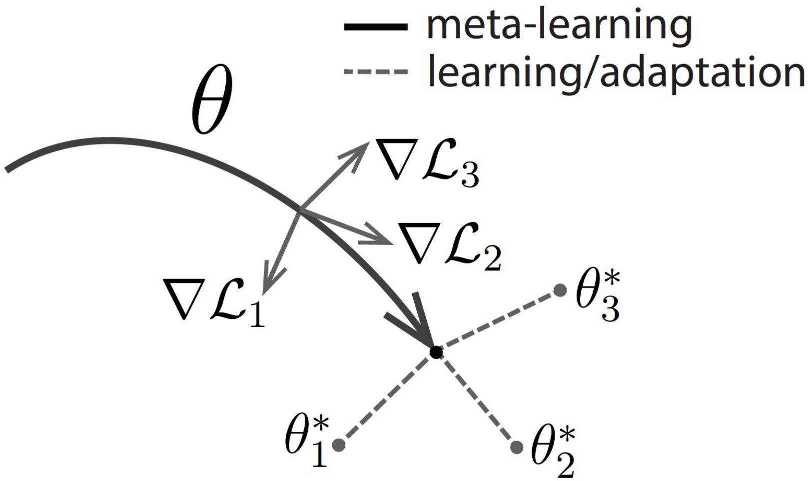
Few-shot Learning with Meta-Learning for Earth Observation

Cesar Luis Aybar Camacho

** Student at Copernicus Master in Digital Earth (EMCDE)*

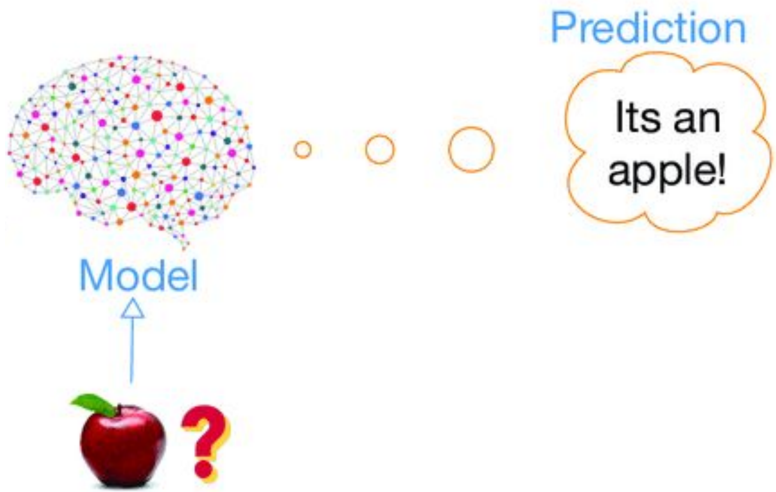
April 22, 2021

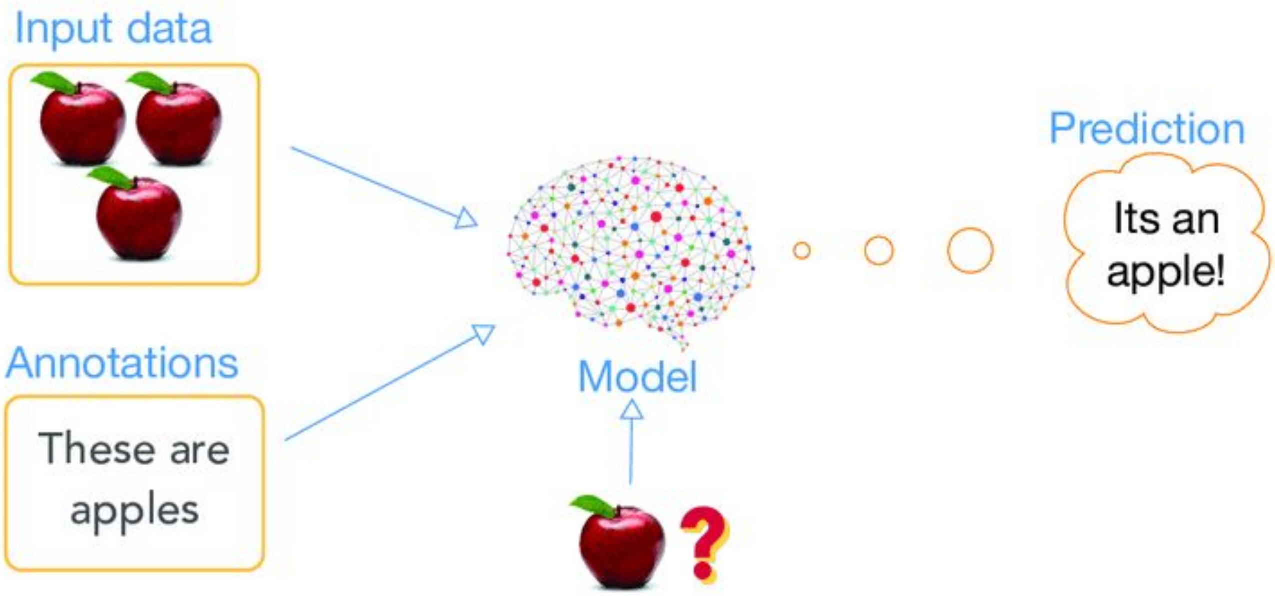


COPERNICUS MASTER IN DIGITAL EARTH

With the support of the Erasmus+ Programme of the European Union

Standard Supervised Learning Problem


Prediction
Its an
apple!
Model

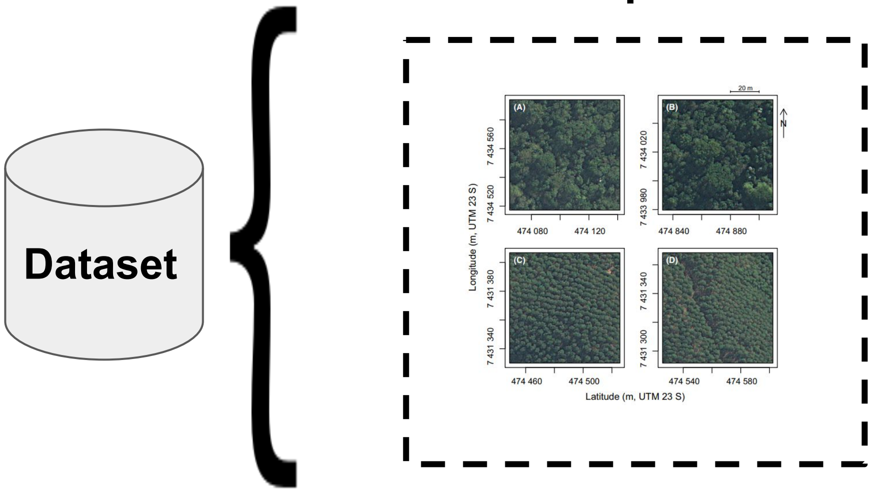

Input data
Annotations
These are apples
Model
Prediction
Its an apple!

How we can map forest disturbance in the Atlantic rainforest?

Fabien H. Wagne, et. al, 2019

How we can map forest disturbance in the Atlantic rainforest?

input

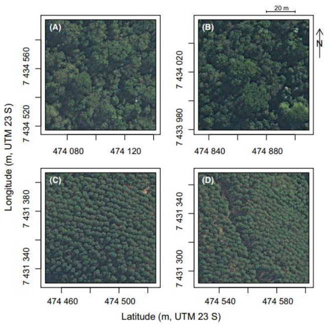


[Fabien H. Wagne, et. al, 2019]

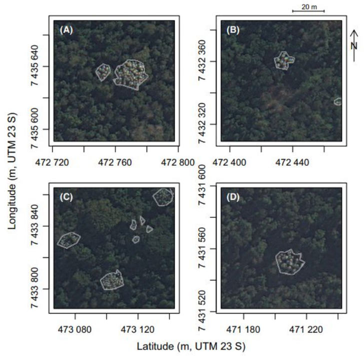
How we can map forest disturbance in the Atlantic rainforest?

Dataset

input



target

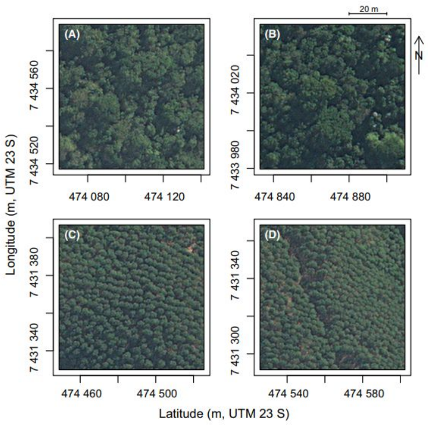


[Fabien H. Wagne, et. al, 2019]

Cesar Luis Aybar Camacho - csaybar@gmail.com
University of Salzburg | Department of Geoinformatics - Z_GIS

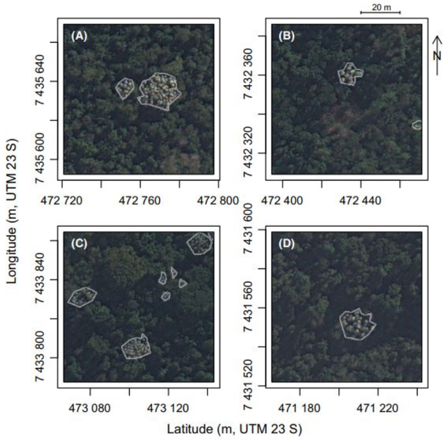
How we can map forest disturbance in the Atlantic rainforest?

