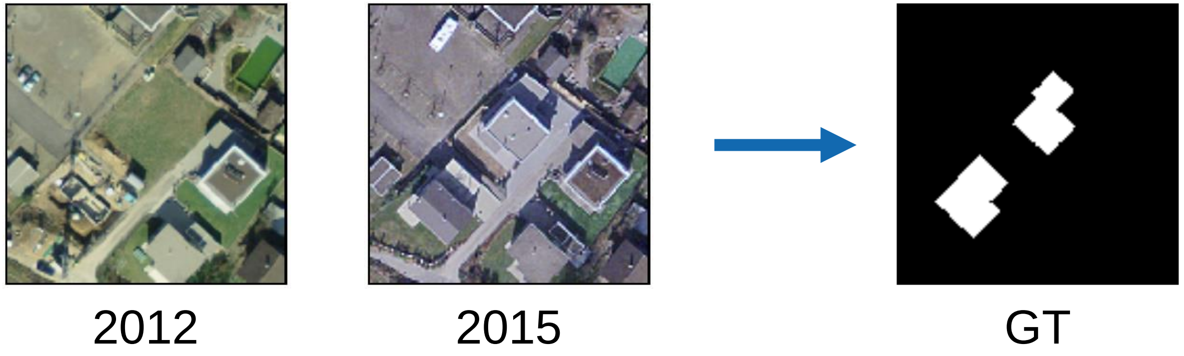
Deep Active Learning in Remote Sensing for data efficient Change Detection

ECML/PKDD Workshop on Machine Learning for Earth Observation, 2020

Vít Růžička, Stefano D'Aronco, Jan Dirk Wegner, Konrad Schindler

- **The Task:** Change detection between two co-registered images



- **The Problem:** Large dataset with **highly unbalanced class distribution**

Number of patches:
1k changed << **83k** unchanged

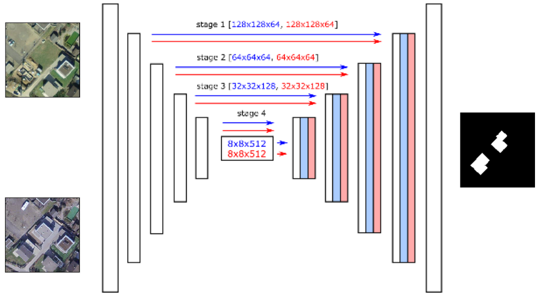
- **The Goal:** **Lower the annotation effort**

Approach: Active Learning

- Annotated **training dataset**

x_1 , x_2 , y

- **Train a model** on training set



Approach: Active Learning

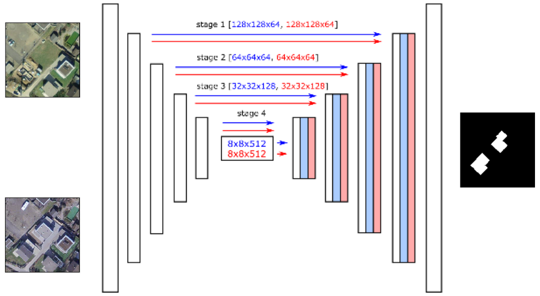
- Annotated **training dataset**

x_1 , x_2 , y

Unlabeled dataset:

x_1 , x_2

- **Train a model** on training set



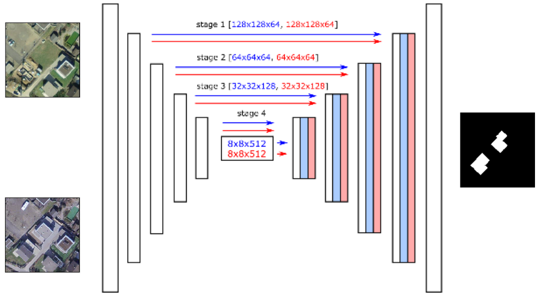
- **Select additional training samples** using trained model representation
- **Idea:** add **samples with high *uncertainty*** to improve the model where it is uncertain

Approach: Active Learning

- Annotated **training dataset**

x_1 , x_2 , y

- **Train a model** on training set



- **Select additional training samples** using trained model representation

- **Idea:** add **samples with high *uncertainty*** to improve the model where it is uncertain

