Optimization-Based Meta-Learning

CS 330

Course Reminders

Project group form due **tonight**.
(for assigning project mentors)

Homework 1 due **Wednesday**

Following up on some high-res feedback:

- optional reading materials posted on website
- coding environment set-up: detailed guidance on Azure for HW2 & HW3
- managing questions in lecture

Plan for Today

Recap
- Meta-learning problem & black-box meta-learning

Optimization Meta-Learning } **Part of Homework 2!**
- Overall approach
- Compare: optimization-based vs. black-box
- Challenges & solutions
- Case study of land cover classification (time-permitting)

Goals for by the end of lecture:
- Basics of optimization-based meta-learning techniques (& how to implement)
- Trade-offs between black-box and optimization-based meta-learning

Problem Settings Recap

Multi-Task Learning

Solve multiple tasks $\mathcal{T}_1, \cdots, \mathcal{T}_T$ at once.

$$\min_{\theta} \sum_{i=1}^{T} \mathcal{L}_i(\theta, \mathcal{D}_i)$$

Transfer Learning

Solve target task $\mathcal{T}_b$ after solving source task $\mathcal{T}_a$ by *transferring* knowledge learned from $\mathcal{T}_a$

Meta-Learning Problem

Transfer Learning with Many Source Tasks

Given data from $\mathcal{T}_1, \ldots, \mathcal{T}_n$, solve new task $\mathcal{T}_{\text{test}}$ more quickly / proficiently / stably

Example Meta-Learning Problem

5-way, **1**-shot image classification (MiniImagenet)

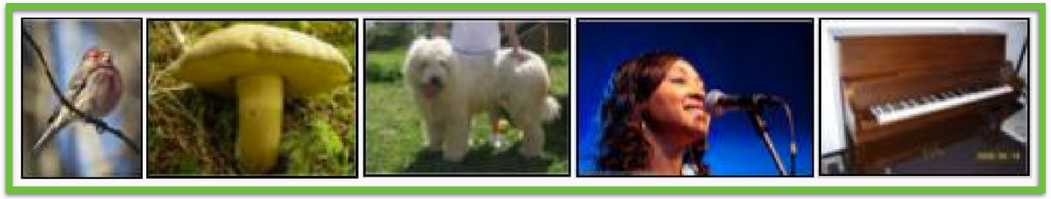
Given 1 example of 5 classes:

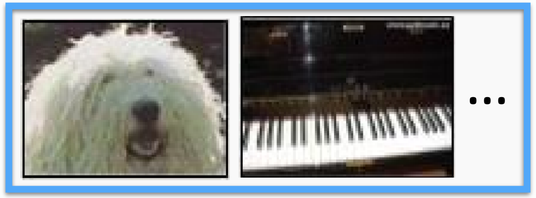
Classify new examples

meta-test

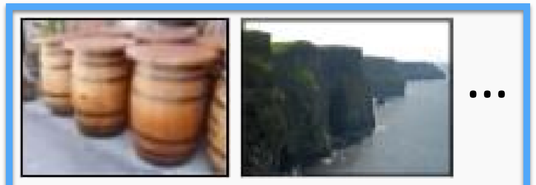
meta-training

$\mathcal{T}_1$



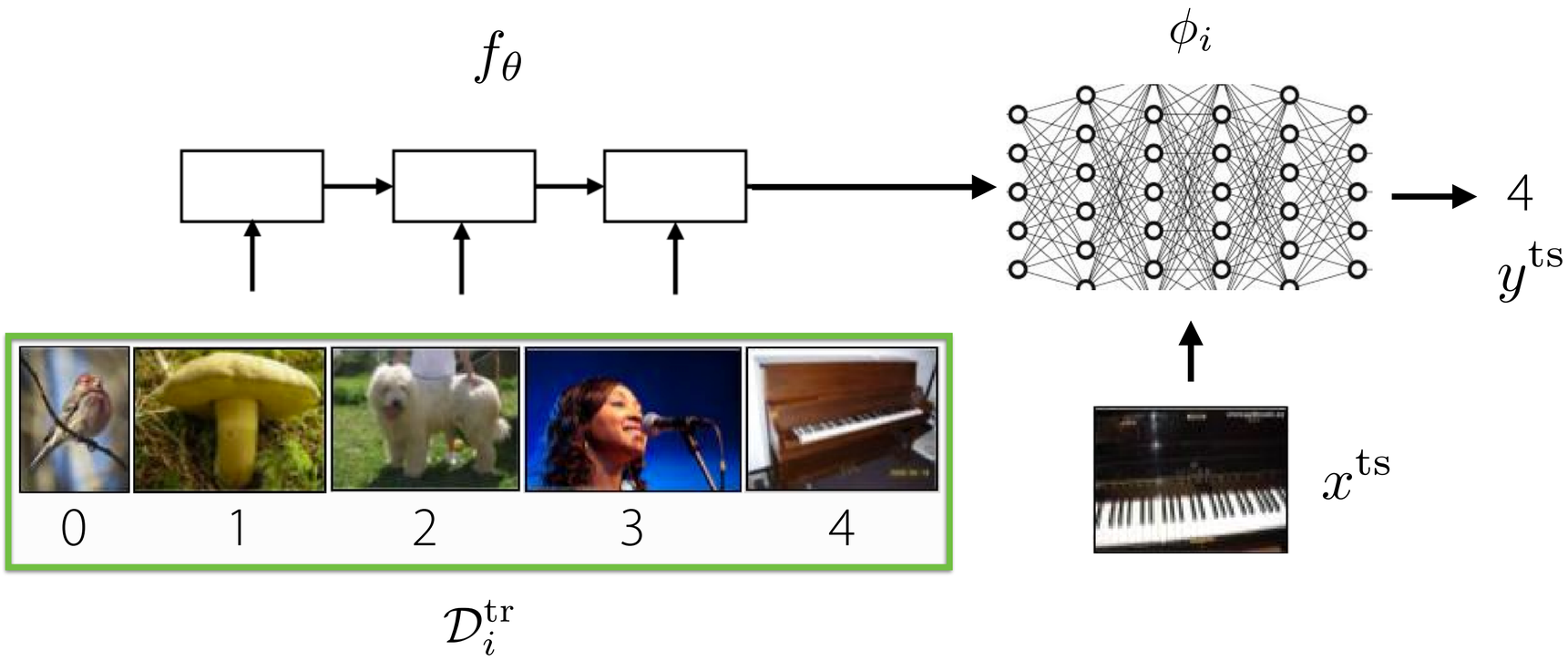


$\mathcal{T}_2$



Can replace image classification with: **regression**, **language generation**, **skill learning**, **any ML problem**

Black-Box Adaptation



general form:

$$y^{\text{ts}} = f_{\text{black-box}}(\mathcal{D}_i^{\text{tr}}, x^{\text{ts}})$$

+ **expressive**

- **challenging optimization** problem

How else can we represent $\phi_i = f_\theta(\mathcal{D}_i^{\text{tr}})$?