input

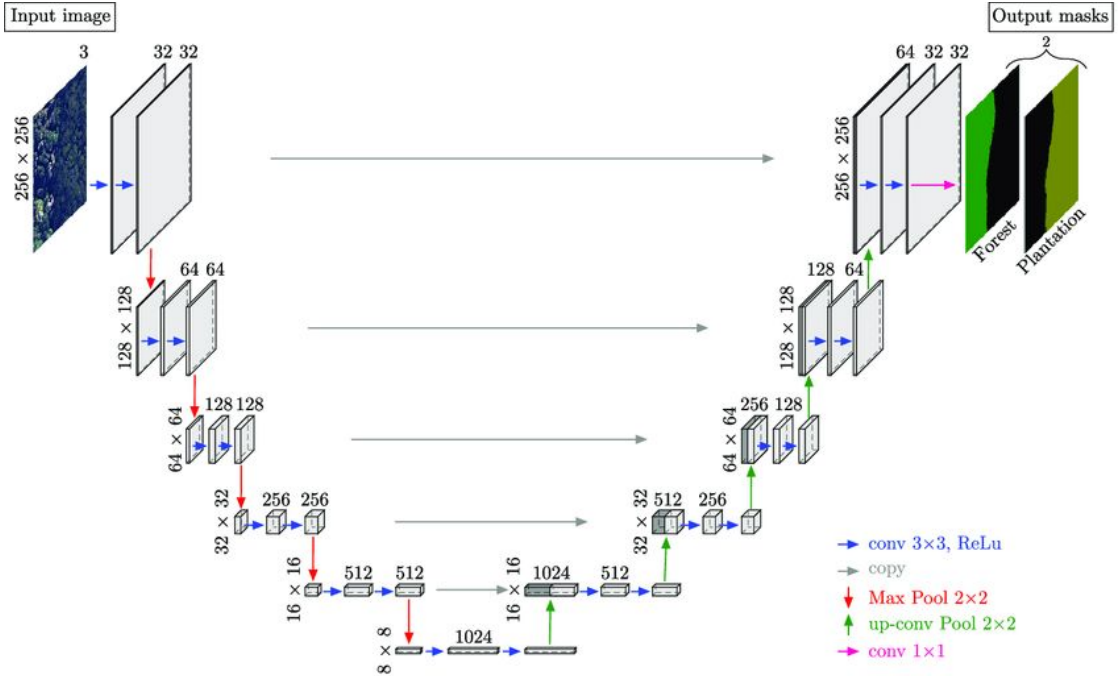


Black Box Model

target

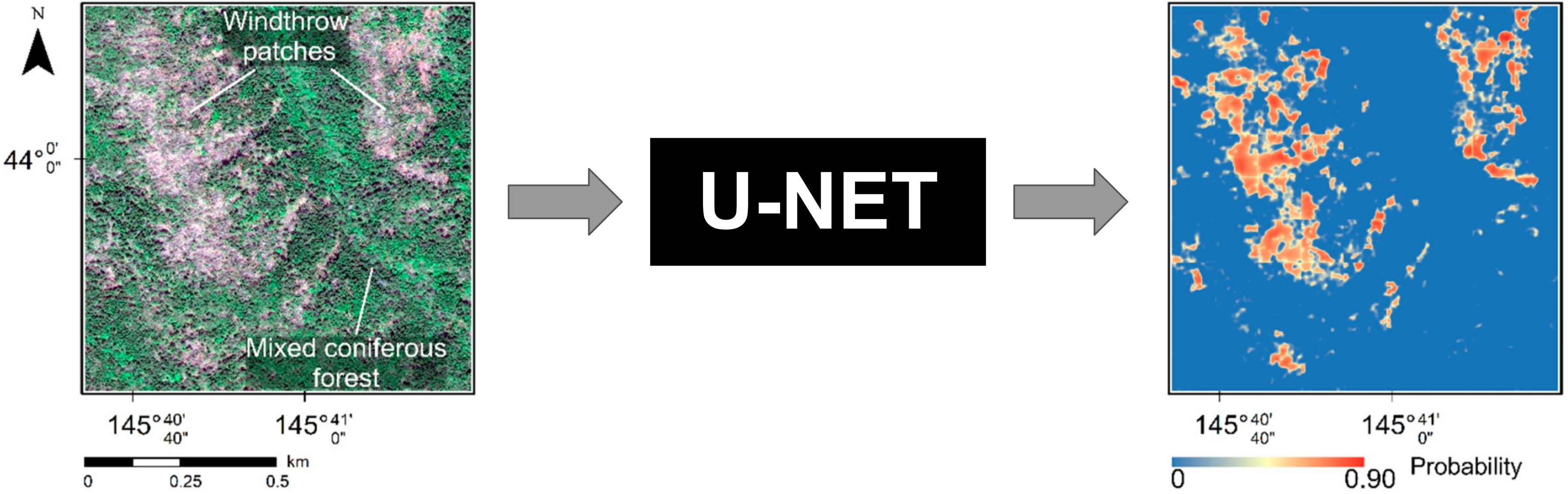


[Fabien H. Wagne, et. al, 2019]



U-NET

Fabien H. Wagne, et. al, 2019



[Fabien H. Wagne, et. al, 2019]

Shortcomings

- Large dataset are not always available.

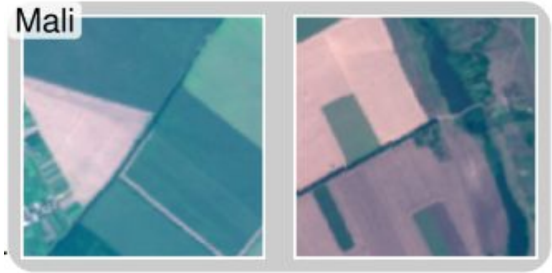
Shortcomings

- Large dataset are not always available.
- Need retrain every time new data is added.

Shortcomings

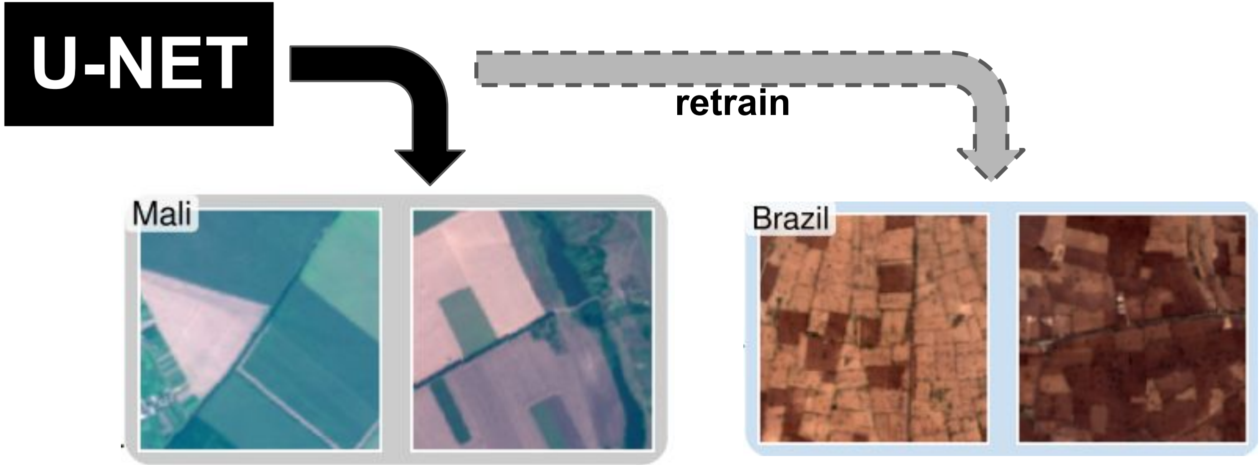
- Large dataset are not always available.
- Need retrain every time new data is added.

U-NET



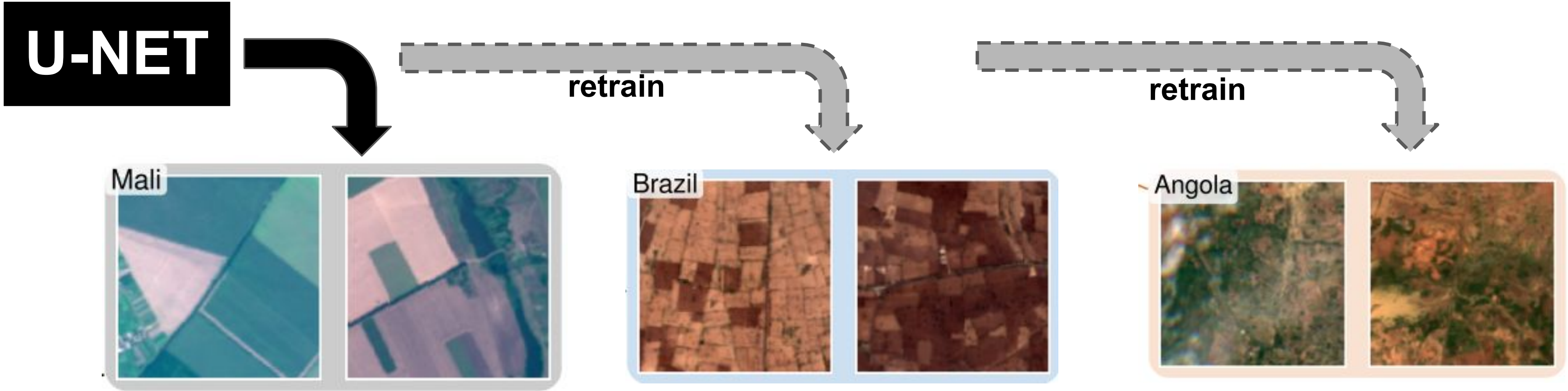
Shortcomings

- Large dataset are not always available.
- Need retrain every time new data is added.



