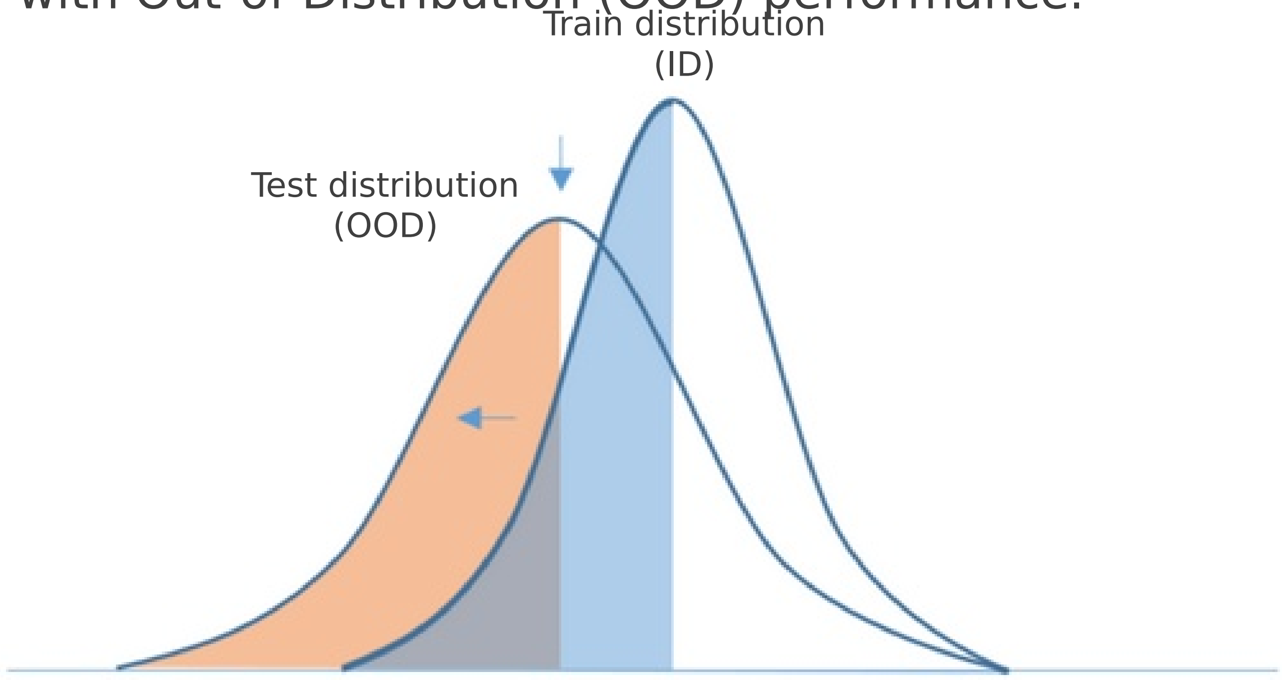
WILDS: Distribution shifts in the wild

FMoW-wilds: Land use classification across different regions and years

The WILD Guess team
William Callaghan
Jérémy Parent
Sorin Muchi
Nathan Alix-Vignola
Mathieu Lamarche

The Distribution shift problem

- Distribution shifts (DS) in ML: « *When training distribution differs from the test distribution* »[1]
- Two distinct shifts problems:
  - Inter-domain shift
  - Subpopulation shift
- Quantify the DS performance drop by comparing In Distribution (ID) performance with Out-of Distribution (OOD) performance.



[1] Koh, Pang Wei, et al. "Wilds: A benchmark of in-the-wild distribution shifts." *International Conference on Machine Learning*. PMLR, 2021.

FMoW dataset from WILDS package

- What is the WILDS package?
  - Benchmark of 10 datasets with naturally occuring distribution shifts.
  - Specifically built to study distribution shift impact on model performance.
  - Includes pre-made scripts, dataloaders, baseline models and basic methods to compensate distribution shift impact.
- Functional Map of the World (FMoW) dataset
  - More than 500k satellite images of human features on earth.
  - Classification problem with 62 categories of building & land use.

