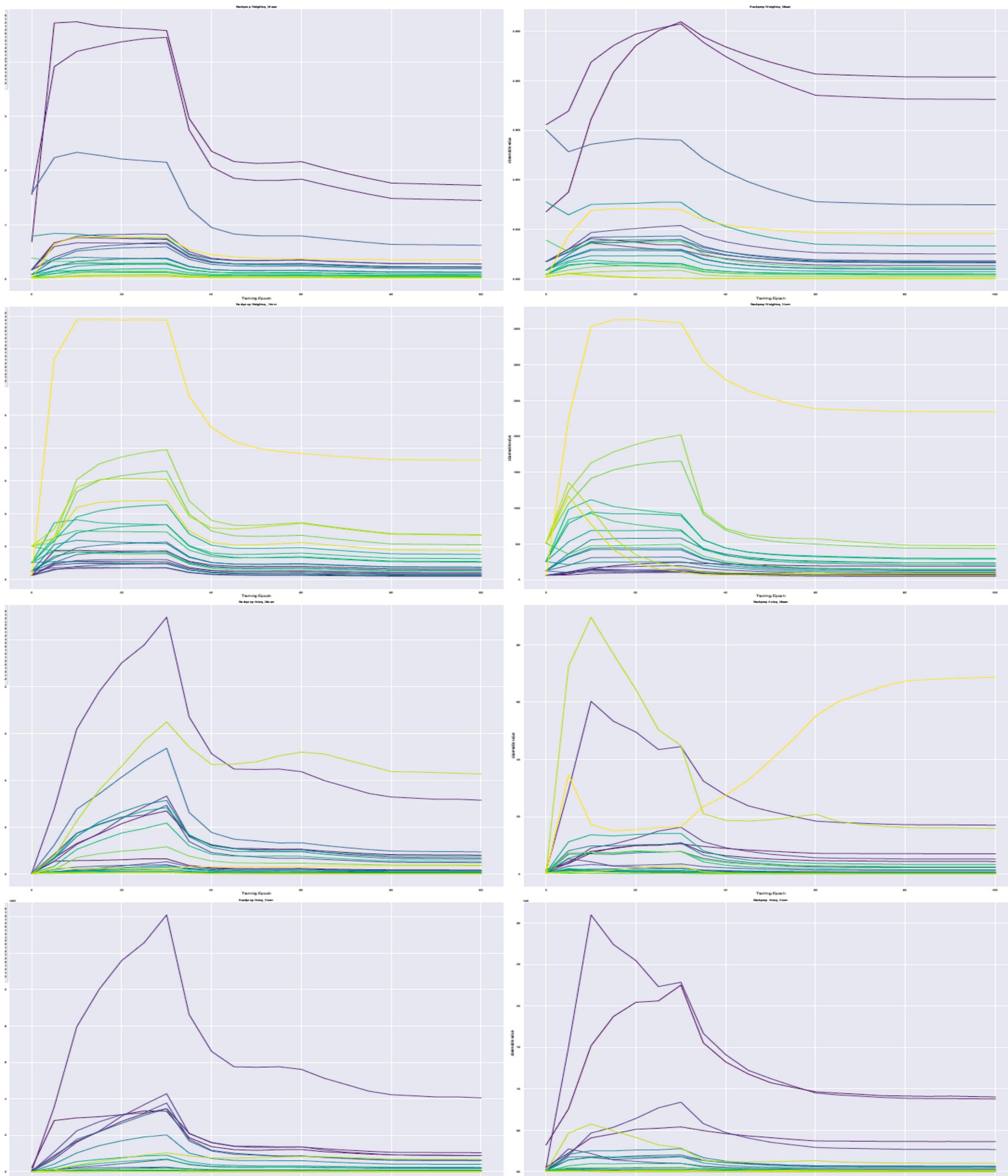
SGDM | Adam

weight-square-mean

weight-square-norm

activation-square-mean

activation-square-norm



Visualizing observables on ImageNet

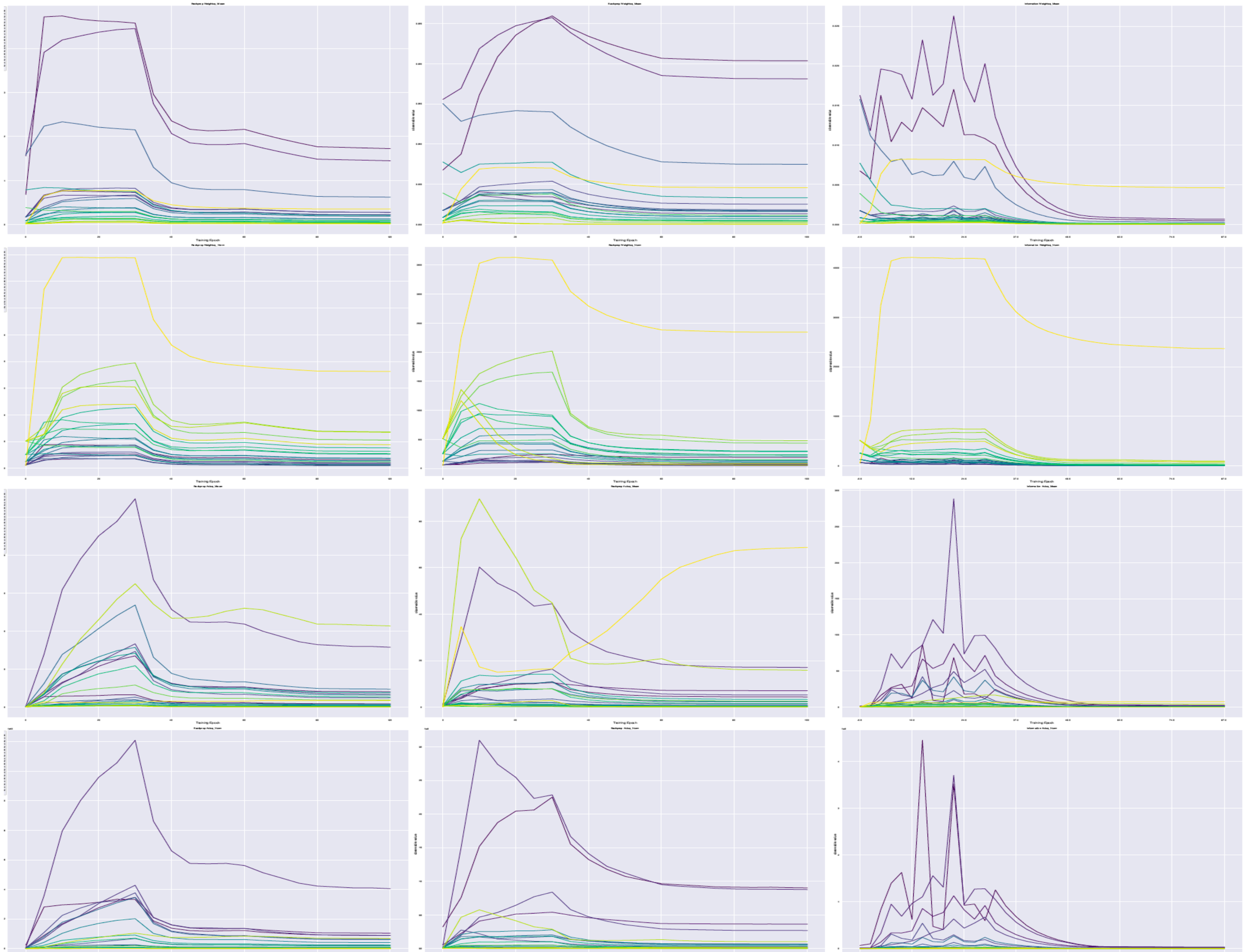
SGDM | Adam | IA

weight-square-mean

weight-square-norm