Shortcomings

- Large dataset are not always available.
- Need retrain every time new data is added.

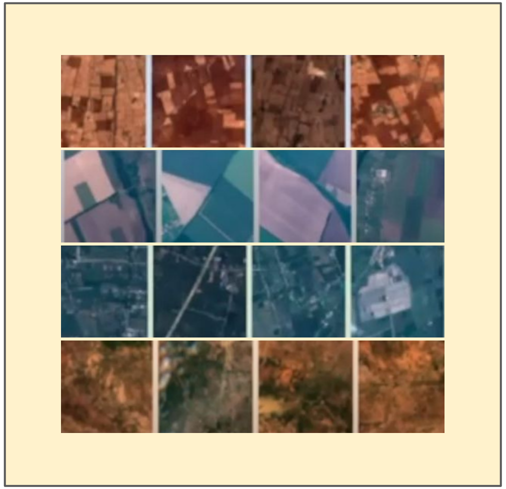


Cesar Luis Aybar Camacho - csaybar@gmail.com

University of Salzburg | Department of Geoinformatics - Z_GIS

One model per region

One model on pooled data

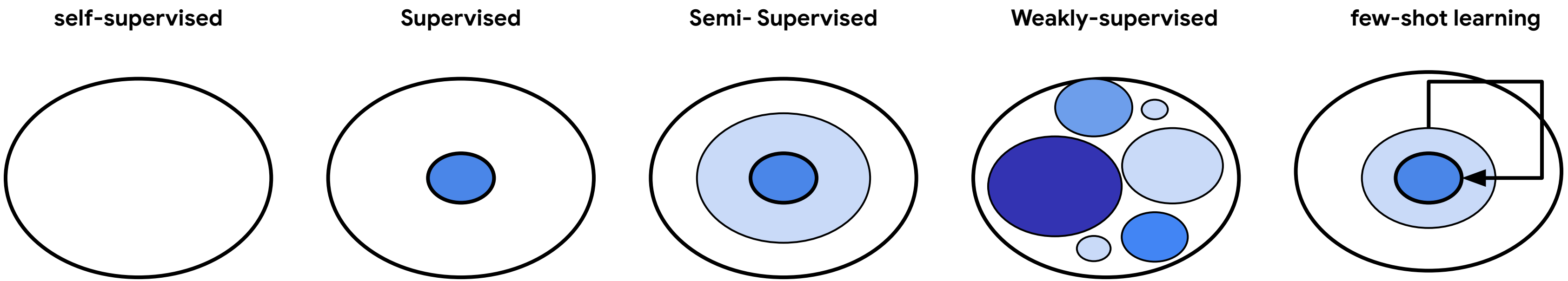


Cesar Luis Aybar Camacho - csaybar@gmail.com

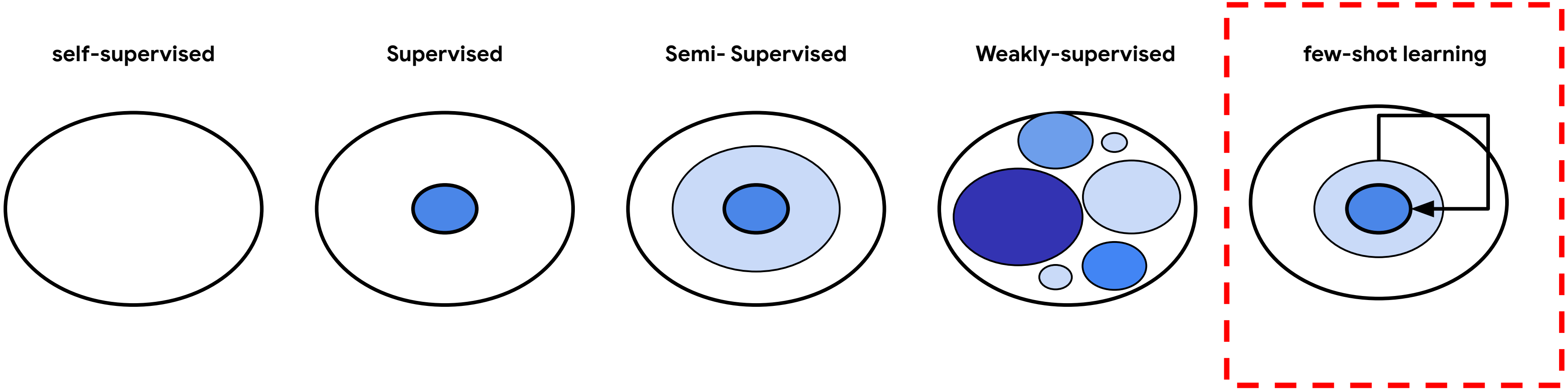
University of Salzburg | Department of Geoinformatics - Z_GIS

Beyond Supervised Learning

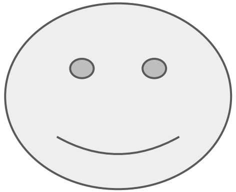
Semantic segmentation approaches:



Semantic segmentation approaches:

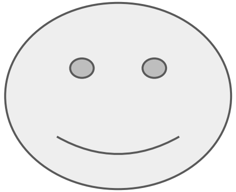


Few-shot Learning 🔥

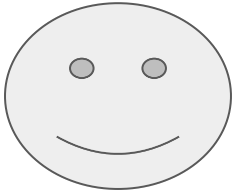


Tom

What is her name?



Tom



Tom

What is her name?

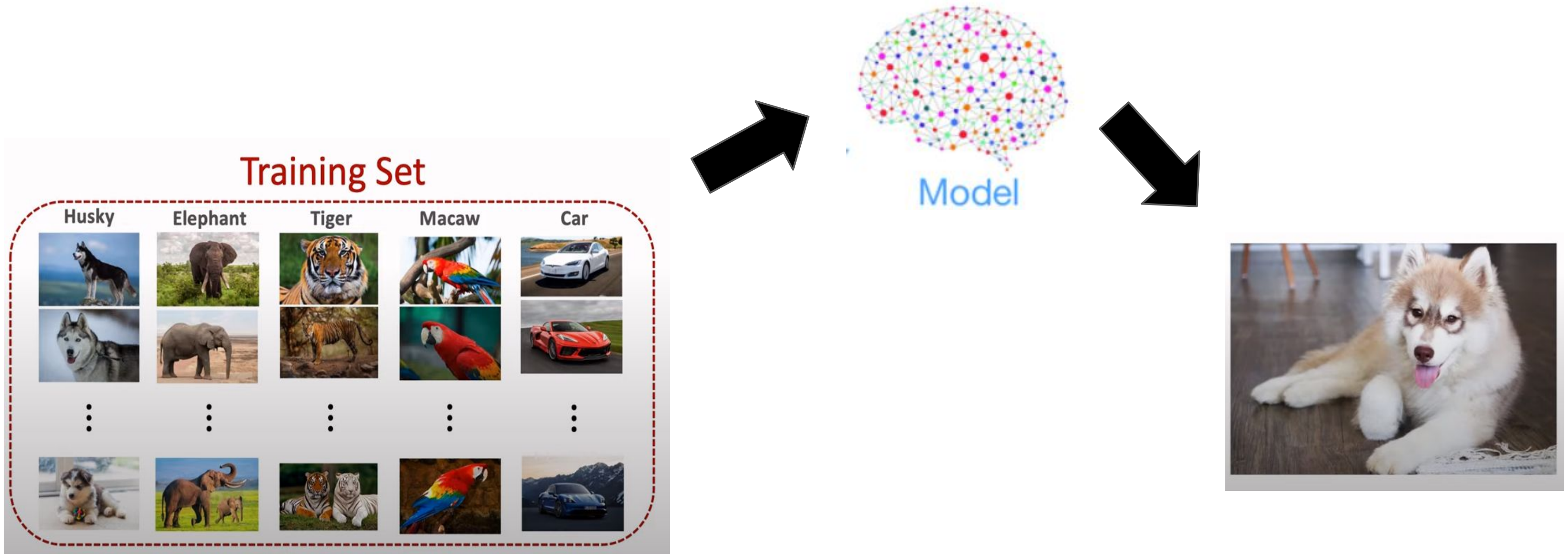
Tom

What is her name?