Category Examples

Tunnel Opening

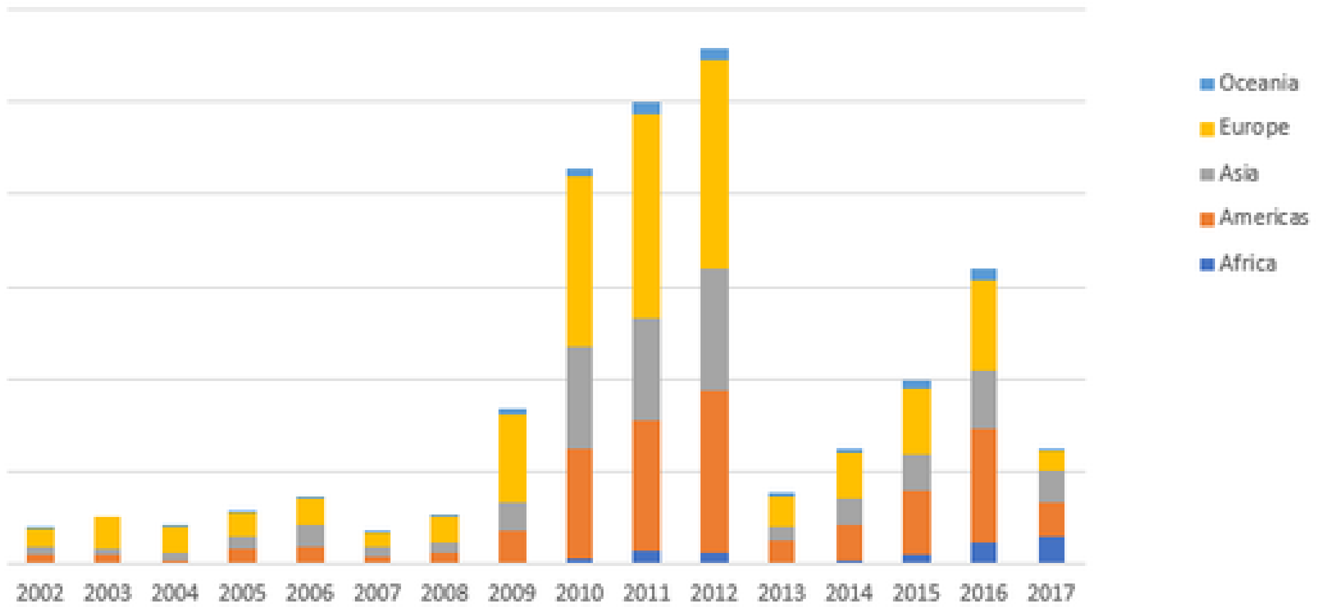
Office Building

Oil or Gas Facility

Dam

FMoW dataset as a Distribution Shift problem

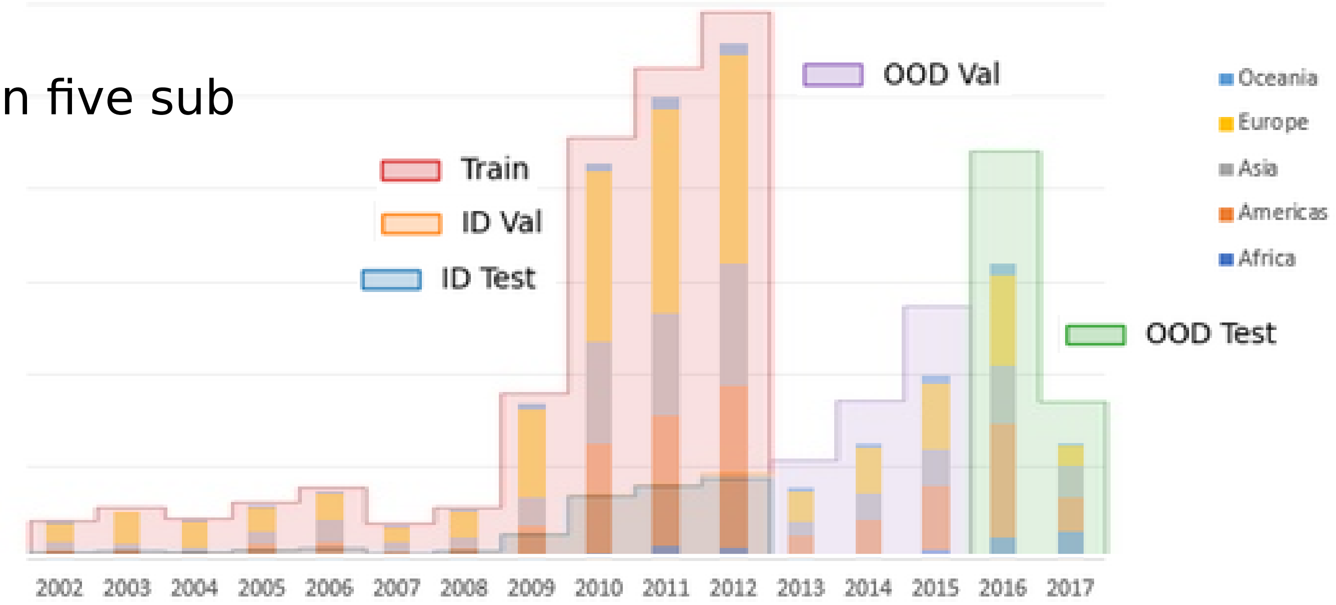
- Sub-population shift problem across regions (PB #1).
- Inter-Domain distribution shift problem across years (PB #2).



FMoW dataset as a Distribution Shift problem

- Sub-population shift problem across regions (PB #1).
- Inter-Domain distribution shift problem across years (PB #2).

Main dataset split in five sub dataset



- Objective:
  - *Maintaining* the overall model predictive power while *uniformizing* its performance per region and per year group.
  - Key metrics:
    - Test ID accuracy
    - Test OOD accuracy
    - Worst region accuracy
    - Average per region accuracy

Method	OOD Test Accuracy	ID Test Accuracy	OOD Test Average Region Accuracy	OOD Test Worst Region Accuracy
ERM Baseline	53.7%	59.7%	52.6%	34.7%

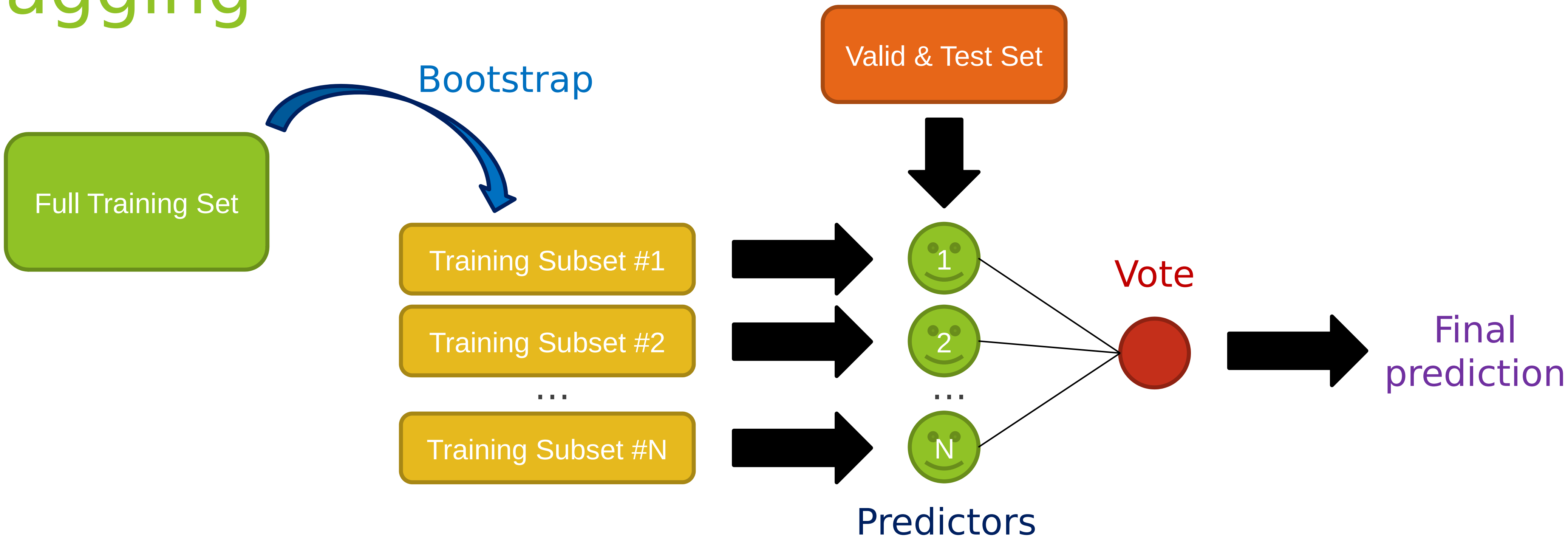
Minimise difference (PB #2)

