# Skada and Skada-Bench: Benchmarking Unsupervised Domain Adaptation Methods with Realistic Validation

Y. Lalou*, T. Gnassounou*, A. Collas*, A. De Mathelin*, O. Kachaiev, A. Odonnat, A. Gramfort, T. Moreau, R. Flamary
* Equal contribution

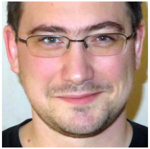
Mind seminar, 2024

# Table of contents

# SKADA Maintainers

Théo
Gnassounou

Oleksii
Kachaiev

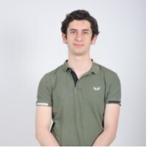


Rémi
Flamary

Antoine
Collas


Yanis
Lalou

Antoine
De Mathelin



Ruben
Bueno

Alexandre
Gramfort

+ all the contributors

# 1. Introduction to **Domain Adaptation** (DA)

# Supervised learning

**Independent** and **identically** distributed data:

$$\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n \sim \mathbb{P}(X, Y)$$

$\boldsymbol{x}_i \in \mathbb{R}^d$, $y_i \in \mathcal{Y}$, e.g. $\{-1, 1\}$ for binary classification.

**Goal** : find a predictor $f : \mathbb{R}^d \to \mathcal{Y}$ by empirical risk minimization

$$\min_{f \in \mathcal{F}} \left\{ R(f) = \frac{1}{n} \sum_{i=1}^{n} \ell\left(y_i, f(\boldsymbol{x}_i)\right) \right\}$$

with $\ell$ a loss function.

- Hyper-parameters tuning with a grid search cross-validation.
- Generalization performance estimation with a cross-validation.
- Easy to implement with Scikit-Learn.

# Domain Adaptation Problem

**Source** domain ($\mathcal{S}$) and **Target** domain ($\mathcal{T}$) with

$$\mathbb{P}_{\mathcal{S}}(X, Y) \neq \mathbb{P}_{\mathcal{T}}(X, Y)$$

Source domain ($\mathcal{S}$): MNIST

Target domain ($\mathcal{T}$): SVHN

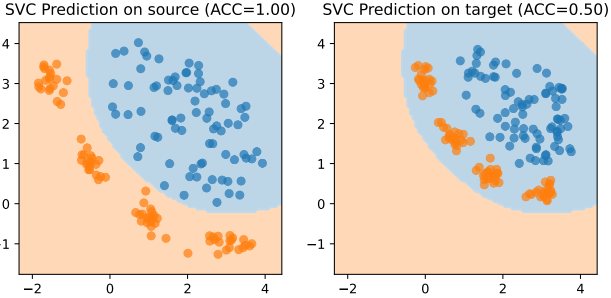
Domain adaptation: MNIST $\rightarrow$ SVHN

# Domain Adaptation Problem

**Source** domain ($\mathcal{S}$) and **Target** domain ($\mathcal{T}$) with

$$\mathbb{P}_{\mathcal{S}}(X, Y) \neq \mathbb{P}_{\mathcal{T}}(X, Y)$$

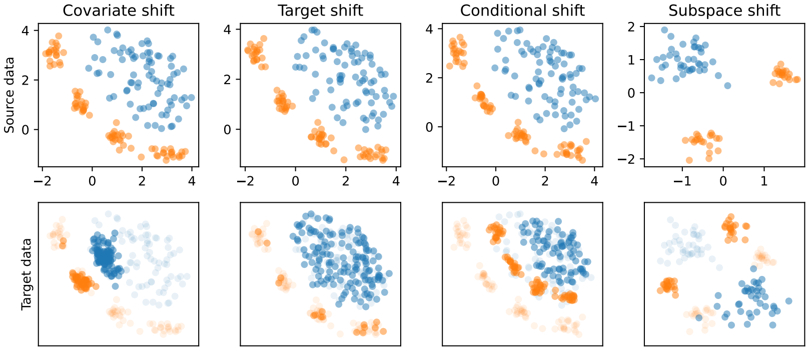
Training on the source domain (classical empirical risk minimization):



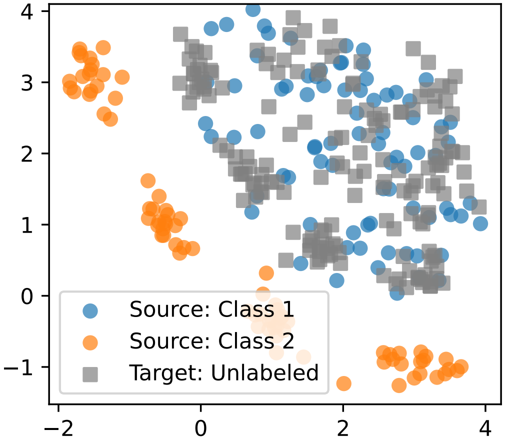

Simple shifts can lead to very bad classifications on the target domain.

# Unsupervised Domain Adaptation problem

- Source domain is **labeled** : $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n_\mathcal{S}} \sim \mathbb{P}_\mathcal{S}(X, Y)$
- Target domain is **unlabeled** : $\{(\boldsymbol{x}_i, .\ )\}_{i=1}^{n_\mathcal{T}} \sim \mathbb{P}_\mathcal{T}(X, Y)$
- Assumptions on the shift between $\mathbb{P}_\mathcal{S}(X, Y)$ and $\mathbb{P}_\mathcal{T}(X, Y)$:

