# Meta-Learning Unsupervised Update Rules

Paper by Luke Metz, Niru Maheswaranathan, Brian Cheung, Jascha Sohl-Dickstein

# Outline

Motivation

Problem Breakdown

Method Overview

Meta-Learning Setup

Inner Loop

Outer Loop

Experimental Results

Critiques

# Motivation



Unsupervised learning enables representation learning on mountains on unlabeled data for downstream tasks

# Motivation



Unsupervised learning enables representation learning on mountains of unlabeled data for downstream tasks.

Unsupervised Learning Rules

- **VAE:** Severe overfitting to training space.
- **GANs:** Great for images, weak on discrete data (ex. text).
- **Both:** Learning rule not unsupervised (ex. surrogate loss).

# Motivation



Unsupervised learning enables representation learning on mountains of unlabeled data for downstream tasks

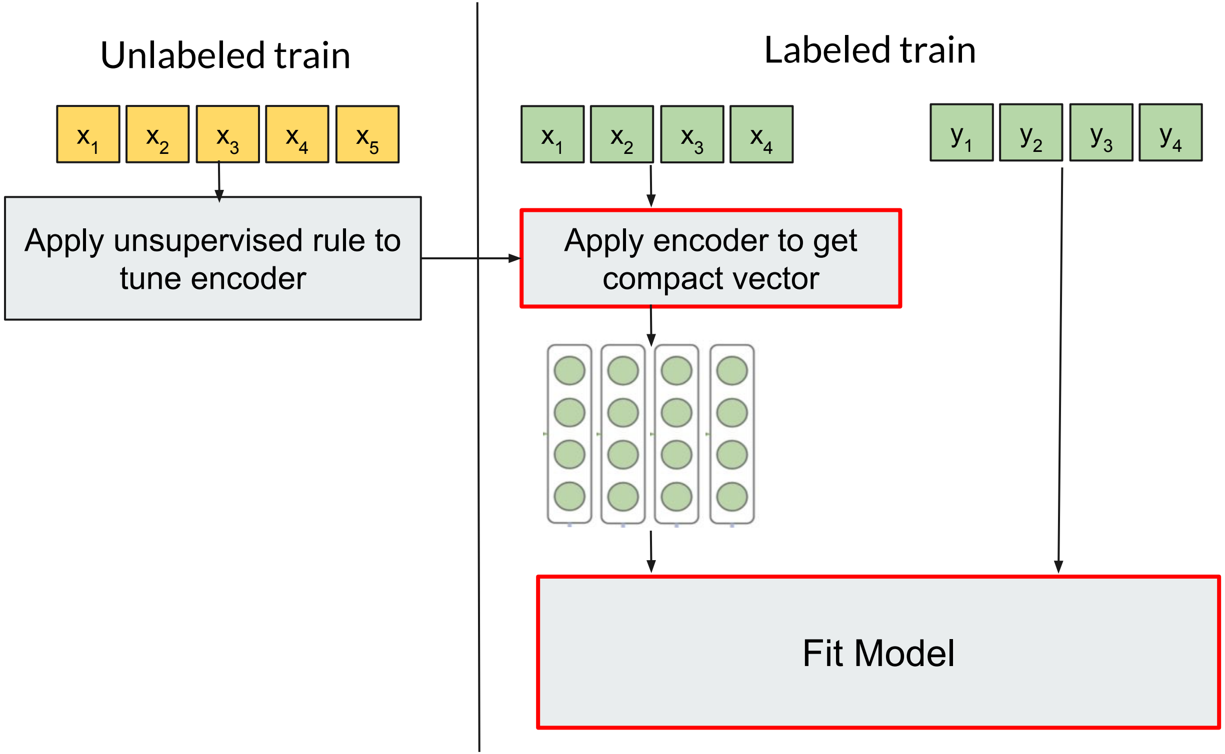
Unsupervised Learning Rules

- **VAE:** Severe overfitting to training space.
- **GANs:** Great for images, weak on discrete data (ex. text).
- **Both:** Learning rule not unsupervised (ex. surrogate loss).

**Question:** Can we meta-learn an unsupervised learning rule?

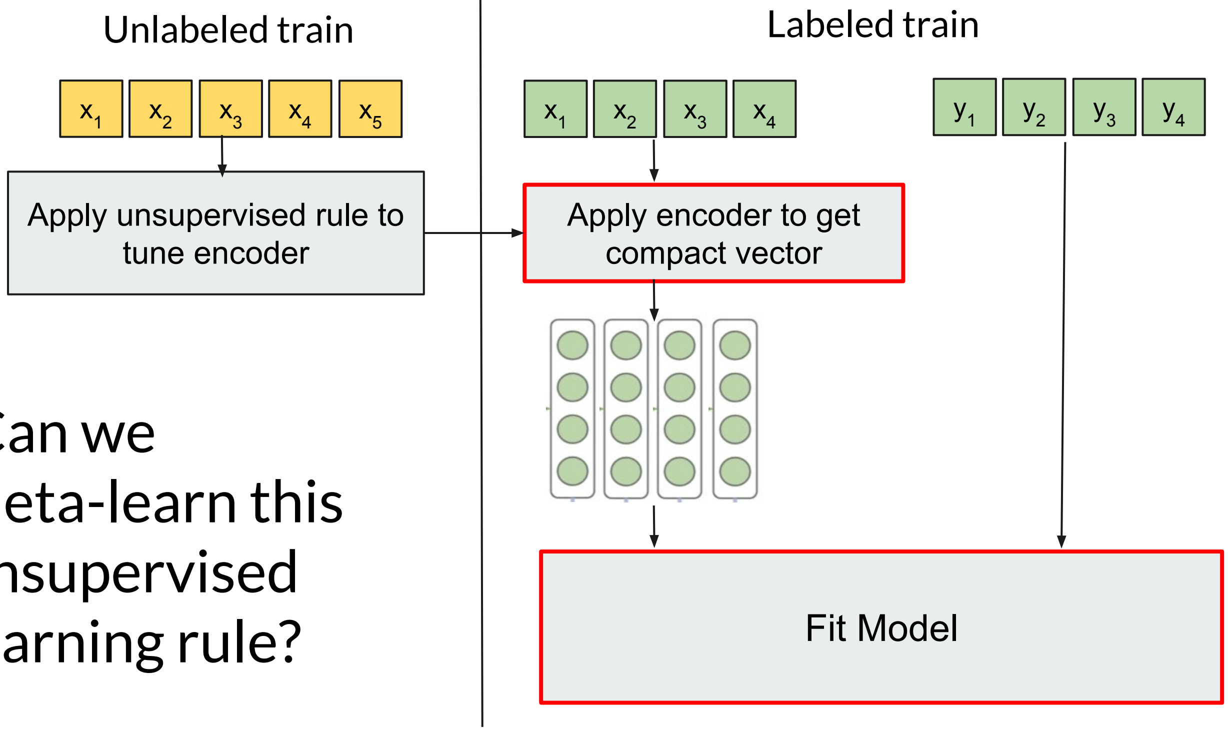
# Semi-Supervised Few-Shot Classification