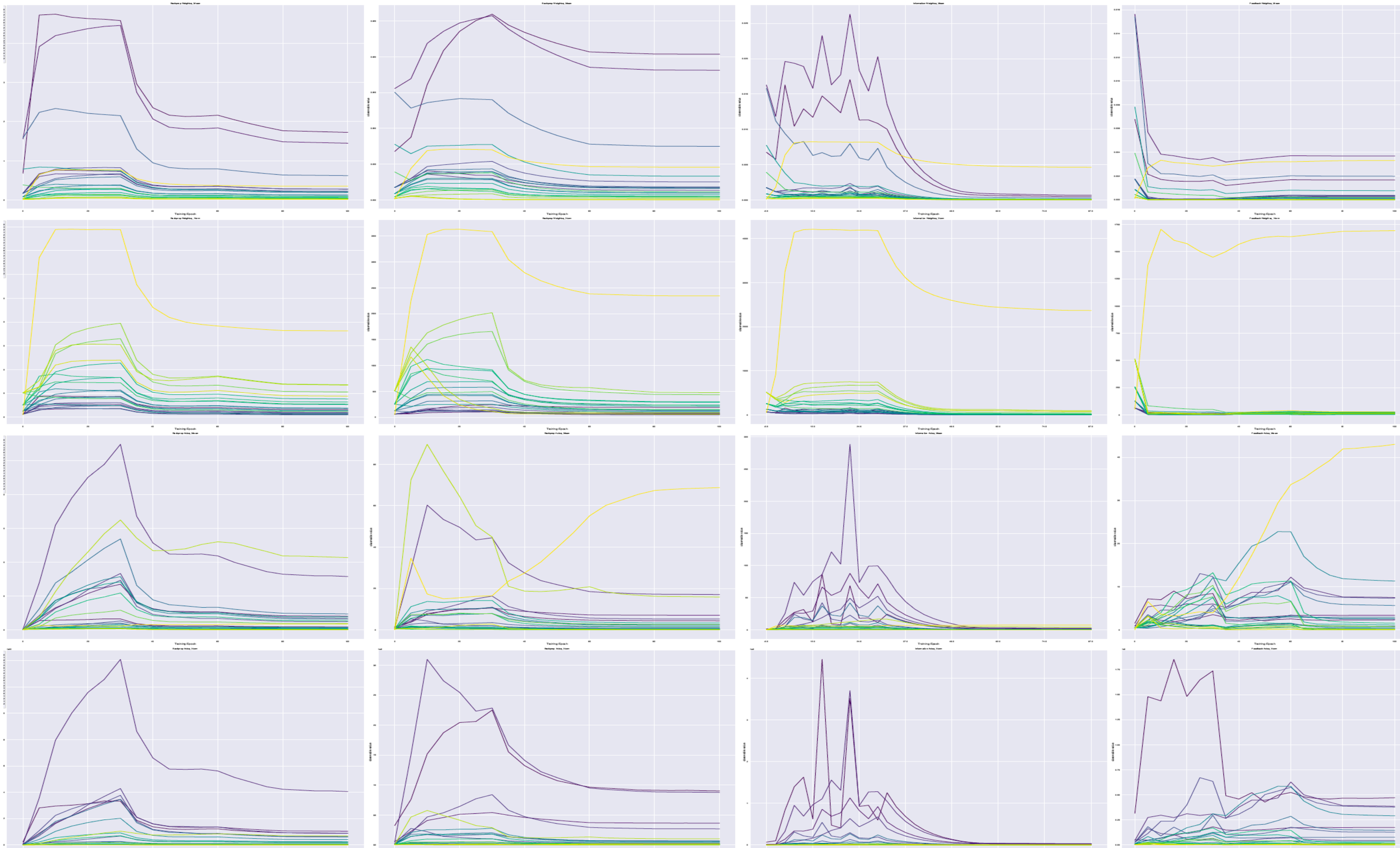
activation-square-mean

activation-square-norm

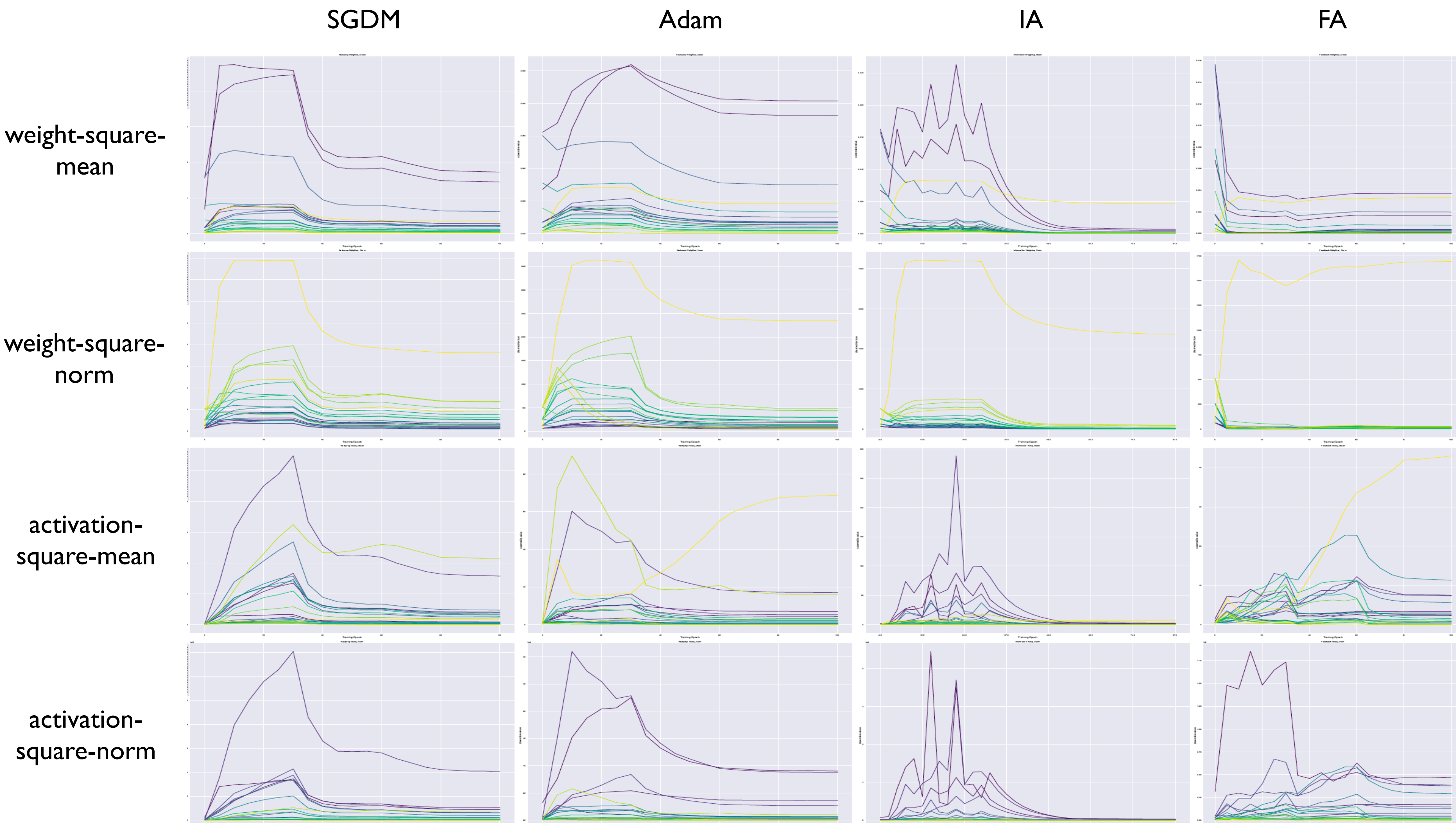


Visualizing observables on ImageNet

	SGDM	Adam	IA	FA
weight-square-mean				
weight-square-norm				
activation-square-mean				
activation-square-norm				