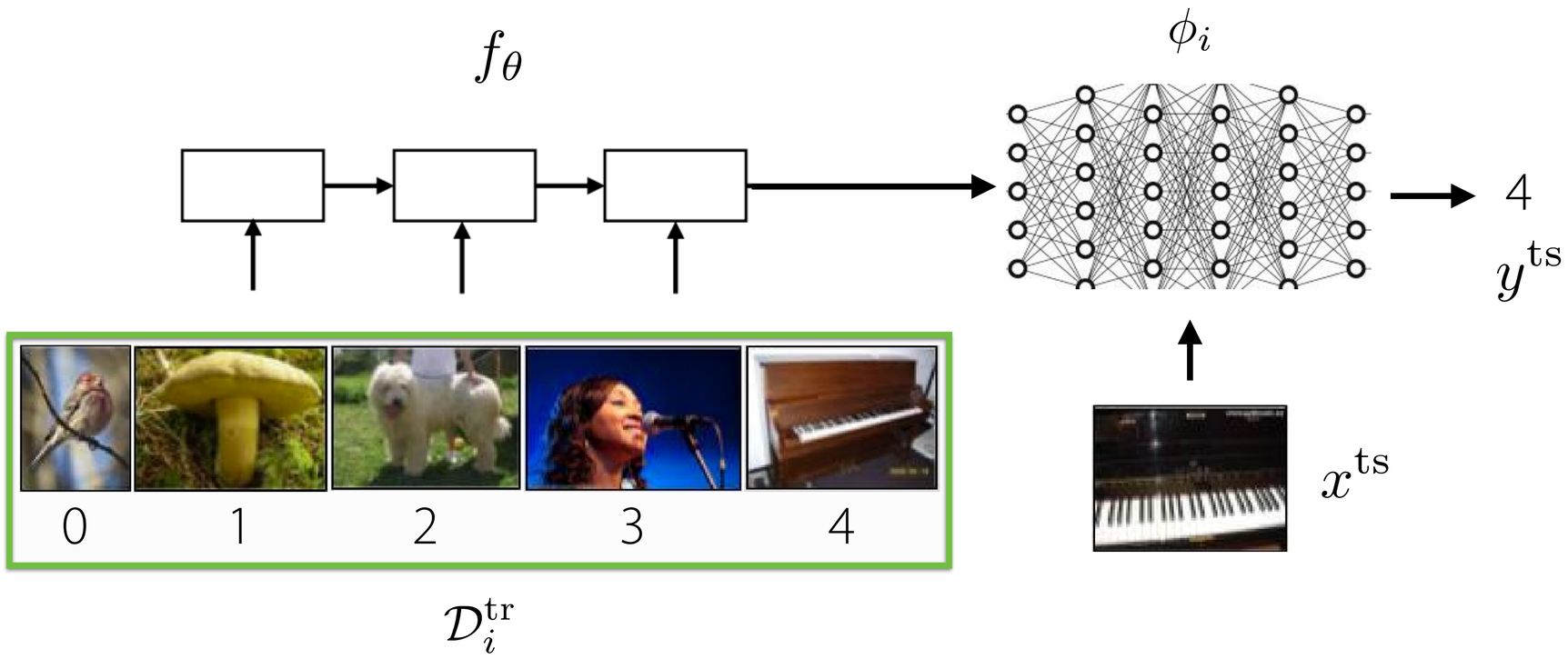
What if we treat it as an **optimization** procedure?

Plan for Today

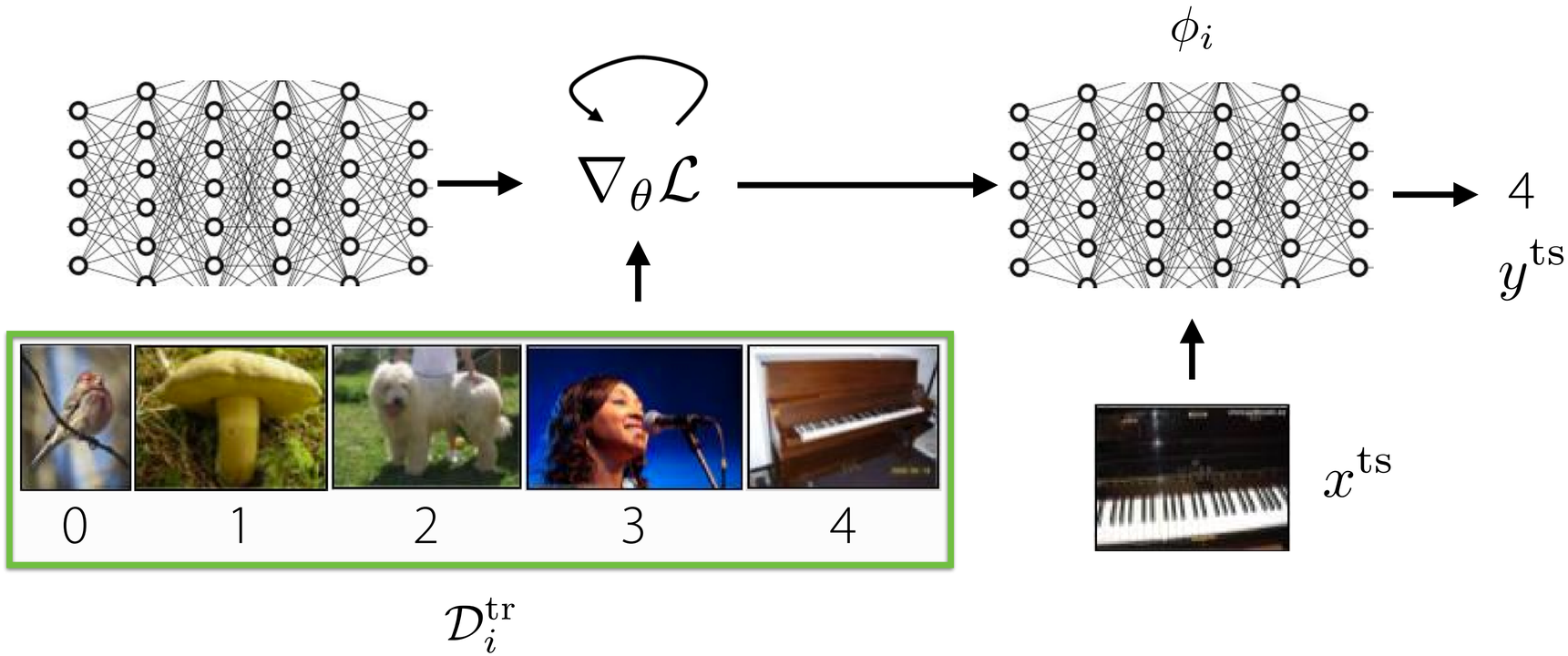
Recap
- Meta-learning problem & black-box meta-learning

Optimization Meta-Learning } Part of Homework 2!
- Overall approach
- Compare: optimization-based vs. black-box
- Challenges & solutions
- Case study of land cover classification (time-permitting)

~~Black-Box~~ ~~Adaptation~~ Optimization-Based Adaptation



~~Black-Box~~ ~~Adaptation~~ Optimization-Based Adaptation



Key idea: embed optimization inside the inner learning process

Why might this make sense?