Acquisition function

Unlabeled dataset:

x_1 , x_2

- **Ask the user to provide labels** for selected samples

Approach: Active Learning

- Annotated **training dataset**

x_1 , x_2 , y

Next AL iteration

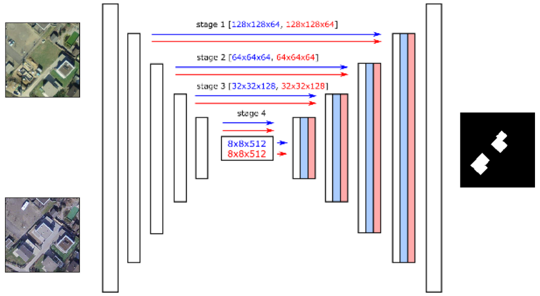
- **Ask the user to provide labels** for selected samples

Unlabeled dataset:

Acquisition function

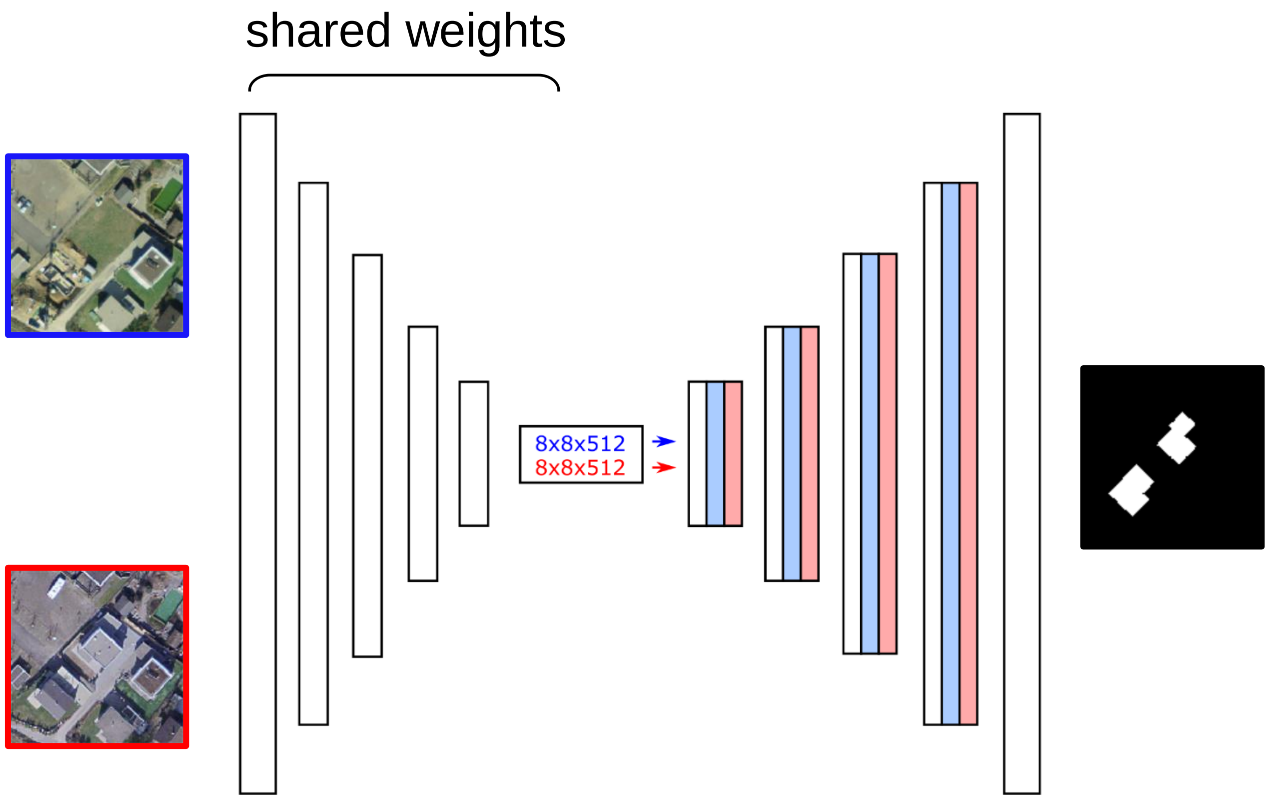
x_1 , x_2

- **Train a model** on training set



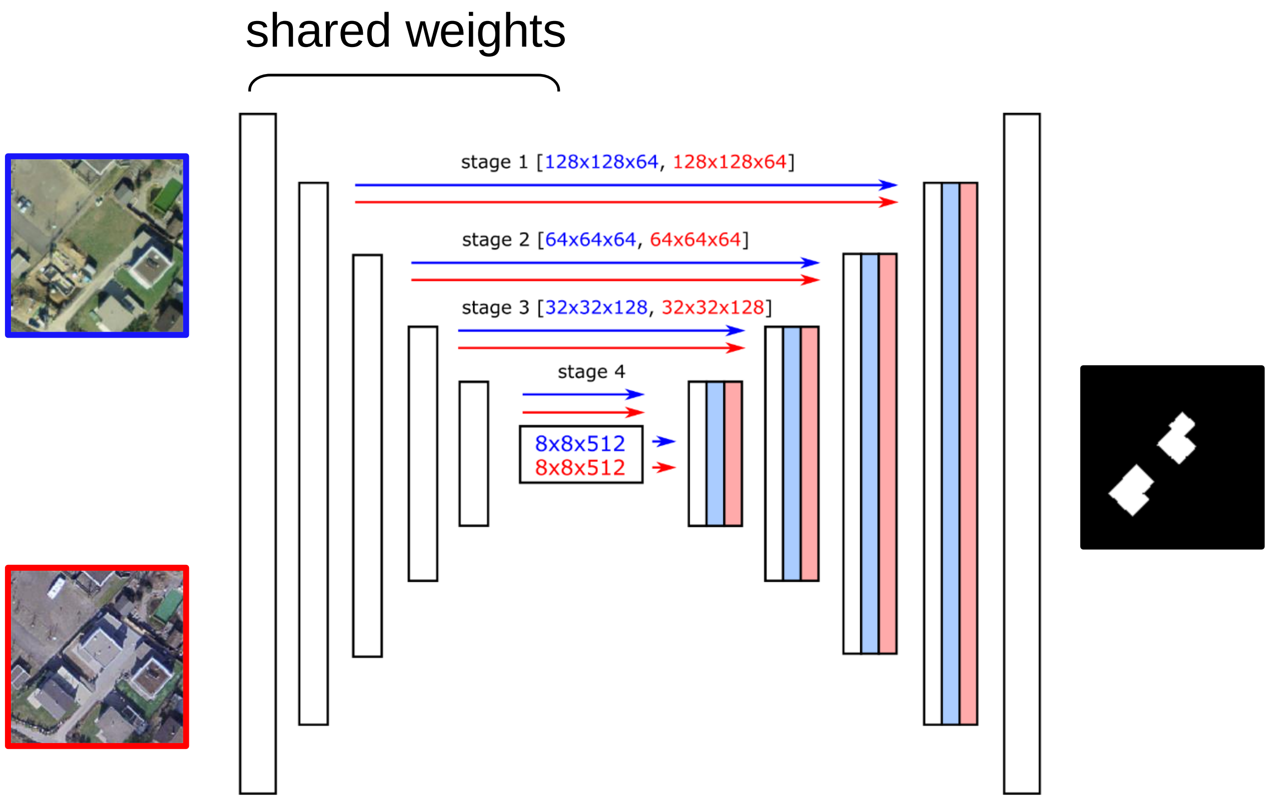
- **Select additional training samples** using trained model representation
- **Idea:** add **samples with high *uncertainty*** to improve the model where it is uncertain