Visualizing observables on ImageNet

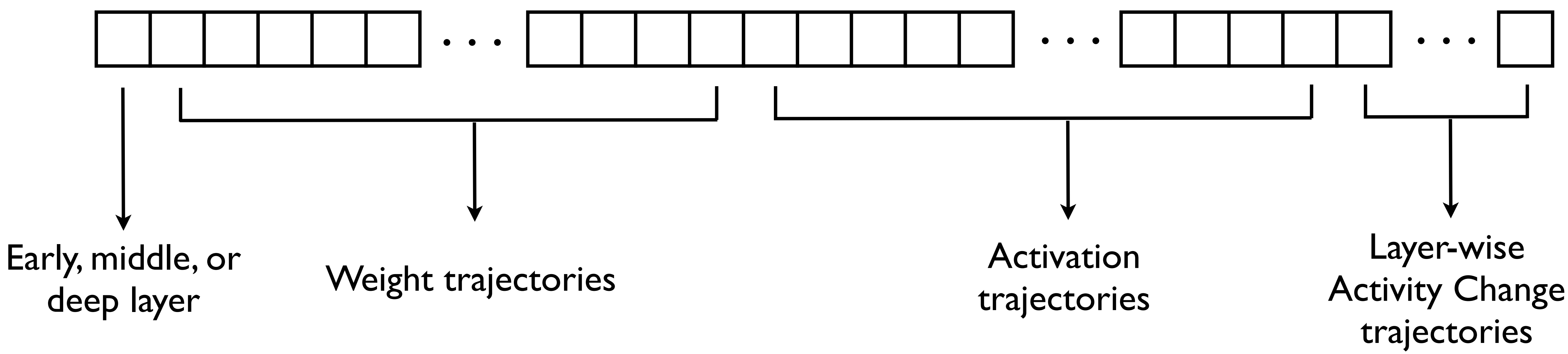


Trajectories across network training appear highly distinctive

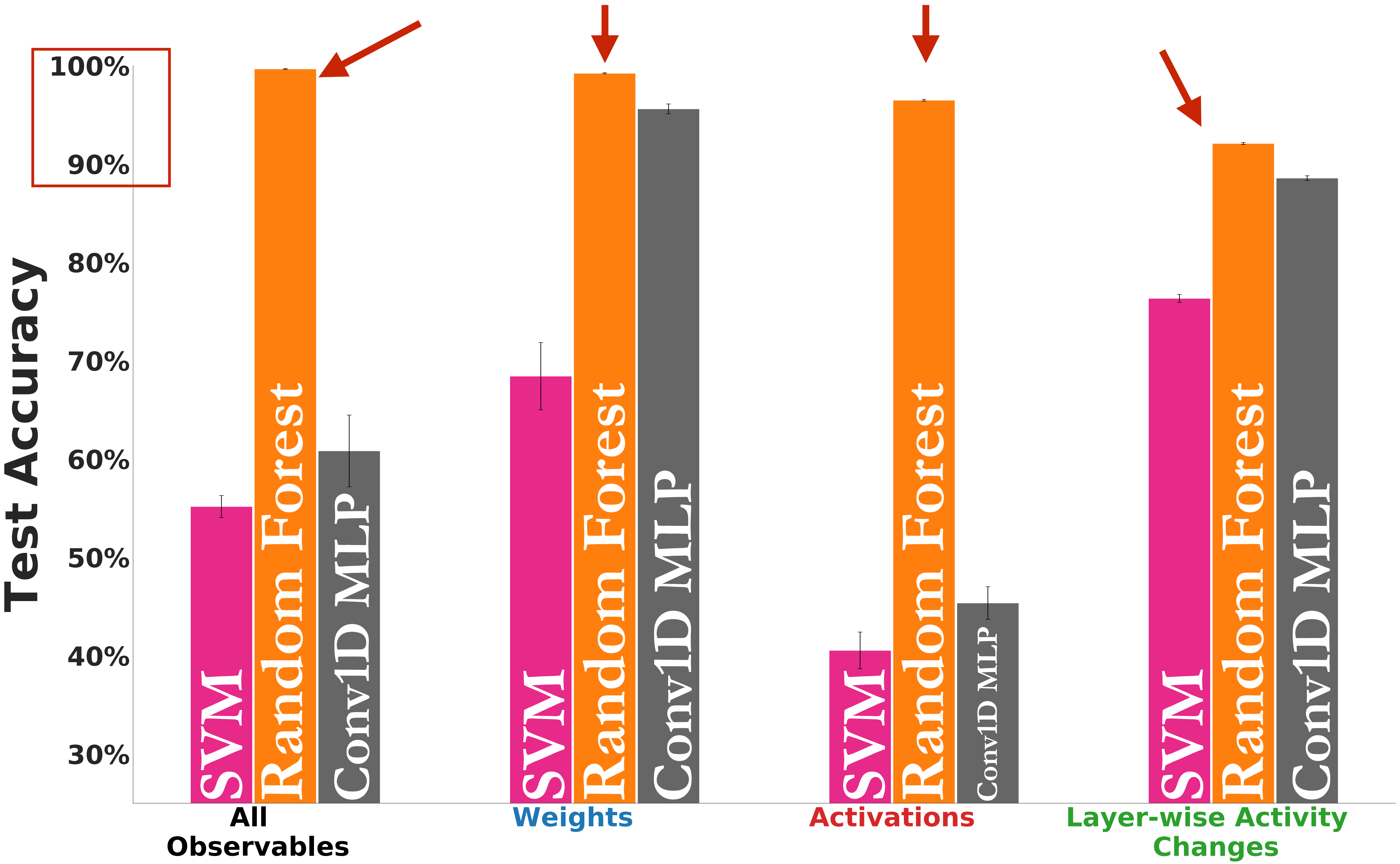
Framing it as a classification problem

How well can we do by framing it as a classification problem?

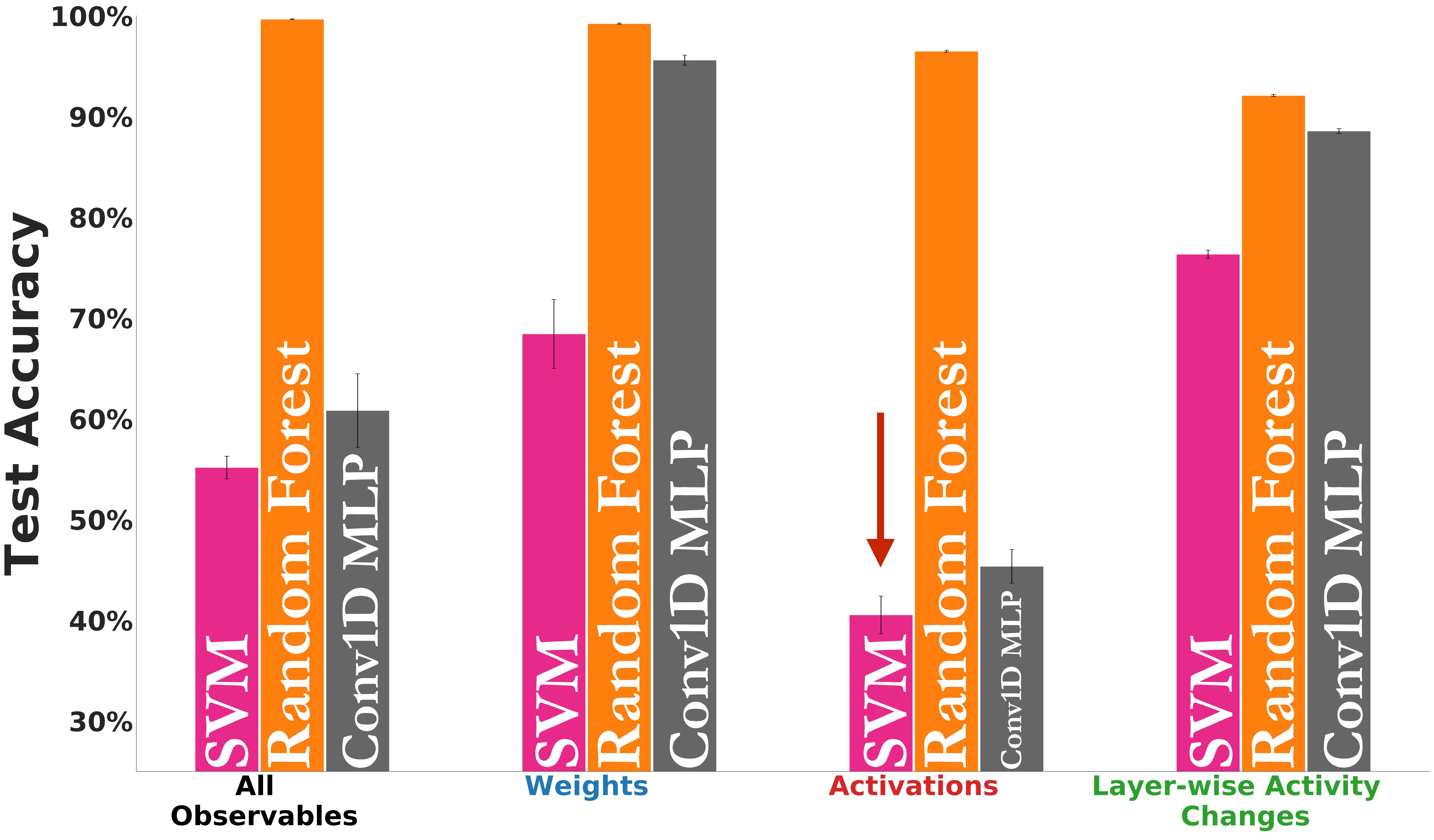
Sample is constructed from one layer of a trained network