Support Dataset *Sn*

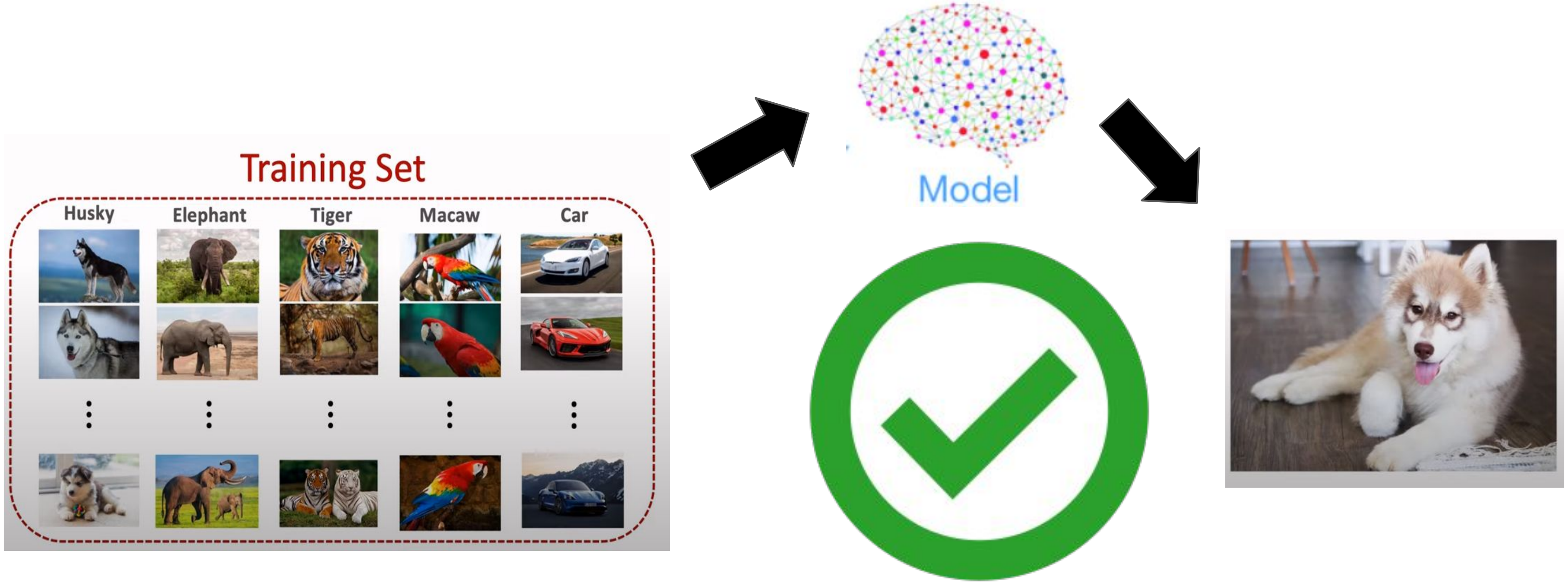
Query Dataset *Qn*

Standard SL



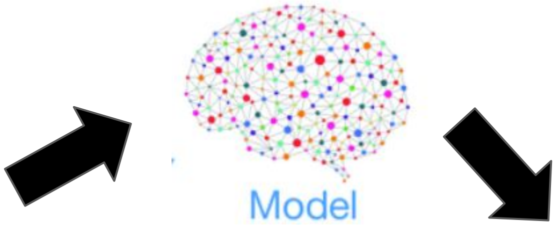
[Shusen Wang video]

Standard SL



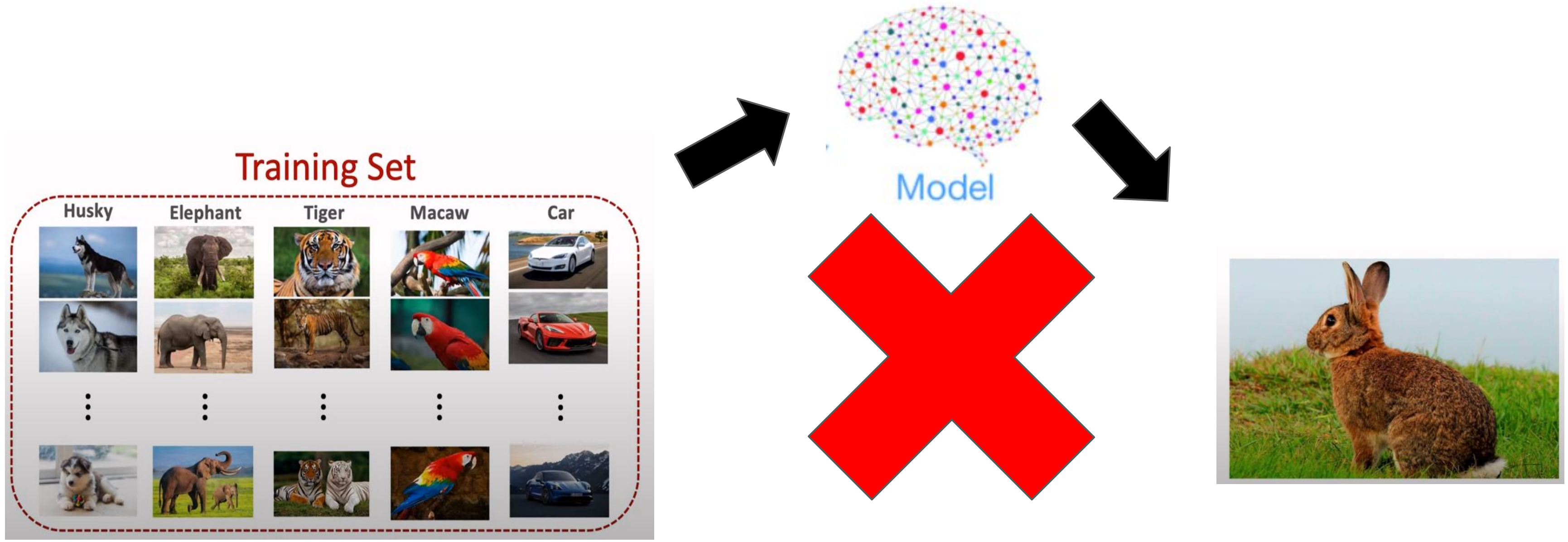
Shusen Wang video

Standard SL



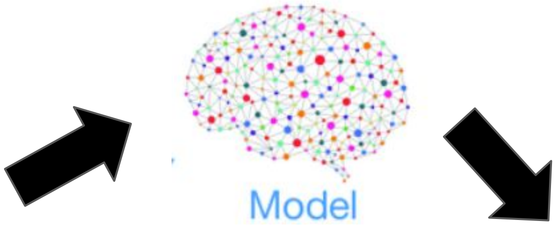
[Shusen Wang video]

Standard SL



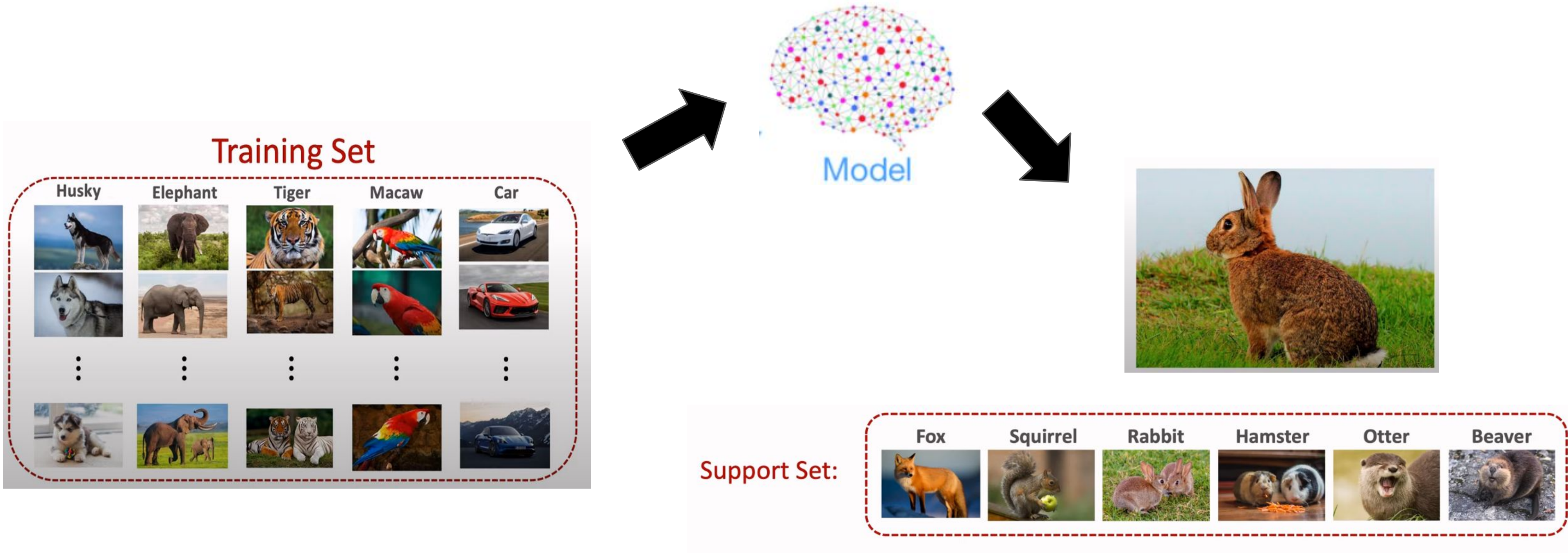
[Shusen Wang video]

