Reproducible Science of Deep Learning:

The Pruning Case Study

Michela Paganini, Facebook AI Research
@WonderMicky

UCI Symposium on Reproducibility in Machine Learning
07/22/2020

FACEBOOK

Science is a verb

My recent work

Bespoke vs. Prêt-à-Porter Lottery Tickets: Exploiting Mask Similarity for Trainable Sub-Network Finding — **under review**

One Ticket to Win Them All: Generalizing Lottery Ticket Initializations Across Datasets and Optimizers — **NeurIPS 2019 poster**

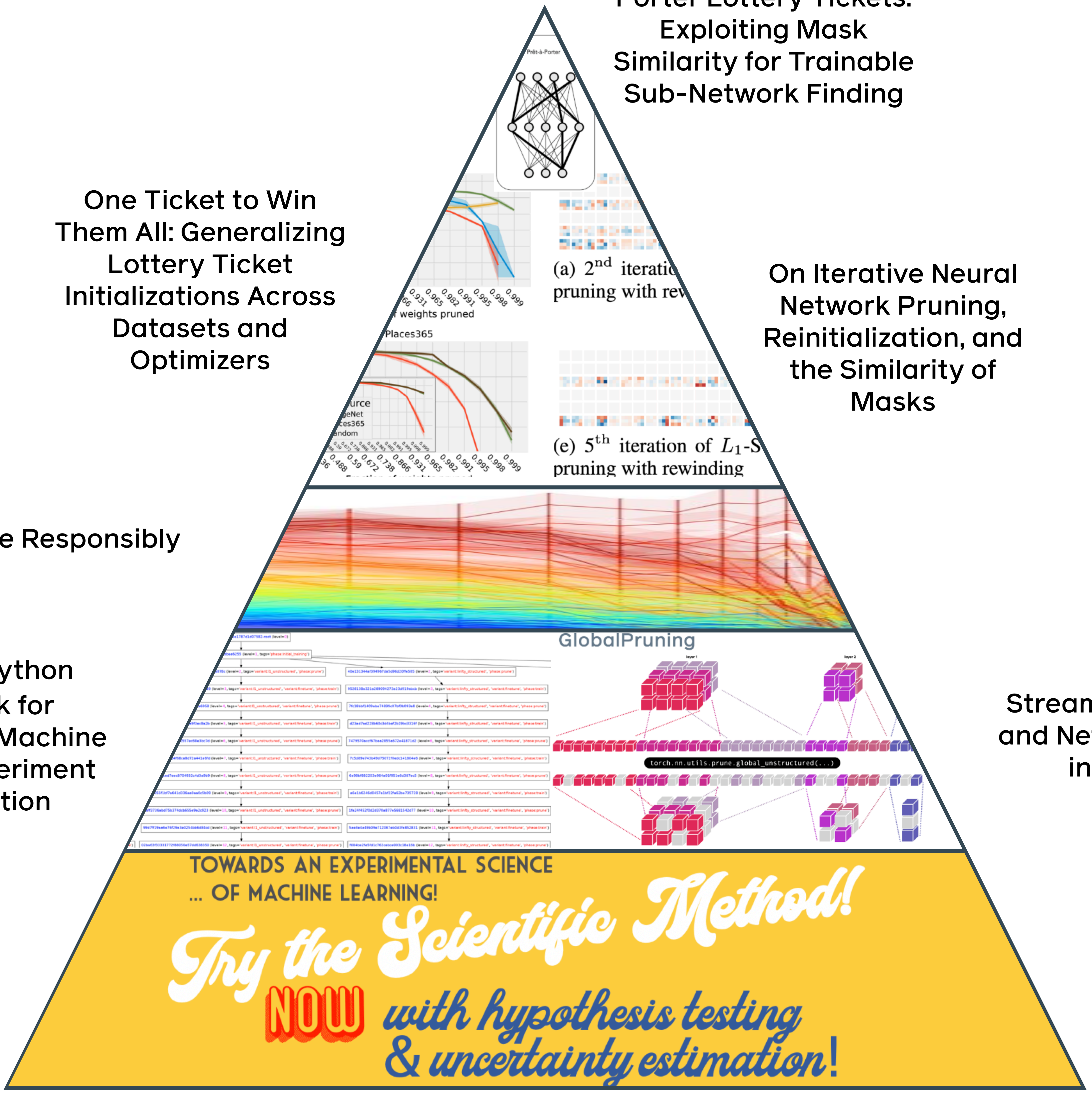
On Iterative Neural Network Pruning, Reinitialization, and the Similarity of Masks — **ICLR 2020 workshop poster**

Prune Responsibly — **under review**

`dagger`: A Python Framework for Reproducible Machine Learning Experiment Orchestration — **under review**

Streamlining Tensor and Network Pruning in PyTorch — **ICLR 2020 workshop oral**

The Scientific Method in the Science of Machine Learning — **ICLR 2019 workshop oral**



Agenda

1. Why reproducibility?
2. The scientific method in the science of ML
3. Pruning for hypothesis testing
4. Dagger
5. Pruning in PyTorch
6. Measuring the disproportionate harm of pruning

1. Why reproducibility?

Trust

Integrity

Confidence

Progress

Reliability

1. Why reproducibility?

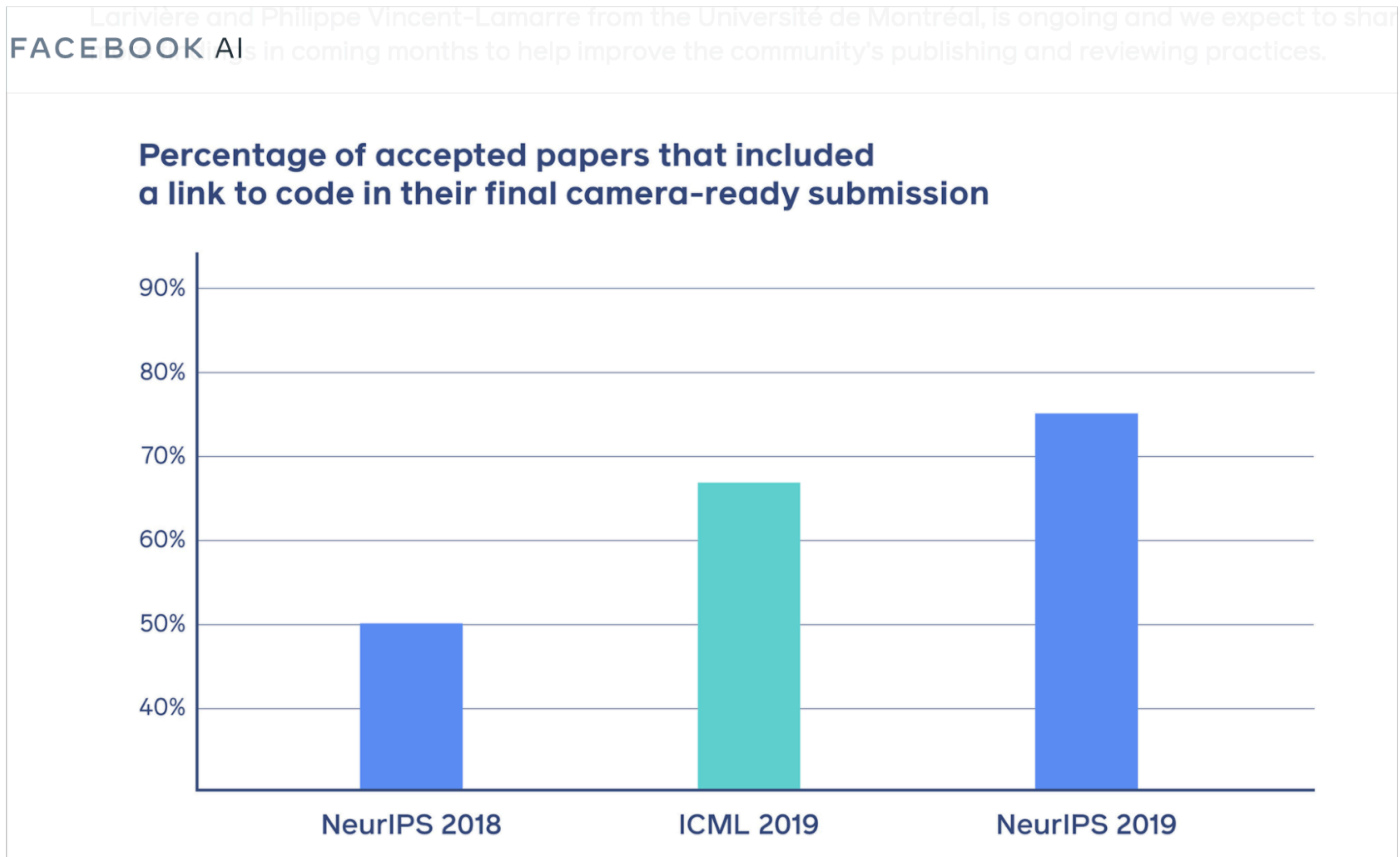
nature > news q&a > article

NEWS Q&A · 19 DECEMBER 2019

This AI researcher is trying to ward off a reproducibility crisis

Joelle Pineau is leading an effort to encourage artificial-intelligence researchers to open up their code.

Elizabeth Gibney



https://ai.facebook.com/blog/how-the-ai-community-can-get-serious-about-reproducibility/

Papers With Code

sotabench

June 10, 2019

Towards Reproducible Research with PyTorch Hub

PYTORCH
HUB

PUBLISHING MODELS

PyTorch Hub supports publishing pre-trained models (model definitions and pre-trained weights) to a GitHub repository by adding a simple `hubconf.py` file.

Discovery

Find the best models related to your research/application!

Reproducibility

Spend minutes instead of days on baselines

Responsibility

Publish solid papers with reproducible results.