Model design for Active Learning



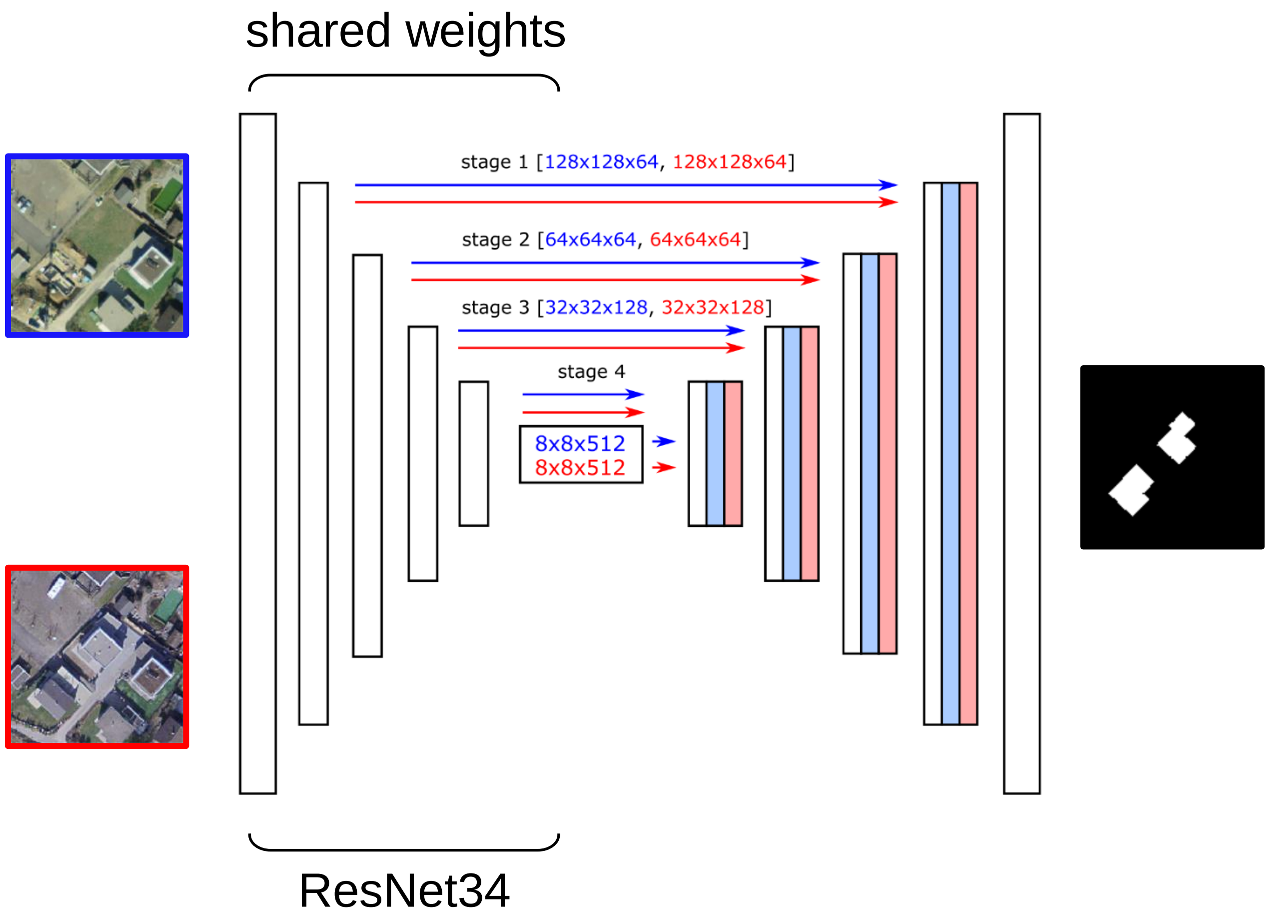
- **Siamese network** for Change Det.
  - with shared encoder weights

Model design for Active Learning



- **Siamese network** for Change Det.
  - with shared encoder weights
- **U-Net model architecture**
  - skip connections preserve detail

Model design for Active Learning

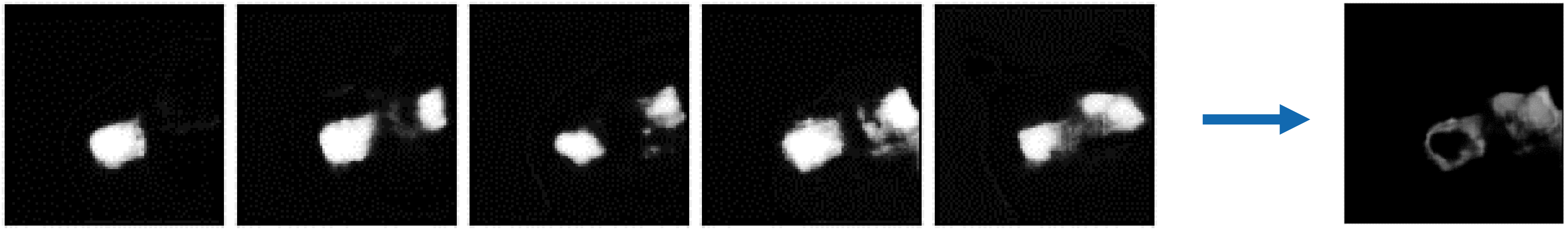


- **Siamese network** for Change Det.
  - with shared encoder weights
- **U-Net model architecture**
  - skip connections preserve detail
- **ResNet34 pre-trained encoder**