Recall: Fine-tuning

Fine-tuning

$$\phi \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}(\theta, \mathcal{D}^{\text{tr}})$$

pre-trained parameters

training data for new task

(typically for many gradient steps)

Universal Language Model Fine-Tuning for Text Classification. Howard, Ruder. ‘18

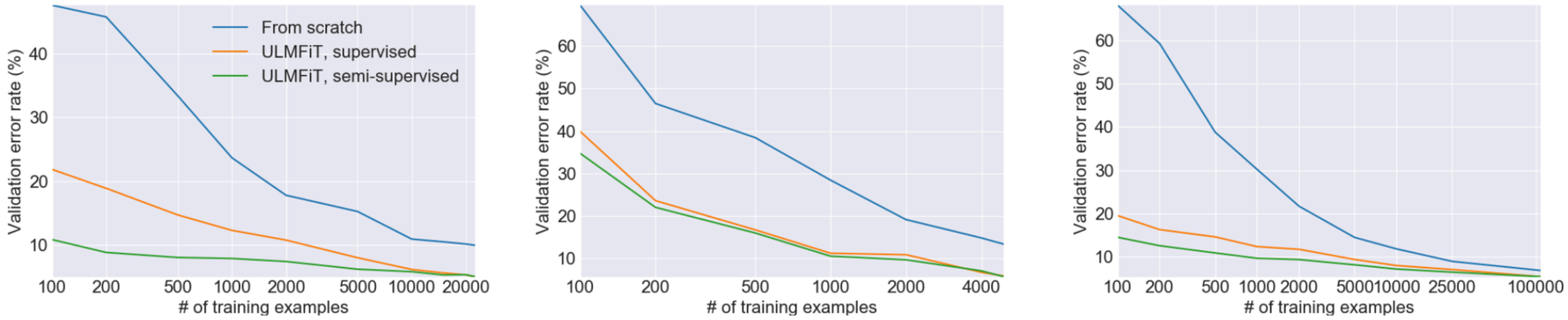


Figure 3: Validation error rates for supervised and semi-supervised ULMFiT vs. training from scratch with different numbers of training examples on IMDb, TREC-6, and AG (from left to right).

Fine-tuning less effective with **very small datasets**.

Optimization-Based Adaptation

Fine-tuning
[test-time]

pre-trained parameters

$$\phi \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}(\theta, \mathcal{D}^{\text{tr}})$$

training data for new task

Meta-learning

$$\min_\theta \sum_{\text{task } i} \mathcal{L}(\theta - \alpha \nabla_\theta \mathcal{L}(\theta, \mathcal{D}_i^{\text{tr}}), \mathcal{D}_i^{\text{ts}})$$

Key idea: Over many tasks, learn parameter vector θ that transfers via fine-tuning

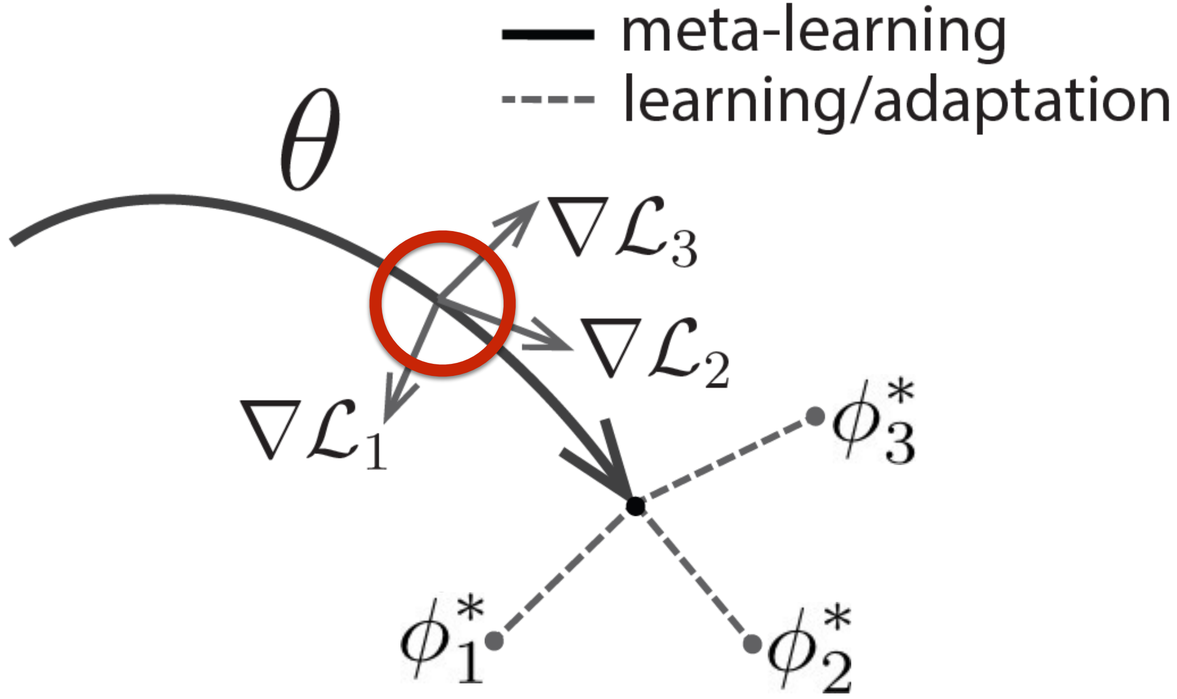
Finn, Abbeel, Levine. Model-Agnostic Meta-Learning. ICML 2017

Optimization-Based Adaptation

$$\min_{\theta} \sum_{\text{task } i} \mathcal{L}(\theta - \alpha \nabla_{\theta} \mathcal{L}(\theta, \mathcal{D}_i^{\text{tr}}), \mathcal{D}_i^{\text{ts}})$$

θ parameter vector being meta-learned

ϕ_i^* optimal parameter vector for task i



Model-**A**gnostic **M**eta-**L**earning

Finn, Abbeel, Levine. Model-Agnostic Meta-Learning. ICML 2017

Optimization-Based Adaptation

Key idea: Acquire ϕ_i through optimization.

General Algorithm:

~~Black box approach~~ Optimization-based approach

1. Sample task $\mathcal{T}_i$ *(or mini batch of tasks)*
2. Sample disjoint datasets $\mathcal{D}_i^{\text{tr}}, \mathcal{D}_i^{\text{test}}$ from $\mathcal{D}_i$
3. ~~Compute $\phi_i \leftarrow f_\theta(\mathcal{D}_i^{\text{tr}})$~~ Optimize $\phi_i \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}(\theta, \mathcal{D}_i^{\text{tr}})$
4. Update θ using $\nabla_\theta \mathcal{L}(\phi_i, \mathcal{D}_i^{\text{test}})$

—> brings up **second-order** derivatives

Do we need to compute the full Hessian? 😱 **-> whiteboard**

Do we get higher-order derivatives with more inner gradient steps?

$$\frac{d}{d\theta}\mathcal{L}(\phi_i, \mathcal{D}_i^{\text{ts}})$$
$$= \nabla_{\bar{\phi}}\mathcal{L}(\bar{\phi}, \mathcal{D}_i^{\text{ts}})|_{\bar{\phi}=\phi_i} \frac{d\phi_i}{d\theta}$$
$$= \nabla_{\bar{\phi}}\mathcal{L}(\bar{\phi}, \mathcal{D}_i^{\text{ts}})|_{\bar{\phi}=\phi_i} \left(I - \alpha\frac{d^2}{d\theta^2}\mathcal{L}(\theta, \mathcal{D}_i^{\text{tr}})\right)$$

PyTorch

Deep learning libraries handle the math for you.