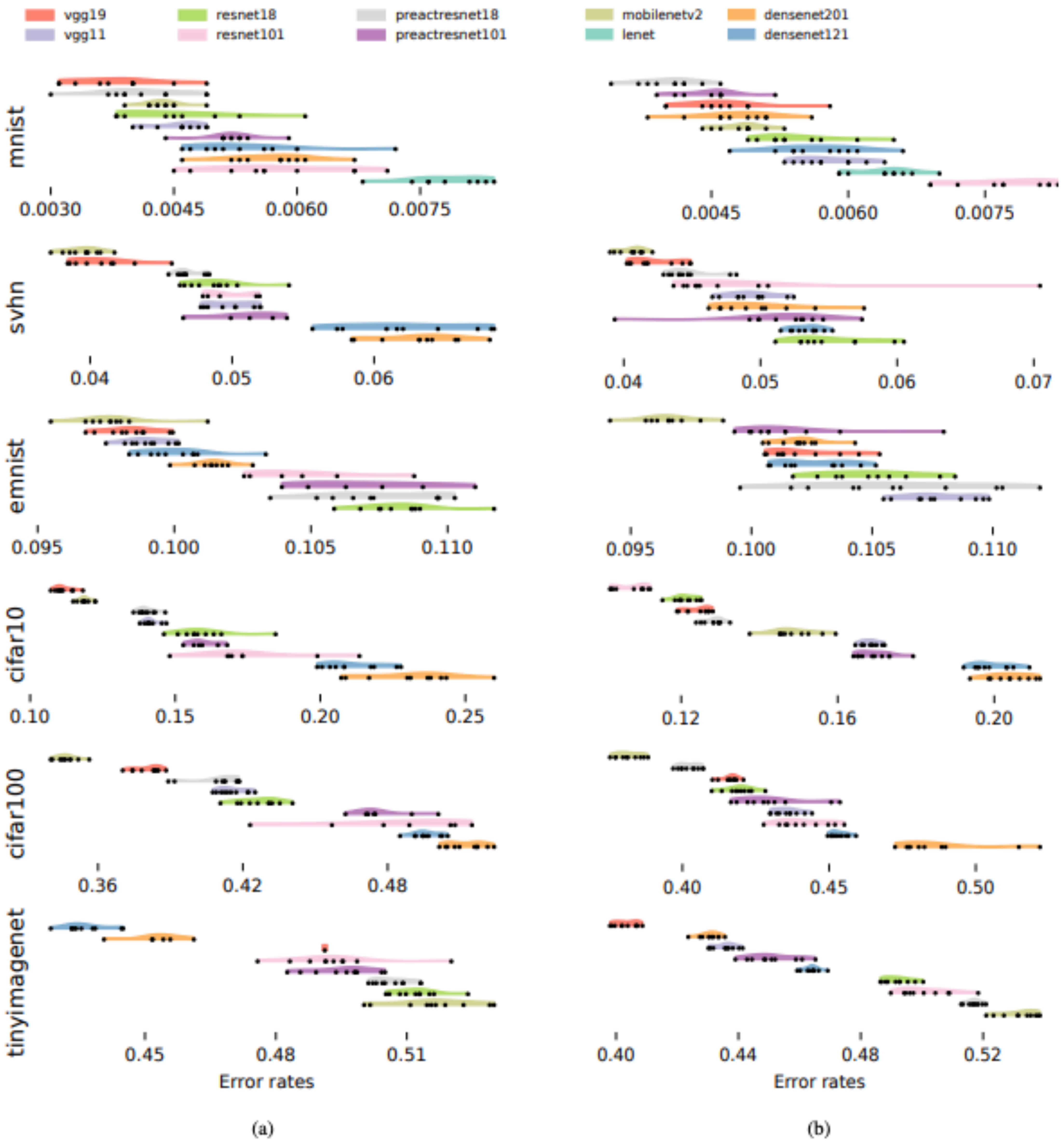
Slide adapted from Ailing Zhang

1. Why reproducibility?

Unreproducible Research is Reproducible

Xavier Bouthillier, César Laurent, Pascal Vincent ; Proceedings of the 36th International Conference on Machine Learning, PMLR 97:725-734, 2019.

- unreproducible findings can be built upon reproducible methods
- not just a matter of deterministic reproducibility of methods and single numerical results
- necessity of ensuring the reproducibility of empirical findings and conclusions by properly accounting for essential sources of variations
- more energy should be devoted to proper empirical research in our community
- promote the use of more rigorous and diversified methodologies



Measurements are affected by sources of variations

What can we learn from the other sciences?

The Scientific Method in the Science of Machine Learning, arXiv:1904.10922

The one and only way to make objective statements?

A social contract among scientists to harmonize workflows and compare findings?

Transparency

Falsifiability

Reproducibility

Intellectual Honesty

Key Steps for Experimental Scientific Research.

hypothesis formulation

statement of expectations

experiment design

statistical analysis

uncertainty estimation

Key Steps for Experimental Scientific Research.

hypothesis formulation	"The null hypothesis is ..., the alternative hypothesis is ..."
statement of expectations	"If the hypothesis is right, then I should expect to observe ..."
experiment design	"I design this experiment to be sensitive to..."
statistical analysis	"Do I observe the expected effect? Is it stronger or weaker than expected?"
uncertainty estimation	"Do I have enough observations and did I account for systematic biases?"

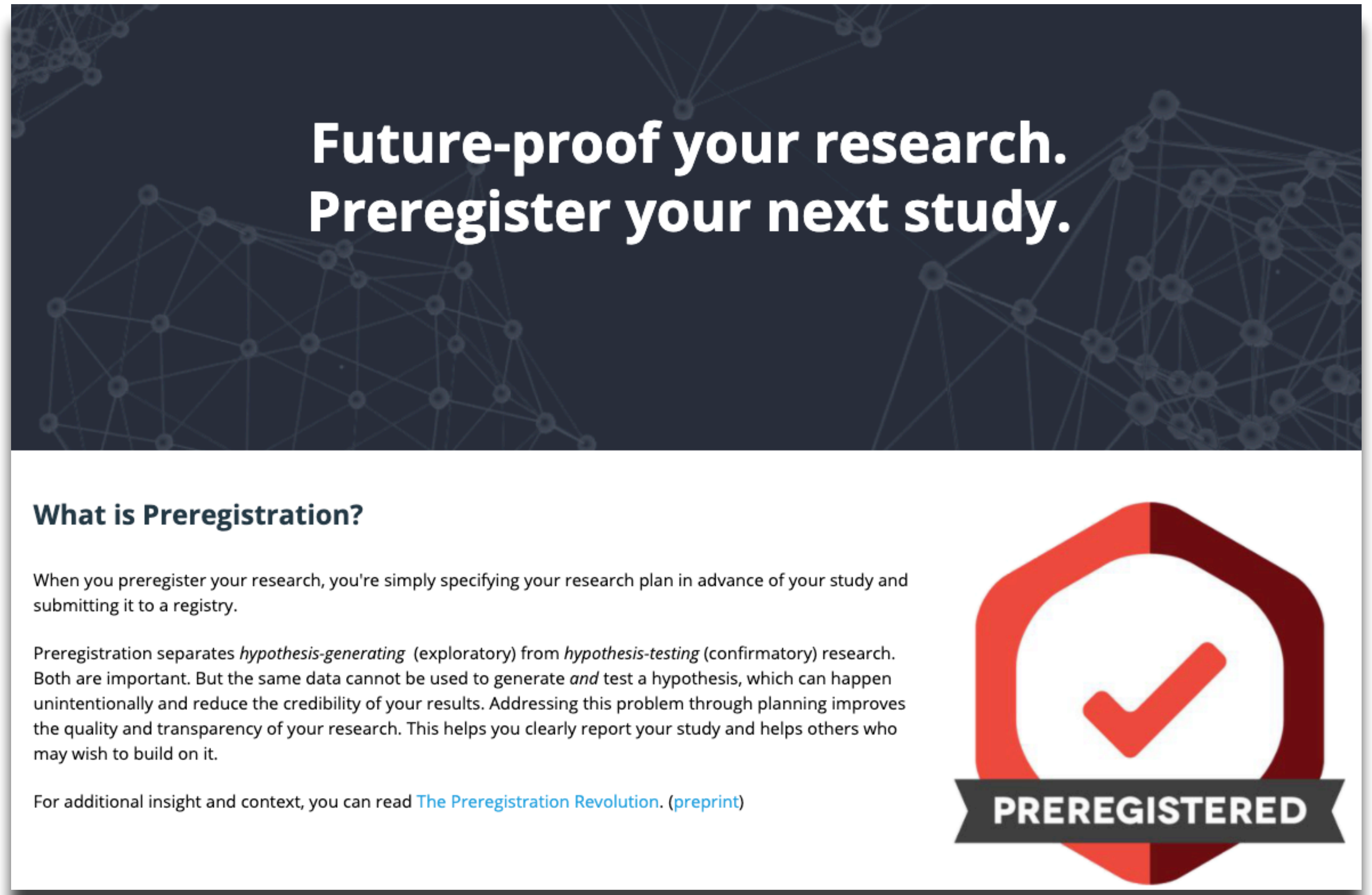
"The first step towards a scientific formulation of ML then demands a more dramatic shift in priorities from drawing and recording single instances of experimental results to collecting enough data to gain an understanding of population statistics."

"it is plausible that a significant percentage of published work claiming state-of-the-art performance actually has no statistical sensitivity to measure their improvement over competing methods."

Blind analysis and pre-registration

Don't judge a paper by its *p*-value.