Uncertainty estimation for deep learning models

- **Explicit ensemble of models:**
  - **Train multiple models**, average their predictions

- **Implicit ensembles:**
  - Use **stochasticity of regularization processes** to simulate multiple forward passes
  - **Monte Carlo Batch Normalization** (Teye et al., 2018) – batch normalization in *training mode* during inference:



Model ensemble predictions

Variance

score = variance / entropy of predicted posterior, summed over all pixels

Experiments

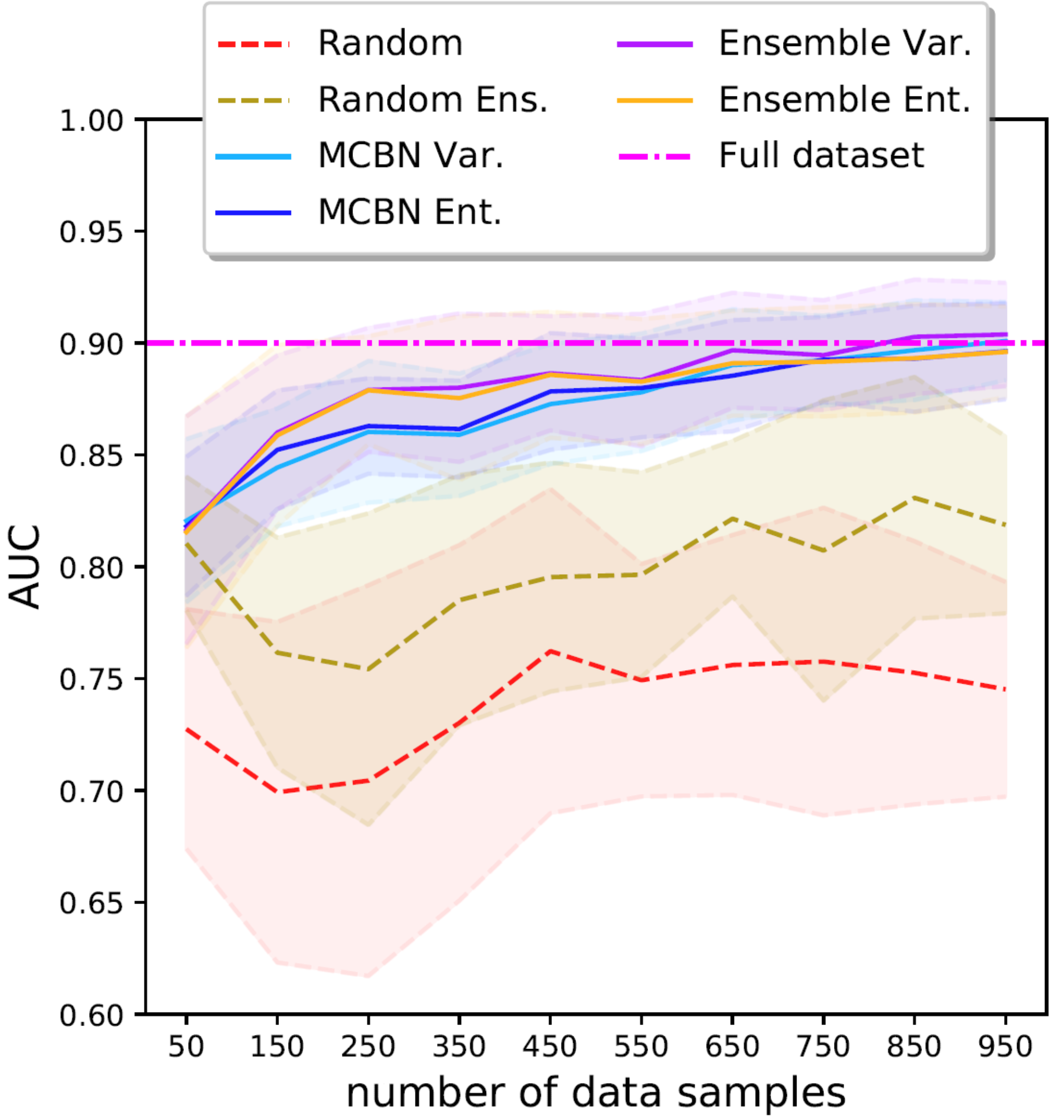
Methods:

- **Ensemble of models** with either variance or entropy metric (N=5 models)
- **Monte Carlo Batch Normalization (MCBN)** with either variance or entropy metric (N=5 forward passes)

Baselines:

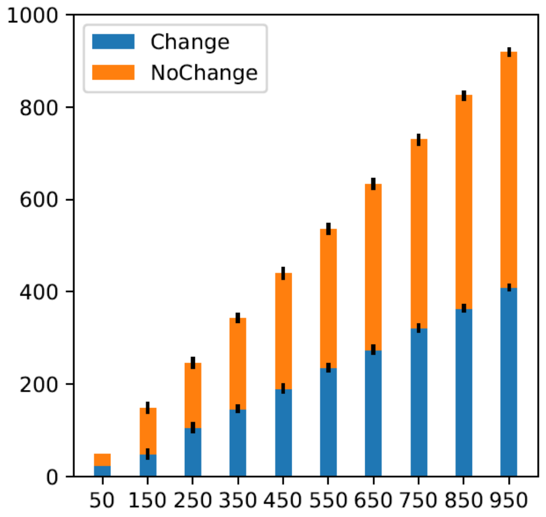
- **Upper bound**: train with all change patches in the (fully annotated) training set, and a matching number of unchanged patches
- **Lower bound**: replace acquisition function with uniform random sampling (with and without ensemble of N=5 models)

Results

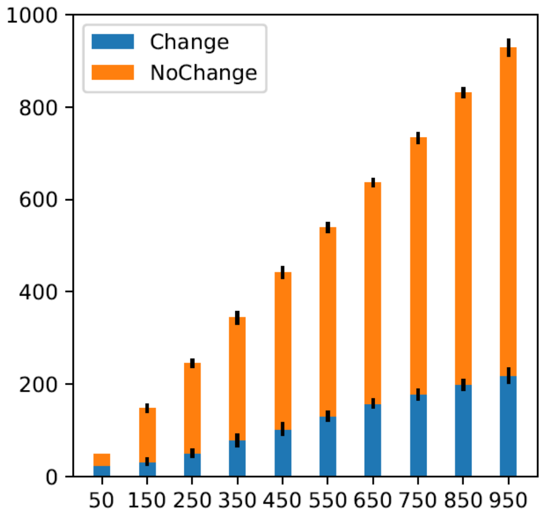


- All tested methods reach the upper bound with **≈1% of the data** (850 samples)
- Ensemble method slightly outperforms MCBN **in early iterations**

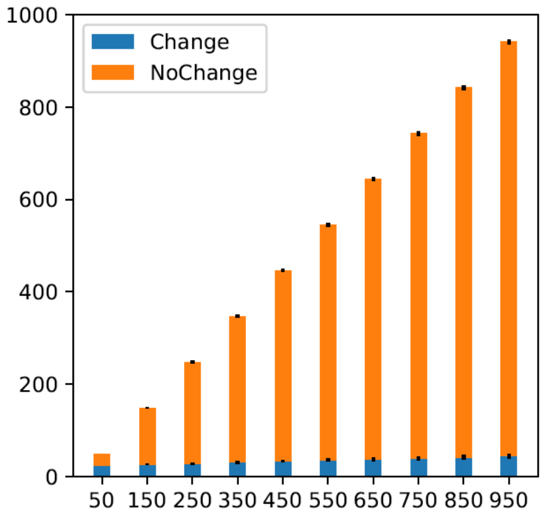
Acquisition function prefers balanced patch distribution



(a) Ensemble Variance



(b) MCBN Variance



(c) Random

- **Both tested methods automatically balance acquisition of uncertain samples** from changed and unchanged patches.
- **Random baseline** converges to the original ratio between classes (1 : 83) which leads to poor performance.

Conclusion

- **Siamese U-Net with ResNet34 encoder** for active learning.
- Matches the performance of manually balanced dataset with all available training patches **with a fraction of the labelling effort (only ≈1 % of the data)**.
- Active sample selection **automatically balances training set**, despite extremely imbalanced input data.
- Standard approach of **explicit ensembles improves faster**, but the novel **MCBN method catches up**.
- **Future work**: faster ways to quantify uncertainty to reduce the time bottleneck.