Few Shot L



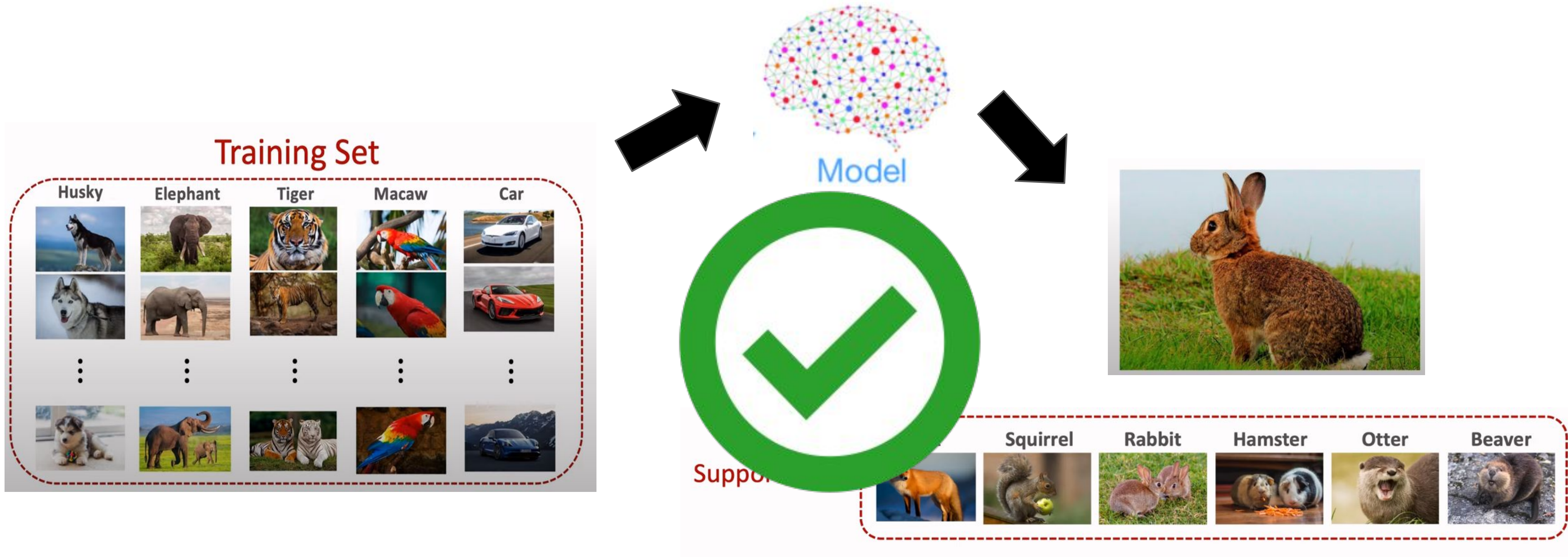
Shusen Wang video

Few Shot L



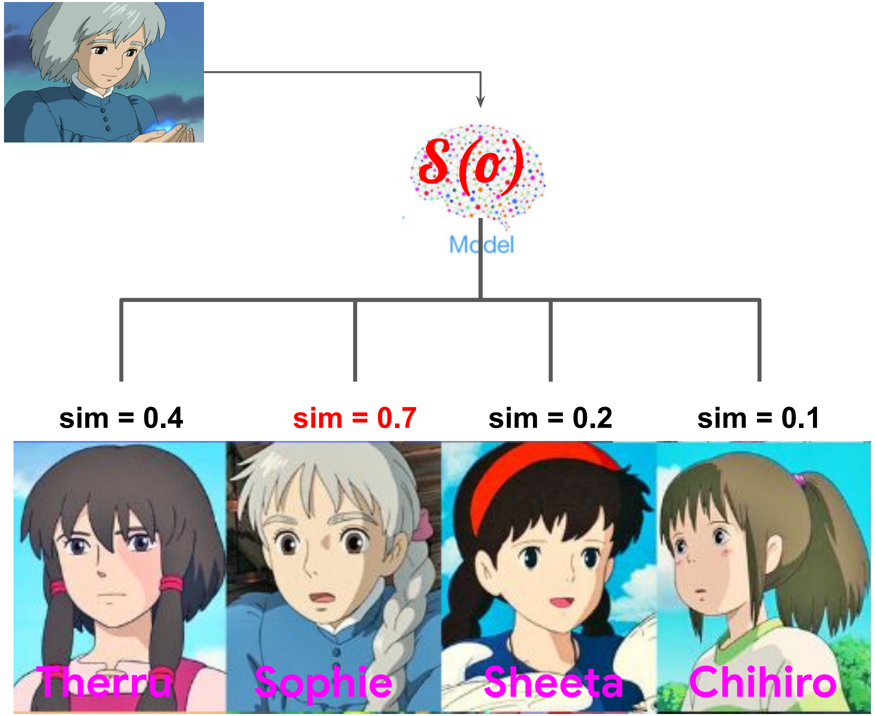
[Shusen Wang video]

Few Shot L



[Shusen Wang video]

What is her name?



Basic Idea

1. Divide the **dataset** in **query** and **support**.
2. Learn a **similarity function**.
3. Apply the **similarity function** to the predictions.

Few shot Learning approaches:

- Feature Transfer: Standard transfer learning, Baseline++ (Chen et al. 2019), Simpleshot (Wang et al. 2019), etc.
- Metric Learning: Matching Networks (Vinyals et al. 2016), Prototypical Networks (Snell et al. 2017), Relation Networks (Sunget et al. 2018), etc.
- **Meta-learning: Model-Agnostic Meta-Learning (MAML, Finn et al. 2017), MAML ++ (Antoniou et al. 2019), Meta-SGD (Li et al. 2017), etc.**
- Bayesian methods: Bayesian MAML (Yoon et al. 2018), VERSA (Gordon et al. 2019), ALPaCA (Harrison et al. 2018), etc.

Bayesian Meta-Learning for the Few-Shot Setting via Deep Kernels - Massimiliano Patacchiola 2020 - NeurIPS 2020

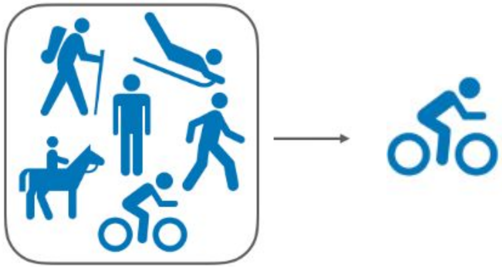
MetaLearning + Few-Shot Learning

In meta-learning models learn how to learn!

Single Task (from scratch)

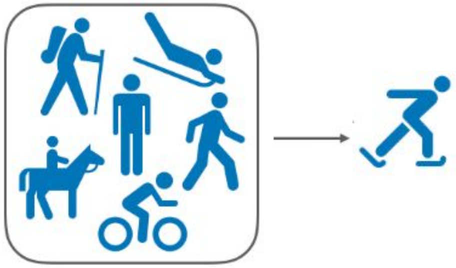
Learn task and perform task

Pretraining and fine-tuning



Refresh task of interest

meta-learning



Quickly learn a new task

[Bayesian Meta-Learning for the Few-Shot Setting via Deep Kernels - Massimiliano Patacchiola 2020 - NeurIPS 2020]

Model-Agnostic Meta-Learning

MAML

Agnostic, in the sense that the method can be used in different contexts, few-shot learning is a particular case.

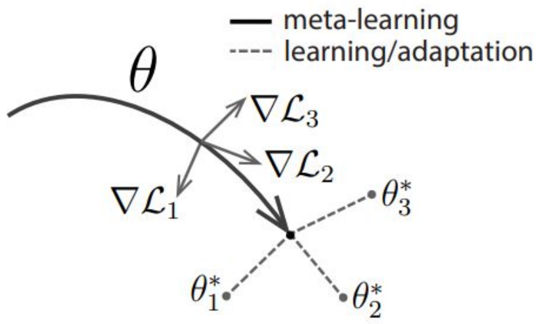
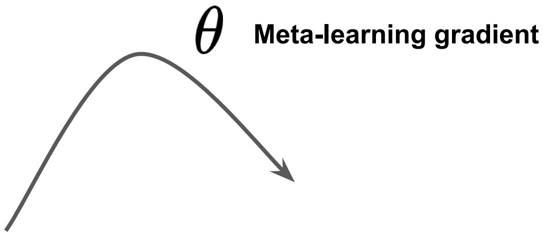


Figure 1. Diagram of our model-agnostic meta-learning algorithm (MAML), which optimizes for a representation θ that can quickly adapt to new tasks.

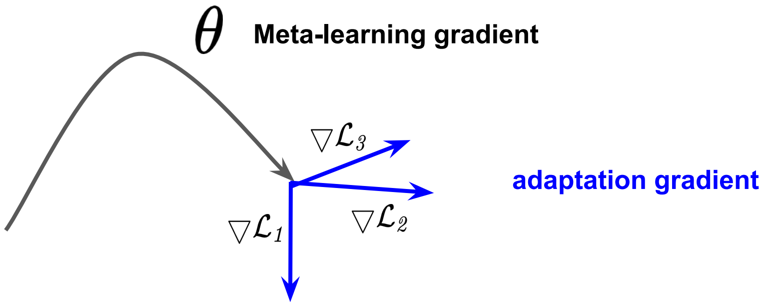
MAML Intuition

$$\min_{\theta}\ \theta - \alpha \bigtriangledown_{\theta} \mathcal{L}(\theta, \mathcal{D})$$

MAML Intuition



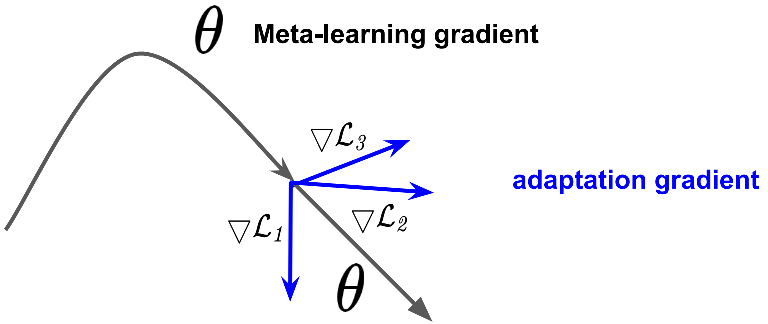
MAML Intuition



Cesar Luis Aybar Camacho - csaybar@gmail.com

University of Salzburg | Department of Geoinformatics - Z_GIS

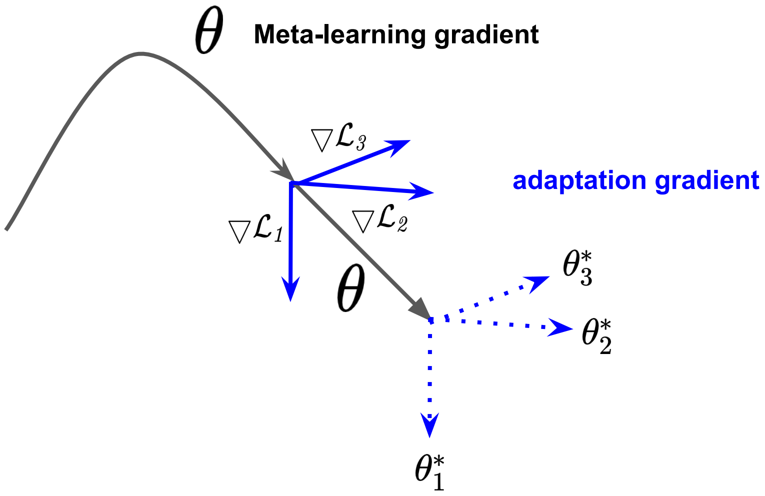
MAML Intuition



Cesar Luis Aybar Camacho - csaybar@gmail.com

University of Salzburg | Department of Geoinformatics - Z_GIS

MAML Intuition



Cesar Luis Aybar Camacho - csaybar@gmail.com

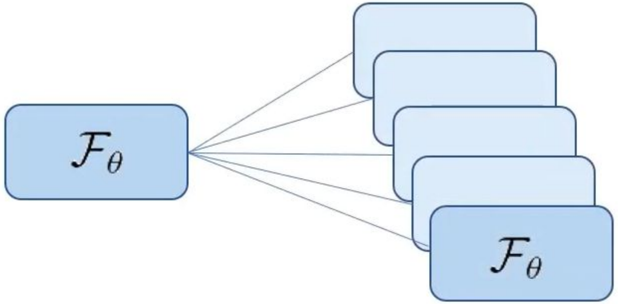
University of Salzburg | Department of Geoinformatics - Z_GIS