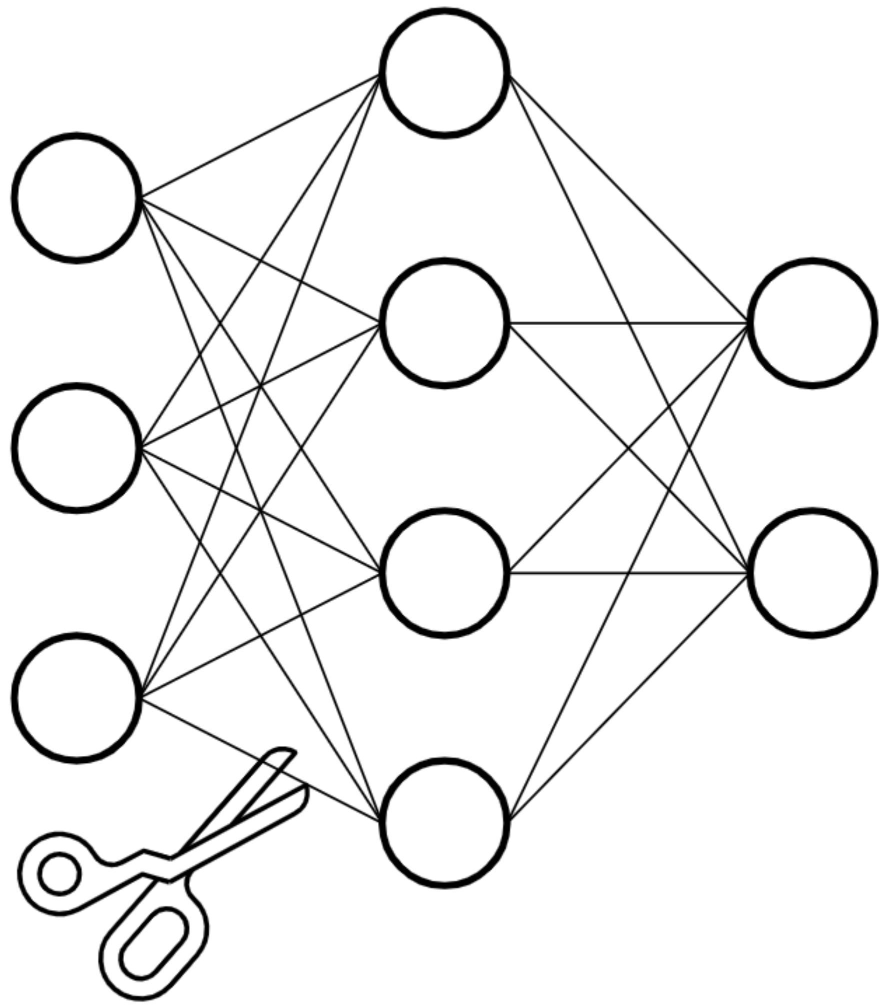
cos.io/prereg/



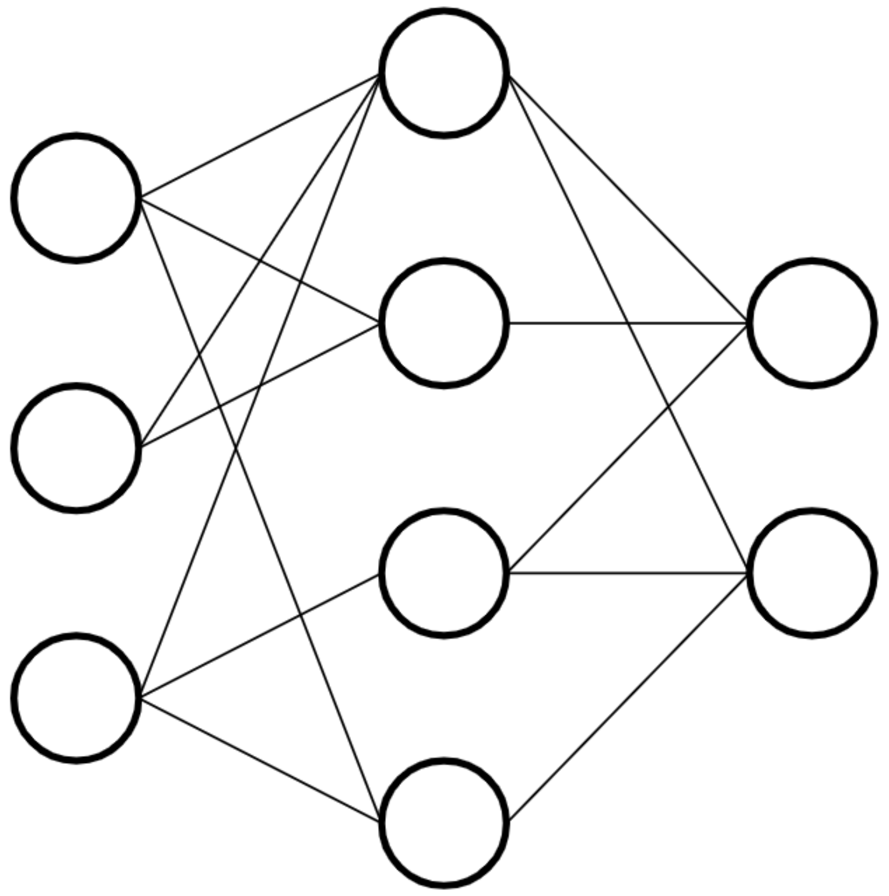
preregister.science

The Pruning Case Study

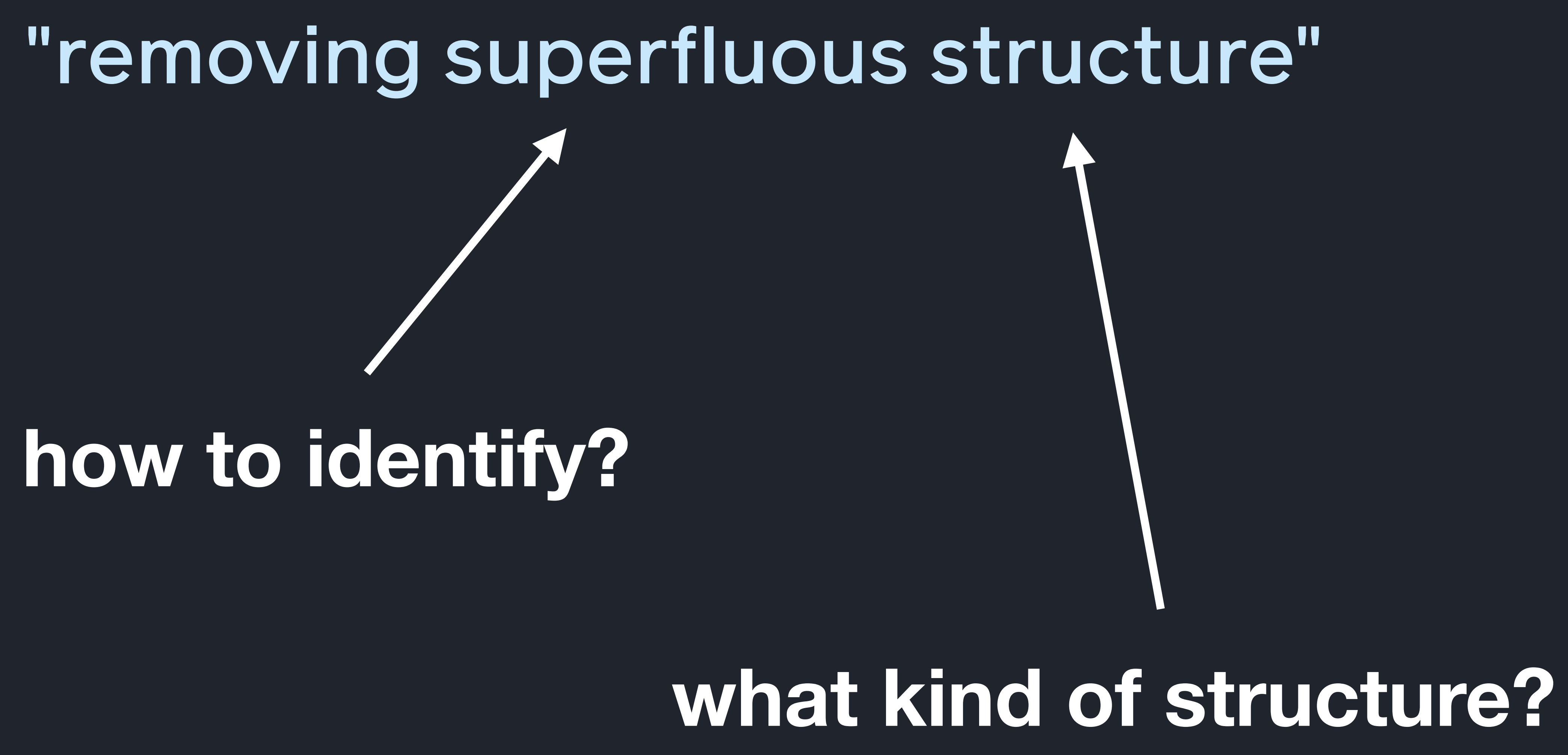
Pruning



Before pruning



After pruning



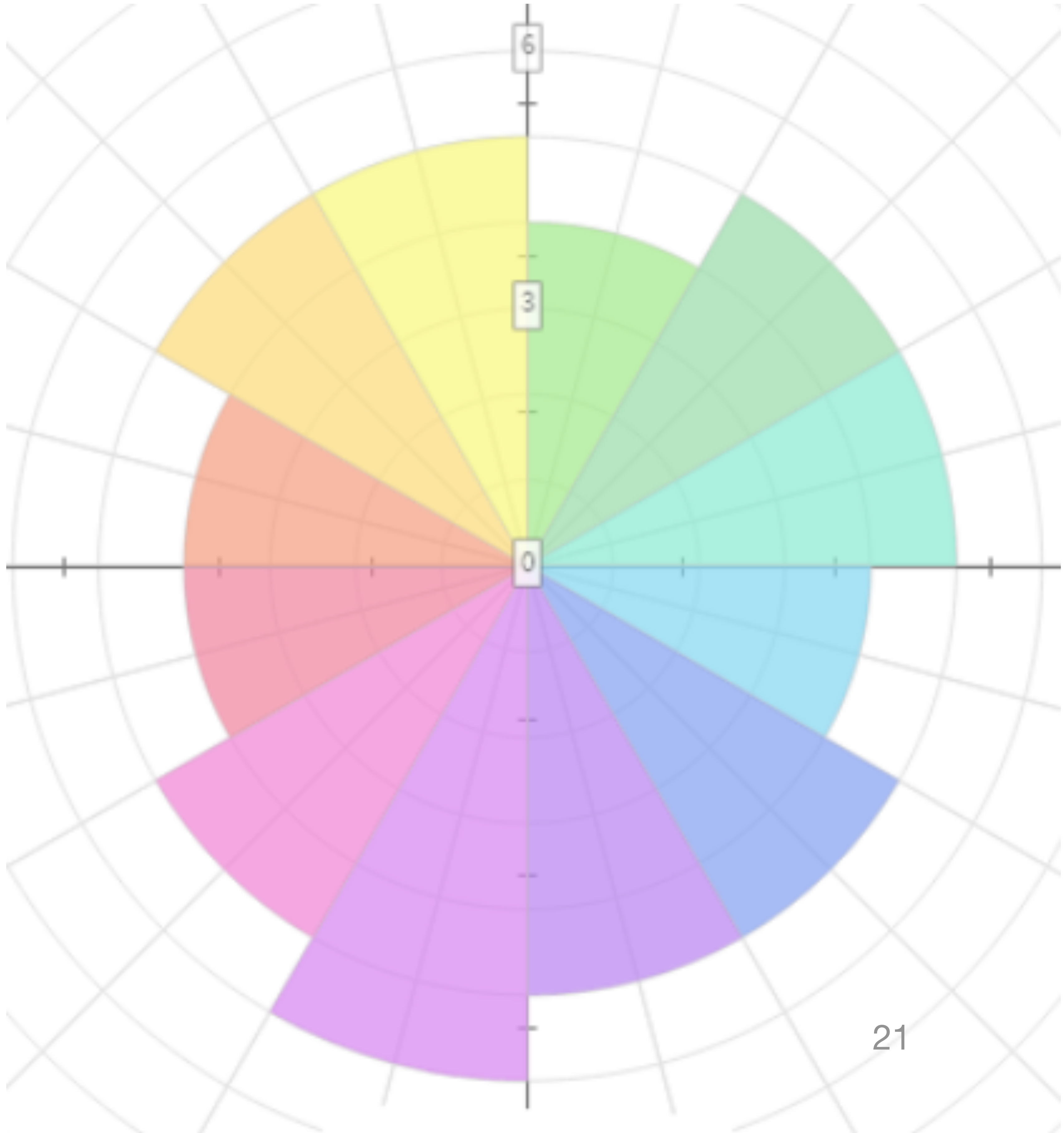
The state of pruning

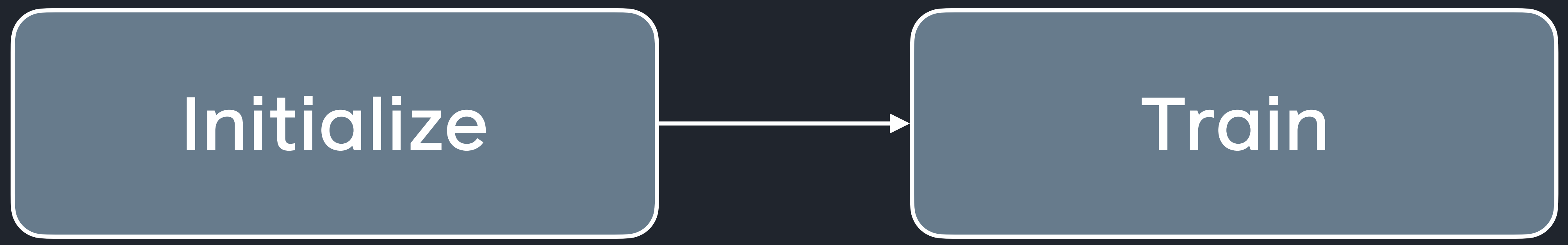
Pruning should remove unnecessary redundancy and unused capacity

Can be executed *before*, *during*, and *after* training

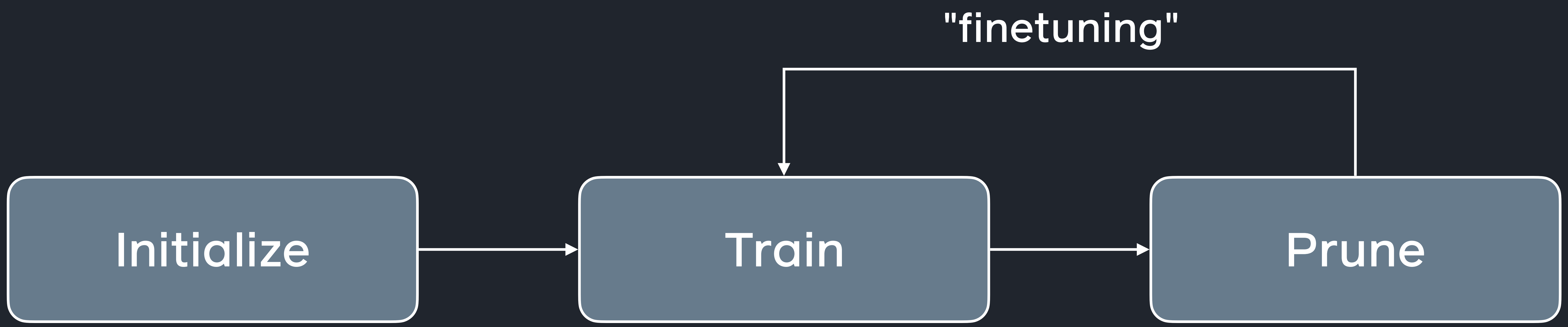
Pruning methods differ across many dimensions:

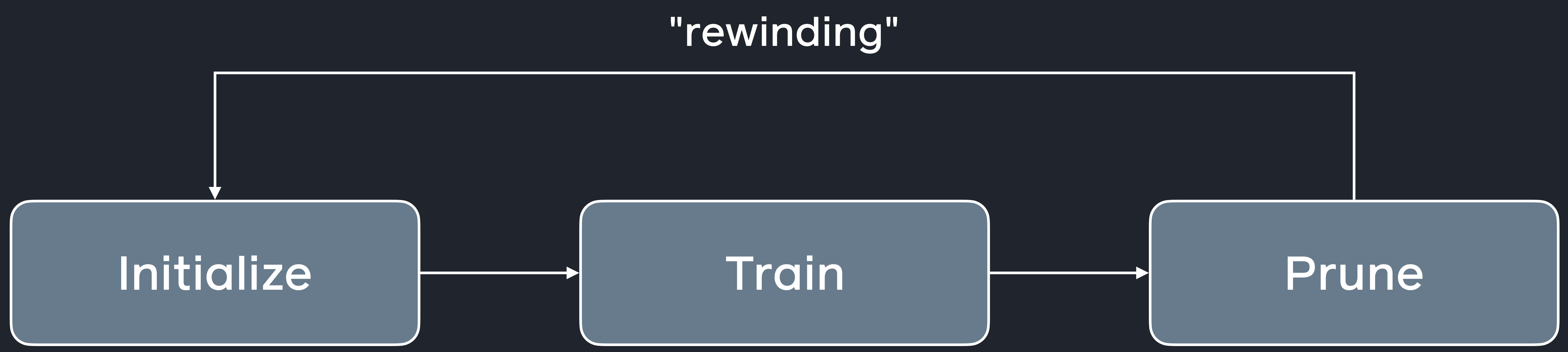
- based on weight magnitude, activations, gradients, Hessian, interpretability measures, credit assignment, random, etc.
- Layer-wise vs global, unstructured vs structured, etc.
- Rule-based, bayesian, differentiable, soft approaches, etc.
- One-shot vs iterative pruning
- Followed by: finetuning, reinitialization, rewinding

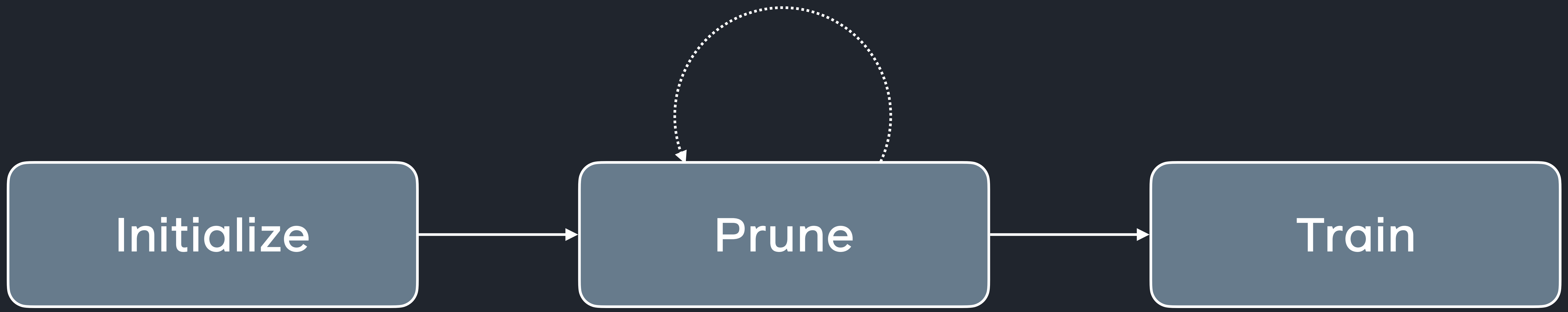



Initialize
Train


Initialize
Train
Prune


"finetuning"
Initialize
Train
Prune


"rewinding"
Initialize
Train
Prune


Initialize
Prune
Train

```
035bb6eaec977e3de60e0853d63cb011.bz2
035bb6eaec977e3de60e0853d63cb011.slim.json
049d06b3f12d4485d2c9555eb4f27b24.bz2
049d06b3f12d4485d2c9555eb4f27b24.slim.json
06cfe230e2b9565ab2d59e864eab7045.bz2
06cfe230e2b9565ab2d59e864eab7045.slim.json
08291cdd4c0f12de74166b02d935eabe.bz2
08291cdd4c0f12de74166b02d935eabe.slim.json
086ae9abf2282a3aa26e3f769f1ef509.bz2
086ae9abf2282a3aa26e3f769f1ef509.slim.json
0a74d268096a65c2a1779414aadba3df.bz2
0a74d268096a65c2a1779414aadba3df.slim.json
20d92fb2765e4edbd8a8f75adbb58696.bz2
20d92fb2765e4edbd8a8f75adbb58696.slim
29ef5352b5d4b460bdaeb15c4c0f6c8c.bz2
29ef5352b5d4b460bdaeb15c4c0f6c8c.slim.json
30195510da3f77e3a12efcfc1d41b63a.bz2
30195510da3f77e3a12efcfc1d41b63a.slim.json
32537b8ea57608d5bf18d82b5b62d078.bz2
32537b8ea57608d5bf18d82b5b62d078.slim.json
344ae86e68def0bfb43bd3cfb8fbec13.bz2
344ae86e68def0bfb43bd3cfb8fbec13.slim.json
373ebdfa64468e1b3a23293e4d04680f.bz2
373ebdfa64468e1b3a23293e4d04680f.slim.json
386885205bcba9494dbebf580afee8b2.bz2
386885205bcba9494dbebf580afee8b2.slim.json
38ef0199861b8a1a5d05556d7fed1e52.bz2
38ef0199861b8a1a5d05556d7fed1e52.slim.json
3b4e8afc175fd03729a830fd136c5684.bz2
3b4e8afc175fd03729a830fd136c5684.slim.json
3cb027ad63585f8eda23639226302f63.bz2
3cb027ad63585f8eda23639226302f63.slim.json
42fa6cdfb64714c43c3419381a8fb821.bz2
42fa6cdfb64714c43c3419381a8fb821.slim.json
4d2060ad068f9c8beb61c909ccedf1fb.bz2
4d2060ad068f9c8beb61c909ccedf1fb.slim.json
50394e6dc4e478c3a9db0ab660937987.bz2
50394e6dc4e478c3a9db0ab660937987.slim.json
59f8a498b7ead01b98767ba912c8bb2f.bz2
59f8a498b7ead01b98767ba912c8bb2f.slim.json
5a3771b0392256a2d5625d71ce663e2b.bz2
5a3771b0392256a2d5625d71ce663e2b.slim.json
5a3c5fbd365bcf3935d7fd89e2145582.bz2
5a3c5fbd365bcf3935d7fd89e2145582.slim.json
61b999ebb35c25c34033a2987c92be76.bz2
61b999ebb35c25c34033a2987c92be76.slim.json
…440a835c5a513…ed4bbc8e.bz2
…835c5…4bbc8e…
6…334…cd28cac62…1c2…2
65…3345f4e5cd28cac62…1c2c.slim.json
…d079674dc4f327048…07a8a.bz2
…40d079674dc4f327…9a07a8a.slim.js…
69dc21cf4ea444271…11f1029917.bz2
69dc21cf4ea44427…a511f1029917.slim…
6b109f29bc047f37…7bed1291e72f.bz2
6b109f29bc047f37b1417bed1291e72f.slim.json
6b6aba4fececaba…dd06e11ca61e.bz2
6b6aba4fececaba…dd06e11ca61e.slim…
76ff392ef7b2748ba3784472f4884bf0.bz2
76ff392ef7b2748ba3784472f4884bf0.slim.json
77bbee568a85d535408c0a3c8d945ed5.bz2
77bbee568a85d535408c0a3c8d945ed5.slim.json
79b8a91c6ed9a9c8b5ea6dea047d906a.bz2
79b8a91c6ed9a9c8b5ea6dea047d906a.slim.json
7a2d512e2e28c2beafc75214bf81a2cb.bz2
7a2d512e2e28c2beafc75214bf81a2cb.slim.json
8110e7ed753ac476cb080ec43cdd8291.bz2
8110e7ed753ac476cb080ec43cdd8291.slim.json
9fd11c13ff649e45082663585e705ce9.bz2
9fd11c13ff649e45082663585e705ce9.slim.json
b420996715b37da7749ce42badbada10.bz2
b420996715b37da7749ce42badbada10.slim.json
ba803121bff54079d3fadaa6ff434c53.bz2
ba803121bff54079d3fadaa6ff434c53.slim.json
bc2887dc4df8d4baab6952ec8ef26323.bz2
bc2887dc4df8d4baab6952ec8ef26323.slim.json
bcb8a78325b9290b49671ece38ad2885.bz2
bcb8a78325b9290b49671ece38ad2885.slim.json
c53d29419b6cfe7ff959b2b63ea782e1.bz2
c53d29419b6cfe7ff959b2b63ea782e1.slim.json
…80fd73ad23cf5790d5170b0087af8fd.bz2
…d73ad23cf5790d5170b0087af8fd.slim.json
c…f414a5f1e8e11322c4643c8b9c4.bz2
c…f414a5f1e8e11322c4643c8b9c4.slim.json
…cb7aaca40af5a918fd9cae80110.bz2
…98cb7aaca40af5a918fd9cae80110.slim.json
cb79aa59933f51df2f577b1bb7fbd429.bz2
cb79aa59933f51df2f577b1bb7fbd429.slim.json
cde95b2e20d54cf0666228ac61985182.bz2
cde95b2e20d54cf0666228ac61985182.slim.json
d5e50946d8d6fb49339b1b3abeca403a.bz2
d5e50946d8d6fb49339b1b3abeca403a.slim.json
dd41bc211d272cc6674ed6662cf962d2.bz2
dd41bc211d272cc6674ed6662cf962d2.slim.json
dd4dc4556488cfaffa3e044d1300d1c5.bz2
dd4dc4556488cfaffa3e044d1300d1c5.slim.json
e090eac0b64924c34acd90adb7300d64.bz2
e090eac0b64924c34acd90adb7300d64.slim.json
e6ac41a441f1a1edaaeaf0f583e921d4.bz2
e6ac41a441f1a1edaaeaf0f583e921d4.slim.json
ec50fc8bd8523301dc7a1cc38e3133e1.bz2
ec50fc8bd8523301dc7a1cc38e3133e1.slim.json
```

???

Reproducible Experiment Orchestration

`facebookresearch/dagger`

`dagger` is a minimal framework for describing trees of network-mutating actions suited to the needs of researchers, allowing fast experimentation as well as maintenance of clear provenance in experiment evolution .

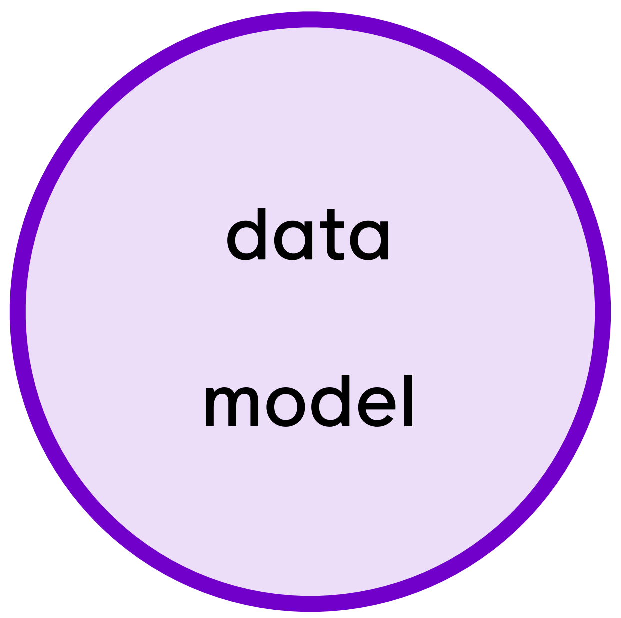
Goals:
- Allow researchers to abstract away *fundamental scientific contributions* from *experiment-tracking boilerplate code*
- *Bookkeeping:* track model state provenance

Concepts:
- **Experiment**: the graph
- **Experiment State**: a node
- **Recipe**: an edge

`dagger` is a minimal framework for describing trees of network-mutating actions suited to the needs of researchers, allowing fast experimentation as well as maintenance of clear provenance in experiment evolution .