# Semi-Supervised Few-Shot Classification






Can we meta-learn this unsupervised learning rule?

# Learning the Learning Rule



Backpropagation:

$$\left\{ \begin{aligned} W_{ij}^{[l]} &\rightarrow W_{ij}^{[l]} - \lambda \frac{\partial \mathbf{L}(W_{ij}^{[l]}, \mathbf{b}^{[l]})}{\partial W_{ij}^{[l]}} \\ b^{[l]} &\rightarrow b^{[l]} - \lambda \frac{\partial \mathbf{L}(W_{ij}^{[l]}, \mathbf{b}^{[l]})}{\partial b^{[l]}} \end{aligned} \right\}$$

Unsupervised Update: $\Delta W = f(\theta, h^{[l-1]})$

# Method Overview



## Outer loop

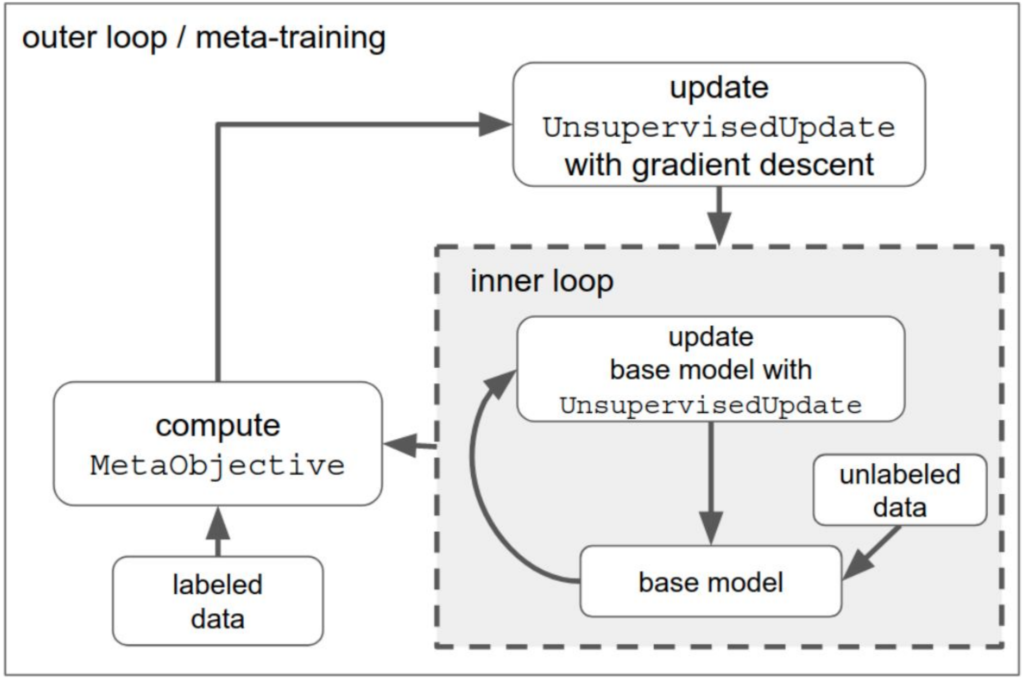
- Optimize meta-objective:

$$\theta^* = \arg\min_{\theta} \mathbf{E}_{\mathrm{task}}\left[\sum_t \mathrm{MetaObjective}(\phi_t)\right]$$

