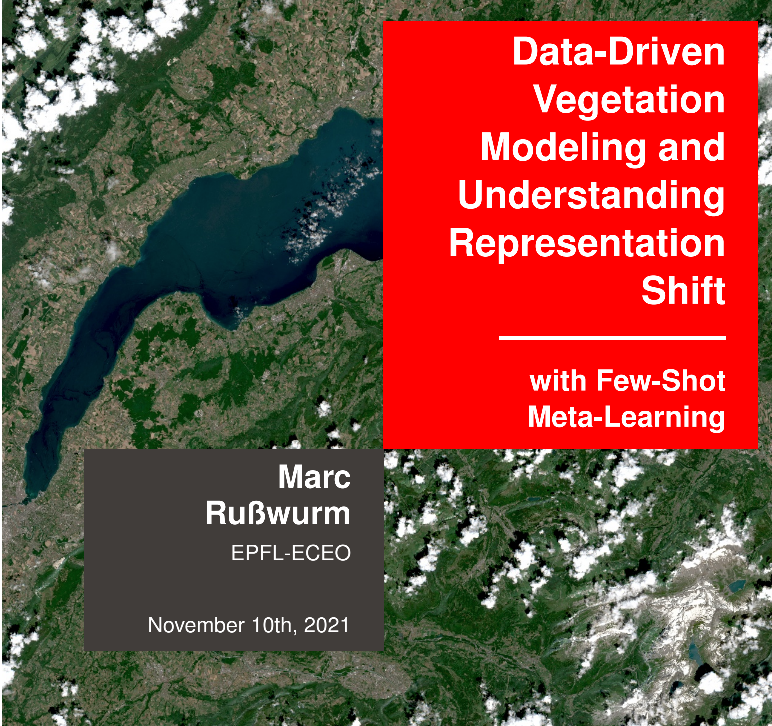
EPFL



Data-Driven Vegetation Modeling and Understanding Representation Shift

with Few-Shot Meta-Learning

Marc Rußwurm
EPFL-ECEO

November 10th, 2021

École polytechnique fédérale de Lausanne

Why do we need Vegetation Modeling?

Zero Hunger — estimating the **crop yield** & predict **food prices**, shortages.

Climate Action — by monitoring **vegetation changes** over time (desertification).

Life on Land — human life depends on **agriculture**, animal life on **biotopes**.

Why do we need Vegetation Modeling?

Zero Hunger	estimating the **crop yield** & predict **food prices**, shortages.
Climate Action	by monitoring **vegetation changes** over time (desertification).
Life on Land	human life depends on **agriculture**, animal life on **biotopes**.

Why do we need Vegetation Modeling?

Zero Hunger	estimating the **crop yield** & predict **food prices**, shortages.
Climate Action	by monitoring **vegetation changes** over time (desertification).
Life on Land	human life depends on **agriculture**, animal life on **biotopes**.

Why do we need Vegetation Modeling?

Zero Hunger	estimating the **crop yield** & predict **food prices**, shortages.
Climate Action	by monitoring **vegetation changes** over time (desertification).
Life on Land	human life depends on **agriculture**, animal life on **biotopes**.

Crop Type Mapping:

Satellite Time Series Data $\rightarrow$ Crop Type Label (e.g. corn, meadow)

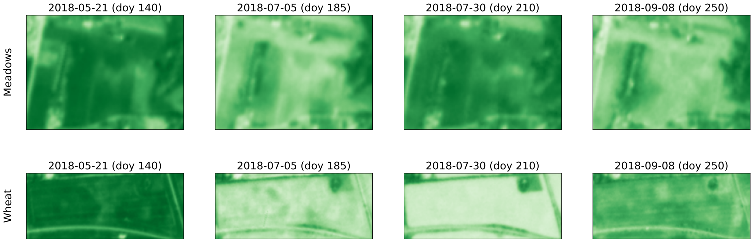
Why do we need Vegetation Modeling?

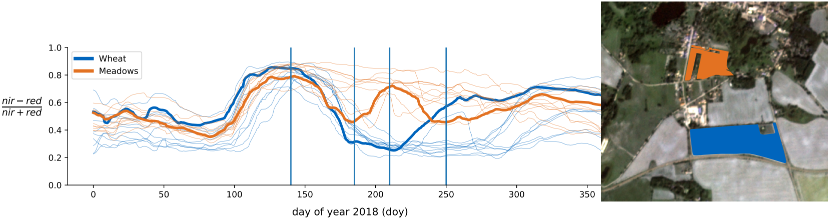
Zero Hunger	estimating the **crop yield** & predict **food prices**, shortages.
Climate Action	by monitoring **vegetation changes** over time (desertification).
Life on Land	human life depends on **agriculture**, animal life on **biotopes**.

Later: Land Cover Classification

Satellite Image $\rightarrow$ Land Cover Label (e.g. urban, forest)

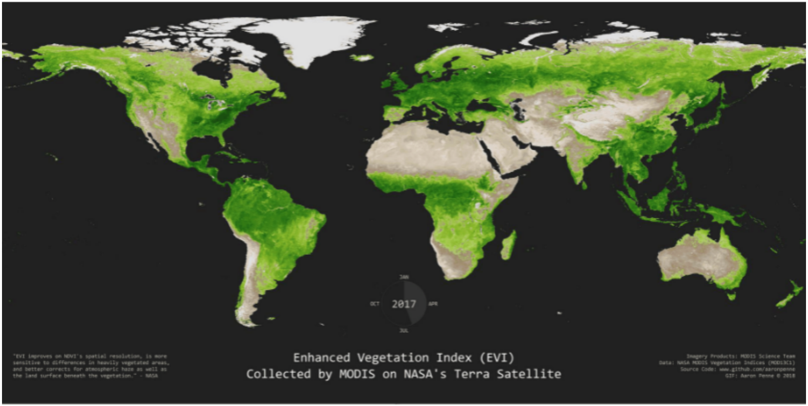
Satellite Time Series → Crop Type





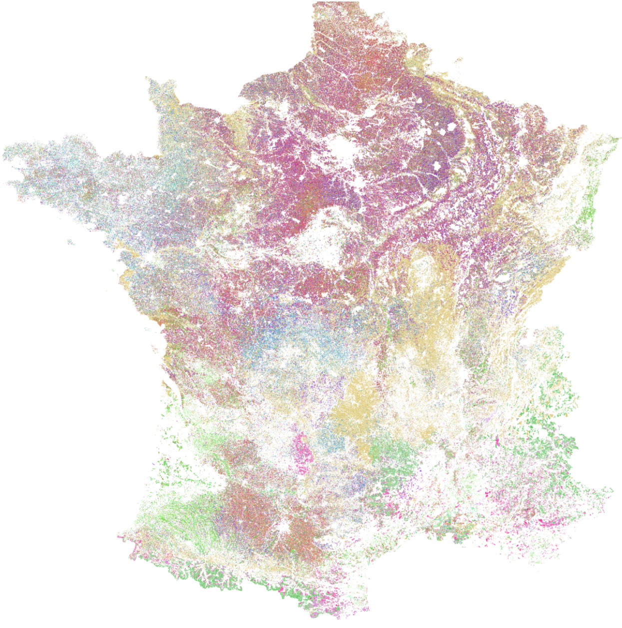
Kondmann, et al., ”DENETHOR: The DynamicEarthNET dataset for Harmonized, inter-Operable, analysis-Ready, daily crop monitoring from space.”. NeurIPS Data Track (2021).