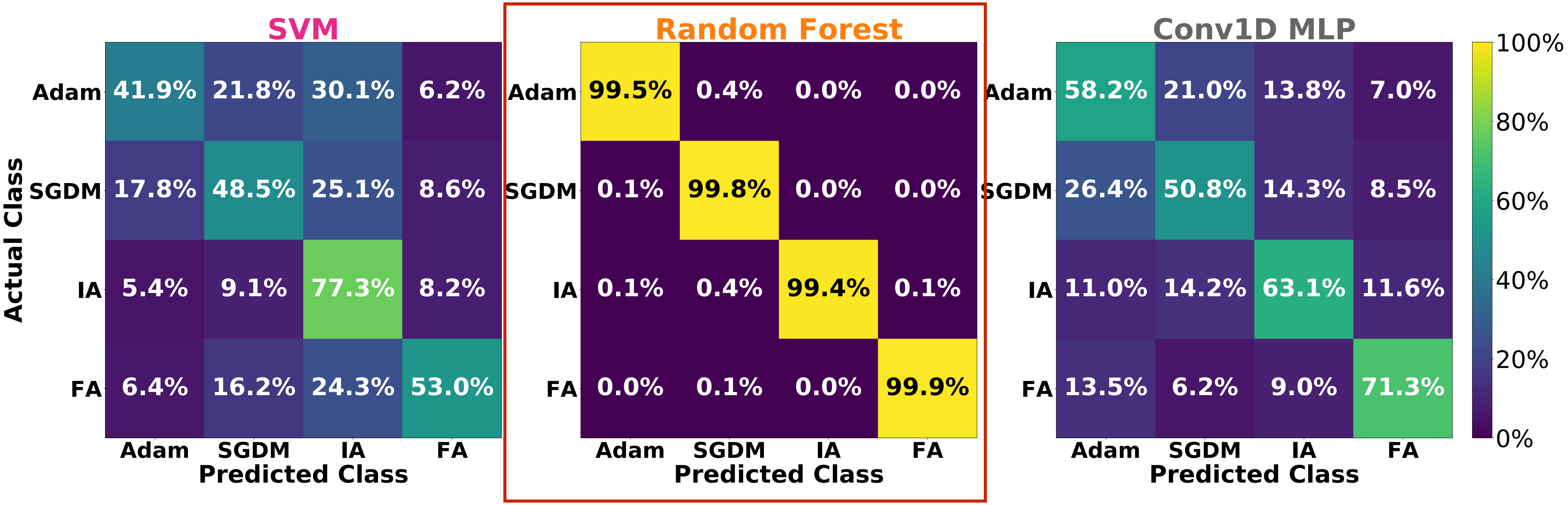
General separability problem *is* tractable



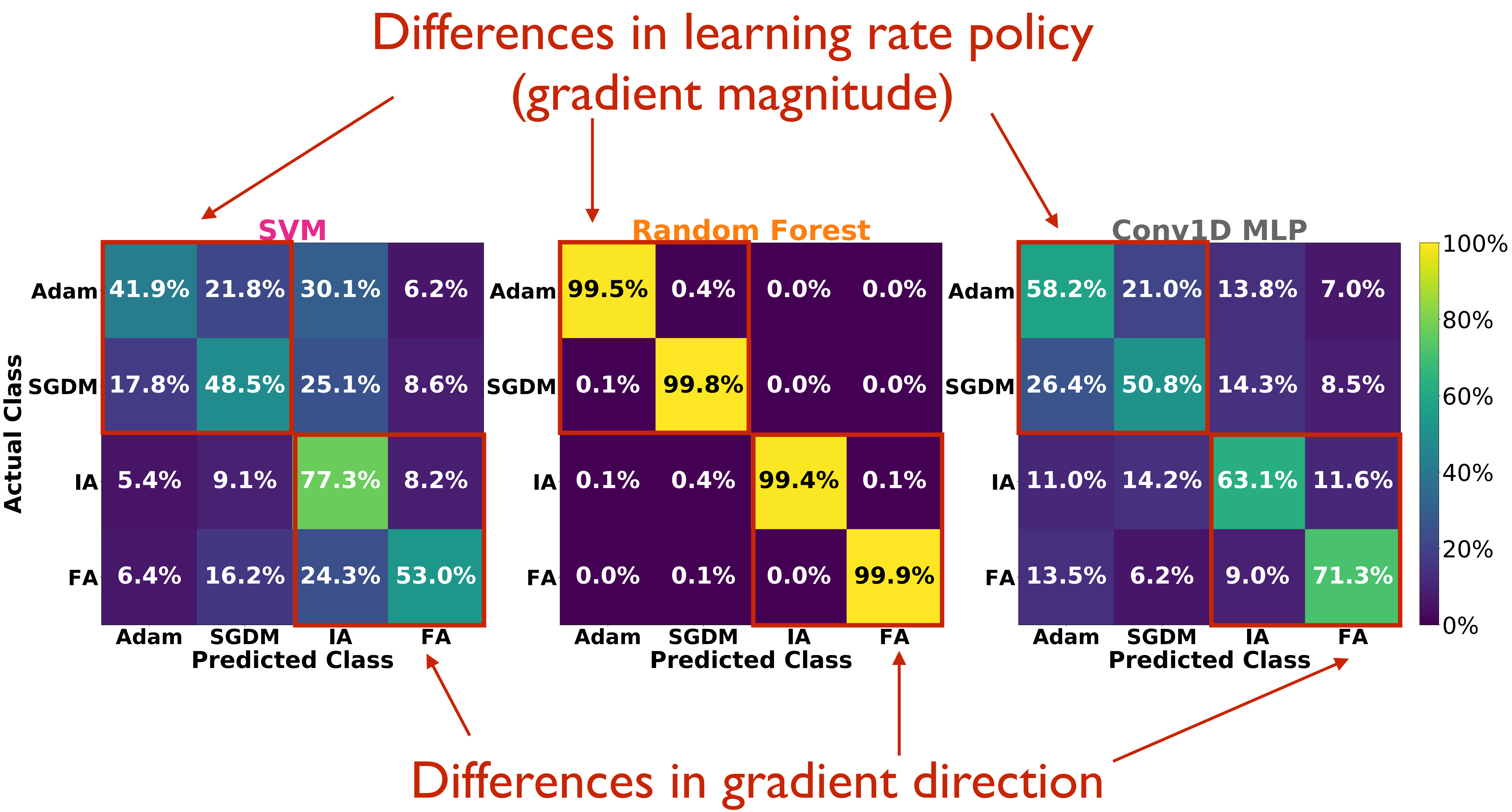
Learning rules are not *linearly* separable from the activations