## Inner loop

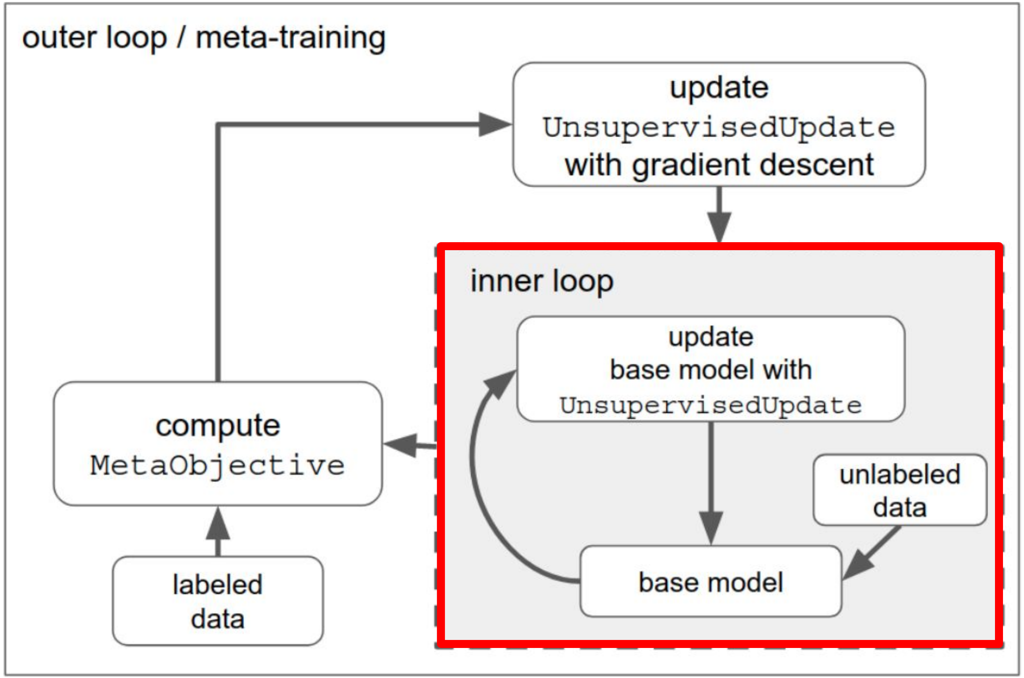
- Learn encoder using unsupervised update rule.

# Meta-Learning Setup



# Meta-Learning Setup

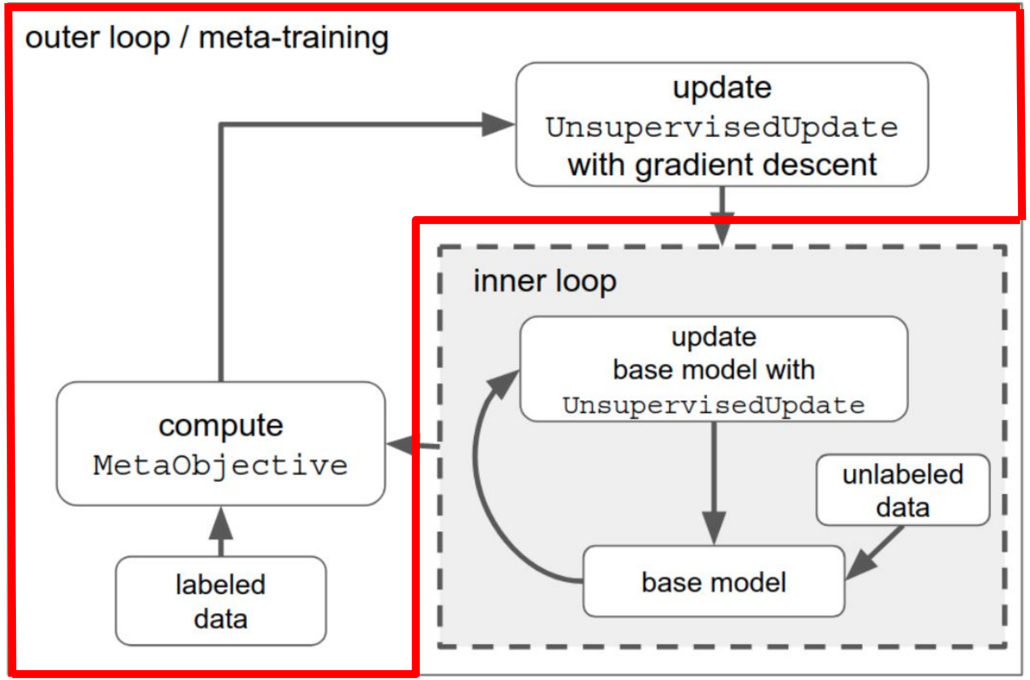





Inner loop applies an unsupervised learning alg. *on unlabeled data*

# Meta-Learning Setup






Inner loop applies an unsupervised learning alg. *on unlabeled data*

Outer loop evaluates unsupervised learning alg. *using labeled data*

# Inner Loop



**Question:** Given a base model, $g(x; \phi)$, which encodes inputs into compact vectors, how do we learn its parameters $\phi$ to give useful features?

# Inner Loop



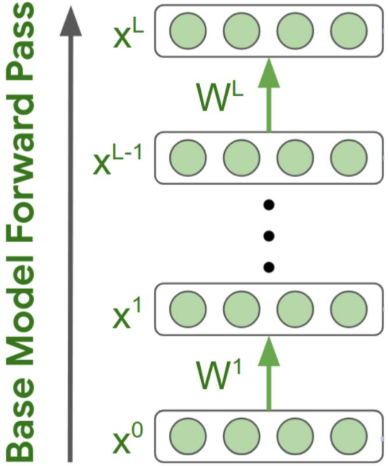
**Question:** Given a base model, $g(x; \phi)$, which encodes inputs into compact vectors, how do we learn its parameters $\phi$ to give useful features?