Satellite Time Series Data in Abundance

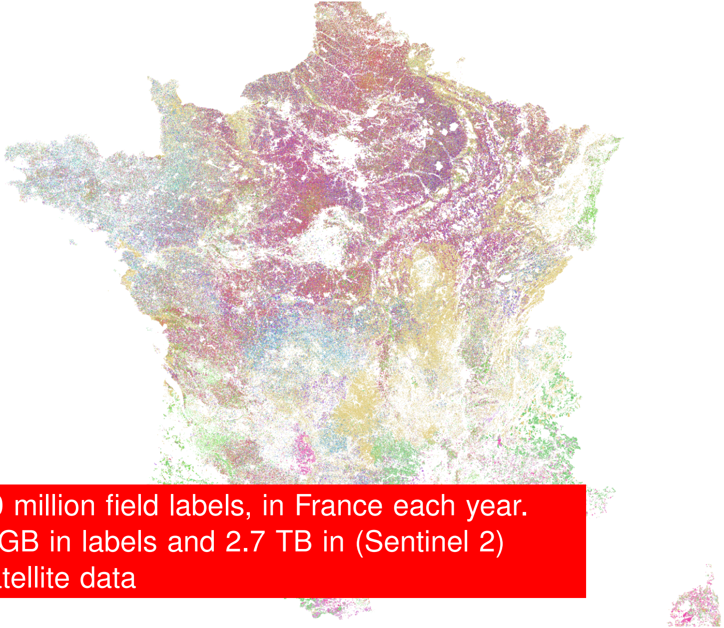


visualization Aaron Penne Github

Label Data in Abundance



Label Data in Abundance



10 million field labels, in France each year.
3 GB in labels and 2.7 TB in (Sentinel 2) satellite data

Problem Solved?

Problem Solved?

Plenty of input data!

Problem Solved?

Plenty of input data!

Plenty of labelled data!

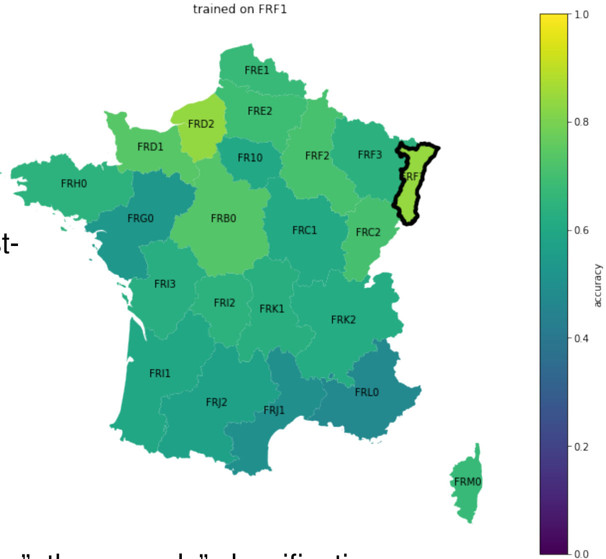
Problem Solved?

Plenty of input data!

Plenty of labelled data!

Let’s train a Deep Learning Model!

The ”Data Dilemma”



training and testing on different regions

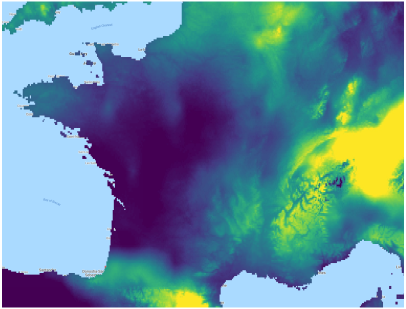
binary ”wheat” vs ”other cereals” classification

crop type time series classification with a simple 1D-CNN and 200 samples per class in each NUTS-2 region

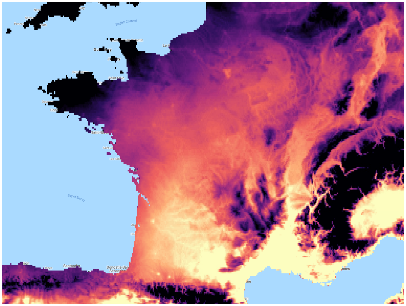
Environmental Conditions

Photosynthesis is a function of the environment.

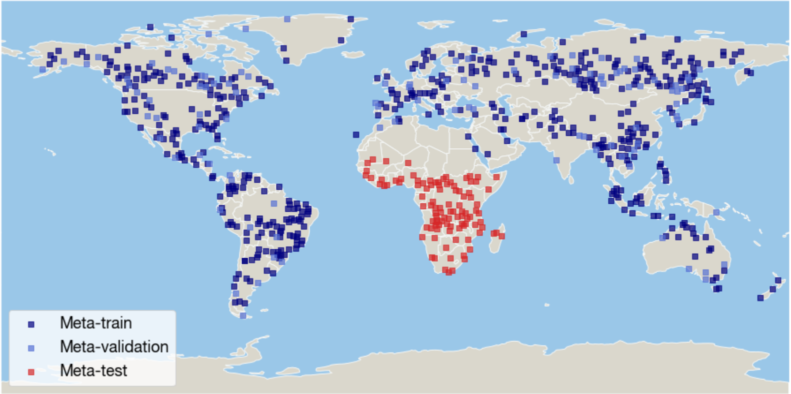
Precipitation 2018-08



Max temp (23-30℃) 2018-08



Apply models on regions of little data



How do we generate data?

Data Generation Model

domain

$$\mathcal{D} = \{\mathcal{X}, P(\mathcal{X})\}$$

Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. IEEE Transactions on knowledge and data engineering, 22(10), 1345-1359.
Qiang Yang; Yu Zhang, Wenyuan Dai; Sinno Jialin Pan (Editors) (2020). Transfer learning. Cambridge University Press. DOI 9781139061773
Shai Ben-David and Shai Shalev-Shwartz (2014). Understanding Machine Learning: From Theory to Algorithms

Data Generation Model

domain

$$\mathcal{D} = \{\mathcal{X}, P(\mathcal{X})\}$$

$$\downarrow$$

$$\mathbf{X} \sim P(\mathcal{X})$$

Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. IEEE Transactions on knowledge and data engineering, 22(10), 1345-1359.