MAML Intuition

$$\boxed{\mathcal{F}_\theta}$$

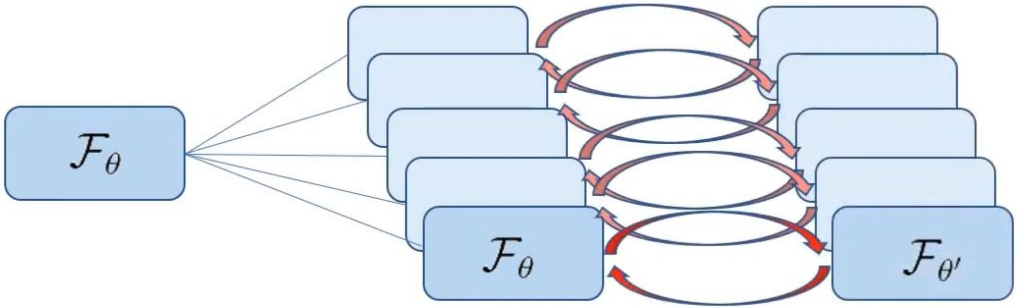
MAML Intuition

1. **Copy Model per task**



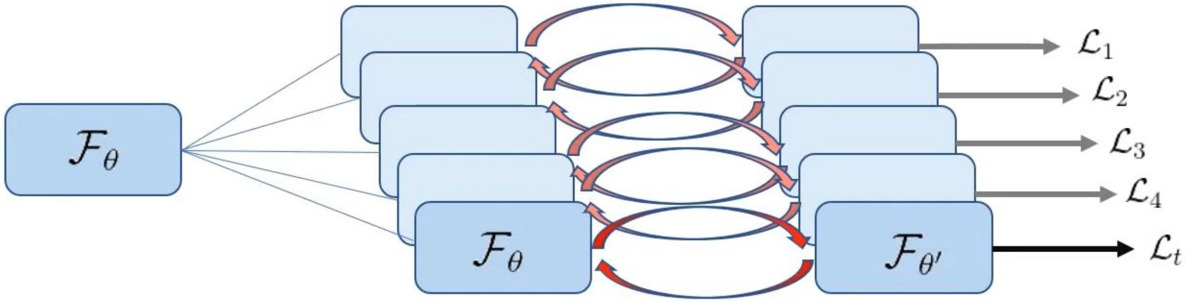
MAML Intuition

1. Copy Model per task **2. Support set train**



MAML Intuition

1. Copy Model per task **2. Support set train** **3. Calculate query set loss**

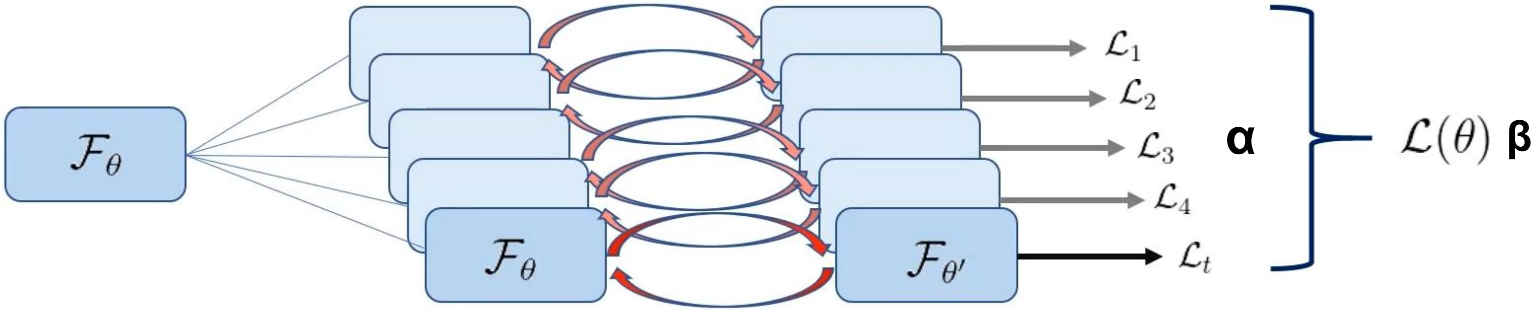


Cesar Luis Aybar Camacho - csaybar@gmail.com

University of Salzburg | Department of Geoinformatics - Z_GIS

MAML Intuition

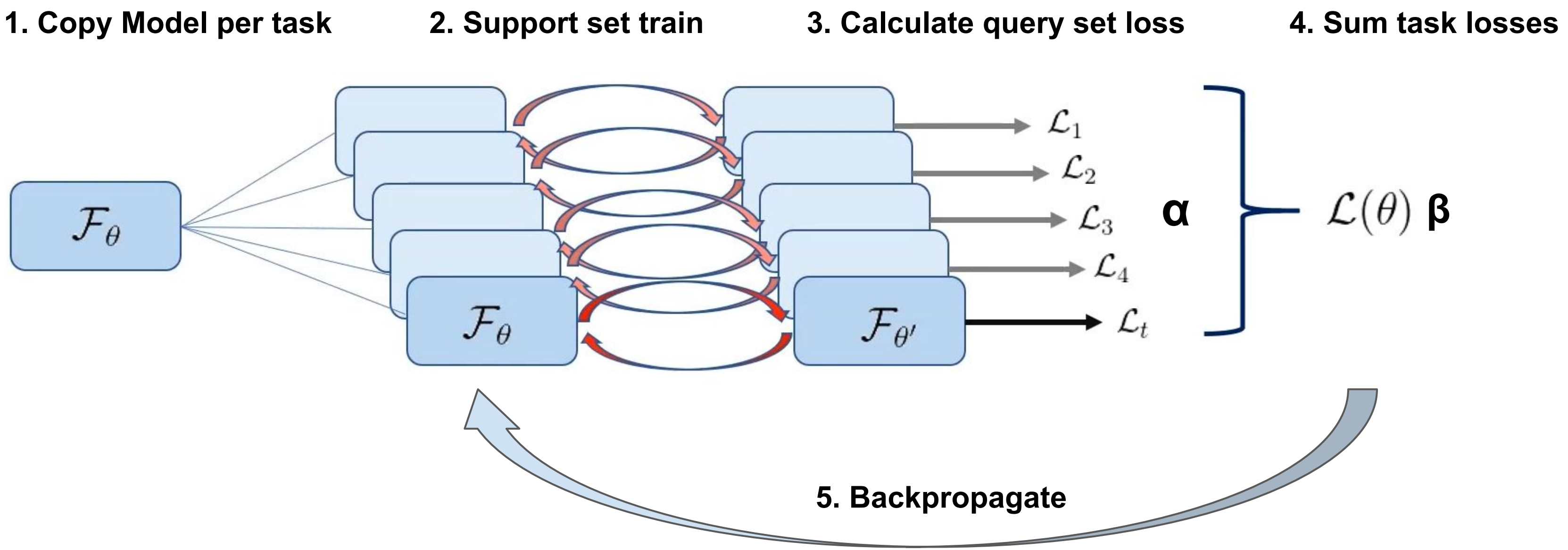
1. Copy Model per task **2. Support set train** **3. Calculate query set loss** **4. Sum task losses**



Cesar Luis Aybar Camacho - csaybar@gmail.com

University of Salzburg | Department of Geoinformatics - Z_GIS

MAML Intuition



MAML Gradient Descent

$$\min_{\theta}\ \theta - \alpha \bigtriangledown_{\theta} \mathcal{L}(\theta, \mathcal{D})$$

MAML Gradient Descent

Loss function

$$\min_{\theta} \theta - \alpha \nabla_{\theta} \mathcal{L}(\theta, \mathcal{D})$$

Data Points

Updated parameter

step size hyperparameter

MAML Gradient Descent

$$\min_{\theta} \sum_{\tau_i \sim p(\tau)} \mathcal{L}\big(\theta - \alpha \bigtriangledown_{\theta} \mathcal{L}\big(\theta,\, \mathcal{D}^{\mathcal{S}}_{\tau_i}\big), \mathcal{D}^{\mathcal{Q}}_{\tau_i}\big) = \sum_{\tau_i \sim p(\tau)} \mathcal{L}\big(\theta',\, \mathcal{D}^{\mathcal{Q}}_{\tau_i}\big)$$