Goals:
- Allow researchers to abstract away *fundamental scientific contributions* from *experiment-tracking boilerplate code*
- *Bookkeeping:* track model state provenance

Concepts:
- **Experiment**: the graph
- **Experiment State**: a node
- **Recipe**: an edge

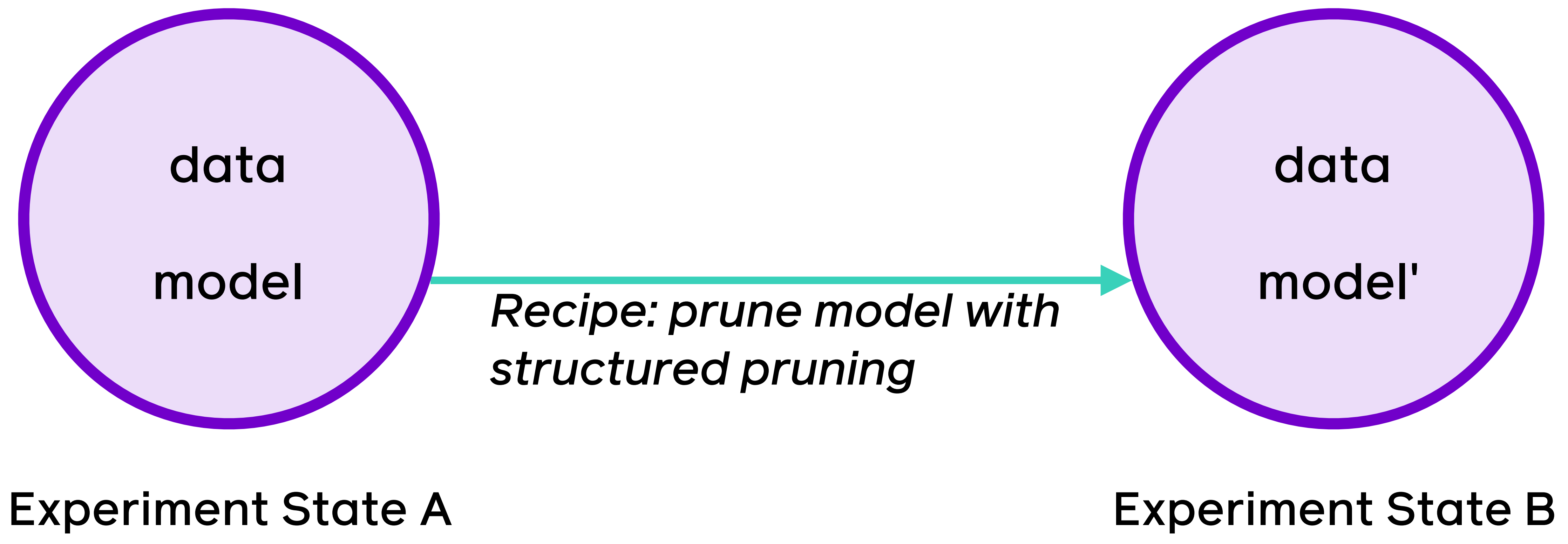


Experiment State A

`dagger` is a minimal framework for describing trees of network-mutating actions suited to the needs of researchers, allowing fast experimentation as well as maintenance of clear provenance in experiment evolution .

Goals:
- Allow researchers to abstract away *fundamental scientific contributions* from *experiment-tracking boilerplate code*
- *Bookkeeping:* track model state provenance

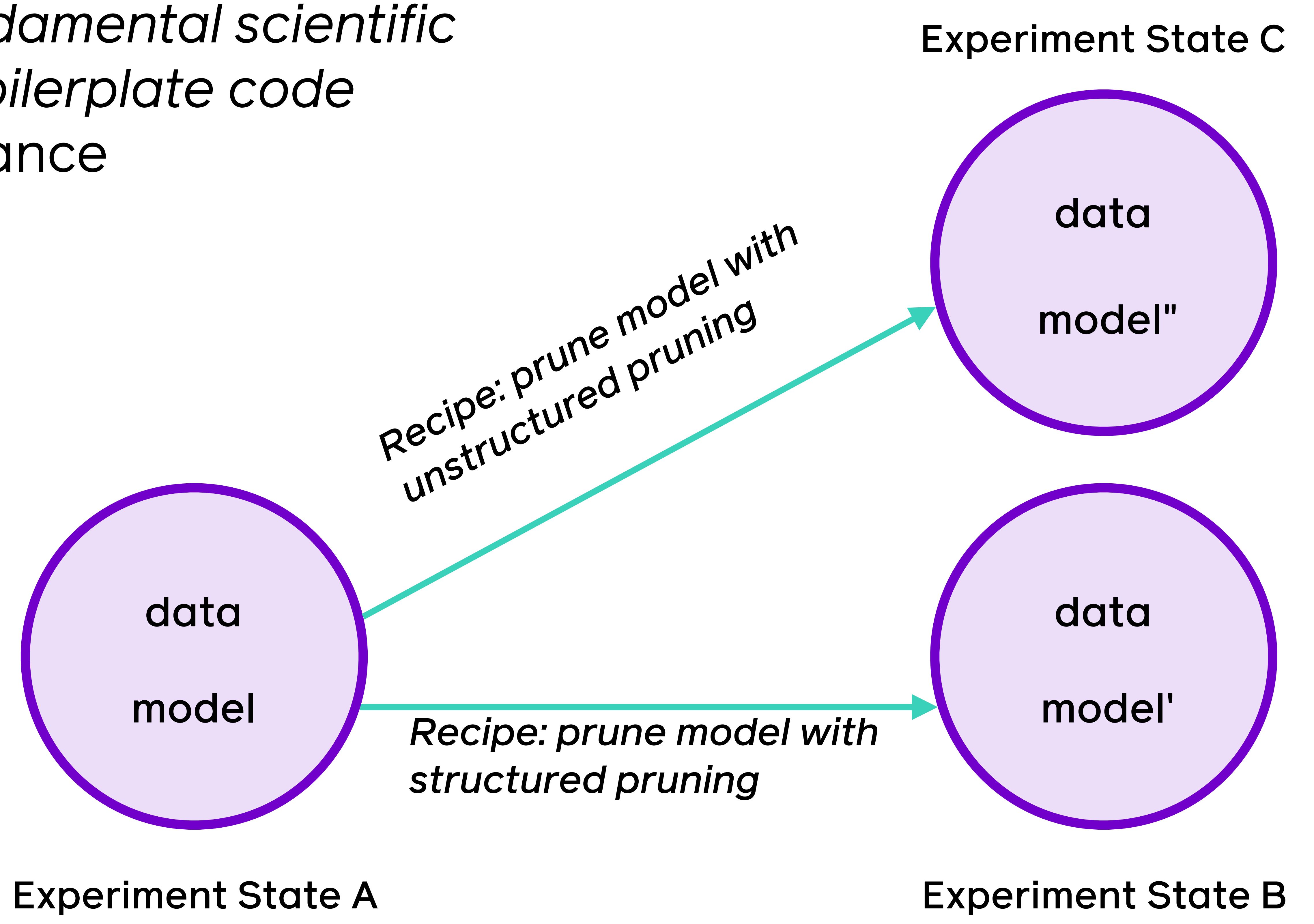
Concepts:
- **Experiment**: the graph
- **Experiment State**: a node
- **Recipe**: an edge



`dagger` is a minimal framework for describing trees of network-mutating actions suited to the needs of researchers, allowing fast experimentation as well as maintenance of clear provenance in experiment evolution .

Goals:
- Allow researchers to abstract away *fundamental scientific contributions* from *experiment-tracking boilerplate code*
- *Bookkeeping:* track model state provenance

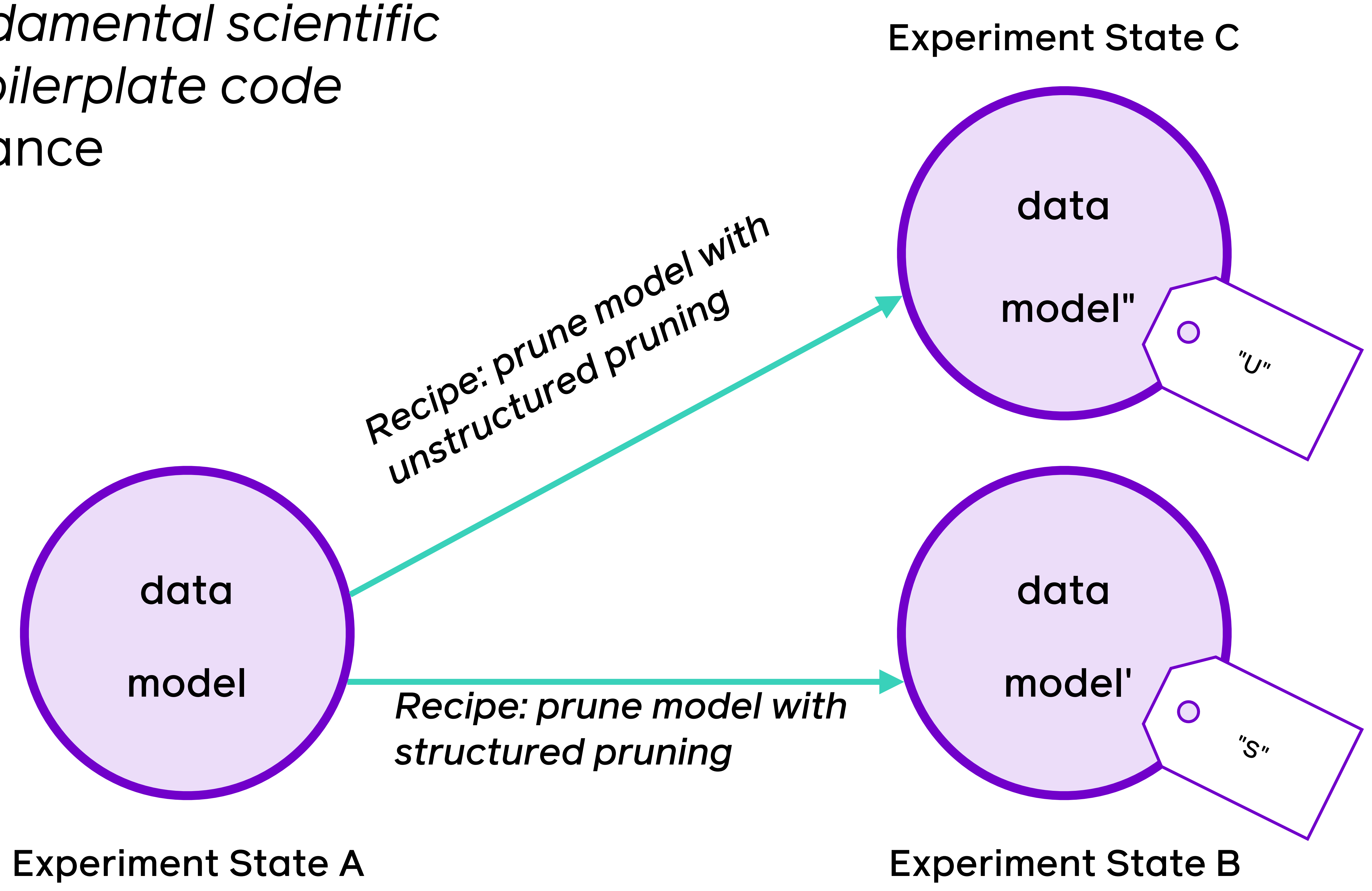
Concepts:
- **Experiment**: the graph
- **Experiment State**: a node
- **Recipe**: an edge



`dagger` is a minimal framework for describing trees of network-mutating actions suited to the needs of researchers, allowing fast experimentation as well as maintenance of clear provenance in experiment evolution .