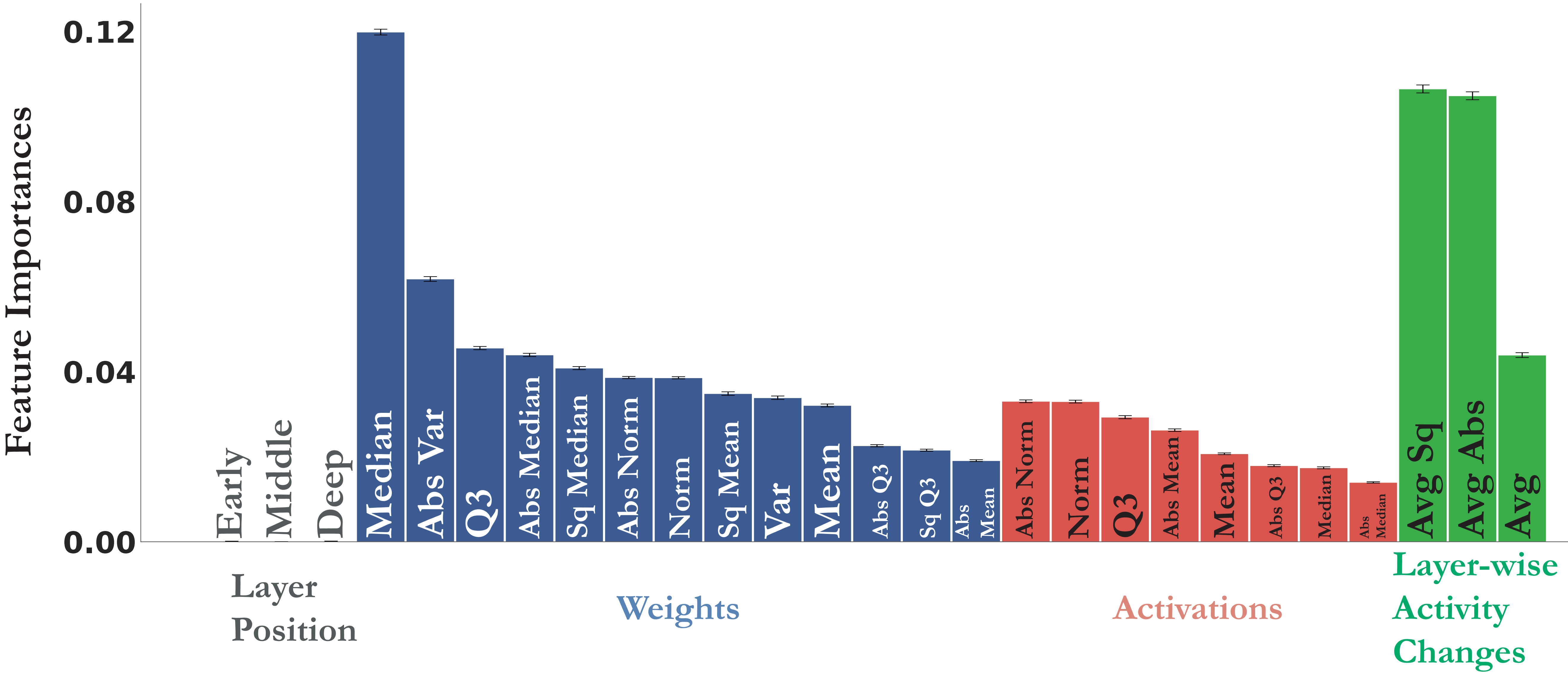
Random Forest makes few mistakes



Differences in learning rate policy harder to distinguish



Not all aggregate statistics are useful



Adding Experimental Realism

Removing certain "animals" or "training curricula": holdouts of entire input classes

Access to only portions of the learning trajectory: subsampling observable trajectories

Incomplete and noisy measurements: subsampling units and Gaussian noise before collecting observables

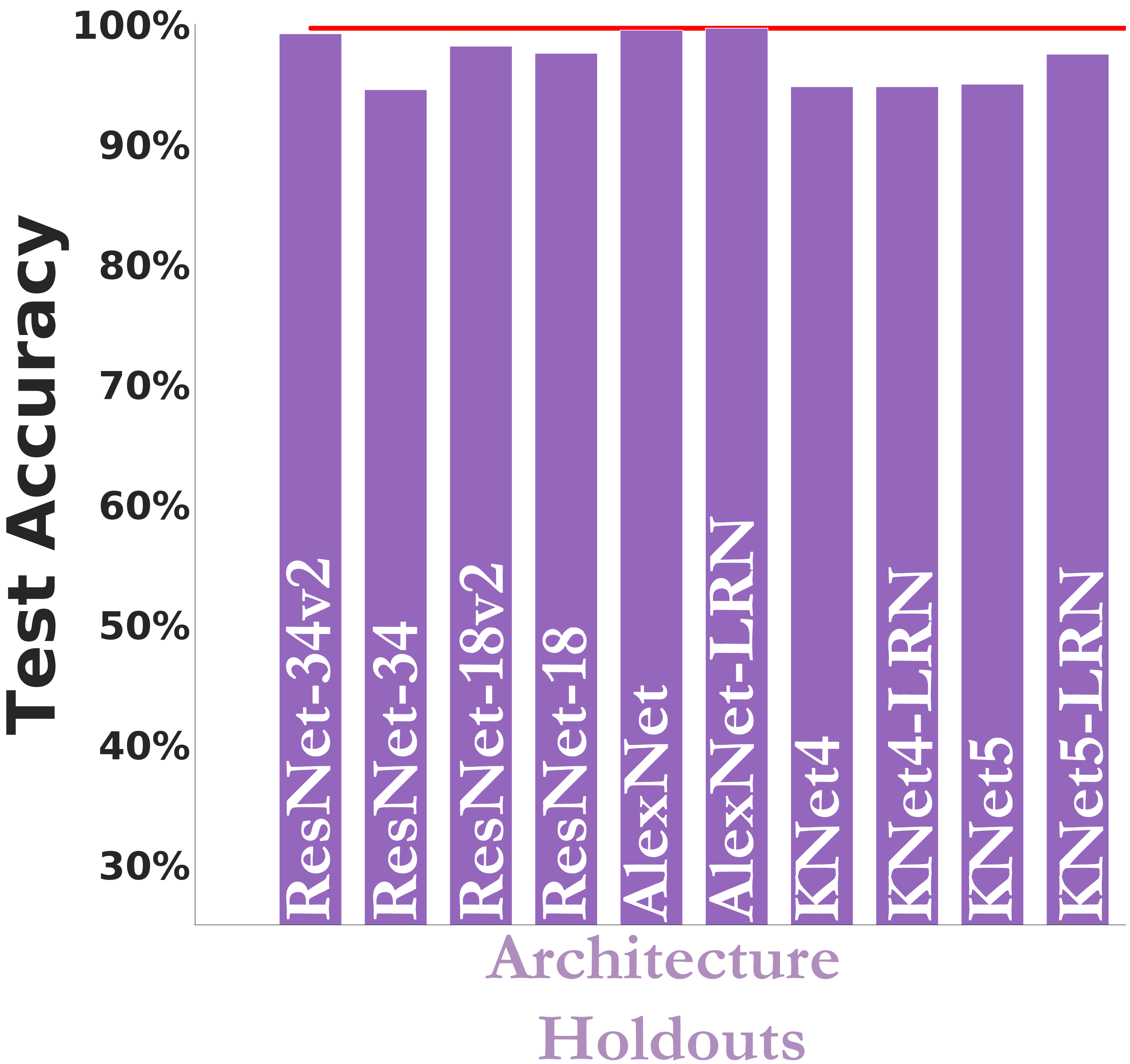
Adding Experimental Realism

Removing certain "animals" or "training curricula": holdouts of entire input classes

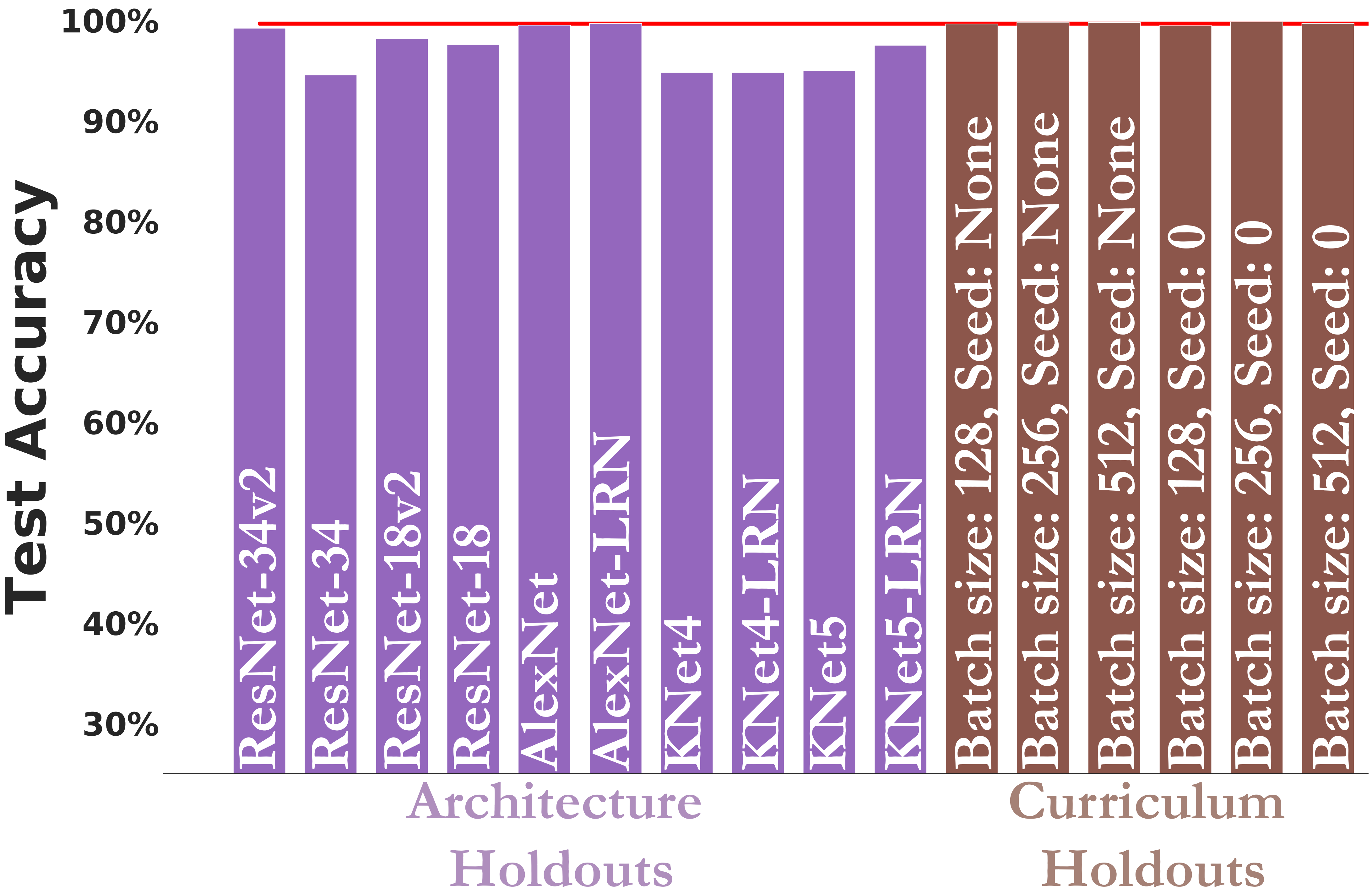
Access to only portions of the learning trajectory: subsampling observable trajectories