MAML Gradient Descent

Loss function MetaLearning

Updated Parameters

$$\min_{\theta} \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}\left(\boxed{\theta - \alpha \bigtriangledown_\theta \mathcal{L}(\theta, \boxed{\mathcal{D}^{\mathcal{S}}_{\mathcal{T}_i}})}, \mathcal{D}^{\mathcal{Q}}_{\mathcal{T}_i}\right) = \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}(\theta', \boxed{\mathcal{D}^{\mathcal{Q}}_{\mathcal{T}_i}})$$

MetaModel parameter

Task

Gradient Descent Adaptation

Support Set

Query Set

Model-Agnostic Meta-Learning

```python
model = ConvolutionalNeuralNetwork(out_features=5) #we suppose a 5-way setting
meta_optimizer = torch.optim.Adam(model.parameters(), lr=1e-3)

#[...] outer loop starts here, sample batch of tasks

for task in batch:
    train_inputs, train_targets = task['support'] #input-output train pairs
    test_inputs, test_targets = task['query'] #input-output test pairs

    train_logit = model(train_input)
    inner_loss = F.cross_entropy(train_logit, train_target) #on train set
    model.zero_grad()
    grads = torch.autograd.grad(inner_loss, model.meta_params(), create_graph=True)
    params = OrderedDict()
    for (name, param), grad in zip(model.meta_named_pars(), grads):
        params[name] = param - step_size * grad
    test_logit = model(test_input, params=params) #assign params to model
    #Notice the `+=` which is used to accumulate the loss for each task
    outer_loss += F.cross_entropy(test_logit, test_target) #on test set

outer_loss.backward()
meta_optimizer.step()
```

Algorithm 2 MAML for Few-Shot Supervised Learning

Require: $p(\mathcal{T})$: distribution over tasks
Require: α, β: step size hyperparameters

1: randomly initialize θ
2: **while** not done **do**
3: Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
4: **for all** $\mathcal{T}_i$ **do**
5: Sample K datapoints $\mathcal{D} = \{\mathbf{x}^{(j)}, \mathbf{y}^{(j)}\}$ from $\mathcal{T}_i$
6: Evaluate $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$ using $\mathcal{D}$ and $\mathcal{L}_{\mathcal{T}_i}$ in Equation (2) or (3)
7: Compute adapted parameters with gradient descent: $\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$
8: Sample datapoints $\mathcal{D}_i' = \{\mathbf{x}^{(j)}, \mathbf{y}^{(j)}\}$ from $\mathcal{T}_i$ for the meta-update
9: **end for**
10: Update $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'})$ using each $\mathcal{D}_i'$ and $\mathcal{L}_{\mathcal{T}_i}$ in Equation 2 or 3
11: **end while**

Cesar Luis Aybar Camacho - csaybar@gmail.com
University of Salzburg | Department of Geoinformatics - Z_GIS

Model-Agnostic Meta-Learning

```python
model = ConvolutionalNeuralNetwork(out_features=5) #we suppose a 5-way setting
meta_optimizer = torch.optim.Adam(model.parameters(), lr=1e-3)

#[...] outer loop starts here, sample batch of tasks

for task in batch:
    train_inputs, train_targets = task['support'] #input-output train pairs
    test_inputs, test_targets = task['query'] #input-output test pairs

    train_logit = model(train_input)
    inner_loss = F.cross_entropy(train_logit, train_target) #on train set
    model.zero_grad()
    grads = torch.autograd.grad(inner_loss, model.meta_params(), create_graph=True)
    params = OrderedDict()
    for (name, param), grad in zip(model.meta_named_pars(), grads):
        params[name] = param - step_size * grad
    test_logit = model(test_input, params=params) #assign params to model
    #Notice the `+=` which is used to accumulate the loss for each task
    outer_loss += F.cross_entropy(test_logit, test_target) #on test set

outer_loss.backward()
meta_optimizer.step()
```

Algorithm 2 MAML for Few-Shot Supervised Learning

Require: $p(\mathcal{T})$: distribution over tasks
Require: α, β: step size hyperparameters
1: randomly initialize θ
2: **while** not done **do**
3: Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
4: **for all** $\mathcal{T}_i$ **do**
5: Sample K datapoints $\mathcal{D} = \{\mathbf{x}^{(j)}, \mathbf{y}^{(j)}\}$ from $\mathcal{T}_i$
6: Evaluate $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$ using $\mathcal{D}$ and $\mathcal{L}_{\mathcal{T}_i}$ in Equation (2) or (3)
7: Compute adapted parameters with gradient descent: $\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$
8: Sample datapoints $\mathcal{D}_i' = \{\mathbf{x}^{(j)}, \mathbf{y}^{(j)}\}$ from $\mathcal{T}_i$ for the meta-update
9: **end for**
10: Update $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'})$ using each $\mathcal{D}_i'$ and $\mathcal{L}_{\mathcal{T}_i}$ in Equation 2 or 3
11: **end while**

MAML pro vs cons

- Elegant and neat.
- Fully differentiable method
- Agnostic (easily adapted to multiple setting).

- Unstable, hard to train.
- High order derivatives.
- Vanishing gradient.

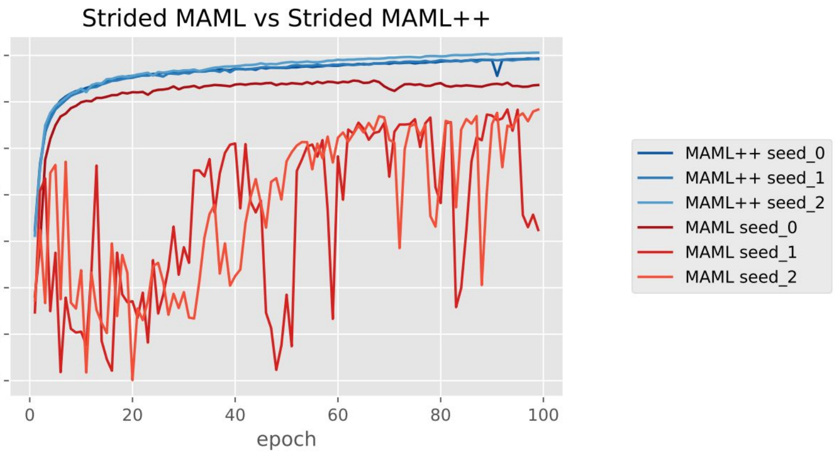
Published as a conference paper at ICLR 2019

How to train your MAML

Antreas Antoniou
University of Edinburgh
{a.antoniou}@sms.ed.ac.uk

Harrison Edwards
OpenAI, University of Edinburgh
{h.l.edwards}@sms.ed.ac.uk

Amos Storkey
University of Edinburgh
{a.storkey}@ed.ac.uk



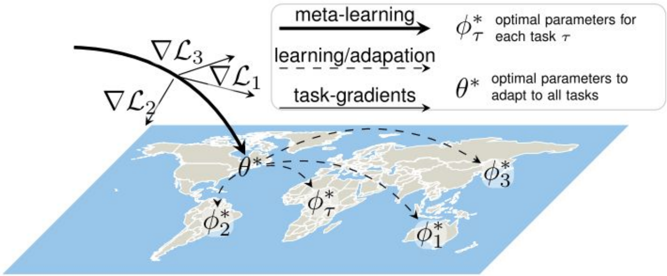
- https://paperswithcode.com/sota/few-shot-image-classification-on-mini-2
- https://paperswithcode.com/sota/few-shot-semantic-segmentation-on-fss-1000

Earth Observation?

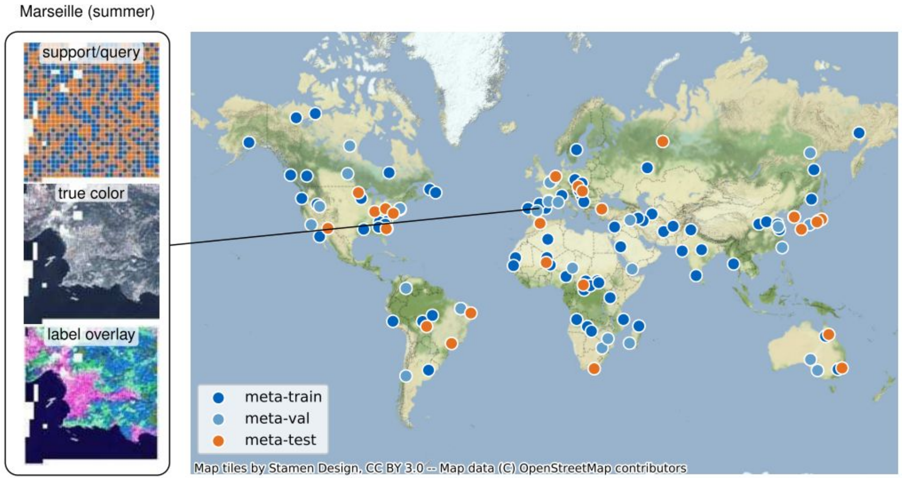
Meta-Learning for Few-Shot Land Cover Classification