Minimise difference (PB #1)

Explored Methods to Compensate Distribution Shift

- Bagging
- Label Shift Corrections
- Black Box Shift Correction (BBSC)
- Distributionally & Outlier Robust Optimisation (DORO)
- ConvNext
- Vision Transformer

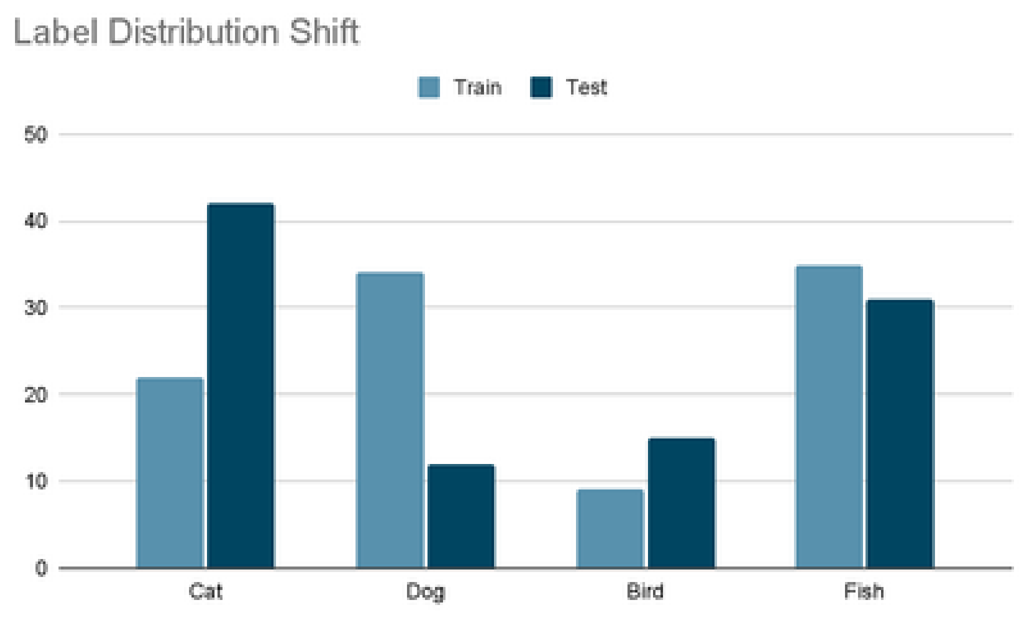
Bagging



Method	Inter-Domain distribution shift across years: OOD Test Accuracy	ID Test Accuracy	ID-OOD Test Average Accuracy Relative Difference	Subpopulation shift across regions: OOD Test Average Region Accuracy	OOD Test Worst Region Accuracy	OOD Test Average-Worst Region Relative Difference
ERM Baseline	53.7%	59.7%	10.2%	52.6%	34.7%	34.0%
Bootstrapped Dataset	49.3%	55.4%	11.2%	49.1%	33.6%	31.6%
Bagging with Bootstrap	53.1%	58.6%	9.4%	51.8%	34.3%	33.8%

Label Shift Correction

- Assumptions
  - p(y) changes
  - P(x|y) stays fixed
- Expectation Maximization + Bias-Corrected Temperature Scaling
  - Estimate the label shift
  - Reweight the predictions accordingly
- Experimental Results
  - Applied Blindly to whole dataset -> Poor results
  - Applied Per Year-Region groups
    - Comparable Results & Improvements on worst region accuracy.



Method	OOD Test Accuracy	ID Test Accuracy	ID-OOD Test Average Accuracy Relative Difference	OOD Test Average Region Accuracy	OOD Test Worst Region Accuracy	OOD Test Average-Worst Region Relative Difference
ERM Baseline	53.7%	59.7%	10.2%	52.6%	34.7%	34.0%
ERM Baseline + EM & BCTS	49.7%	54.6%	9.8%	50.2%	39.1%	22.1%
DORO	51.6%	59.5%	13.3%	50%	32.7%	35%
DORO + EM & BCTS	51.7%	59.2%	12.6%	51.4%	33.5%	34.7%
Bootstrap	50.9%	57.2%	11%	49.1%	33.4%	31.5%

Black Box Shift Correction (BBSC)

- Similar assumptions as Label Shift Correction
  - p(y) changes
  - P(x|y) stays fixed
  - Training data should contain labels from every class.
- Correcting Label Shift
  - Estimates the ratio w = q(y)/p(y) for each label.
  - w is used in importance-weighted ERM to obtain a new predictor.
- Experimental Results
  - Applied blindly to whole dataset -> Poor results
  - Greater label subpopulation shift than global label shift
- Areas for Improvement
  - Run method separately on each region to produce a set of weights corresponding to each region.
    - Objective becomes the average of weighted losses across regions.
    - Could also have a weighted average of weighted losses across regions (similar to groupDRO).