Qiang Yang; Yu Zhang, Wenyuan Dai; Sinno Jialin Pan (Editors) (2020). Transfer learning. Cambridge University Press. DOI 9781139061773

Shai Ben-David and Shai Shalev-Shwartz (2014). Understanding Machine Learning: From Theory to Algorithms

Data Generation Model

domain

$$\mathcal{D} = \{\mathcal{X}, P(\mathcal{X})\}$$

task

$$\mathcal{T} = \{\mathcal{Y}, f(\cdot)\}$$

$$\downarrow$$

$$\mathbf{X} \sim P(\mathcal{X})$$

Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. IEEE Transactions on knowledge and data engineering, 22(10), 1345-1359.

Qiang Yang; Yu Zhang, Wenyuan Dai; Sinno Jialin Pan (Editors) (2020). Transfer learning. Cambridge University Press. DOI 9781139061773

Shai Ben-David and Shai Shalev-Shwartz (2014). Understanding Machine Learning: From Theory to Algorithms

Data Generation Model

domain

$$\mathcal{D} = \{\mathcal{X}, P(\mathcal{X})\}$$

task

$$\mathcal{T} = \{\mathcal{Y}, f(\cdot)\}$$

$$\mathbf{X} \sim P(\mathcal{X}) \xrightarrow{\text{predictive function f}} y = f(\mathbf{X})$$

Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. IEEE Transactions on knowledge and data engineering, 22(10), 1345-1359.

Qiang Yang; Yu Zhang, Wenyuan Dai; Sinno Jialin Pan (Editors) (2020). Transfer learning. Cambridge University Press. DOI 9781139061773

Shai Ben-David and Shai Shalev-Shwartz (2014). Understanding Machine Learning: From Theory to Algorithms

Data Generation Model

domain

$$\mathcal{D} = \{\mathcal{X}, P(\mathcal{X})\}$$

task

$$\mathcal{T} = \{\mathcal{Y}, f(\cdot)\}$$

$$\mathbf{X} \sim P(\mathcal{X}) \xrightarrow{\text{predictive function f}} y = f(\mathbf{X})$$

joint data distribution $P(\mathcal{X} \times \mathcal{Y})$

Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. IEEE Transactions on knowledge and data engineering, 22(10), 1345-1359.
Qiang Yang; Yu Zhang, Wenyuan Dai; Sinno Jialin Pan (Editors) (2020). Transfer learning. Cambridge University Press. DOI 9781139061773
Shai Ben-David and Shai Shalev-Shwartz (2014). Understanding Machine Learning: From Theory to Algorithms

Data Generation Model

domain	task
$\mathcal{D} = \{\mathcal{X}, P(\mathcal{X})\}$	$\mathcal{T} = \{\mathcal{Y}, f(\cdot)\}$

$$\mathbf{X} \sim P(\mathcal{X}) \xrightarrow{\text{predictive function f}} y = f(\mathbf{X})$$

joint data distribution $P(\mathcal{X} \times \mathcal{Y})$

Dataset: $\{(\mathbf{X}_i, y_i)\}_{i=1}^{N} \overset{\text{i.i.d}}{\sim} P(\mathcal{X} \times \mathcal{Y})$

Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. IEEE Transactions on knowledge and data engineering, 22(10), 1345-1359.
Qiang Yang; Yu Zhang, Wenyuan Dai; Sinno Jialin Pan (Editors) (2020). Transfer learning. Cambridge University Press. DOI 9781139061773
Shai Ben-David and Shai Shalev-Shwartz (2014). Understanding Machine Learning: From Theory to Algorithms

Data Generation Model

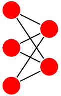
domain $\mathcal{D} = \{\mathcal{X}, P(\mathcal{X})\}$

task $\mathcal{T} = \{\mathcal{Y}, f(\cdot)\}$

$$\mathbf{X} \sim P(\mathcal{X}) \xrightarrow{\text{predictive function f}} y = f(\mathbf{X})$$

joint data distribution $P(\mathcal{X} \times \mathcal{Y})$

Dataset: $\{(\mathbf{X}_i, y_i)\}_{i=1}^{N} \overset{\text{i.i.d}}{\sim} P(\mathcal{X} \times \mathcal{Y})$

input $\mathbf{X}$ → feature learning f_θ  → label y

Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. IEEE Transactions on knowledge and data engineering, 22(10), 1345-1359.
Qiang Yang; Yu Zhang, Wenyuan Dai; Sinno Jialin Pan (Editors) (2020). Transfer learning. Cambridge University Press. DOI 9781139061773
Shai Ben-David and Shai Shalev-Shwartz (2014). Understanding Machine Learning: From Theory to Algorithms

In- and Out-of-Dist. Generalization

The I.I.D assumption is usually enforced via random splitting a single dataset D into training and testing partitions

I.I.D Machine Learning

$$D \overset{\text{i.i.d}}{\sim} P(\mathcal{X} \times \mathcal{Y})$$

random split

$$D_{\text{train}} \qquad D_{\text{test}}$$

$$f_\theta \xrightarrow{\text{compare with } f(\cdot)} \mathcal{L}_{\text{in}}$$

train model

"in-domain" or "in-distribution" generalization

In- and Out-of-Dist. Generalization

The I.I.D assumption is usually enforced via random splitting a single dataset D into training and testing partitions

I.I.D Machine Learning

$$D \overset{\text{i.i.d}}{\sim} P(\mathcal{X} \times \mathcal{Y})$$

random split

D_{train} $\quad$ D_{test}

Data of another distribution

$$\mathbf{X}, y \sim Q(\mathcal{X} \times \mathcal{Y})$$

$$\mathcal{D}_Q = \{\mathcal{X}, P_Q(\mathcal{X})\}$$
$$\mathcal{T}_Q = \{\mathcal{Y}, f_Q(\cdot)\}$$

$$f_\theta \xrightarrow{\text{compare with } f(\cdot)} \mathcal{L}_{\text{in}} \xrightarrow{\text{compare with } f_Q(\cdot)} \mathcal{L}_{\text{out}}$$