Incomplete and noisy measurements: subsampling units and Gaussian noise before collecting observables

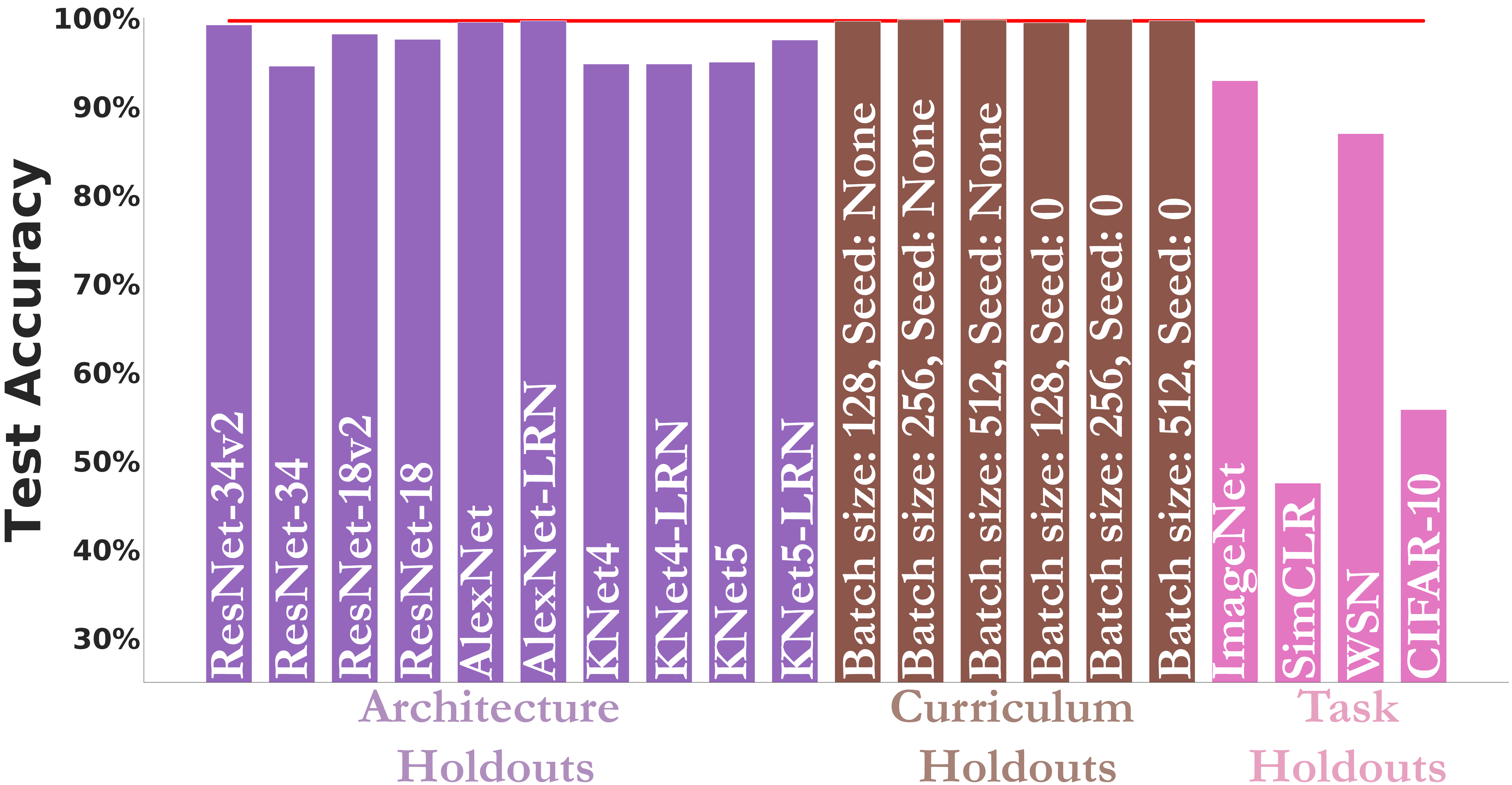
Generalization to held-out “animals”



Generalization to held-out "training curricula"



Quantifying learning differences between tasks



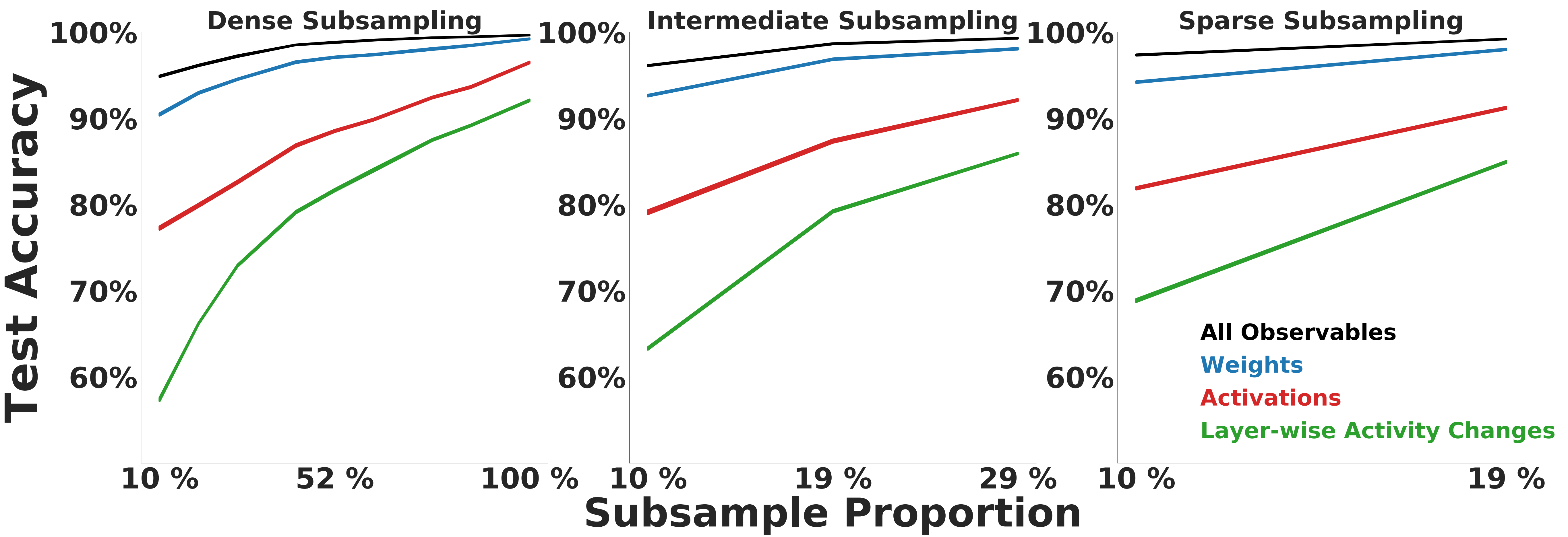
Adding Experimental Realism

Removing certain "animals" or "training curricula": holdouts of entire input classes

Access to only portions of the learning trajectory: subsampling observable trajectories