Optimization-Based Adaptation

Key idea: Acquire ϕ_i through optimization.

Meta-Test Time:

Optimization-based approach

1. Given task $\mathcal{T}_j$
2. Given training data $\mathcal{D}_j^{\text{tr}}$
3. Fine-tune $\phi_j \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}(\theta, \mathcal{D}_j^{\text{tr}})$
4. Make predictions on new datapoints $f_{\phi_j}(x)$

Plan for Today

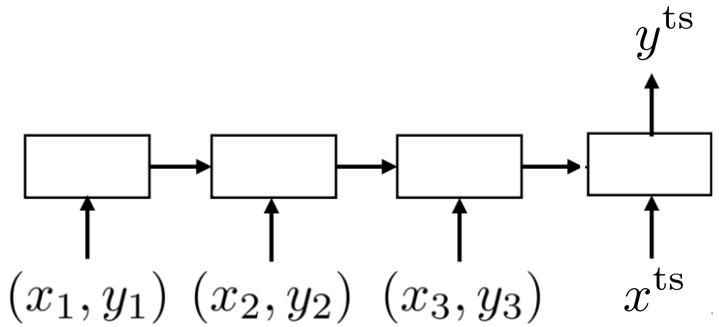
Recap
- Meta-learning problem & black-box meta-learning

Optimization Meta-Learning } Part of Homework 2!
- Overall approach
- **Compare: optimization-based vs. black-box**
- Challenges & solutions
- Case study of land cover classification (time-permitting)

Optimization vs. Black-Box Adaptation

Black-box adaptation

general form: $y^{\text{ts}} = f_{\text{black-box}}(\mathcal{D}_i^{\text{tr}}, x^{\text{ts}})$



Model-agnostic meta-learning

$$y^{\text{ts}} = f_{\text{MAML}}(\mathcal{D}_i^{\text{tr}}, x^{\text{ts}})$$
$$= f_{\phi_i}(x^{\text{ts}})$$

$$\text{where } \phi_i = \theta - \alpha \nabla_\theta \mathcal{L}(\theta, \mathcal{D}_i^{\text{tr}})$$

MAML can be viewed as **computation graph**, with embedded gradient operator

Note: Can mix & match components of computation graph

Learn initialization but replace gradient update with learned network

$$\text{where } \phi_i = \theta - \alpha \cancel{\nabla_\theta \mathcal{L}(\theta, \mathcal{D}_i^{\text{tr}})}$$
$$f(\theta, \mathcal{D}_i^{\text{tr}}, \nabla_\theta \mathcal{L})$$

Ravi & Larochelle ICLR '17

(actually precedes MAML)

This **computation graph view** of meta-learning will come back again!

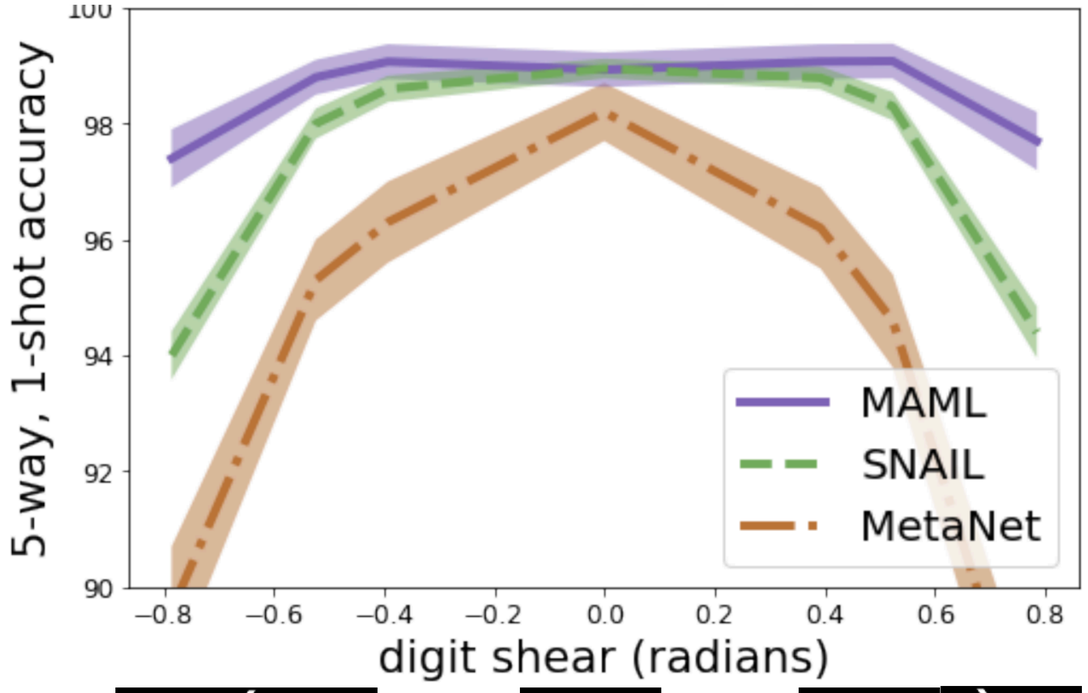
Optimization vs. Black-Box Adaptation

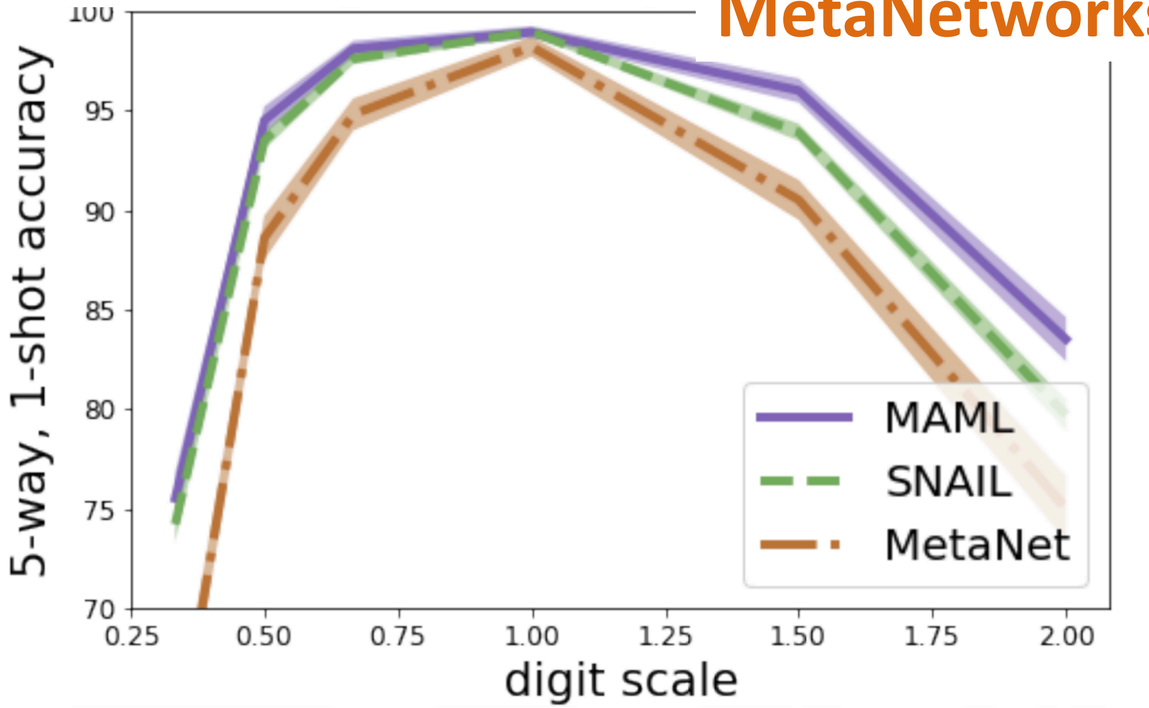
How well can learning procedures generalize to similar, but extrapolated tasks?