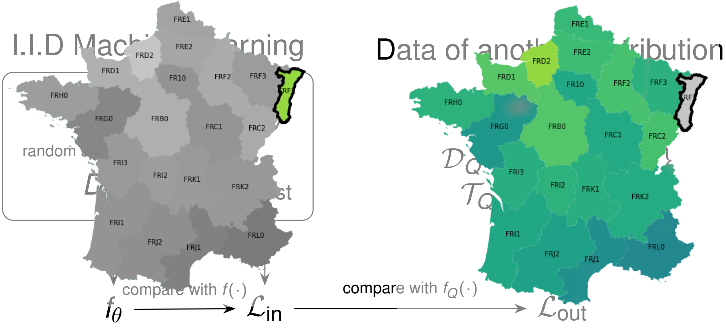
train model

"in-domain" or "in-distribution" generalization

"out-of-domain" or "out-of-distribution" generalization

In- and Out-of-Dist. Generalization

The I.I.D assumption is usually enforced via random splitting a single dataset D into training and testing partitions



train model

”in-domain” or ”in-distribution” generalization

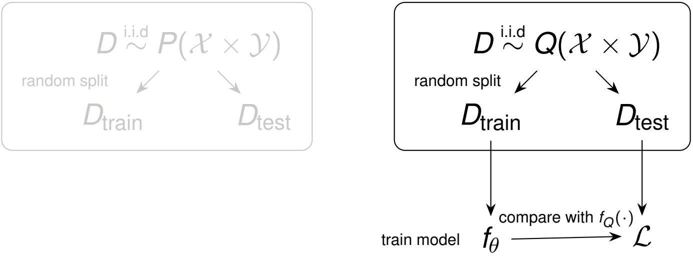
”out-of-domain” or ”out-of-distribution” generalization

Improve on the target Task/Domain

Option 1) Just train on target data (we will call ”random initialization”)

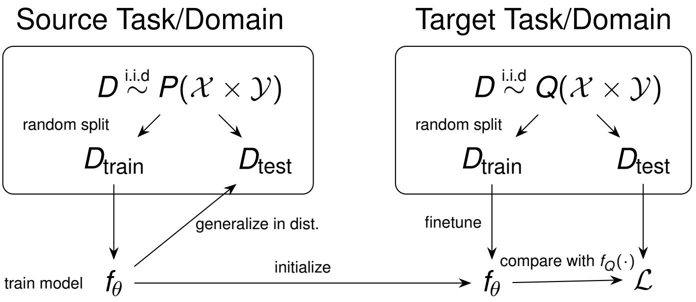
Source Task/Domain

Target Task/Domain



Improve on the target Task/Domain

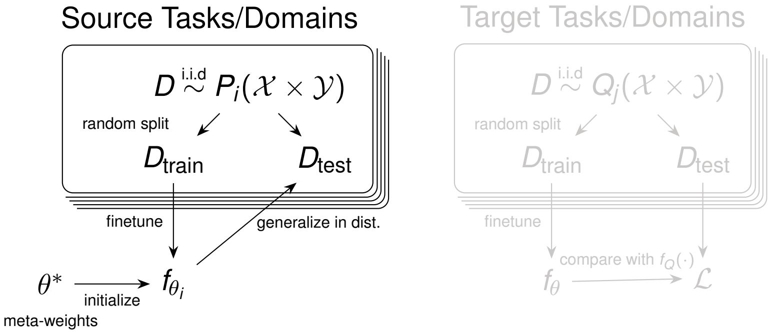
Option 2) "Pretrain" on Source finetune on Target



Improve on the target Task/Domain

Option 3) ML framework that allows not ident. joint dists and in particular Model-Agnostic Meta-Learning

(Finn et al., 2017).

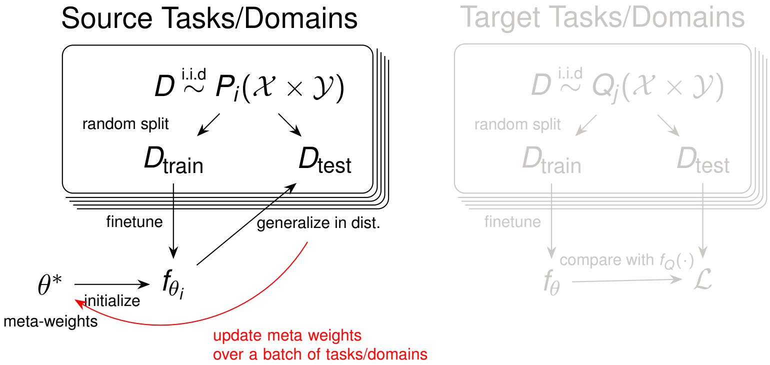


Finn, C., Abbeel, P., & Levine, S. (2017, July). Model-agnostic meta-learning for fast adaptation of deep networks. In International Conference on Machine Learning (pp. 1126-1135). PMLR.

Improve on the target Task/Domain

Option 3) ML framework that allows not ident. joint dists and in particular Model-Agnostic Meta-Learning

(Finn et al., 2017).

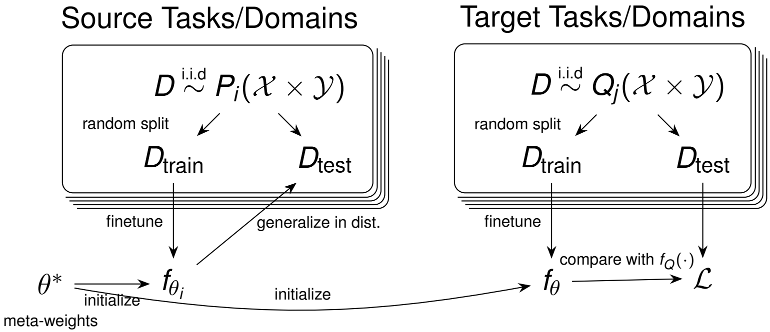


Finn, C., Abbeel, P., & Levine, S. (2017, July). Model-agnostic meta-learning for fast adaptation of deep networks. In International Conference on Machine Learning (pp. 1126-1135). PMLR.

Improve on the target Task/Domain

Option 3) ML framework that allows not ident. joint dists and in particular Model-Agnostic Meta-Learning

(Finn et al., 2017).



Finn, C., Abbeel, P., & Levine, S. (2017, July). Model-agnostic meta-learning for fast adaptation of deep networks. In International Conference on Machine Learning (pp. 1126-1135). PMLR.