Method	OOD Test Accuracy	ID Test Accuracy	ID-OOD Test Average Accuracy Relative Difference	OOD Test Average Region Accuracy	OOD Test Worst Region Accuracy	OOD Test Average-Worst Region Relative Difference
ERM Baseline	53.7%	59.7%	10.2%	52.6%	34.7%	34.0%
BBSC	50.7%	57.8%	12.3%	49.7%	28.6%	42.5%

adapt to Wilds project

Distributionally & Outlier Robust Optimisation (DORO)

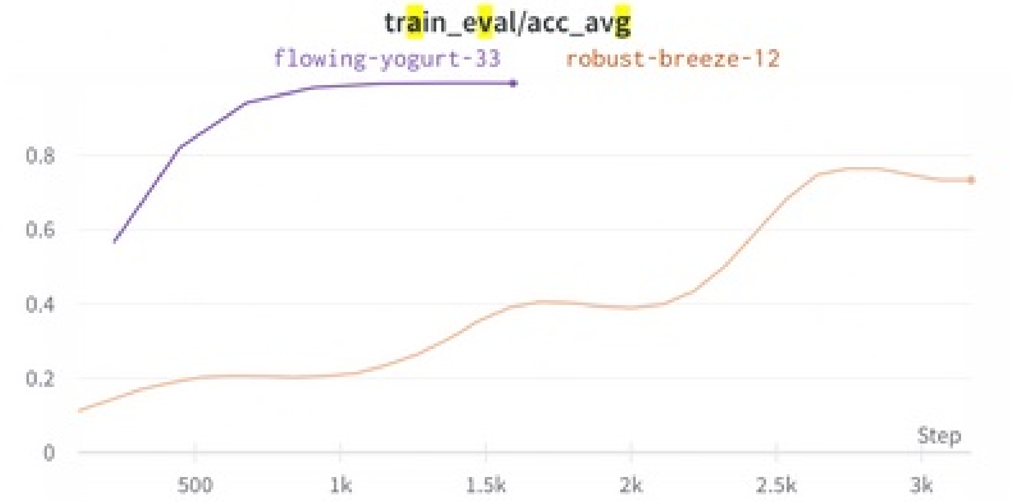
- Extension of Distributionally Robust Optimisation (DRO)
  - Aims to minimize worst-case training loss over pre-defined groups.
  - Assigns more weight to “harder” instances.
- However, DRO sensitive to outliers.
  - Intuitively “hard” instances that incur higher losses than inliers.
  - DORO filters out a fraction of data (epsilon) based on one of two methods:
    - Conditional Value at Risk (CVaR)
    - Chi-Squared Risk
- Experimental Results
  - Did not perform as well as ERM baseline.
  - Possibly due to other shortcomings of DRO -> learning spurious correlations leading to high loss on some groups.
- Areas of Improvement
  - Extend DORO to groupDORO
  - Run with larger batch size

Method	OOD Test Accuracy	ID Test Accuracy	ID-OOD Test Average Accuracy Relative Difference	OOD Test Average Region Accuracy	OOD Test Worst Region Accuracy	OOD Test Average-Worst Region Relative Difference
ERM Baseline	53.7%	59.7%	10.2%	52.6%	34.7%	34.0%
DORO	51.6%	59.5%	11.8%	50.0%	32.5%	35.0%

2.pdf
adapt to Wilds project

ConvNext method

- Can bigger model improve OOD shift ?
- ConvNext is a CNN based architecture similar to the baseline of the FMoW dataset (DenseNet)
  - Larger Kernel Size (7x7)
  - GELU instead of ReLU
  - Layer Normalization instead of Batch Normalization
  - Inverted bottlenecks
- F



Method	OOD Test Accuracy	ID Test Accuracy	ID-OOD Test Average Accuracy Relative Difference	OOD Test Average Region Accuracy	OOD Test Worst Region Accuracy	OOD Test Average-Worst Region Relative Difference
ERM Baseline	53.7%	59.7%	10.2%	52.6%	34.7%	34.0%
ConvNext	60.2%	67.2%	10.4%	58.9%	38.6%	34.5%

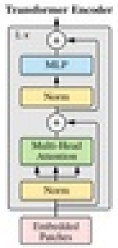
Vision Transformer (ViT) method

Grid search:

- architectures: B/16, B/32, L/16, L/32
- weights initialization: random, pre-trained
- learning approaches: see table

Best architecture:
ViT-B/16 pretrained on ImageNet-21k & Noisy Student

Model	Layers	Hidden size D	MLP size	Heads	Params
ViT-Base	12	768	3072	12	86M
ViT-Large	24	1024	4096	16	307M
ViT-Huge	32	1280	5120	16	632M



Links:

- Vision Transformer Paper
  :
- PyTorch Implementation
- Pre-trainted Weights: (ImageNet-21k)

	Inter-Domain distribution shift across years			Subpopulation shift across regions		
Method	**OOD Test Accuracy**	**ID Test Accuracy**	**ID-OOD Test Average Accuracy Relative Difference**	**OOD Test Average Region Accuracy**	**OOD Test Worst Region Accuracy**	**OOD Test Average-Worst Region Relative Difference**
ERM Baseline	53.7%	59.7%	**10.2%**	52.6%	**34.7%**	**34.0%**
ViT & ERM	52.5%	60%	13.2%	52.5%	30.0%	42.8%
ViT & groupDRO	31.5%	56%	12.9%	31.5%	31.5%	**35.1%**
ViT & deepCORAL	52.6%	60%	12.9%	52.6%	32.0%	39.1%
ViT & IRM	38.2%	45%	14.5%	38.2%	24.6%	35.6%
ViT & DANN	46.7%	54%	13.3%	46.7%	28.1%	39.9%
ViT & FixMatch	53.0%	**62%**	14.4%	53.0%	32.1%	39.3%
ViT & PseudoLabel	52.8%	61%	13.8%	52.8%	33.0%	37.5%

Conclusion

- Best overall model: ConvNext
- Best overall method for Distribution Shift compensation: ERM Baseline + EM & BCTS