Omniglot image classification

MAML SNAIL, MetaNetworks

performance





Does this structure come at a cost?

Finn & Levine ICLR ’18

Black-box adaptation

$$y^{\text{ts}} = f_{\text{black-box}}(\mathcal{D}_i^{\text{tr}}, x^{\text{ts}})$$

Optimization-based (MAML)

$$y^{\text{ts}} = f_{\text{MAML}}(\mathcal{D}_i^{\text{tr}}, x^{\text{ts}})$$

Does this structure come at a cost?

For a sufficiently deep network,

MAML function can approximate any function of $\mathcal{D}_i^{\text{tr}}, x^{\text{ts}}$

Finn & Levine, ICLR 2018

Assumptions:

- nonzero α
- loss function gradient does not lose information about the label
- datapoints in $\mathcal{D}_i^{\text{tr}}$ are unique

Why is this interesting?

MAML has benefit of inductive bias without losing expressive power.

Plan for Today

Recap
- Meta-learning problem & black-box meta-learning

Optimization Meta-Learning } Part of Homework 2!
- Overall approach
- Compare: optimization-based vs. black-box
- **Challenges & solutions**
- Case study of land cover classification (time-permitting)

Optimization-Based Adaptation

Challenges. Bi-level optimization can exhibit instabilities.

Idea: Automatically learn inner vector learning rate, tune outer learning rate
(Li et al. Meta-SGD, Behl et al. AlphaMAML)

Idea: Optimize only a subset of the parameters in the inner loop
(Zhou et al. DEML, Zintgraf et al. CAVIA)

Idea: Decouple inner learning rate, BN statistics per-step (Antoniou et al. MAML++)

Idea: Introduce context variables for increased expressive power.
(Finn et al. bias transformation, Zintgraf et al. CAVIA)

Takeaway: a range of simple tricks that can help optimization significantly

Optimization-Based Adaptation

Challenges. Backpropagating through many inner gradient steps is compute- & memory-intensive.

Idea: [Crudely] approximate $\frac{d\phi_i}{d\theta}$ as identity

(Finn et al. first-order MAML '17, Nichol et al. Reptile '18)

Surprisingly works for simple few-shot problems, but (anecdotally) not for more complex meta-learning problems.

Idea: Only optimize the *last layer* of weights.

ridge regression, *logistic regression*

(Bertinetto et al. R2-D2 '19)

support vector machine

(Lee et al. MetaOptNet '19)

—> leads to a closed form or convex optimization on top of meta-learned features

Idea: Derive meta-gradient using the implicit function theorem

(Rajeswaran, Finn, Kakade, Levine. Implicit MAML '19)

—> compute full meta-gradient ***without differentiating through optimization path***

Optimization-Based Adaptation

Challenges. How to choose architecture that is effective for inner gradient step?

Idea: Progressive neural architecture search + MAML

(Kim et al. Auto-Meta)

- finds highly non-standard architecture (deep & narrow)
- different from architectures that work well for standard supervised learning

MiniImagenet, 5-way 5-shot

MAML, basic architecture: 63.11%

MAML + AutoMeta: **74.65%**

Optimization-Based Adaptation

Key idea: Acquire ϕ_i through optimization.

Takeaways: Construct *bi-level optimization* problem.

+ positive inductive bias at the start of meta-learning
+ tends to extrapolate better via structure of optimization
+ maximally expressive with sufficiently deep network
+ model-agnostic (easy to combine with your favorite architecture)
- typically requires second-order optimization
- usually compute and/or memory intensive

Plan for Today

Recap
- Meta-learning problem & black-box meta-learning

Optimization Meta-Learning } Part of Homework 2!
- Overall approach
- Compare: optimization-based vs. black-box
- Challenges & solutions
- **Case study of land cover classification** (time-permitting)

Case Study

Meta-Learning for Few-Shot Land Cover Classification

Marc Rußwurm[1,*,†], Sherrie Wang[2,3,*], Marco Körner[1], and David Lobell[2]

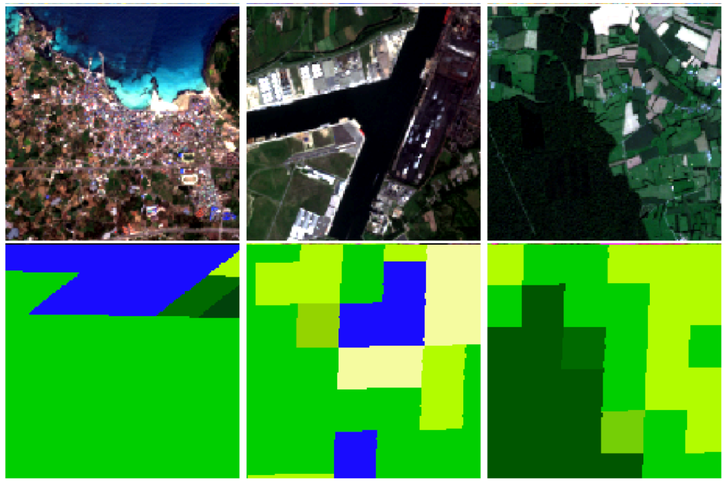
[1]Technical University of Munich, Chair of Remote Sensing Technology
[2]Stanford University, Center on Food Security and the Environment
[3]Stanford University, Institute for Computational and Mathematical Engineering

CVPR 2020 EarthVision Workshop

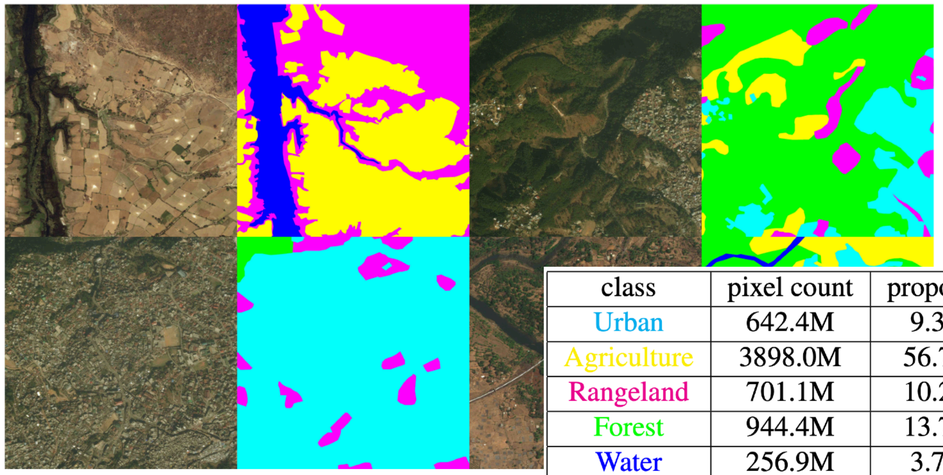
Link: https://arxiv.org/abs/2004.13390

Problem: Map land covering from satellite images

SEN12MS dataset
(Schmitt et al. 2019)