A Visual Comparison

Algorithm 1: Model-Agnostic Meta-Learning

$p(\mathcal{T})$: distribution over tasks;
α, β step size hyperparameters;
randomly initialize θ;
while *not done* **do**
  sample batch of tasks $\tau \sim p(\mathcal{T})$;
  foreach $\tau^{(i)} \in \tau$ **do**
    initialize ϕ_i with θ;
    evaluate training loss $\boldsymbol{g} = \nabla_{\phi_i} \mathcal{L}_{\tau^{(i)}}(f_{\phi_i}, \mathcal{D}^{(i)}_{\text{support}})$;
    adapt parameters $\phi_i \leftarrow \phi_i - \alpha \boldsymbol{g}$;
    evaluate test loss $\mathcal{L}_{\tau^{(i)}}(f_{\phi_i}, \mathcal{D}^{(i)}_{\text{query}})$;
  end
  update $\theta \leftarrow \theta - \beta \sum_{\tau^{(i)} \sim p(\tau)} \nabla_\theta \mathcal{L}_{\tau^{(i)}}(f_{\phi_i}, \mathcal{D}^{(i)}_{\text{query}})$;
end

Meta-Learned Initialization

f_{P_1}

f_{P_2}

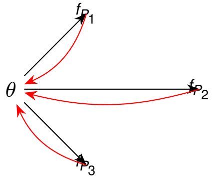
f_{P_3}

A Visual Comparison

Algorithm 1: Model-Agnostic Meta-Learning

$p(\mathcal{T})$: distribution over tasks;
α, β step size hyperparameters;
randomly initialize θ;
while *not done* **do**
  sample batch of tasks $\tau \sim p(\mathcal{T})$;
  foreach $\tau^{(i)} \in \tau$ **do**
    initialize ϕ_i with θ;
    evaluate training loss $\boldsymbol{g} = \nabla_{\phi_i} \mathcal{L}_{\tau^{(i)}}(f_{\phi_i}, \mathcal{D}^{(i)}_{\text{support}})$;
    adapt parameters $\phi_i \leftarrow \phi_i - \alpha \boldsymbol{g}$;
    evaluate test loss $\mathcal{L}_{\tau^{(i)}}(f_{\phi_i}, \mathcal{D}^{(i)}_{\text{query}})$;
  end
  update $\theta \leftarrow \theta - \beta \sum_{\tau^{(i)} \sim p(\tau)} \nabla_\theta \mathcal{L}_{\tau^{(i)}}(f_{\phi_i}, \mathcal{D}^{(i)}_{\text{query}})$;
end

Meta-Learned Initialization

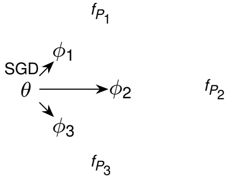


A Visual Comparison

Algorithm 1: Model-Agnostic Meta-Learning

$p(\mathcal{T})$: distribution over tasks;
α, β step size hyperparameters;
randomly initialize θ;
while *not done* **do**
    sample batch of tasks $\tau \sim p(\mathcal{T})$;
    foreach $\tau^{(i)} \in \tau$ **do**
        initialize ϕ_i with θ;
        evaluate training loss $\boldsymbol{g} = \nabla_{\phi_i} \mathcal{L}_{\tau^{(i)}}(f_{\phi_i}, \mathcal{D}^{(i)}_{\text{support}})$;
        adapt parameters $\phi_i \leftarrow \phi_i - \alpha \boldsymbol{g}$;
        evaluate test loss $\mathcal{L}_{\tau^{(i)}}(f_{\phi_i}, \mathcal{D}^{(i)}_{\text{query}})$;
    end
    update $\theta \leftarrow \theta - \beta \sum_{\tau^{(i)} \sim p(\tau)} \nabla_\theta \mathcal{L}_{\tau^{(i)}}(f_{\phi_i}, \mathcal{D}^{(i)}_{\text{query}})$;
end

Meta-Learned Initialization

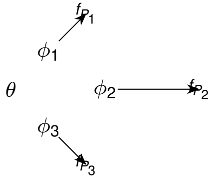


A Visual Comparison

Algorithm 1: Model-Agnostic Meta-Learning

$p(\mathcal{T})$: distribution over tasks;
α, β step size hyperparameters;
randomly initialize θ;
while *not done* **do**
  sample batch of tasks $\tau \sim p(\mathcal{T})$;
  foreach $\tau^{(i)} \in \tau$ **do**
    initialize ϕ_i with θ;
    evaluate training loss $\boldsymbol{g} = \nabla_{\phi_i} \mathcal{L}_{\tau^{(i)}}(f_{\phi_i}, \mathcal{D}^{(i)}_{\text{support}})$;
    adapt parameters $\phi_i \leftarrow \phi_i - \alpha \boldsymbol{g}$;
    evaluate test loss $\mathcal{L}_{\tau^{(i)}}(f_{\phi_i}, \mathcal{D}^{(i)}_{\text{query}})$;
  end
  update $\theta \leftarrow \theta - \beta \sum_{\tau^{(i)} \sim p(\tau)} \nabla_\theta \mathcal{L}_{\tau^{(i)}}(f_{\phi_i}, \mathcal{D}^{(i)}_{\text{query}})$;
end

Meta-Learned Initialization

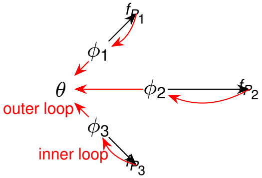


A Visual Comparison

Algorithm 1: Model-Agnostic Meta-Learning

$p(\mathcal{T})$: distribution over tasks;
α, β step size hyperparameters;
randomly initialize θ;
while *not done* **do**
  sample batch of tasks $\tau \sim p(\mathcal{T})$;
  foreach $\tau^{(i)} \in \tau$ **do**
    initialize ϕ_i with θ;
    evaluate training loss $\boldsymbol{g} = \nabla_{\phi_i} \mathcal{L}_{\tau^{(i)}}(f_{\phi_i}, \mathcal{D}^{(i)}_{\text{support}})$;
    adapt parameters $\phi_i \leftarrow \phi_i - \alpha \boldsymbol{g}$;
    evaluate test loss $\mathcal{L}_{\tau^{(i)}}(f_{\phi_i}, \mathcal{D}^{(i)}_{\text{query}})$;
  end
  update $\theta \leftarrow \theta - \beta \sum_{\tau^{(i)} \sim p(\tau)} \nabla_\theta \mathcal{L}_{\tau^{(i)}}(f_{\phi_i}, \mathcal{D}^{(i)}_{\text{query}})$;
end

Meta-Learned Initialization



A Visual Comparison

Algorithm 1: Model-Agnostic Meta-Learning

$p(\mathcal{T})$: distribution over tasks;
α, β step size hyperparameters;
randomly initialize θ;
while *not done* **do**
  sample batch of tasks $\tau \sim p(\mathcal{T})$;
  foreach $\tau^{(i)} \in \tau$ **do**
    initialize ϕ_i with θ;
    evaluate training loss $\boldsymbol{g} = \nabla_{\phi_i} \mathcal{L}_{\tau^{(i)}}(f_{\phi_i}, \mathcal{D}^{(i)}_{\text{support}})$;
    adapt parameters $\phi_i \leftarrow \phi_i - \alpha \boldsymbol{g}$;
    evaluate test loss $\mathcal{L}_{\tau^{(i)}}(f_{\phi_i}, \mathcal{D}^{(i)}_{\text{query}})$;
  end
  update $\theta \leftarrow \theta - \beta \sum_{\tau^{(i)} \sim p(\tau)} \nabla_\theta \mathcal{L}_{\tau^{(i)}}(f_{\phi_i}, \mathcal{D}^{(i)}_{\text{query}})$;
end