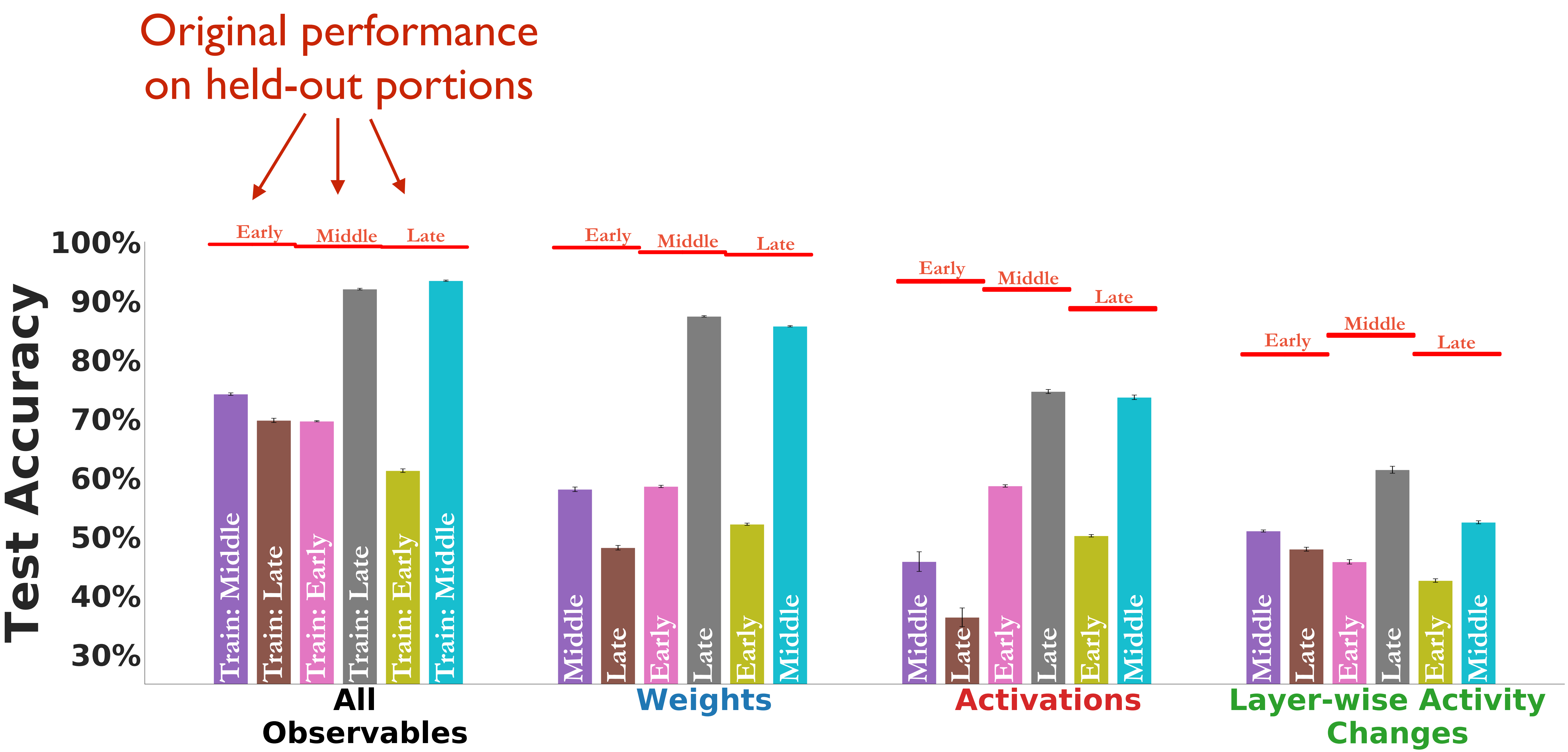
Incomplete and noisy measurements: subsampling units and Gaussian noise before collecting observables

Sparse subsampling *across* learning trajectory robust to trajectory undersampling



Least Robust → Most Robust

Sampling *across* learning trajectory is important for robustness to undersampling



Adding Experimental Realism

Removing certain "animals" or "training curricula": holdouts of entire input classes

Access to only portions of the learning trajectory: subsampling observable trajectories

Incomplete and noisy measurements: subsampling units and Gaussian noise before collecting observables

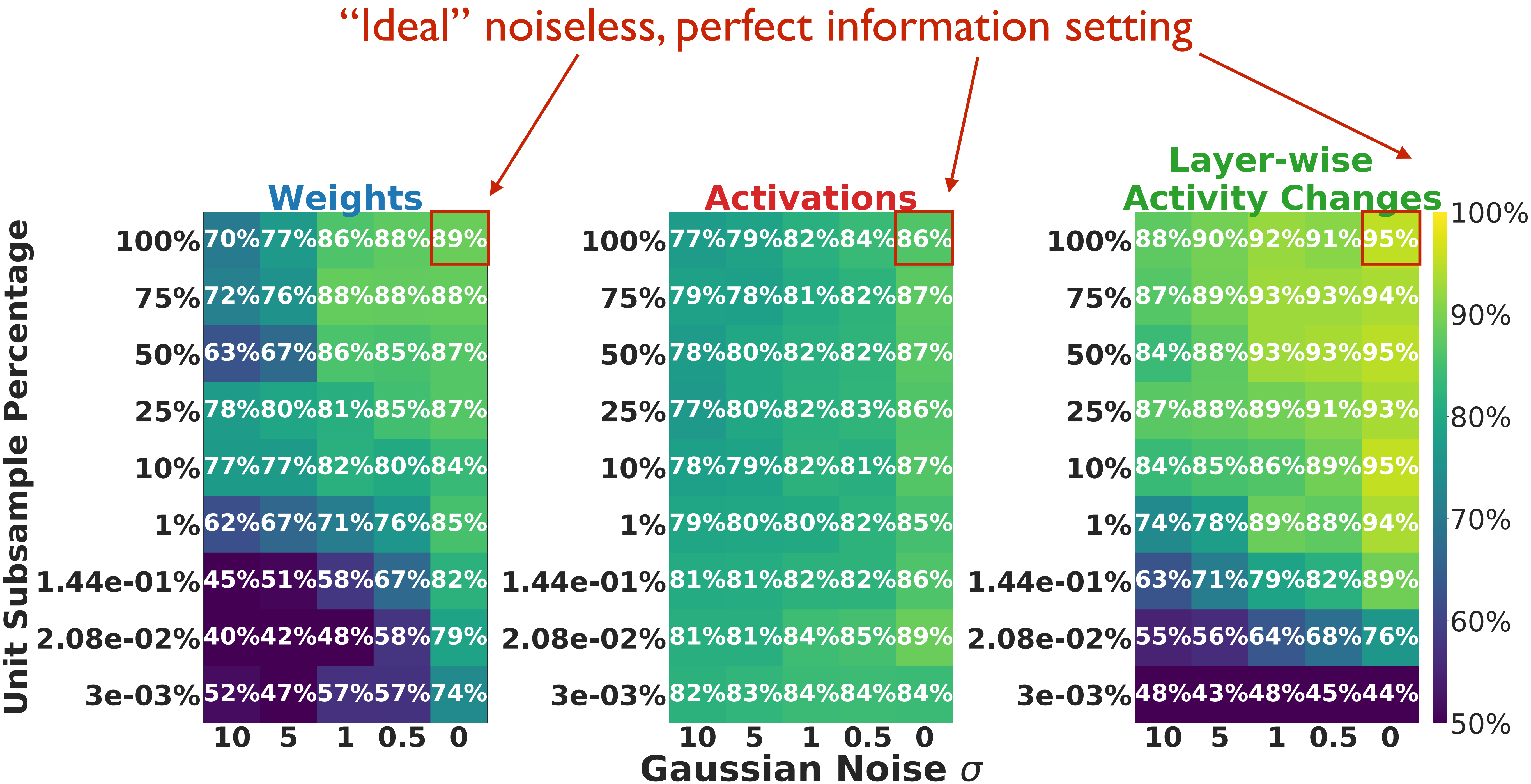
What insights could this approach potentially provide?