Goals:
- Allow researchers to abstract away *fundamental scientific contributions* from *experiment-tracking boilerplate code*
- *Bookkeeping:* track model state provenance

Concepts:
- **Experiment**: the graph
- **Experiment State**: a node
- **Recipe**: an edge



Experiment Loop

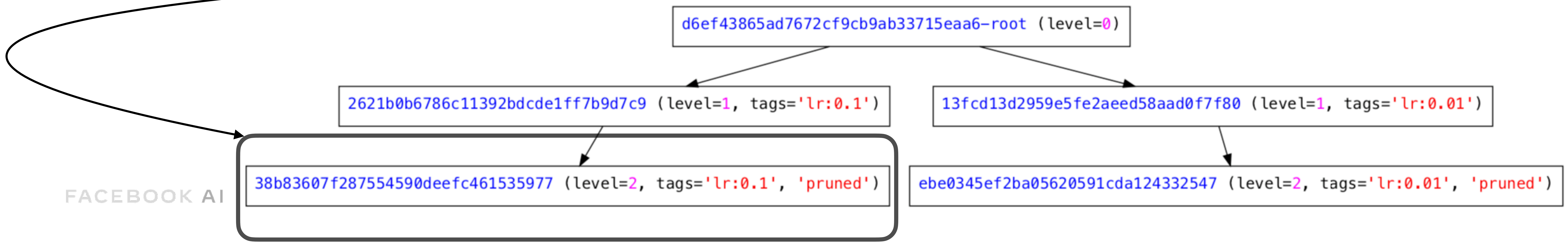
```python
exp = dg.Experiment("/path/to/experiment/folder", state_class=State)
root_state = exp.spawn_new_tree(dataset_name="cifar-10", model_name="vgg-11")

for lr in [0.01, 0.1]:
    train = TrainRecipe(nb_epochs=100, lr=lr)
    prune = PruneRecipe(pruning_technique="lowest_magnitude", pruning_fraction=0.2)
    s = root_state
    with exp.tag(f"lr:{lr}"):
        s = train(s)
        eval_fn(s)
        with exp.tag("pruned"):
            s = prune(s)
exp.run()
```

Custom Definitions

```python
import dagger as dg
from yourlib import get_data, get_model, train_model, prune_model, eval_model

class State(dg.ExperimentState):

    PROPERTIES = ["dataset_name", "model_name"]
    NONHASHED_ATTRIBUTES = ["train_data", "eval_data", "model"]

    def initialize_state(self, **kwargs):
        self.train_data, self.eval_data = get_data(self.dataset_name)
        self.model = get_model(self.model_name)

class TrainRecipe(dg.Recipe):

    PROPERTIES = ["nb_epochs", "lr"]

    def run(self, state):
        train_model(state.model, state.train_data, self.nb_epochs, self.lr)
        return state

class PruneRecipe(dg.Recipe):

    PROPERTIES = ["pruning_technique", "pruning_fraction"]

    def run(self, state):
        prune_model(state.model, self.pruning_technique, self.pruning_fraction)
        return state

@dg.function
def eval_fn(state):
    eval_acc = eval_model(state.model, state.eval_data)
    print(f"Experiment: {state.tags}, Accuracy: {eval_acc}")
```

Experiment Analysis

```python
>>> exp = Experiment.restore("/path/to/experiment/folder", slim=True)
>>> exp.graph.draw()  # Draws the graph in Figure 1
>>> s = exp.graph.nodes.filter("pruned") & exp.graph.nodes.filter("lr:0.1")
>>> s[0].restore()
```



Centralized Pruning in PyTorch

`torch.nn.utils.prune`

torch.nn.utils.prune

Different tensor pruning techniques enabled under a unified framework

BasePruningMethod

CLASS `torch.nn.utils.prune.BasePruningMethod` [SOURCE]

Abstract base class for creation of new pruning techniques.

- CLASSMETHOD `apply(module, name, *args, **kwargs)` [SOURCE]
- `apply_mask(module)` [SOURCE]
- ABSTRACT `compute_mask(t, default_mask)` [SOURCE]
- `prune(t, default_mask=None)` [SOURCE]
- `remove(module)` [SOURCE]

New pruning technique?

Just subclass `BasePruningMethod` and implement `compute_mask`!

PruningContainer

CLASS `torch.nn.utils.prune.PruningContainer(*args)` [SOURCE]

Container holding a sequence of pruning methods for iterative pruning. Keeps track of the order in which pruning methods are applied and handles combining successive pruning calls.

Identity

CLASS `torch.nn.utils.prune.Identity` [SOURCE]

Utility pruning method that does not prune any units but generates the pruning parametrization with a mask of ones.

RandomUnstructured

CLASS `torch.nn.utils.prune.RandomUnstructured(amount)` [SOURCE]

Prune (currently unpruned) units in a tensor at random.

L1Unstructured

CLASS `torch.nn.utils.prune.L1Unstructured(amount)` [SOURCE]

Prune (currently unpruned) units in a tensor by zeroing out the ones with the lowest L1-norm.

RandomStructured

CLASS `torch.nn.utils.prune.RandomStructured(amount, dim=-1)` [SOURCE]

Prune entire (currently unpruned) channels in a tensor at random.

LnStructured

CLASS `torch.nn.utils.prune.LnStructured(amount, n, dim=-1)` [SOURCE]

Prune entire (currently unpruned) channels in a tensor based on their Ln-norm.

CustomFromMask

CLASS `torch.nn.utils.prune.CustomFromMask(mask)` [SOURCE]

torch.nn.utils.prune

Fetches the mask and the original, unpruned tensor to compute the pruned tensor during the forward pass → op is accounted for in the backward pass, too

BasePruningMethod

CLASS `torch.nn.utils.prune.BasePruningMethod` [SOURCE]

Abstract base class for creation of new pruning techniques.

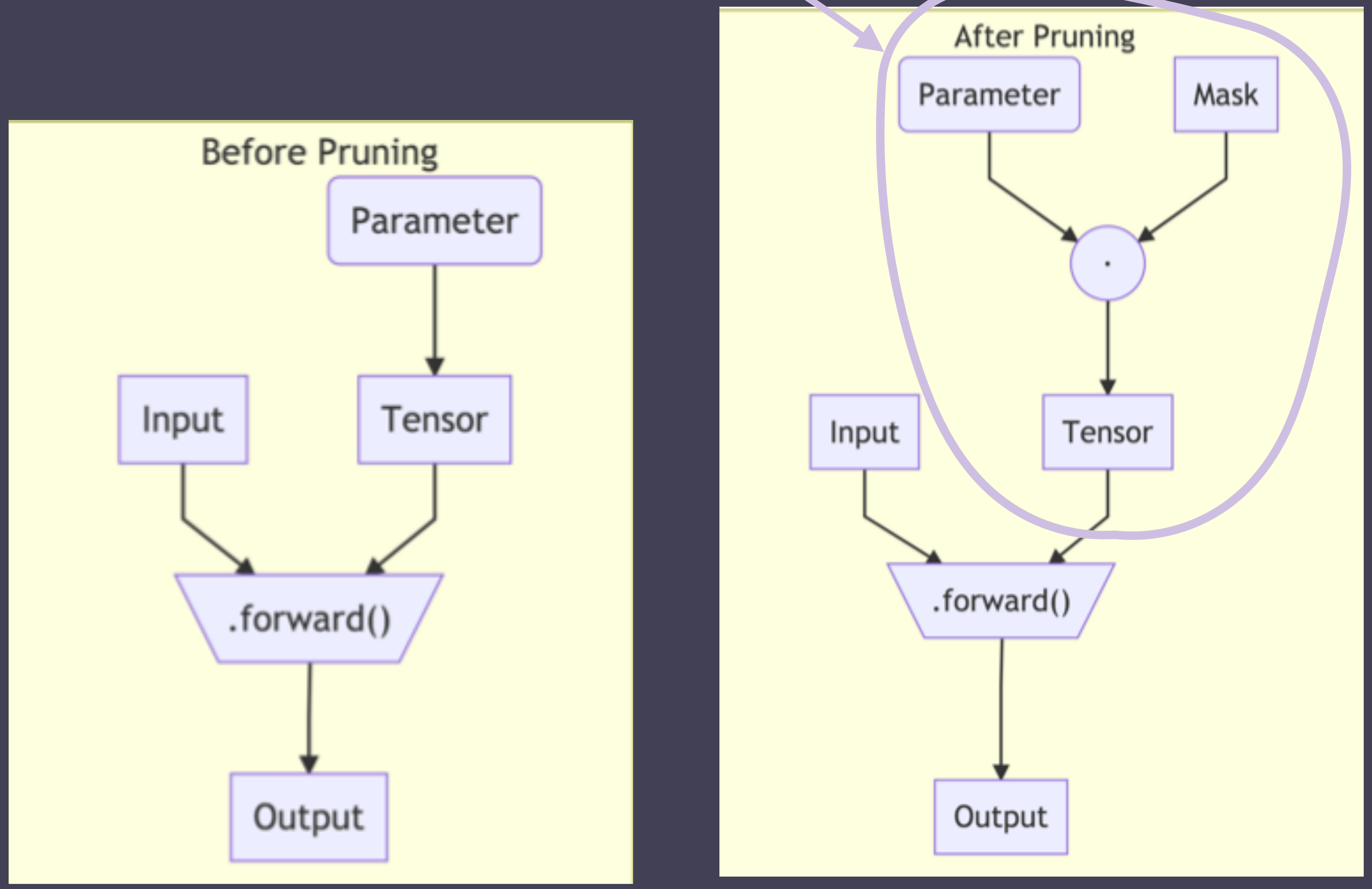
CLASSMETHOD `apply`(*module, name, *args, **kwargs*) [SOURCE]

`apply_mask`(*module*) [SOURCE]

ABSTRACT `compute_mask`(*t, default_mask*) [SOURCE]

`prune`(*t, default_mask=None*) [SOURCE]

`remove`(*module*) [SOURCE]



`torch.nn.utils.prune`

BasePruningMethod

CLASS `torch.nn.utils.prune.BasePruningMethod` [SOURCE]

Abstract base class for creation of new pruning techniques.