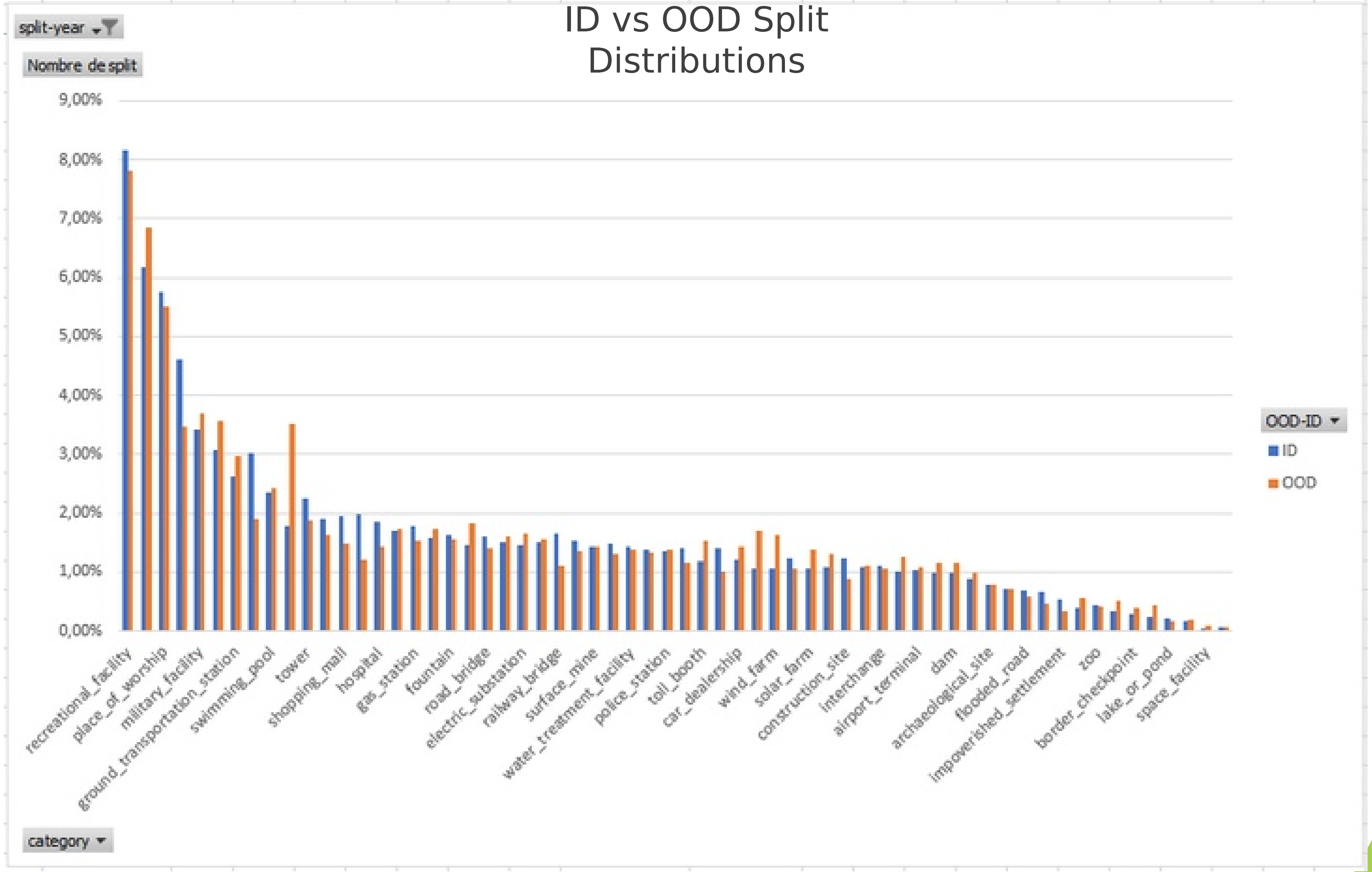
	Inter-Domain distribution shift across years			Subpopulation shift across regions		
Method	**OOD Test Accuracy**	**ID Test Accuracy**	**ID-OOD Test Average Accuracy Relative Difference**	**OOD Test Average Region Accuracy**	**OOD Test Worst Region Accuracy**	**OOD Test Average-Worst Region Relative Difference**
ERM Baseline	53.7%	59.7%	10.2%	52.6%	34.7%	34.0%
ERM Baseline + EM & BCTS	49.7%	54.6%	9.8%	50.2%	39.1%	22.1%
Bagging w/o Bootstrap	57.2%	64.1%	10.8%	56.4%	35.8%	36.5%
ViT & ERM	52.5%	60%	13.2%	52.5%	30.0%	42.8%
ViT & Noisy Student	53.8%	62%	12.6%	53.8%	33.3%	38.2%
BBSC	50.7%	57.8%	12.3%	49.7%	28.6%	42.5%
DORO	51.6%	59.5%	11.8%	50.0%	32.5%	35.0%
ConvNext	60.2%	67.2%	10.4%	58.9%	38.6%	34.5%