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

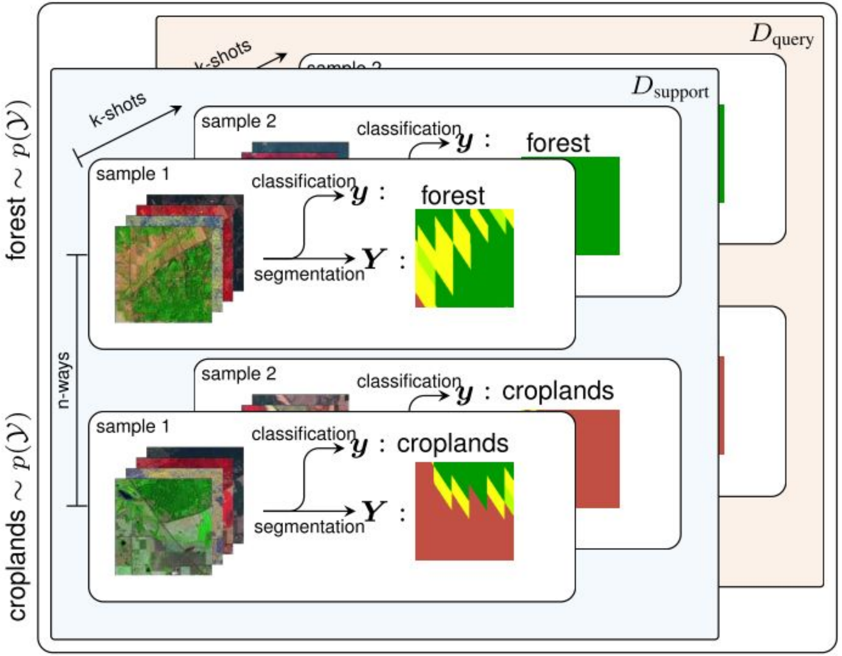


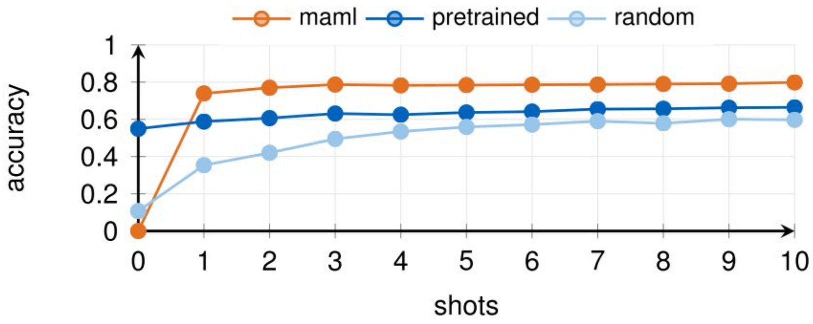
SEN12MS



- “Global” dataset
- Sentinel2 + MODIS
- 125 image tiles

Geographic regions as meta-learning task



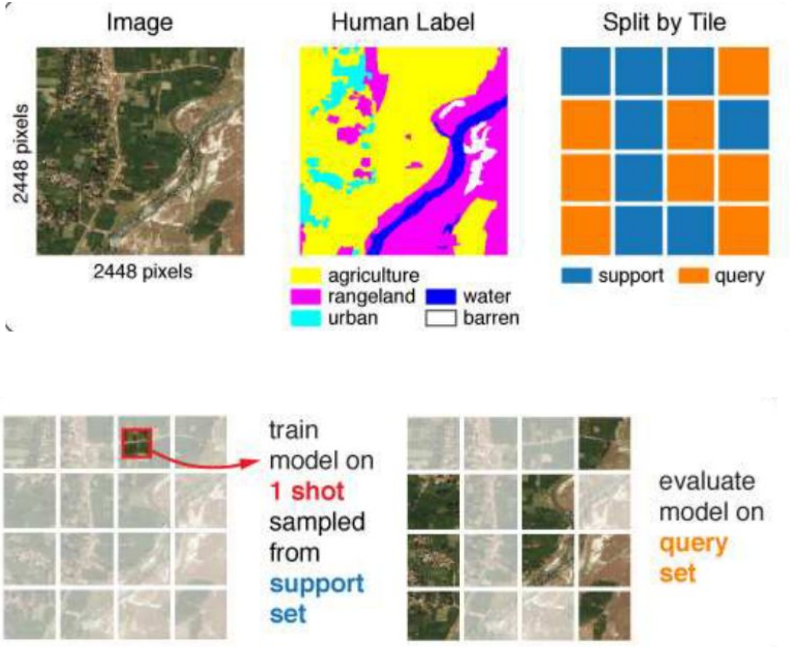


- **MAML adjusts** to new distribution in a single shot and outperforms baselines.

Cesar Luis Aybar Camacho - csaybar@gmail.com

University of Salzburg | Department of Geoinformatics - Z_GIS

DeepGloble

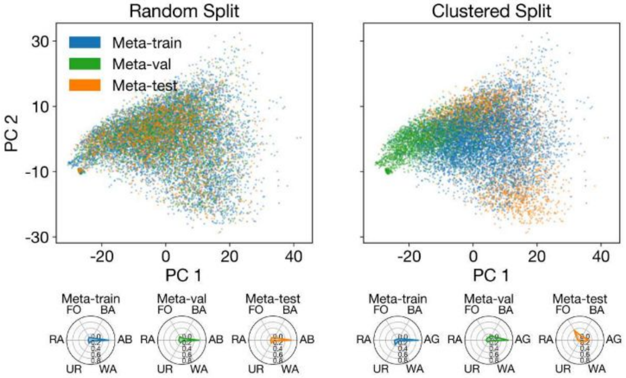


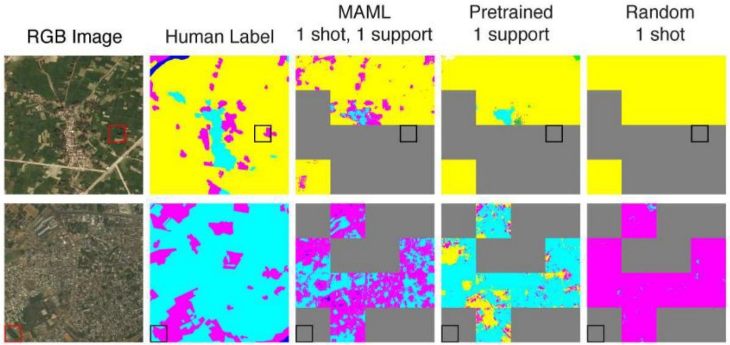
- RGB
- High Resolution (0.5 m)
- Semantic segmentation

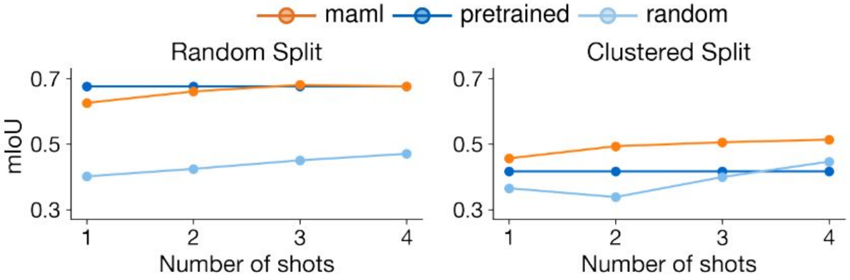
When:

$$P_{train}(X, y) \neq P_{test}(X, y)$$

MAML outperforms pretraining



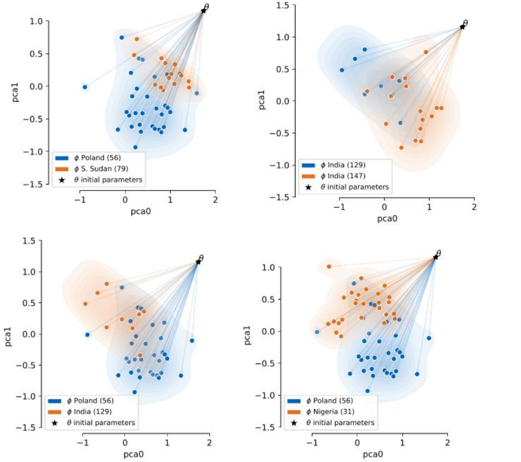


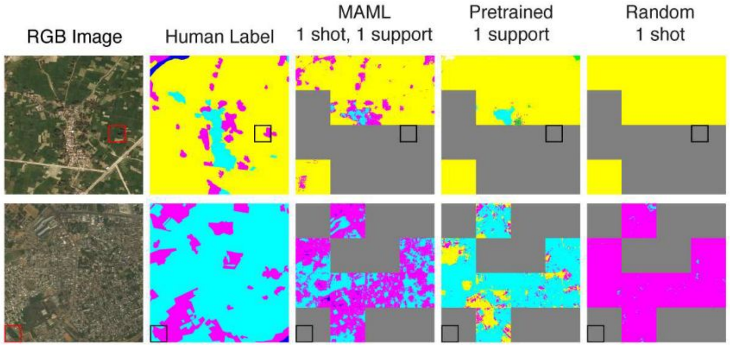


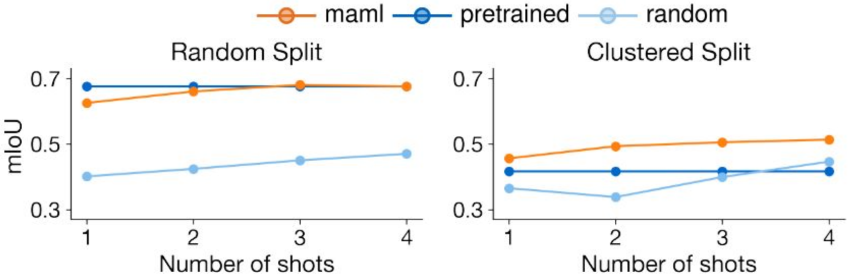
When:

$$P_{train}(X, y) \neq P_{test}(X, y)$$

MAML outperforms pretraining







Cesar Luis Aybar Camacho - csaybar@gmail.com

University of Salzburg | Department of Geoinformatics - Z_GIS

Conclusion

Conclusions

- Results in computer vision paper show us that **meta-learning outperforms** pretraining and fine-tuning when the **meta-task tasks have data distribution that are different from meta-train tasks**.
- Current EO Deep Learning Dataset are a limitation.
- meta-learning framework can lead deep learning in Earth observation to a new era.

Muchas Gracias