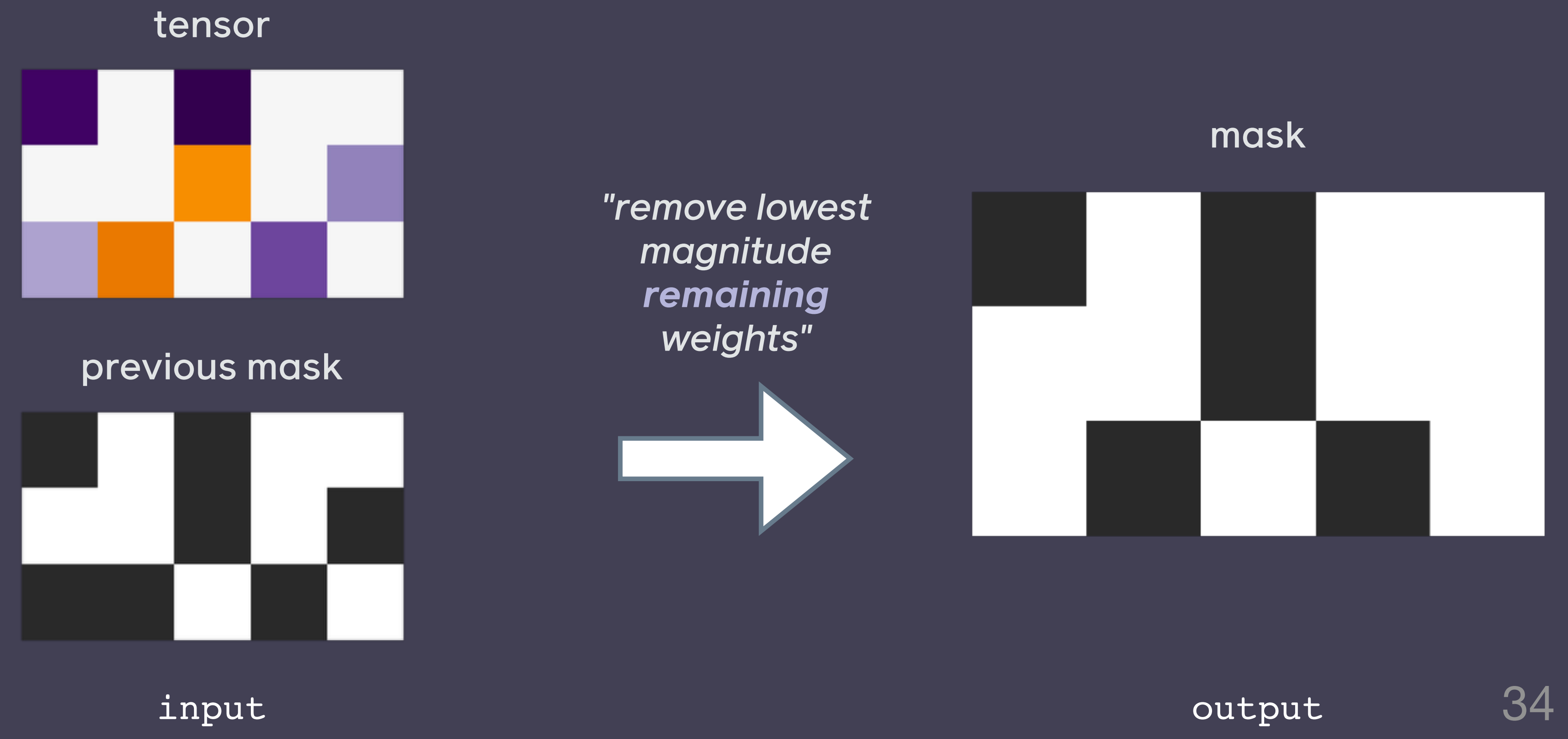
- CLASSMETHOD `apply(module, name, *args, **kwargs)` [SOURCE]
- `apply_mask(module)` [SOURCE]
- ABSTRACT `compute_mask(t, default_mask)` [SOURCE]
- `prune(t, default_mask=None)` [SOURCE]
- `remove(module)` [SOURCE]

defines the interface → concrete subclasses must implement the logic

For example, in `prune.L1Unstructured`:

implements the logic that defines which portions of the tensors will be zeroed out while accounting for previously pruned entries

(through a `prune.PruningContainer`) it handles the case in which the tensor had previously been pruned by computing the valid entries in the tensor that can still be pruned and then applying the new pruning technique exclusively on those entries



Easy to use

```
model = LeNet()  # unpruned model

# L_2 structured pruning will remove 50% of channels across axis 0
prune.ln_structured(
    module=model.conv1,
    name="weight",
    amount=0.5,
    n=2,
    dim=0
)
```

Iterative pruning made easy

`prune.PruningContainer` handles the combination of successive masks for you

```
for _ in range(10):
    # Remove 2 connections per iteration
    prune.l1_unstructured(module=model.fc1, name="bias", amount=2)
```

Global pruning made easy

```
parameters_to_prune = (
    (model.conv1, "weight"),
    (model.conv2, "weight"),
    (model.fc1, "weight"),
)

prune.global_unstructured(
    parameters_to_prune,
    pruning_method=prune.L1Unstructured,
    amount=0.2,
)
```

Easy to extend

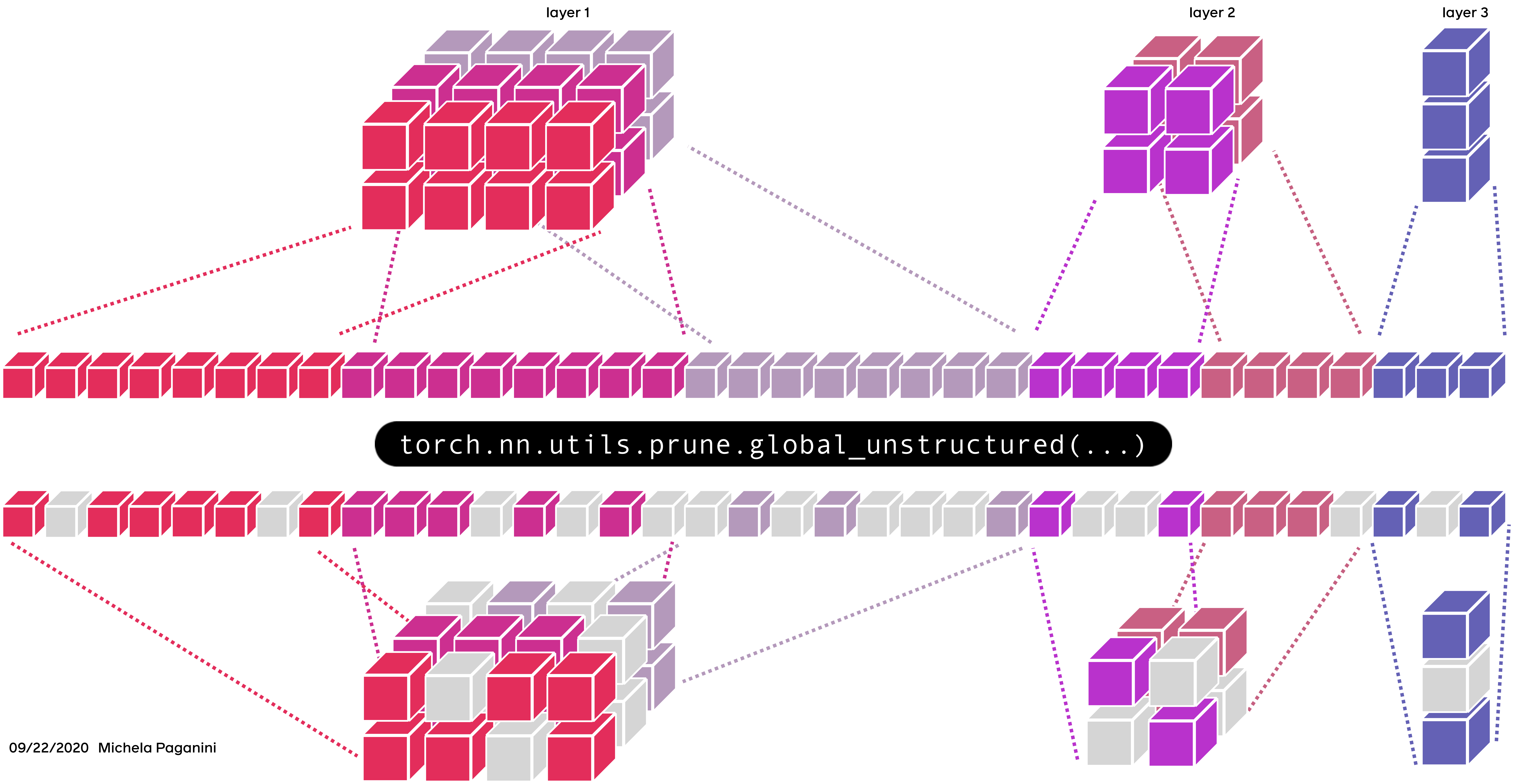
```
class FooBarPruningMethod(prune.BasePruningMethod):
    """Prune every other entry in a tensor
    """
    PRUNING_TYPE = 'unstructured'

    def compute_mask(self, t, default_mask):
        mask = default_mask.clone()
        mask.view(-1)[::2] = 0
        return mask
```

supports 3 PRUNING_TYPEs: `'global'`, `'structured'`, and `'unstructured'` (to determine how to combine masks if pruning is applied iteratively)

instructions on how to compute the mask for the given tensor according to the logic of your pruning technique

```
def foobar_unstructured(module, name):
    FooBarPruningMethod.apply(module, name)
    return module
```

GlobalPruning



torch.nn.utils.prune

BasePruningMethod

CLASS `torch.nn.utils.prune.BasePruningMethod` [SOURCE]

Abstract base class for creation of new pruning techniques.

CLASSMETHOD `apply(module, name, *args, **kwargs)` [SOURCE]

`apply_mask(module)` [SOURCE]

ABSTRACT `compute_mask(t, default_mask)` [SOURCE]

`prune(t, default_mask=None)` [SOURCE]

`remove(module)` [SOURCE]

`torch.nn.utils.prune` is designed to act on a `torch.nn.Module`

provides an interface for acting directly on a tensor

```
tensor = torch.randn([3, 5])
p = torch.nn.utils.prune.LnStructured(amount=1, dim=1, n=2)
masked_tensor = p.prune(tensor)
```

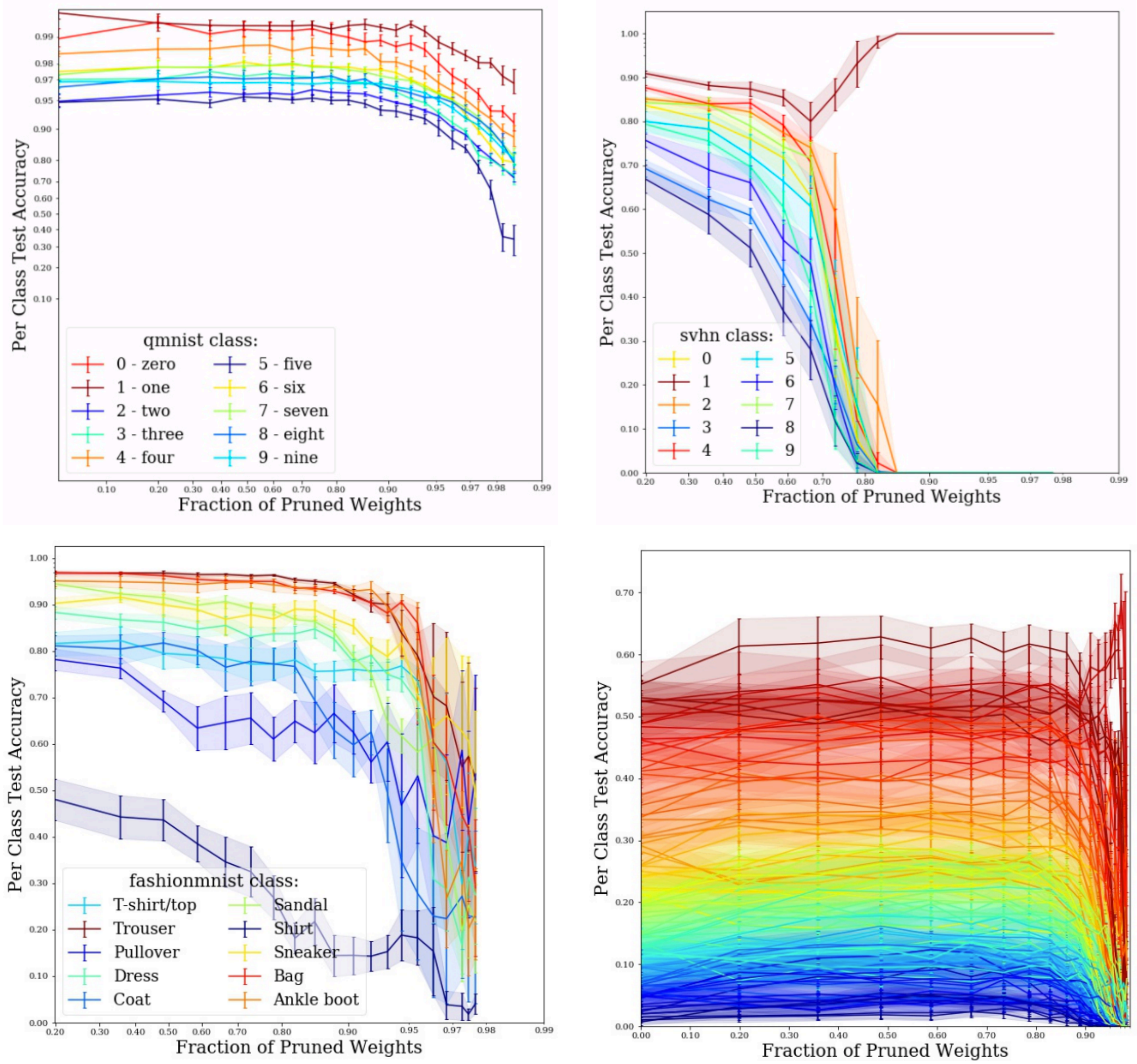
`torch.nn.utils.prune`

Tutorials

Prune Responsibly

arXiv:2009.09936

Test hypotheses that class complexity, difficulty, and representation matter in determining the accuracy after pruning