**Idea:** What if we use another neural network to generate a neuron-specific error signal?

*Then* we can learn its parameters $\theta$ (the meta-parameters) to produce useful error signals

# Inner Loop: Forward Pass



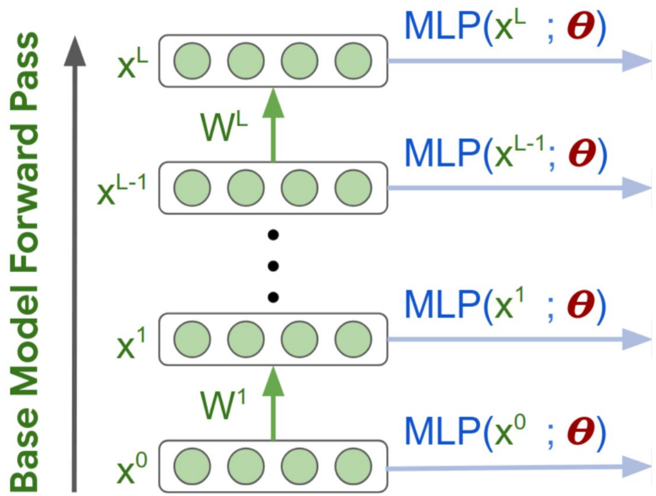



1) Take an input

2) Generate intermediate activations

3) Produce a feature representation

# Inner Loop: Generate Error Signal

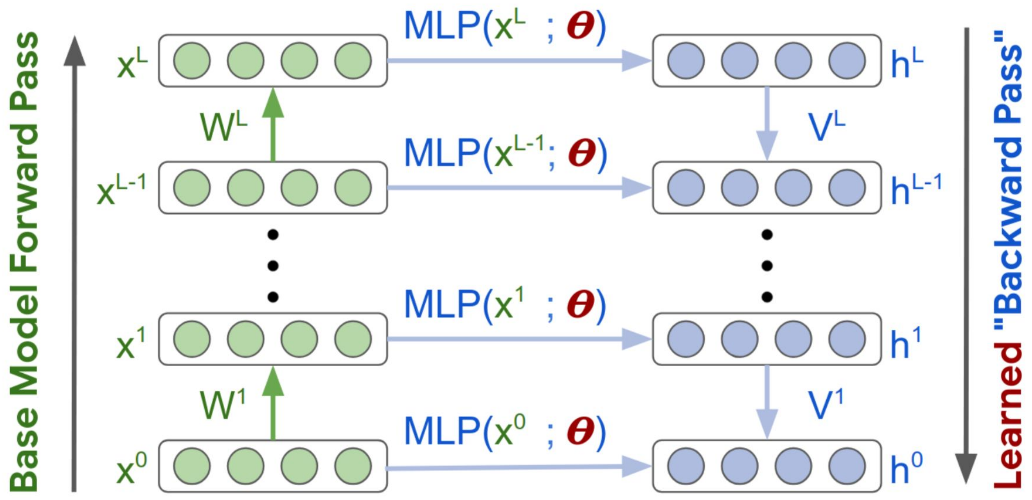



1) Input each layer's activation through an MLP

2) Output error vector

# Inner Loop: Backward Pass



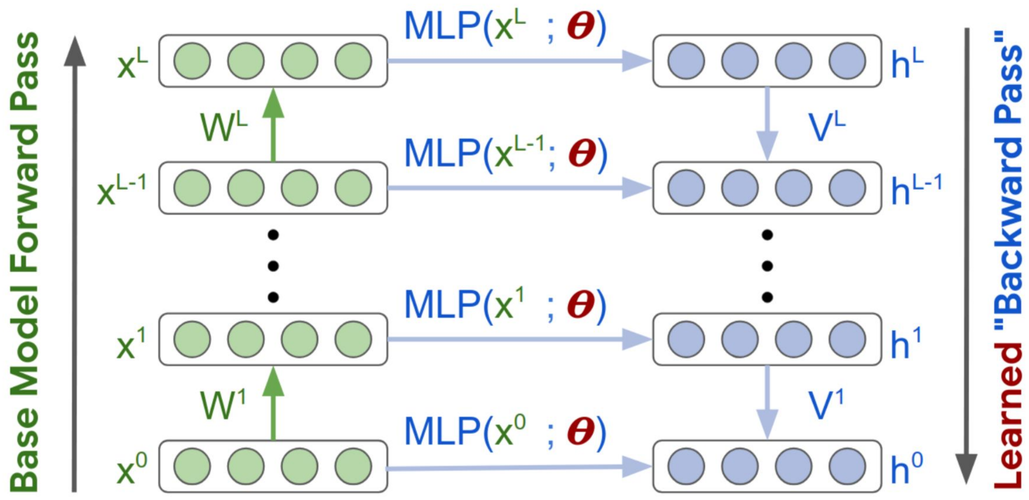



1) Initialize top-level error with output of MLP
2) Backprop the error

3) Linearly combine output from MLP with backpropagated error

# Inner Loop: Update $\phi$






$\phi$ consists of all base model parameters $W^i$, $V^i$, and $b^i$

Updates like $\Delta W^i$, $\Delta V^i$ are linear* functions of local error quantities $h^{i-1}$ and $h^i$

**There are also nonlinear normalizations within this function*

# Inner Loop: Key Points

# Inner Loop: Key Points



- Error generating network replicates the mechanics of backprop for unsupervised learning