DeepGlobe dataset
(Demir et al. 2018)

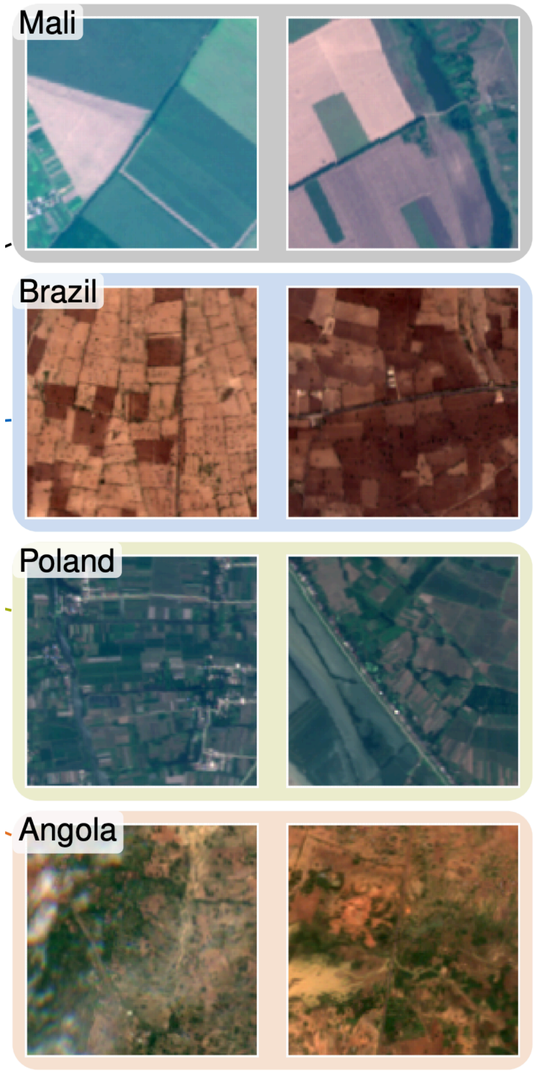


class	pixel count	proportion
Urban	642.4M	9.35%
Agriculture	3898.0M	56.76%
Rangeland	701.1M	10.21%
Forest	944.4M	13.75%
Water	256.9M	3.74%
Barren	421.8M	6.14%
Unknown	3.0M	0.04%

Applications in global urban planning, climate change research

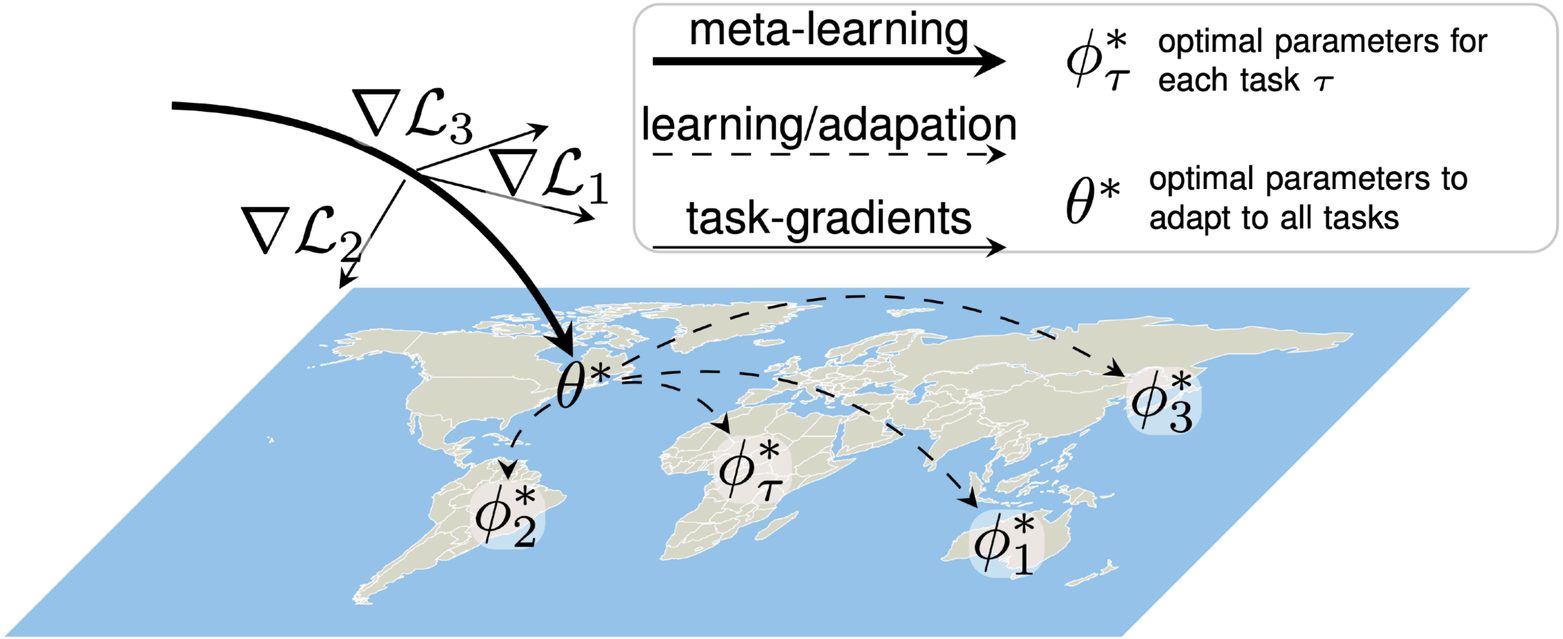
Challenges: Labeling data is expensive.
Different regions look different & have different land use proportions

Framing land cover mapping as a meta-learning problem



Different tasks: different regions of the world

Goal: Segment/classify images from a new region with a small amount of data



Croplands from four countries.