Q & A

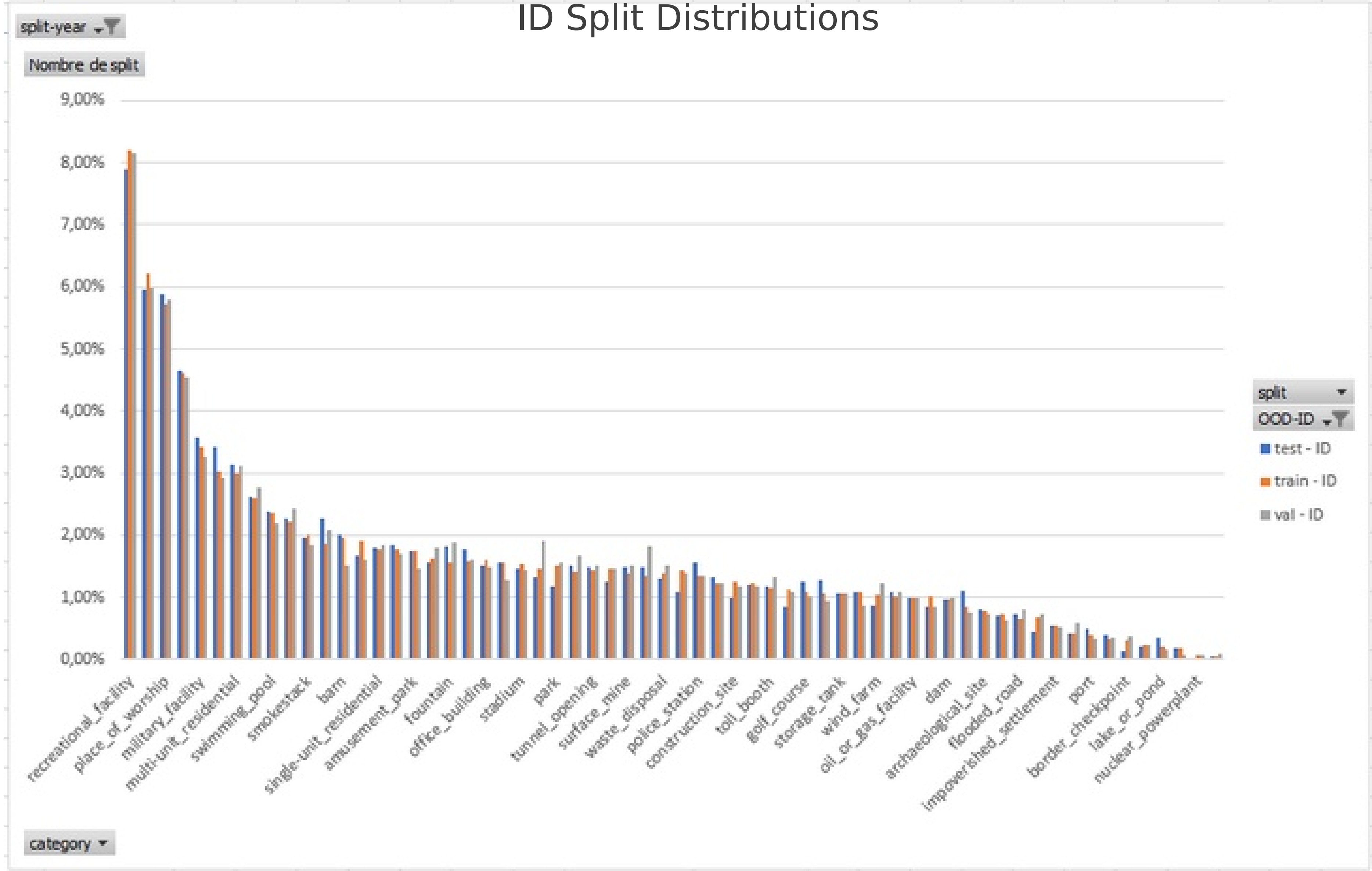
?