# Inner Loop: Key Points



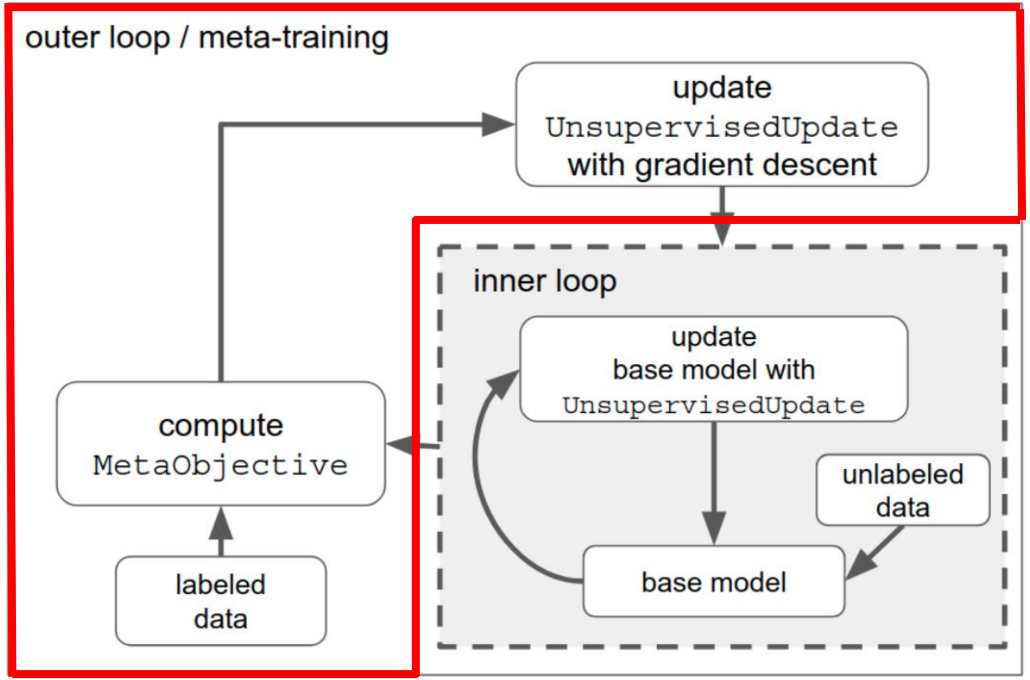
- Error generating network replicates the mechanics of backprop for unsupervised learning
- An iterative updates tune $\phi$ for some higher-level objective

# Inner Loop: Key Points



- Error generating network replicates the mechanics of backprop for unsupervised learning
- An iterative updates tune $\phi$ for some higher-level objective
- Outer loop sets objective by modifying the error generating function

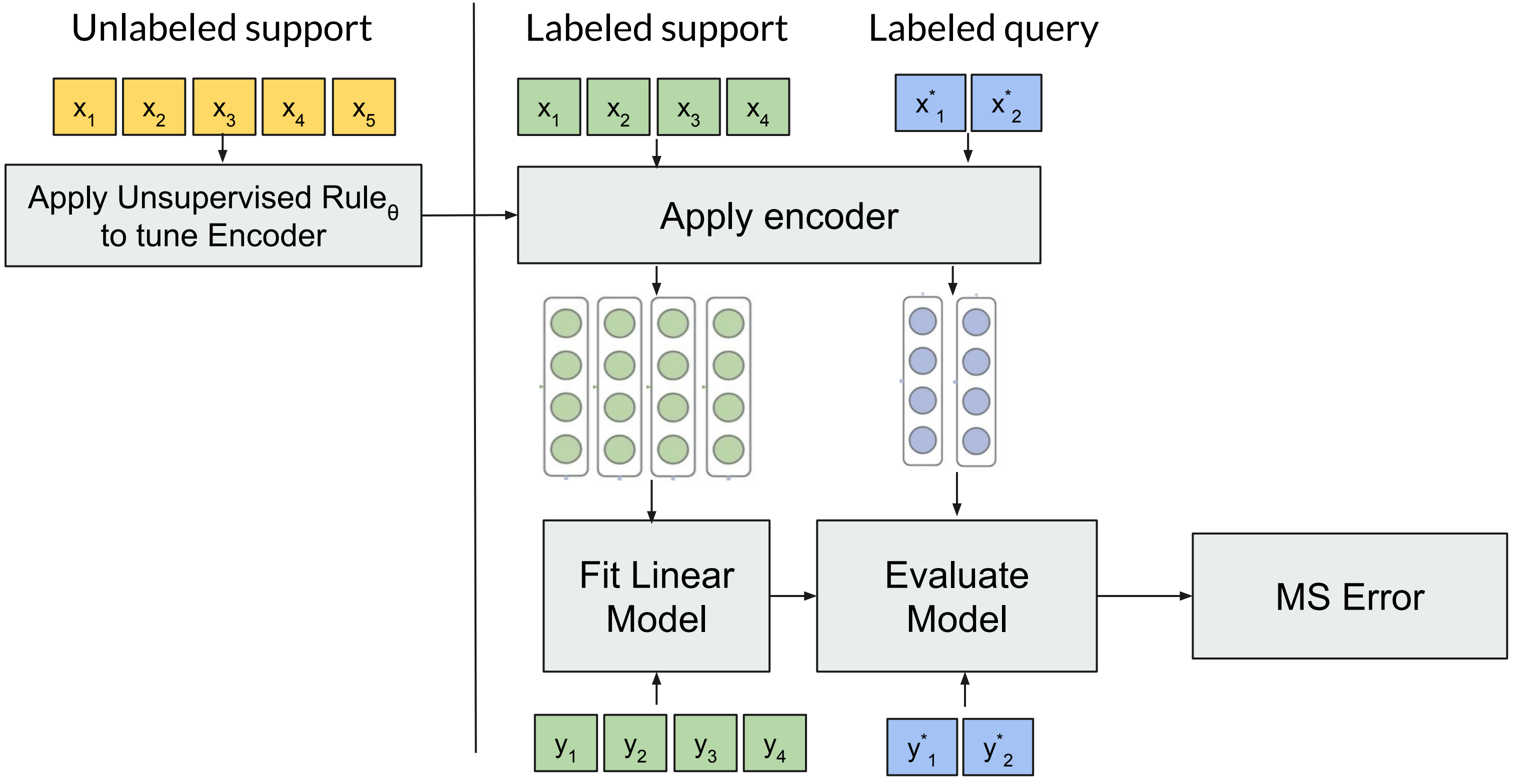
# Outer Loop



outer loop / meta-training
update
UnsupervisedUpdate
with gradient descent
inner loop
update
base model with
UnsupervisedUpdate
compute
MetaObjective
unlabeled
data
base model
labeled
data