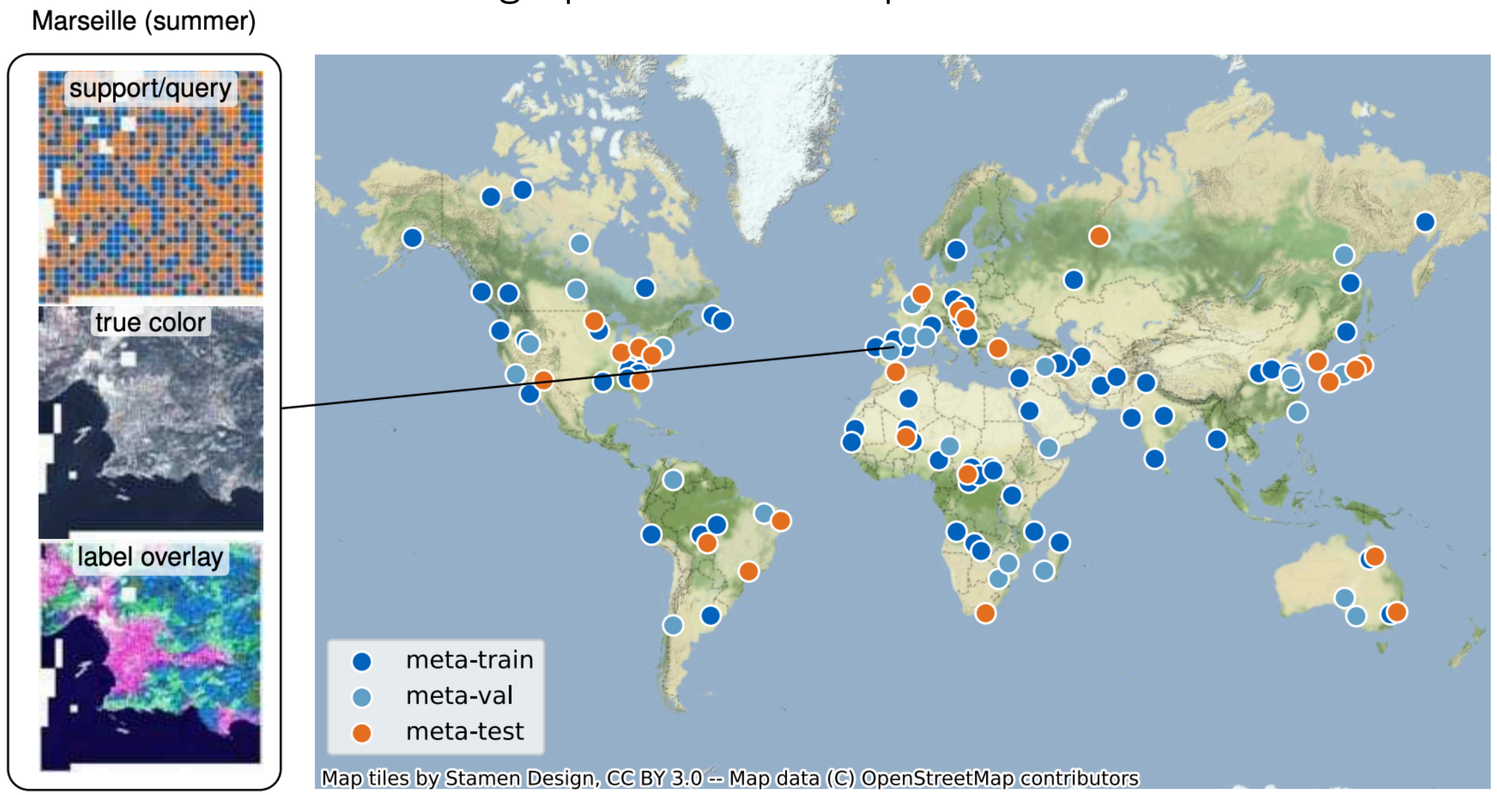
Framing land cover mapping as a meta-learning problem

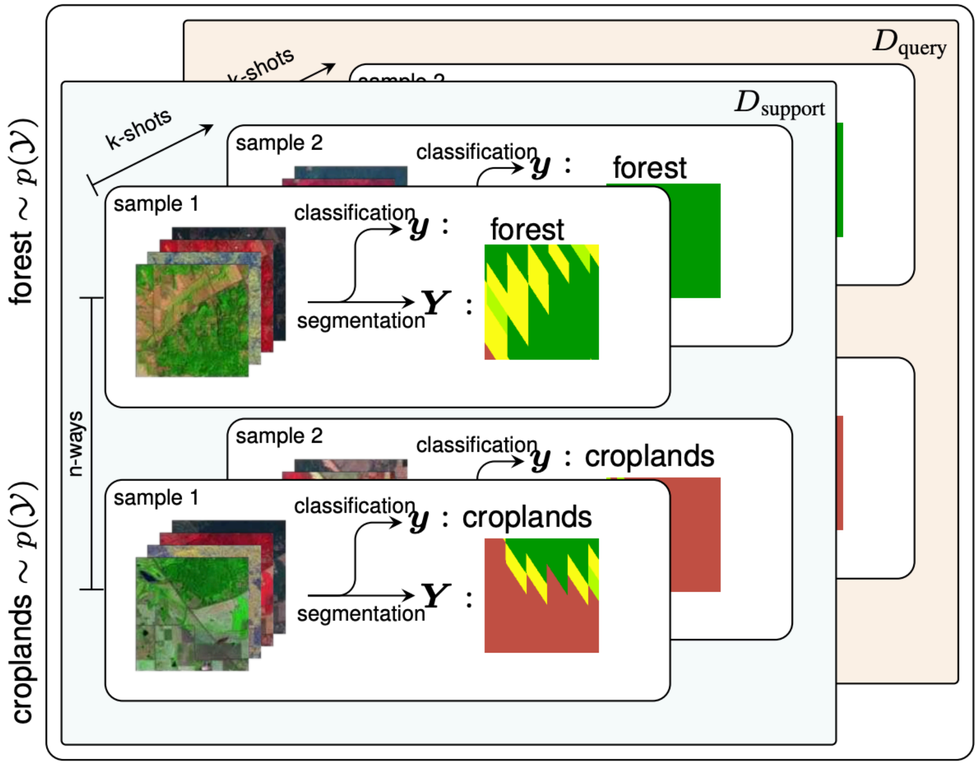
Goal: Segment/classify images from a new region with a small amount of data

SEN12MS dataset (Schmitt et al. 2019)