Annex

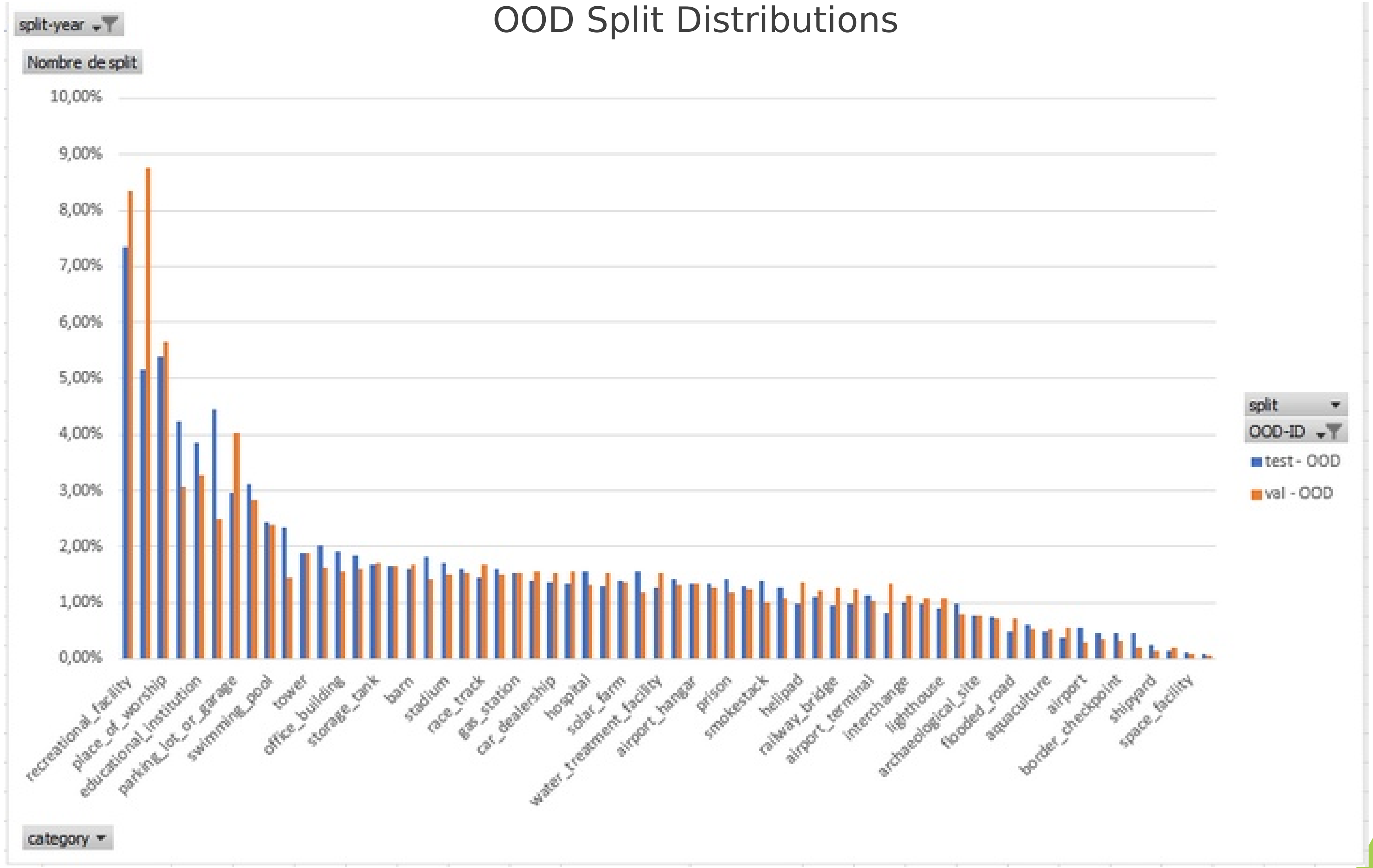
Dataset exploration



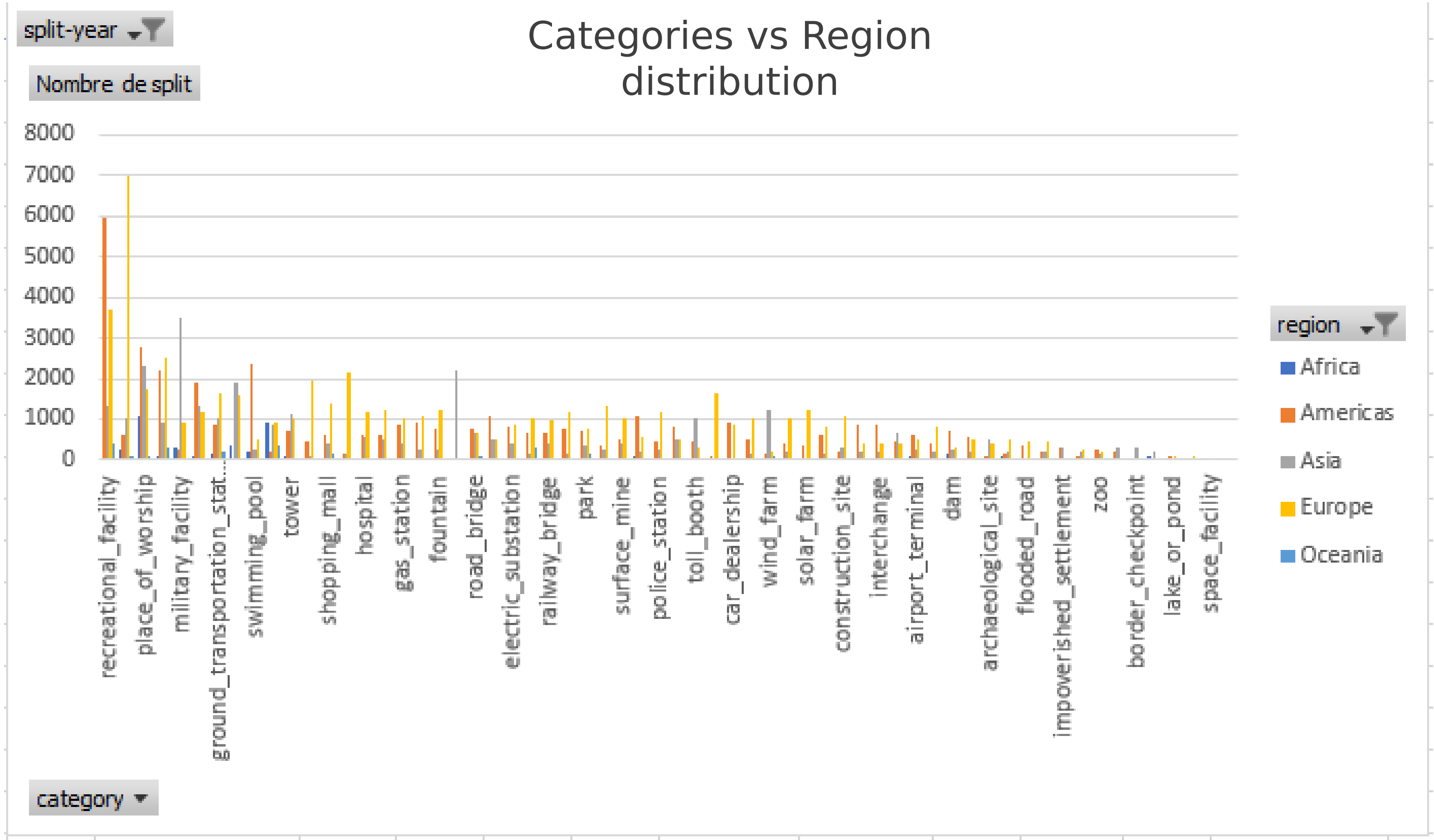
Dataset exploration



Dataset exploration

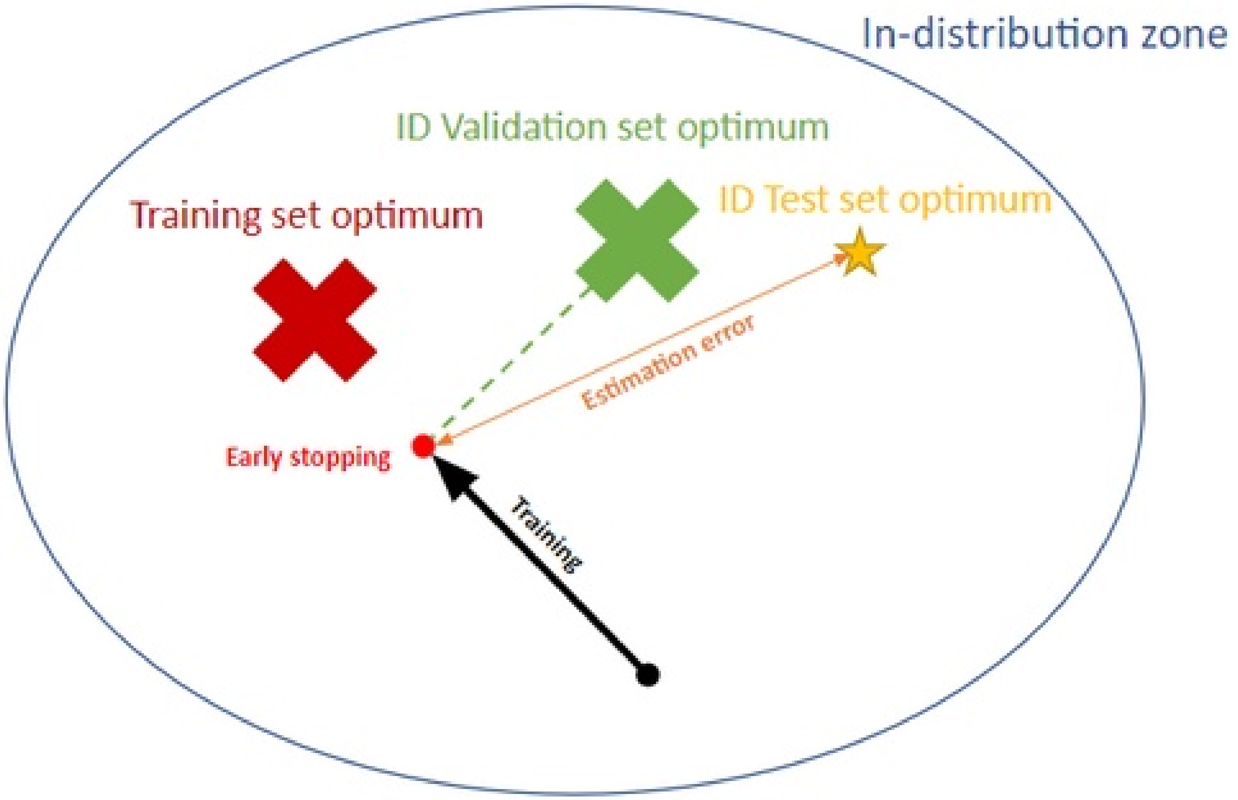


Dataset exploration



Parameter exploration space
Ideal In-Distribution Case

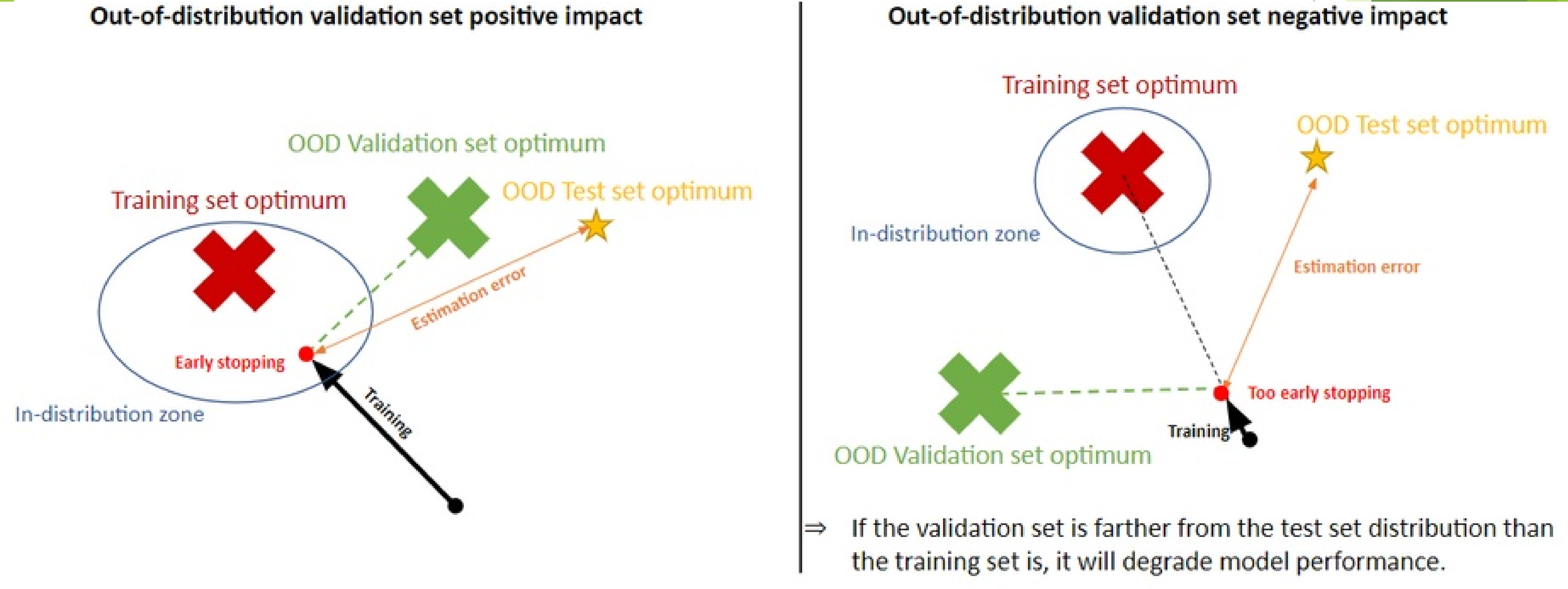
⇒ Validation and Test sets are both from the same distribution as the Training set



⇒ Validation-Test sets delta has information on how far could the Test set optimum be standing.

Parameter exploration space Distribution Shift problem

⇒ Test set is from a different (unknown) distribution than the Training set (Out-of-Distribution).
⇒ Select the validation set to be in also OOD or keep it ID.
⇒ Result in increased estimation error.



Model Training

ERM: - Empirical Risk Minimization *(default / standard training approach)*

groupDRO: Group distributionally robust optimization,
objective: minimize the worst-case training loss over a set of pre-defined groups
+ aggressive regularization (L2 & early-stopping)

deepCORAL: CORrelation ALignment, unsupervised adaptation
minimizes domain shift by aligning the second-order statistics of source and target distributions,
without requiring any target labels

IRM: Invariant Risk Minimization,
learns a data representation such that the optimal classifier,
on top of that data representation, matches for all training distributions

DANN: Domain-Adversarial Training of Neural Networks
trained on labeled data from the source domain and unlabeled data from the target domain

AFN: Adaptive Feature Norm
"progressively adapting the feature norms of the two domains to a large range of values can result in significant transfer gains, implying that those task-specific features with larger norms are more transferable"

PseudoLabel: naive method: dynamically generates pseudolabels and updates the model each batch

FixMatch: FixMatch, semi-supervised
"adds consistency regularization on top of the Pseudo-Label algorithm. Specifically, it generates pseudolabels on a weakly augmented view of the unlabeled data, and then minimizes the loss of the model's prediction on a strongly augmented view"

NoisyStudent: Student-Teacher architecture, semi-supervised
teacher phase generates pseudolabels, and student phases trains to convergence on the (pseudo)labeled data