Different experimental tools have different limitations

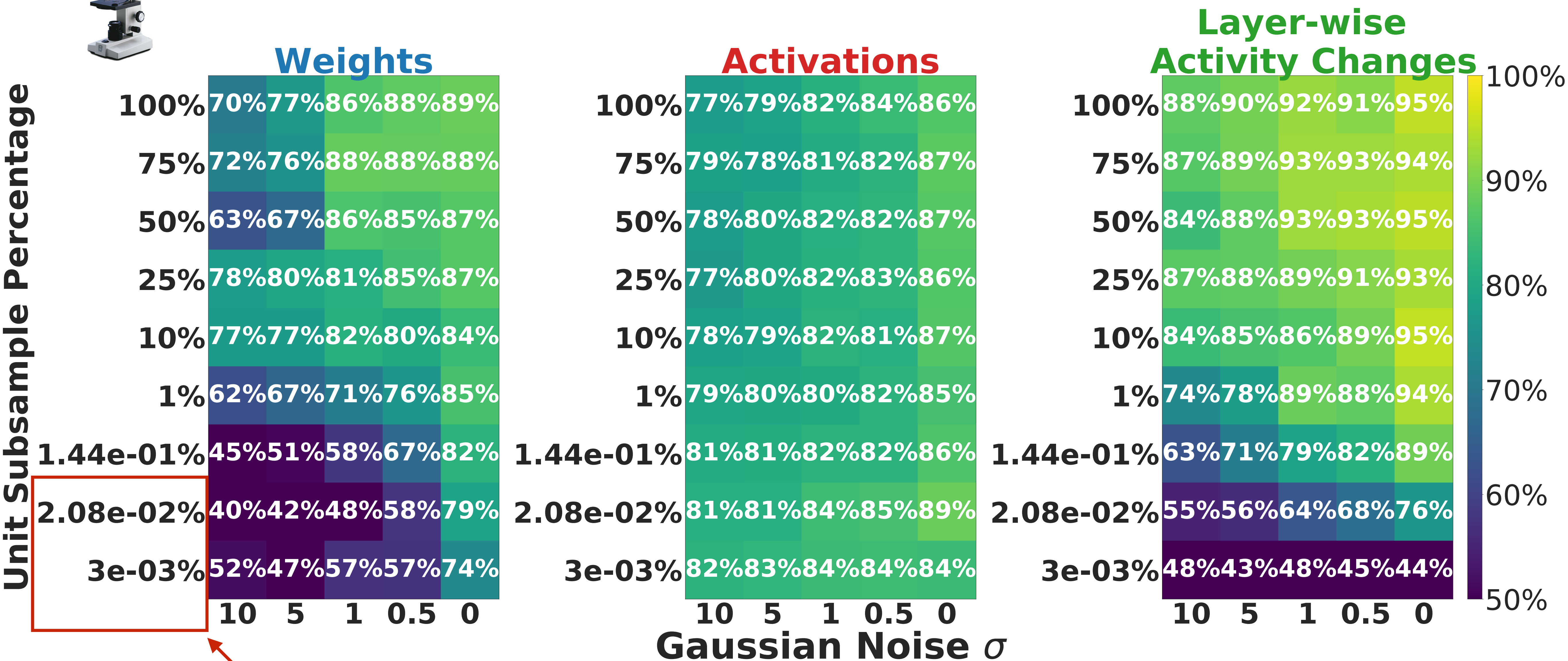
Optical imaging techniques usually give us simultaneous access to thousands of units but can have lower temporal resolution and signal-to-noise

Electrophysiological recordings can have higher signal-to-noise and better temporal resolution, but can lack the coverage to thousands of units

Modeling unit subsampling and measurement noise

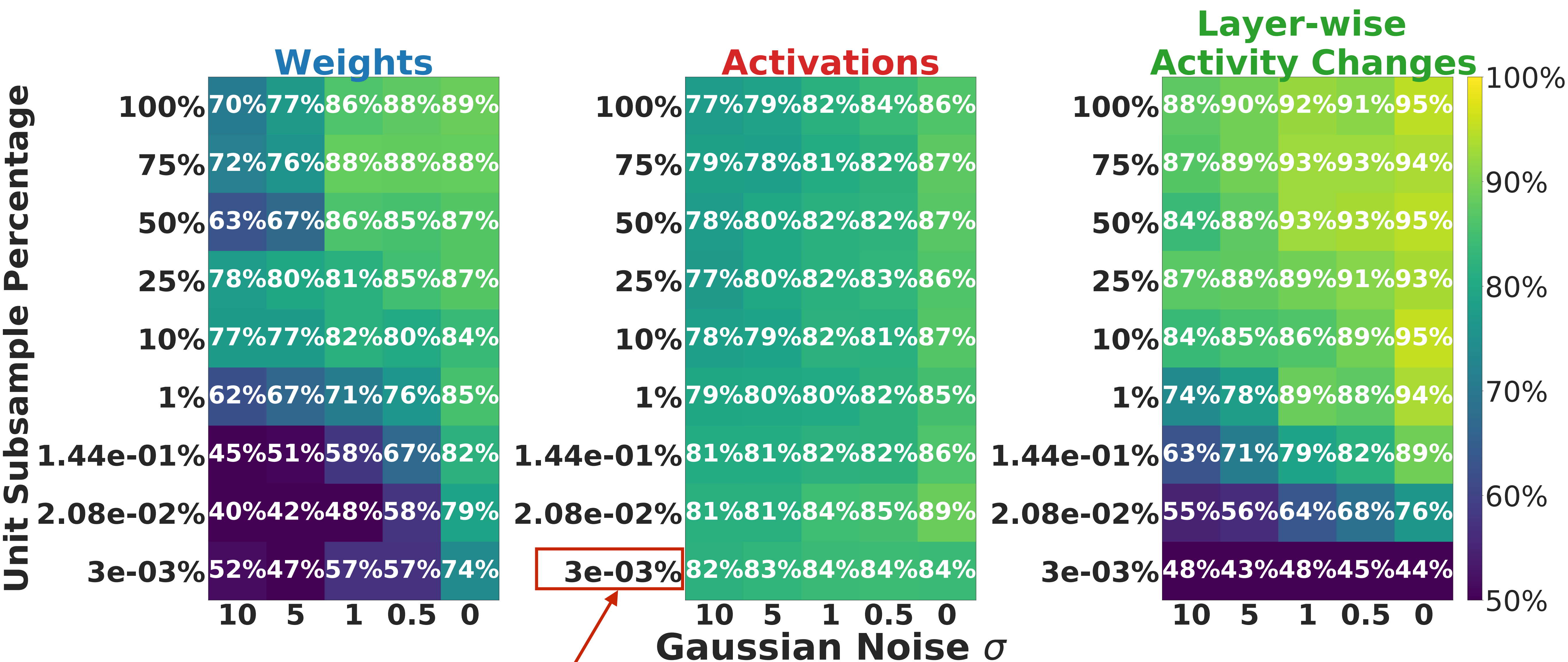


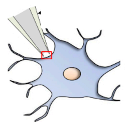
Weights are *not* robust to measurement noise and unit undersampling



Within typical imaging range of several hundred to several thousand synapses

Activations are the most robust to measurement noise and unit undersampling





Within typical electrophysiological range of several hundred units

Conclusions

Conclusions

We can identify learning rules *only* on the basis of aggregate statistics of observable measures: weights, activations, or layer-wise activity changes

Conclusions

We can identify learning rules *only* on the basis of aggregate statistics of observable measures: weights, activations, or layer-wise activity changes

Simple (non-linear) classifier such as Random Forest generalizes across certain held-out classes of input types ("animal" and "training curricula" holdouts)

Conclusions

We can identify learning rules *only* on the basis of aggregate statistics of observable measures: weights, activations, or layer-wise activity changes

Simple (non-linear) classifier such as Random Forest generalizes across certain held-out classes of input types ("animal" and "training curricula" holdouts)

Measurements temporally spaced further apart are more robust to *trajectory undersampling*, for each observable measure

Conclusions

We can identify learning rules *only* on the basis of aggregate statistics of observable measures: weights, activations, or layer-wise activity changes

Simple (non-linear) classifier such as Random Forest generalizes across certain held-out classes of input types ("animal" and "training curricula" holdouts)

Measurements temporally spaced further apart are more robust to *trajectory undersampling*, for each observable measure

Aggregate statistics across units of the network's activation patterns are most robust to *unit undersampling* and *measurement noise*

Conclusions

We can identify learning rules *only* on the basis of aggregate statistics of observable measures: weights, activations, or layer-wise activity changes

Simple (non-linear) classifier such as Random Forest generalizes across certain held-out classes of input types ("animal" and "training curricula" holdouts)

Measurements temporally spaced further apart are more robust to *trajectory undersampling*, for each observable measure

Aggregate statistics across units of the network's activation patterns are most robust to *unit undersampling* and *measurement noise*

Hypothesis: *in vivo* electrophysiological recordings of <u>post-synaptic activities</u> from a neural circuit on the order of several hundred units, frequently measured at wider intervals during the course of learning, may provide a good basis on which to identify learning rules

Conclusions

We can identify learning rules *only* on the basis of aggregate statistics of observable measures: weights, activations, or layer-wise activity changes

Simple (non-linear) classifier such as Random Forest generalizes across certain held-out classes of input types ("animal" and "training curricula" holdouts)

Measurements temporally spaced further apart are more robust to *trajectory undersampling*, for each observable measure

Aggregate statistics across units of the network's activation patterns are most robust to *unit undersampling* and *measurement noise*

Hypothesis: *in vivo* electrophysiological recordings of post-synaptic activities from a neural circuit on the order of several hundred units, frequently measured at wider intervals during the course of learning, may provide a good basis on which to identify learning rules

Code & Dataset: https://github.com/neuroailab/lr-identify