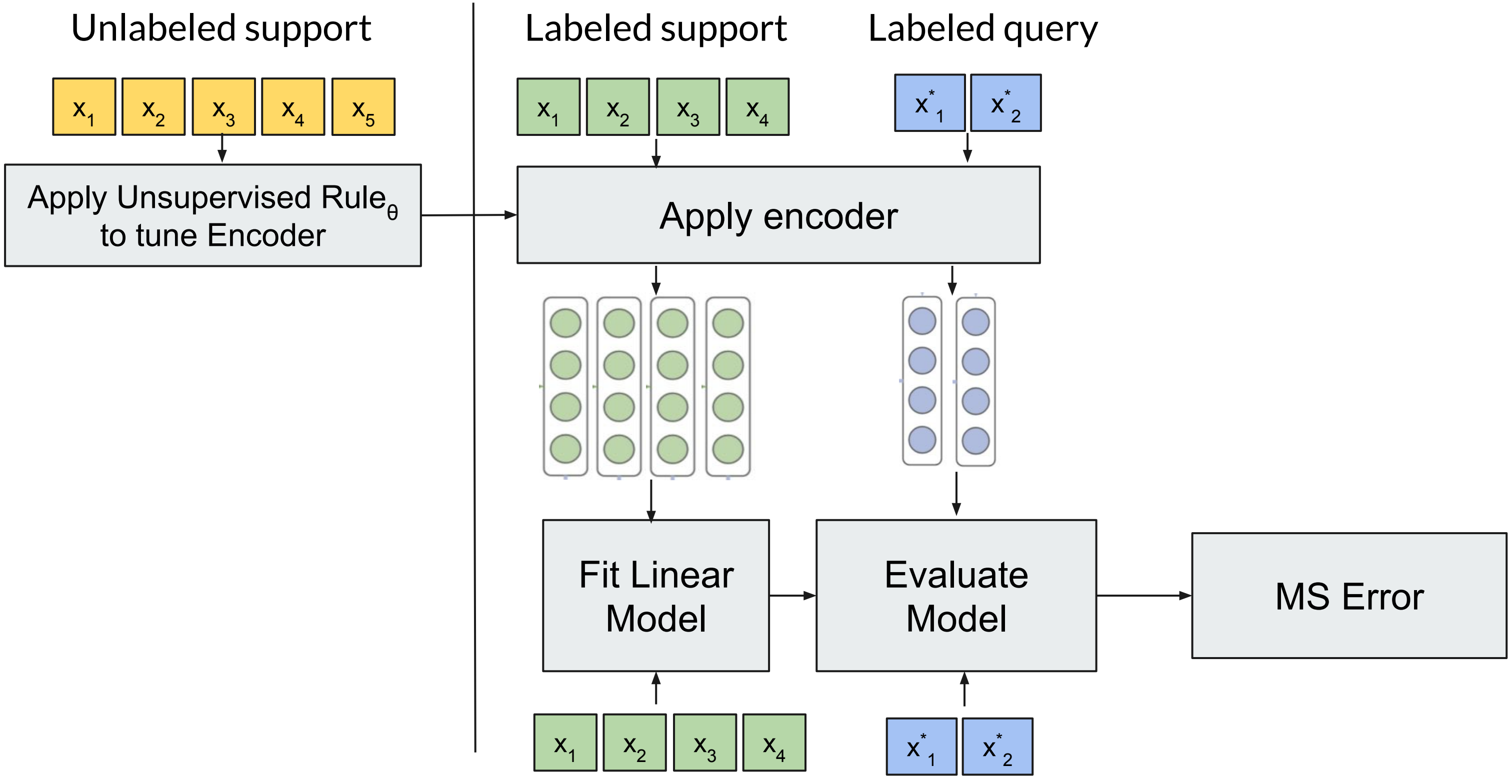
# Outer Loop: Compute MetaObjective





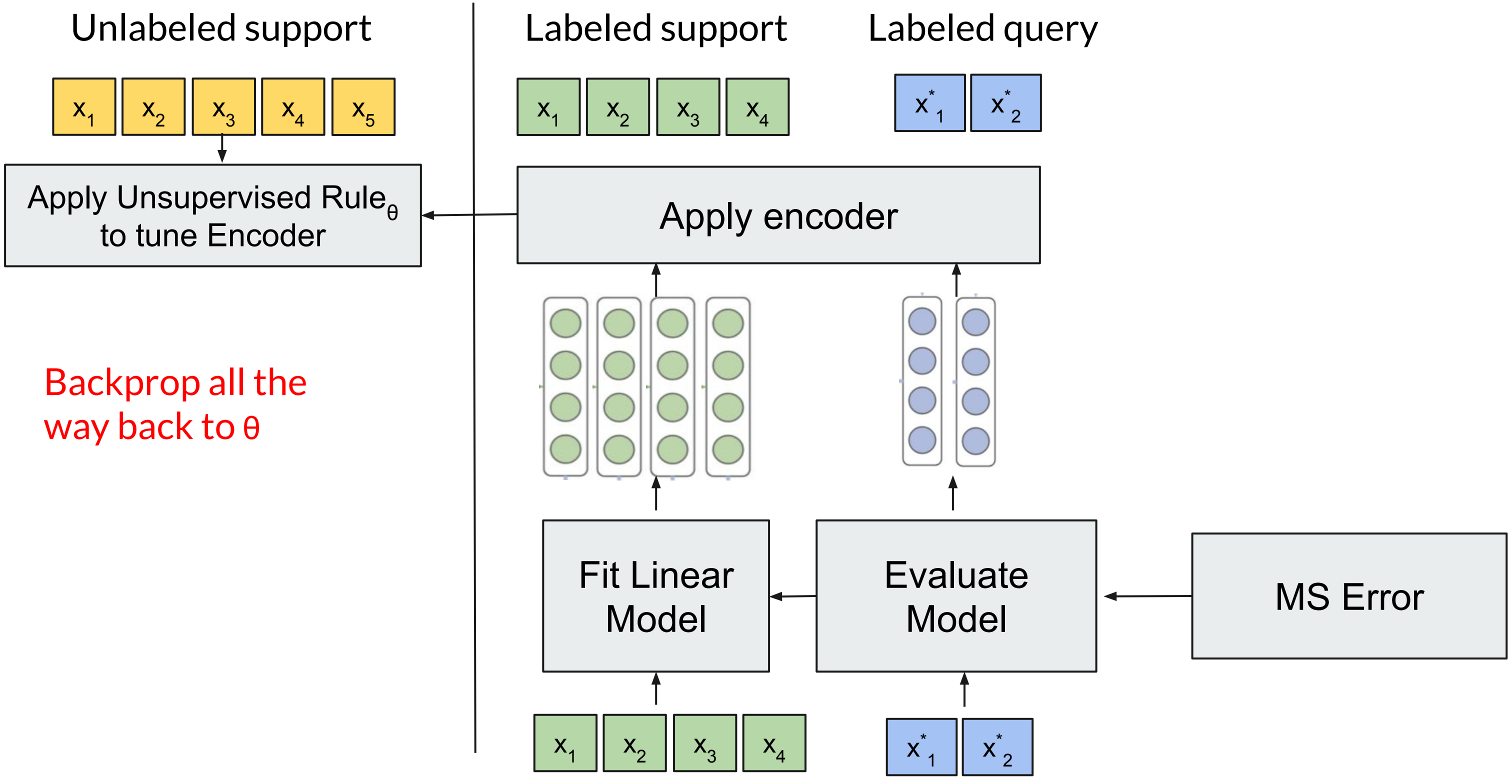
# Outer Loop: Compute MetaObjective





# Outer Loop: Compute MetaObjective

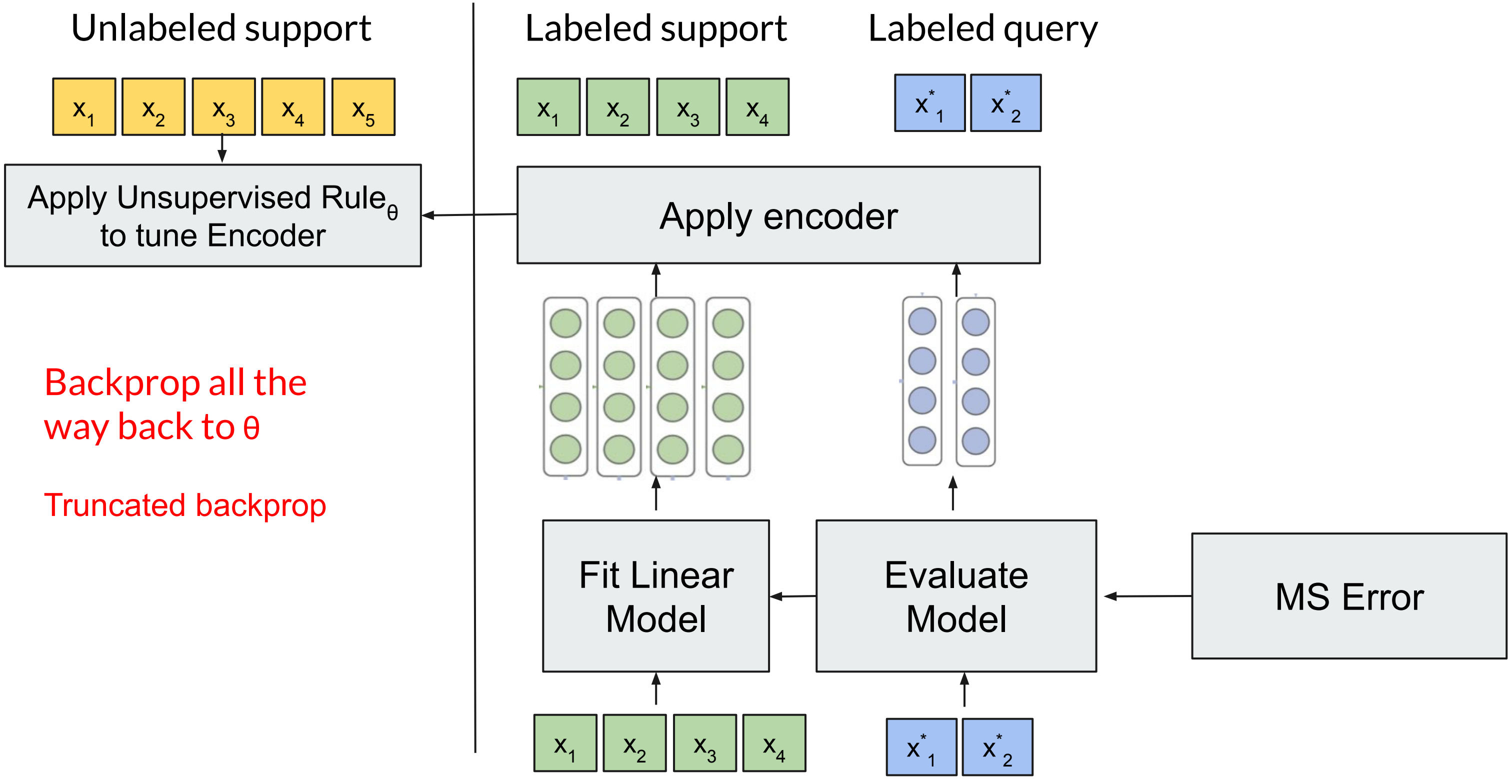





Backprop all the way back to θ

# Outer Loop: Compute MetaObjective





# Results

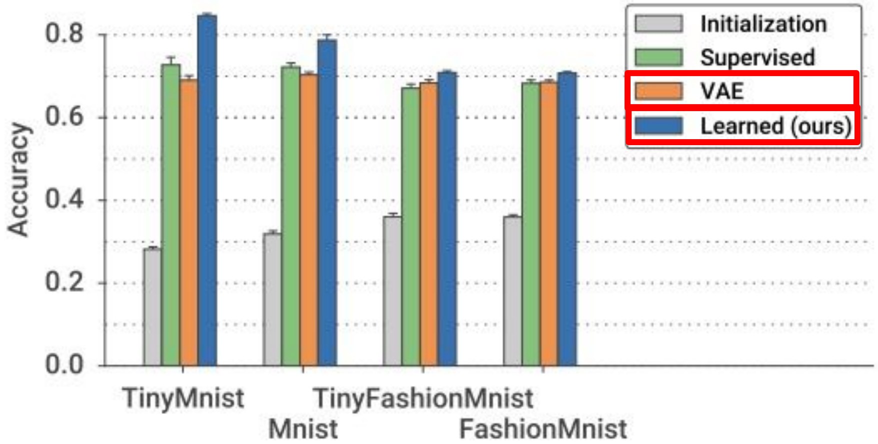


Training Data: CIFAR10 & Imagenet.

- Generalization over datasets.
- Generalization over domains