Bonus: Active Learning Timing

Avg. Training [min]; last 3 iterations

MCBN	107	121	131
Ensemble	266	299	334

-61%

Avg. Acquisition [min]; last 3 iterations

MCBN	162	160	154
Ensemble	100	103	103

+35%

- **Training** a single MCBN model is **61% faster** on average than N=5 Ensemble models
- **Acquisition** evaluation over the whole unlabeled dataset is **35% slower** (note that MCBN is running in unoptimized *training mode* with prediction batch size matching the training batch size)

*) All experiments run on a single GeForce GTX 1080Ti **GPU** (11GB RAM) with Xeon E5-2630v4 **CPU** using Keras.

Bonus: Active Learning Timing

Avg. Training [min]; last 3 iterations

MCBN	107	121	131
Ensemble	266	299	334

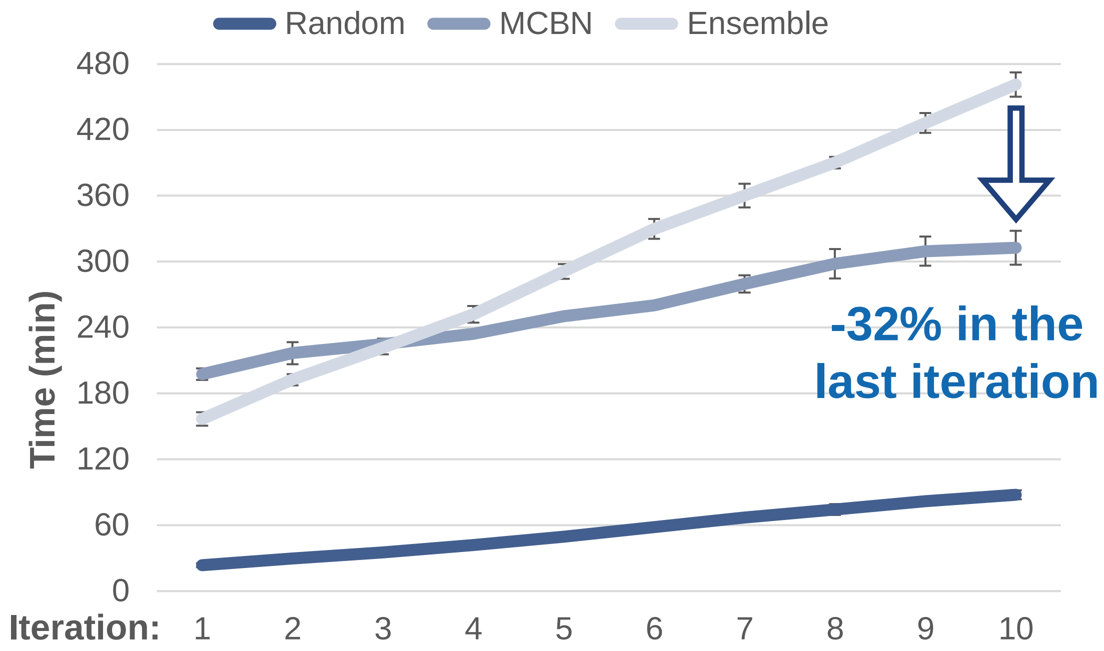
-61%

Avg. Acquisition [min]; last 3 iterations

MCBN	162	160	154
Ensemble	100	103	103

+35%

Whole run [min]; 10 iterations, not cumulative



- **Training** a single MCBN model is **61% faster** on average than N=5 Ensemble models
- **Acquisition** evaluation over the whole unlabeled dataset is **35% slower** (note that MCBN is running in unoptimized *training mode* with prediction batch size matching the training batch size)

*) All experiments run on a single GeForce GTX 1080Ti **GPU** (11GB RAM) with Xeon E5-2630v4 **CPU** using Keras.

Deep Active Learning in Remote Sensing for data efficient Change Detection

ECML/PKDD Workshop on Machine Learning for Earth Observation, 2020

Vít Růžička, Stefano D'Aronco, Jan Dirk Wegner, Konrad Schindler

Thank you for your attention!

Any questions?