Meta-Learned Initialization

f_{P_1}

$\theta \longrightarrow \theta$ f_{P_2}

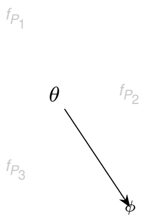
f_{P_3}

A Visual Comparison

Algorithm 1: Model-Agnostic Meta-Learning

$p(\mathcal{T})$: distribution over tasks;
α, β step size hyperparameters;
randomly initialize θ;
while *not done* **do**
  sample batch of tasks $\tau \sim p(\mathcal{T})$;
  foreach $\tau^{(i)} \in \tau$ **do**
    initialize ϕ_i with θ;
    evaluate training loss $\boldsymbol{g} = \nabla_{\phi_i} \mathcal{L}_{\tau^{(i)}}(f_{\phi_i}, \mathcal{D}^{(i)}_{\text{support}})$;
    adapt parameters $\phi_i \leftarrow \phi_i - \alpha \boldsymbol{g}$;
    evaluate test loss $\mathcal{L}_{\tau^{(i)}}(f_{\phi_i}, \mathcal{D}^{(i)}_{\text{query}})$;
  end
  update $\theta \leftarrow \theta - \beta \sum_{\tau^{(i)} \sim p(\tau)} \nabla_\theta \mathcal{L}_{\tau^{(i)}}(f_{\phi_i}, \mathcal{D}^{(i)}_{\text{query}})$;
end

Meta-Learned Initialization



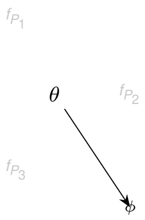
Optimized **to adapt** well to source distributions

A Visual Comparison

Algorithm 1: Model-Agnostic Meta-Learning

$p(\mathcal{T})$: distribution over tasks;
α, β step size hyperparameters;
randomly initialize θ;
while *not done* **do**
  sample batch of tasks $\tau \sim p(\mathcal{T})$;
  foreach $\tau^{(i)} \in \tau$ **do**
    initialize ϕ_i with θ;
    evaluate training loss $\boldsymbol{g} = \nabla_{\phi_i} \mathcal{L}_{\tau^{(i)}}(f_{\phi_i}, \mathcal{D}^{(i)}_{\text{support}})$;
    adapt parameters $\phi_i \leftarrow \phi_i - \alpha \boldsymbol{g}$;
    evaluate test loss $\mathcal{L}_{\tau^{(i)}}(f_{\phi_i}, \mathcal{D}^{(i)}_{\text{query}})$;
  end
  update $\theta \leftarrow \theta - \beta \sum_{\tau^{(i)} \sim p(\tau)} \nabla_\theta \mathcal{L}_{\tau^{(i)}}(f_{\phi_i}, \mathcal{D}^{(i)}_{\text{query}})$;
end

Meta-Learned Initialization

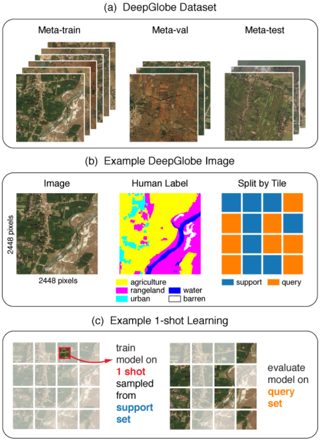


Optimized **to adapt** well to source distributions

while regular **pretraining** is optimized **to perform** well on source distributions.

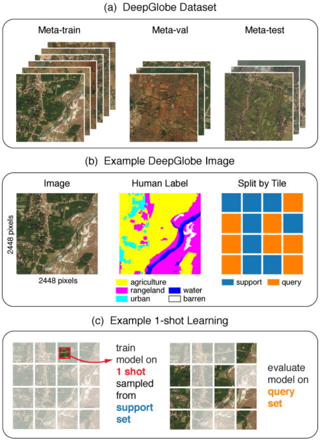
Experiments

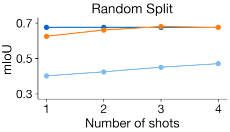
Deepglobe Land Cover Classification



Rußwurm, M., Wang, S., Körner, M., & Lobell, D. (2020). Meta-learning for few-shot land cover classification. In Proceedings of the ieee/cvf conference on computer vision and pattern recognition workshops (pp. 200-201).

Deepglobe Land Cover Classification

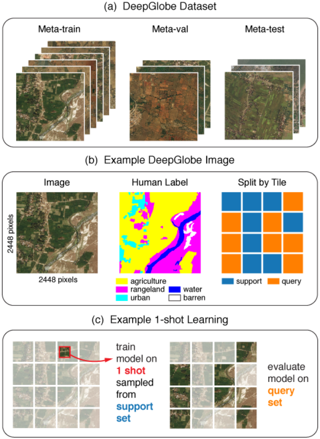


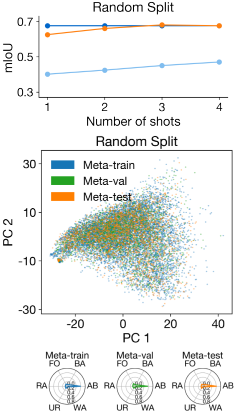


random pretrained MAML

Rußwurm, M., Wang, S., Körner, M., & Lobell, D. (2020). Meta-learning for few-shot land cover classification. In Proceedings of the ieee/cvf conference on computer vision and pattern recognition workshops (pp. 200-201).

Deepglobe Land Cover Classification

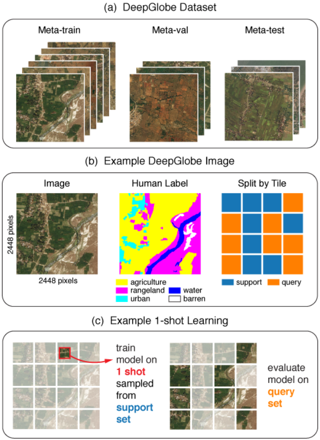


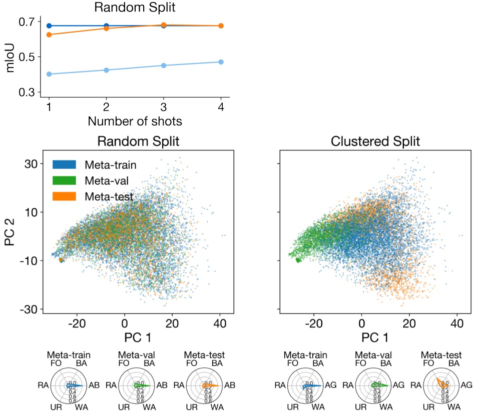


Rußwurm, M., Wang, S., Körner, M., & Lobell, D. (2020). Meta-learning for few-shot land cover classification. In Proceedings of the ieee/cvf conference on computer vision and pattern recognition workshops (pp. 200-201).