Geographic meta-data provided

Example 2-way 2-shot classification task



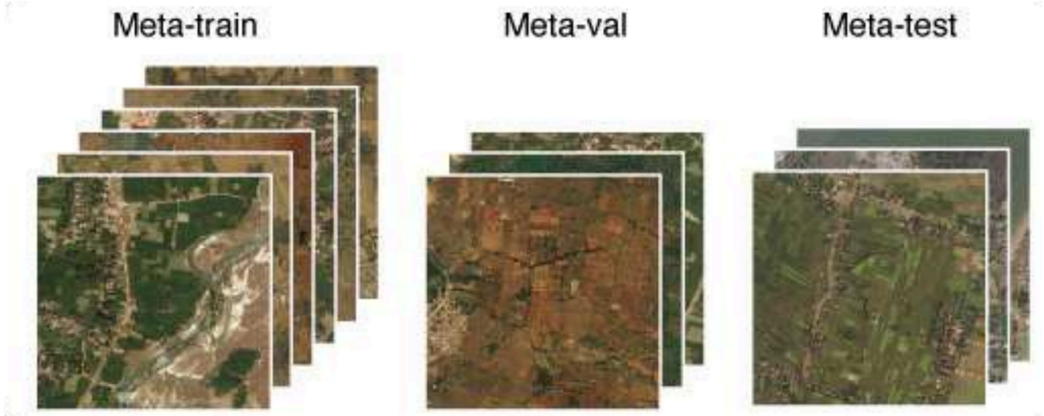


Framing land cover mapping as a meta-learning problem

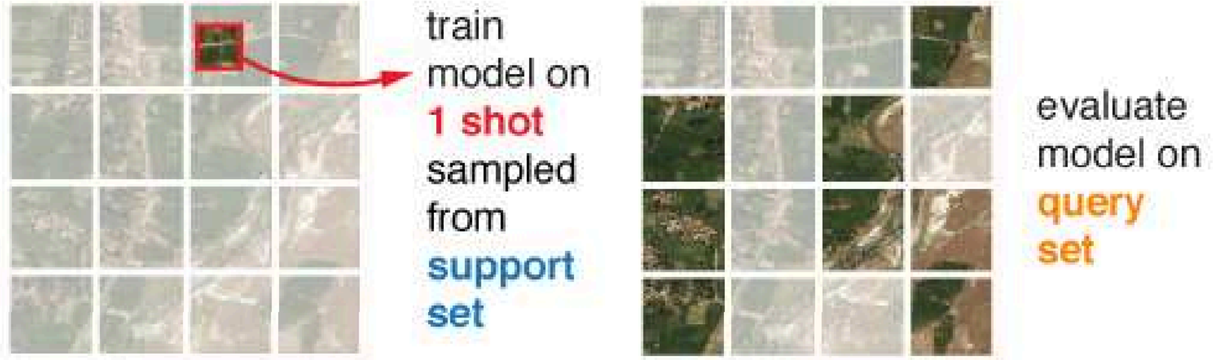
Goal: Segment/classify images from a new region with a small amount of data

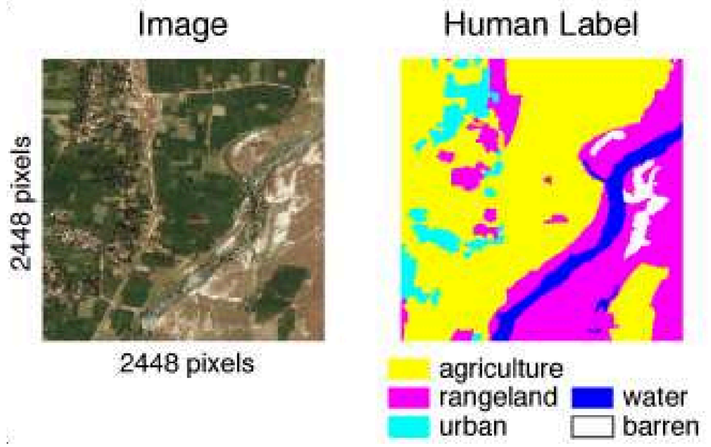
DeepGlobe dataset (Demir et al. 2018)

No geographic metadata, used clustering to guess region



Example 1-shot learning segmentation task.





Evaluation

Meta-training data: $\{\mathcal{D}_1, \ldots, \mathcal{D}_T\}$

Meta-test time: small amount of data from new region: $\mathcal{D}_j^{\text{tr}}$
(meta-test training set / meta-test support set)

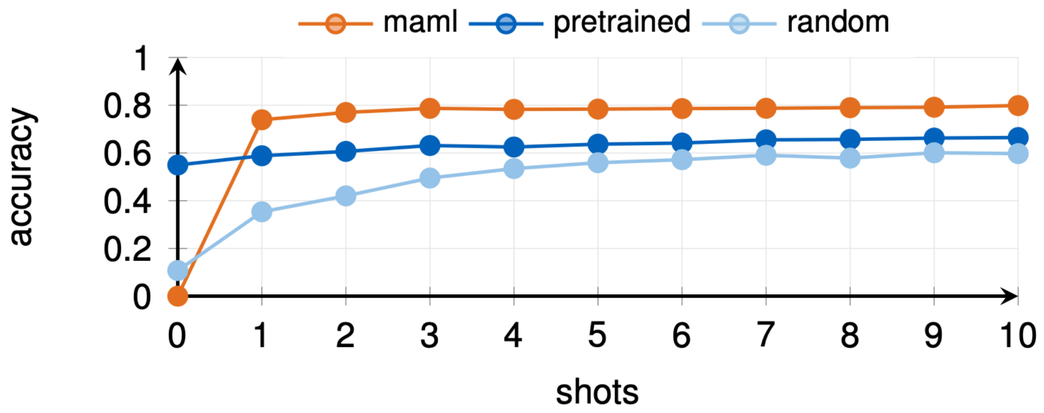
Compare:

Random init: Train from scratch on $\mathcal{D}_j^{\text{tr}}$

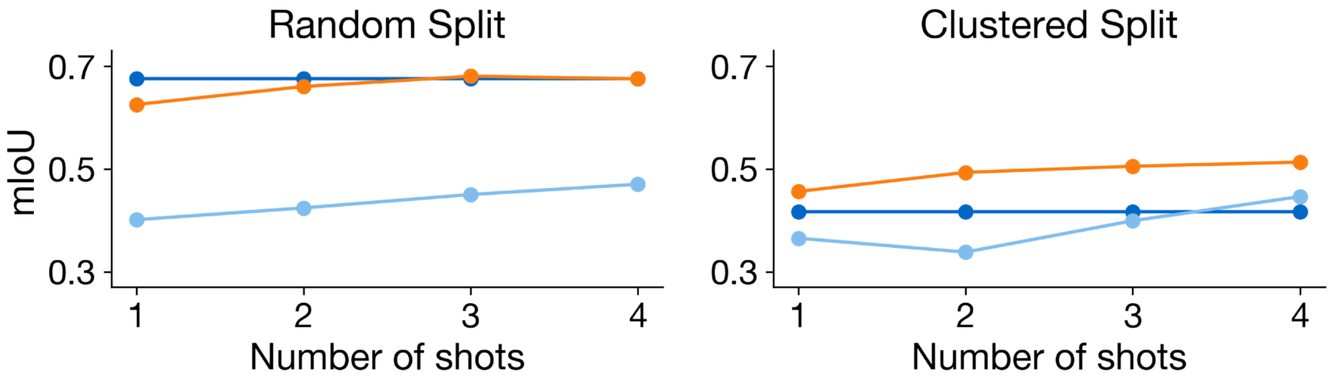
Pre-train on meta-training data $\mathcal{D}_1 \cup \ldots \cup \mathcal{D}_T$, **fine-tune** on $\mathcal{D}_j^{\text{tr}}$

MAML on meta-training data $\{\mathcal{D}_1, \ldots, \mathcal{D}_T\}$, adapt with $\mathcal{D}_j^{\text{tr}}$

SEN12MS dataset



DeepGlobe dataset



More visualizations and analysis in the paper!

Plan for Today

Recap
- Meta-learning problem & black-box meta-learning

Optimization Meta-Learning } **Part of Homework 2!**
- Overall approach
- Compare: optimization-based vs. black-box
- Challenges & solutions
- Case study of land cover classification (time-permitting)

Goals for by the end of lecture:
- Basics of optimization-based meta-learning techniques (& how to implement)
- Trade-offs between black-box and optimization-based meta-learning

Roadmap for upcoming lectures

Wednesday: Non-parametric few-shot learners, comparison of approaches

Next week: Unsupervised pre-training for few-shot learning

Following week: Advanced meta-learning topics (e.g. memorization, large-scale meta-optimization)

Course Reminders

Project group form due **tonight**.
(for assigning project mentors)

Homework 1 due **Wednesday**