Prune and measure class accuracy for over 1M classes across over 100k models

Fit a linear model for class accuracy as a function of:

- **unpruned model class accuracy**
- **class entropy**
- **class representation**
- sparsity
- dataset
- model
- pruning technique
- weight treatment after pruning

Reject hypothesis that coefficients = 0

Closing Remarks

Papers with Code @paperswithcode · Sep 4

🔥 Introducing... the ML Reproducibility Challenge 2020! The 4th annual edition now expands to cover papers from 7 major ML conferences: NeurIPS, EMNLP, ACL, ICML, ICLR, CVPR and ECCV. Find more here at Papers With Code:

Papers with Code - ML Reproducibility Challenge 2020
The ML Reproducibility Challenge 2020 covering paper published in seven major ML conferences: NeurIPS, ACL, EMNLP, ICLR, ICML, CV...
paperswithcode.com

Michela Paganini ✊🏾🙋🏽‍♀️🖤 @WonderMicky · Sep 2

Upset at the latest #NeurIPS2020 reviews? Interested in an alternative publication model for machine learning research? Looking for hypothesis-based science in AI? 👩‍🔬

Check out and submit to the preregister.science workshop at NeurIPS. #SCIENCE!
Deadline: Oct 7. More info 👇

NeurIPS2020 pre-registration workshop
Testing whether pre-registration can help fix our peer review system.
preregister.science

Thanks!

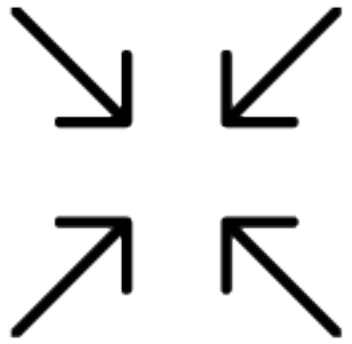
Questions? Contact me: michela@fb.com

WonderMicky

Learn from other Sciences.

Theoretical Science

Experimental Science



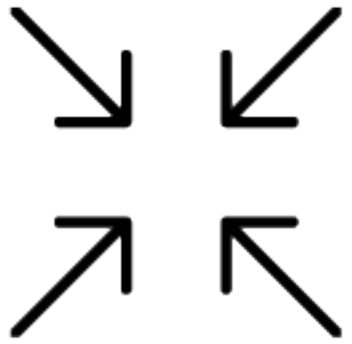
Engineering

0101
1001
0110

Learn from other Sciences.

Theoretical Science

Experimental Science

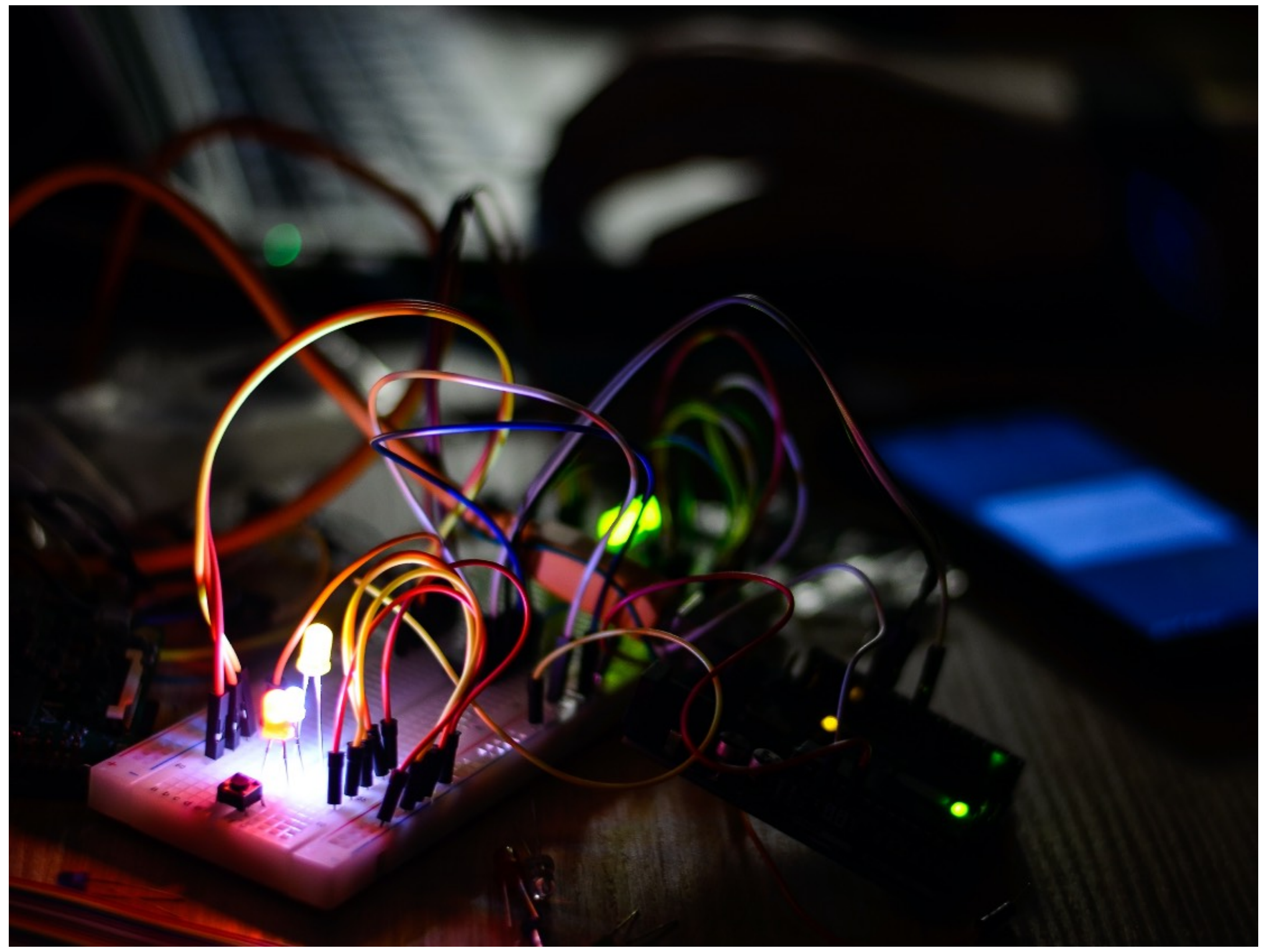


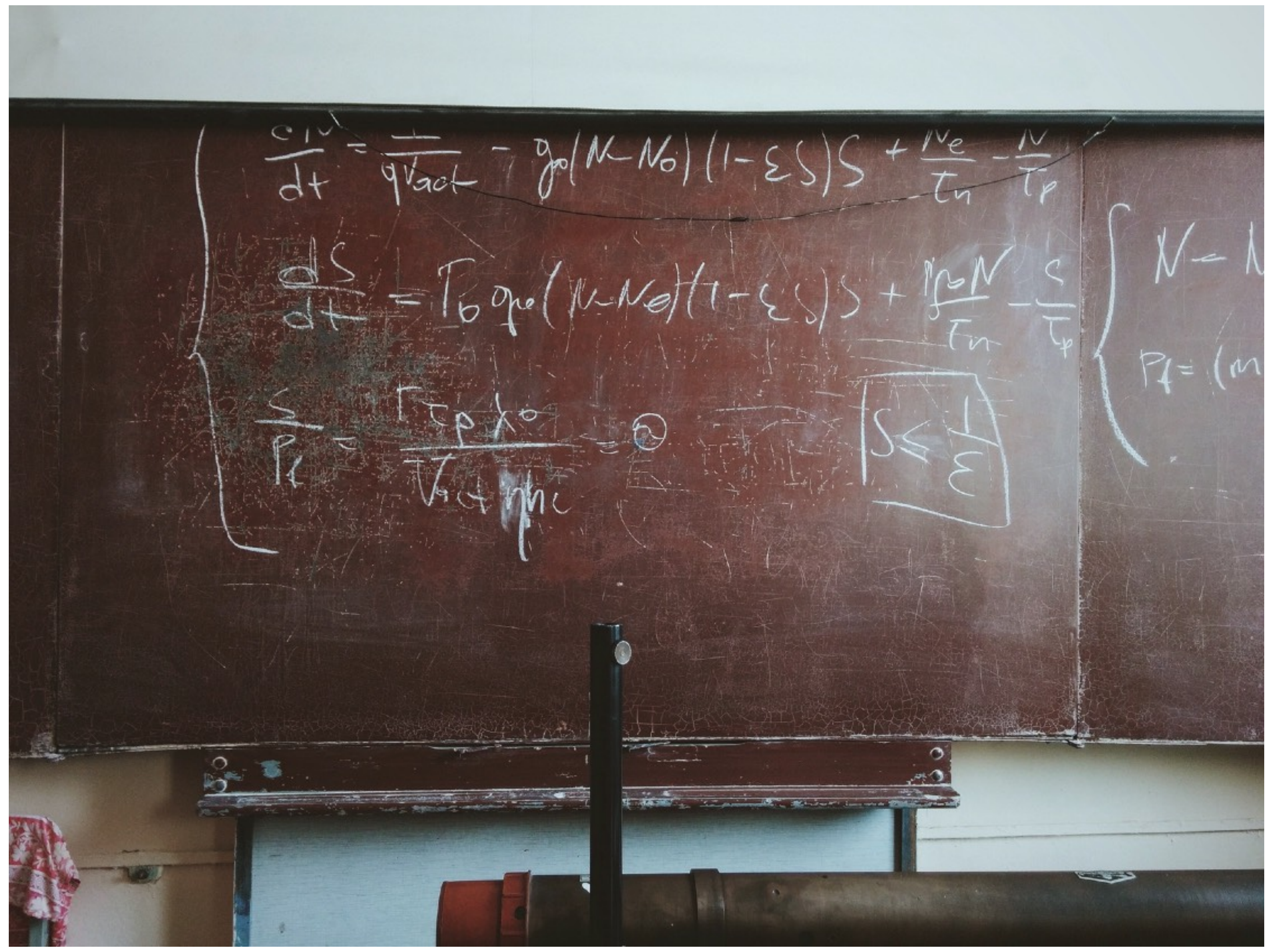
Engineering

0101
1001
0110

Neural Networks can be thought of as **physical objects** obeying laws of dynamics.

CAN STUDY THE INTERACTIONS OF THEIR FUNDAMENTAL COMPONENTS USING EXPERIMENTAL PROCEDURES.





"Grounding ML research in statistically sound hypothesis testing with careful control of nuisance parameters may encourage the publication of advances that stand the test of time."

Code Submission Policies

- ICML 2019 and NeurIPS 2019 rolled out explicit code-submission policies
- Many concerns regarding Dataset confidentiality, Proprietary software, Computation infrastructure, Replication of mistakes…
- NeurIPS 2019/2020 code submission policy leaves significant time and flexibility - "*expects code only for accepted papers, and only by the camera-ready deadline*"