EPFL

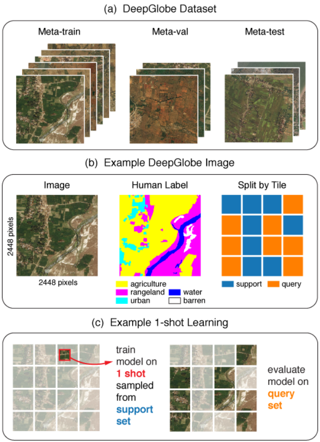
Deepglobe Land Cover Classification

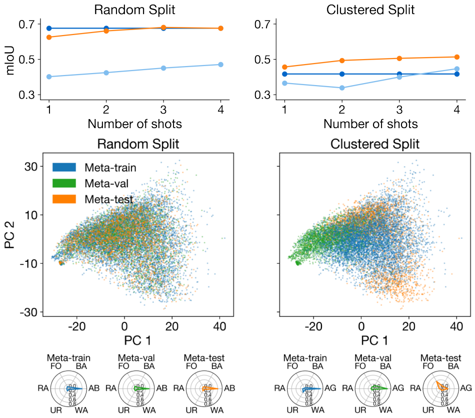




Rußwurm, M., Wang, S., Körner, M., & Lobell, D. (2020). Meta-learning for few-shot land cover classification. In Proceedings of the ieee/cvf conference on computer vision and pattern recognition workshops (pp. 200-201).

Deepglobe Land Cover Classification

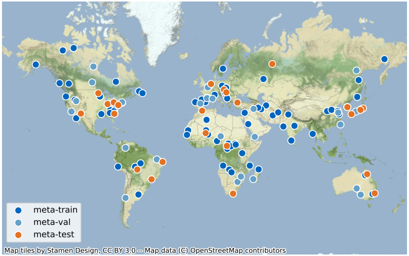




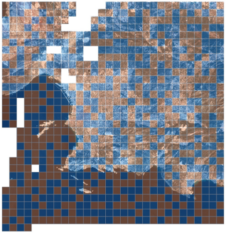
Rußwurm, M., Wang, S., Körner, M., & Lobell, D. (2020). Meta-learning for few-shot land cover classification. In Proceedings of the ieee/cvf conference on computer vision and pattern recognition workshops (pp. 200-201).

Sen12MS — Land Cover Classification

all Sen12MS regions



one region split in train/test

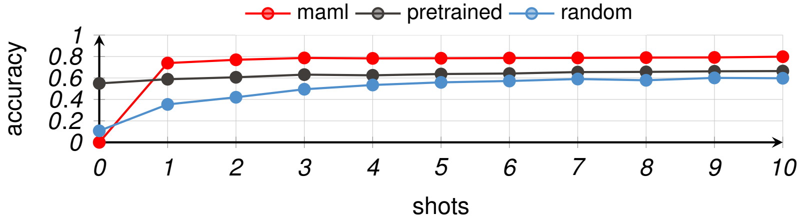


Schmitt, M., Hughes, L. H., Qiu, C., & Zhu, X. X. (2019). SEN12MS–A Curated Dataset of Georeferenced Multi-Spectral Sentinel-1/2 Imagery for Deep Learning and Data Fusion. arXiv preprint arXiv:1906.07789.

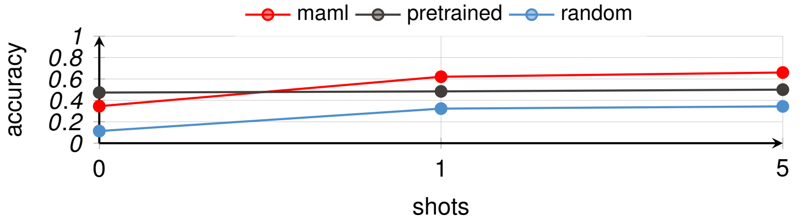
Sen12MS – Few-Shot Performance

Model: 7 stacked CNN layers 64 x 3 x 3 kernels + BN + ReLU + max-pooling

Classification

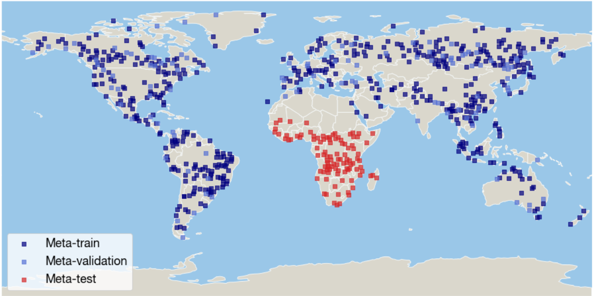


Segmentation



Rußwurm, M., Wang, S., Körner, M., & Lobell, D. (2020). Meta-learning for few-shot land cover classification. In Proceedings of the ieee/cvf conference on computer vision and pattern recognition workshops (pp. 200-201).

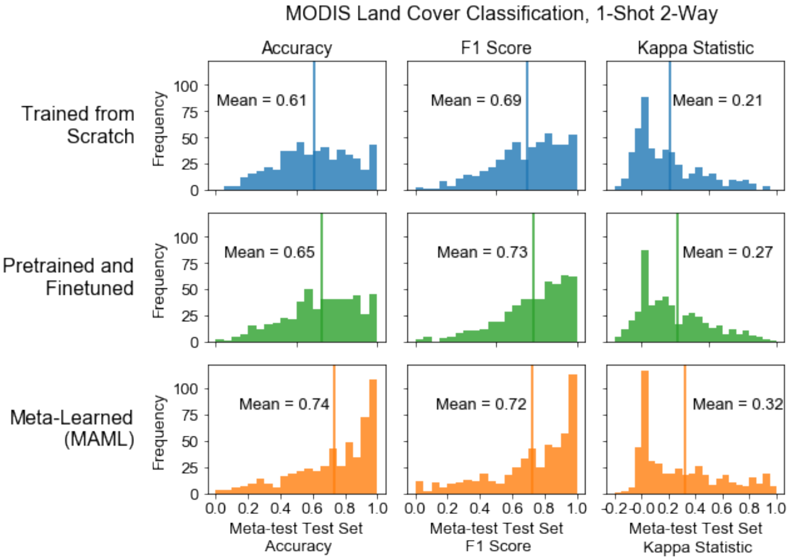
Modis Time Series Classification



1-shot 2-way, 515 meta-test tasks from Africa

Wang, S., Rußwurm, M., Körner, M., & Lobell, D. B. Meta-Learning For Few-Shot Time Series Classification. In IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium (pp. 7041-7044). IEEE.