- Generalization over network architectures

# Results: Generalization over datasets



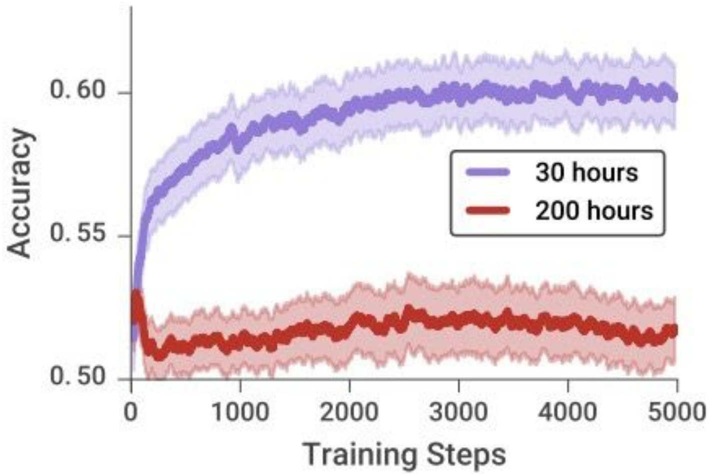



**What's going on?**

- Evaluation of unsupervised learning rule on different datasets
- Comparison to other methods.

# Results: Generalization over Domains

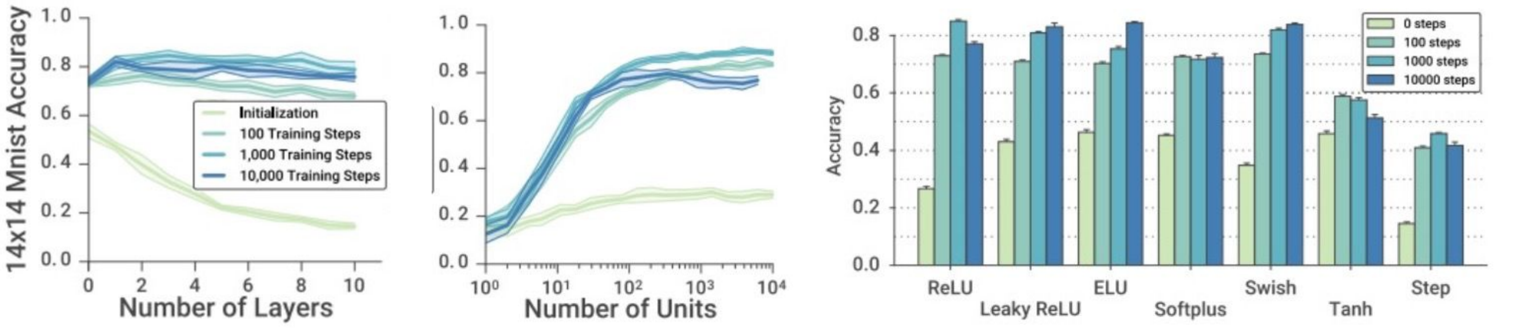





**What's going on?**
Evaluation of unsupervised learning rule on 2-way text classification. 30h vs 200h of meta-training.

# Results: Generalization over Networks






**What's going on?**
Evaluation of unsupervised learning rule on different network architectures.

# Critiques: Limitations

Computationally expensive. 8 days, 512 workers.

Many, many tricks.

Lack of ablative analysis.

Reproducibility. # labeled examples? # unlabeled?

# Critiques: Suggestions



Ablative analysis

Implicit MAML?

Investigate generalization to CNN and attention-based models.

Better way to encode learning rule? Is this architecture expressive?