Challenging Problem: Mixed results



Summary: When to Transfer?

”When to transfer?” of Yang et al., (2020).

- If distributions are too similar, transfer learning is not necessary. (e.g. DeepGlobe, unclustered)
- if distributions are related, transfer learning is effective (e.g. Sen12MS)
- If they are to different, there is no common knowledge to be learned. (e.g. Africa-testset: in Modis Time Series)

Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. IEEE Transactions on knowledge and data engineering, 22(10), 1345-1359.
Qiang Yang; Yu Zhang, Wenyuan Dai; Sinno Jialin Pan (Editors) (2020). Transfer learning. Cambridge University Press. DOI 9781139061773

Summary: When to Transfer?

”When to transfer?” of Yang et al., (2020).

- If distributions are too similar, transfer learning is not necessary. (e.g. DeepGlobe, unclustered)
- if distributions are related, transfer learning is effective (e.g. Sen12MS)
- If they are to different, there is no common knowledge to be learned. (e.g. Africa-testset: in Modis Time Series)

Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. IEEE Transactions on knowledge and data engineering, 22(10), 1345-1359.
Qiang Yang; Yu Zhang, Wenyuan Dai; Sinno Jialin Pan (Editors) (2020). Transfer learning. Cambridge University Press. DOI 9781139061773

Summary: When to Transfer?

”When to transfer?” of Yang et al., (2020).

- If distributions are too similar, transfer learning is not necessary. (e.g. DeepGlobe, unclustered)
- if distributions are related, transfer learning is effective (e.g. Sen12MS)
- If they are to different, there is no common knowledge to be learned. (e.g. Africa-testset: in Modis Time Series)

Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. IEEE Transactions on knowledge and data engineering, 22(10), 1345-1359.
Qiang Yang; Yu Zhang, Wenyuan Dai; Sinno Jialin Pan (Editors) (2020). Transfer learning. Cambridge University Press. DOI 9781139061773

Outlook

- Out-of-distribution generalization remains challenging.
- Few-shot meta-learning well-suited for some cases (i.e., many uniformly distributed datasets around the globe).
- We may need to decide on the learning-framework on a case by case basis

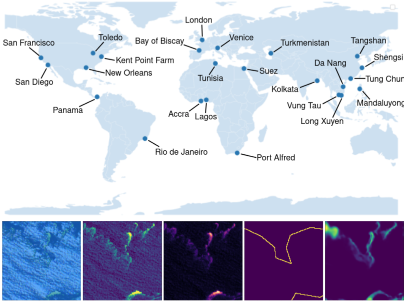
Outlook

- Out-of-distribution generalization remains challenging.
- Few-shot meta-learning well-suited for some cases (i.e., many uniformly distributed datasets around the globe).
- We may need to decide on the learning-framework on a case by case basis

Outlook

- Out-of-distribution generalization remains challenging.
- Few-shot meta-learning well-suited for some cases (i.e., many uniformly distributed datasets around the globe).
- We may need to decide on the learning-framework on a case by case basis

Marine Litter Detection

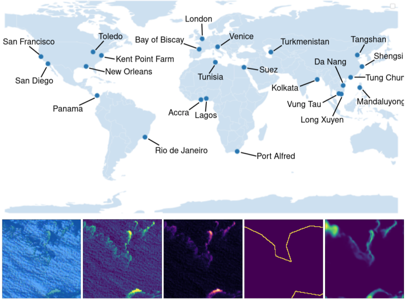


Mifdal, J., Longépé, N., and Rußwurm, M. (2021) Towards detecting floating objects on a global scale with learned spatial features using Sentinel 2, ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci., V-3-2021, 285–293.

Outlook

- Out-of-distribution generalization remains challenging.
- Few-shot meta-learning well-suited for some cases (i.e., many uniformly distributed datasets around the globe).
- We may need to decide on the learning-framework on a case by case basis

Marine Litter Detection



Mifdal, J., Longépé, N., and Rußwurm, M. (2021) Towards detecting floating objects on a global scale with learned spatial features using Sentinel 2, ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci., V-3-2021, 285–293.

what features? ➢ what DL models? ➢ what learning algorithms?

Thank you!

Marc Rußwurm @marccoru

marc.russwurm@epfl.ch
EPFL-ECEO epfl.ch/labs/eceo

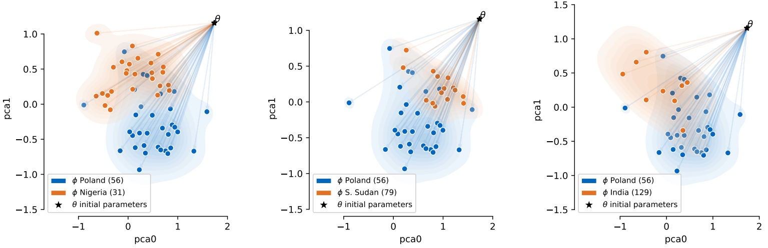
ISPRS Thematic Session on Time Series ST_STIS

with C. Pelletier, Z. Zhou

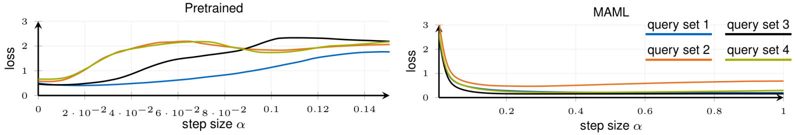
Rußwurm, M., Wang, S., Körner, M., & Lobell, D. (2020). Meta-learning for few-shot land cover classification. In Proceedings of the ieee/cvf conference on computer vision and pattern recognition workshops (pp. 200-201).
Wang, S., Rußwurm, M., Körner, M., & Lobell, D. B. Meta-Learning For Few-Shot Time Series Classification. In IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium (pp. 7041-7044). IEEE.

Similar weights after adaptation?

PCA-representations of the neural network weights before (θ) and after adaptation (ϕ).



Traversing the loss landscape from θ to ϕ.



Colors

rouge	montrose	acier	chartreuse
leman	perle	soufre	marron
grosseille	vertedeau	carotte	ardoise
canard	rose	zinzolin	taupe