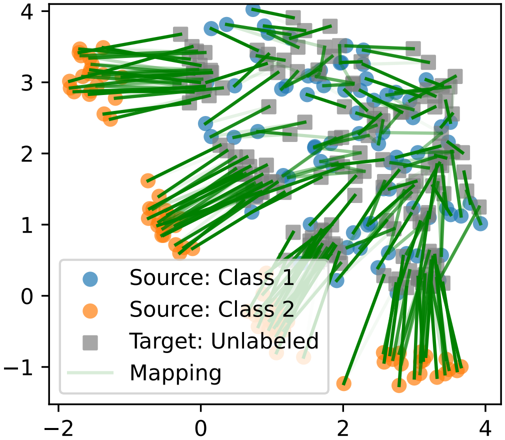


# Toy example of Unsupervised DA



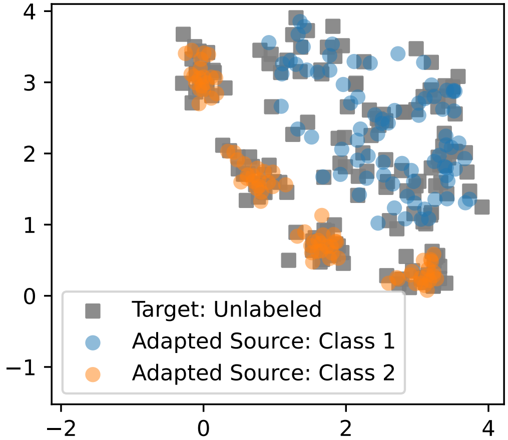

Problem setting

# Toy example of Unsupervised DA



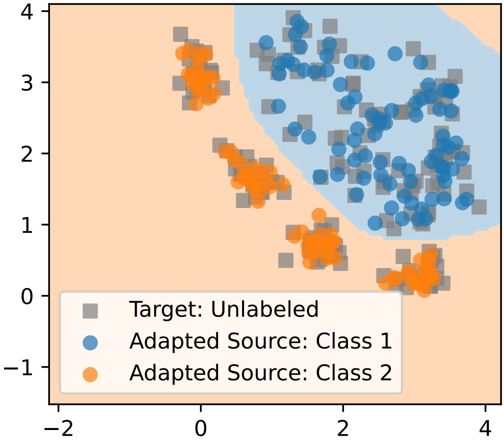

Step 1: Minimize a discrepancy between source and target domains

# Toy example of Unsupervised DA




Step 2: Adapt source distribution to target distribution

# Toy example of Unsupervised DA




Step 3: Train on adapted source and predict on target (ACC=1.00)

# DA methods

## Mapping

- Learn a mapping function $m$ to align source and target domains.
- The domain-adapted predictor becomes:

$$f_{\mathrm{DA}} = f \circ m$$

## Reweighting

$$f_{\mathrm{DA}} \in \arg\min_{f \in \mathcal{F}} \frac{1}{n_{\mathcal{S}}} \sum_{i=1}^{n_{\mathcal{S}}} w_i \, \ell(f(\mathbf{x}_i), y_i)$$

where $w_i$ are importance weights depending on source and target domains.

## End-to-End Deep Learning

- Jointly learn feature representations and a classifier using deep networks.
- Train with adversarial or discrepancy-based objectives to align source and target distributions.

# DA scorers

**Huge variety of DA methods** : mapping, subspace, reweighting, ...
with many different hyper-parameters to tune.
$\hookrightarrow$ **DA scorers** to validate hyper-parameters without using target label.

$$\mathrm{DA\_scorer}(\underbrace{f_{\mathrm{DA}}}_{\text{Predictive model}}, \underbrace{\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n_{\mathcal{S}}}}_{\text{Source data}}, \underbrace{\{\boldsymbol{x}_i\}_{i=1}^{n_{\mathcal{T}}}}_{\text{Target data}}) \in \mathbb{R}$$

Rely on **proxys** of the target accuracy.

# Many extensions

- Multi-source DA
- DA for regression problems
- Deep DA: deep learning models for DA
- Semi-supervised DA: few labeled target data
- Test-time DA: adapt to a new target domain with forgotten source data
- Heterogeneous DA: different feature spaces
- ...

# 2. **Skada** : a Scikit-Learn compatible library for shallow and deep DA

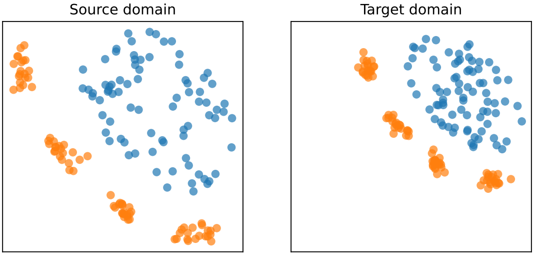
# Scikit-Adaptation: Skada

Implementation of Python library for domain adaptation, including:

- Homogeneous API for all DA methods: **Shallow and Deep learning** .
- **Sklearn-like API** with estimator class (.fit, .predict, ...), pipeline, grid search ...
- **DA scorer** to validate hyper-parameters without using target label.

# Example of conditional shift

```
from skada.datasets import make_shifted_datasets

dataset = make_shifted_datasets(
    n_samples_source=20,
    n_samples_target=20,
    shift="concept_drift",
    return_dataset=True
)
```



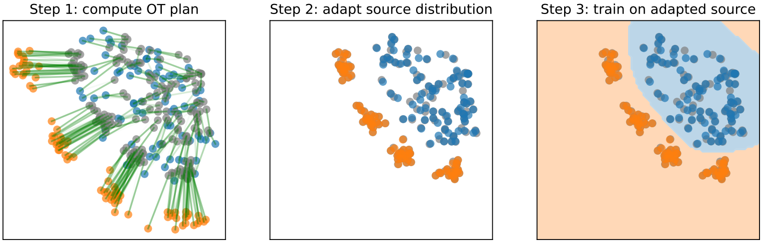
# Training OT Mapping model

```python
from skada import OTMapping

X, y, sample_domain = dataset.pack_train(as_sources=["s"], as_targets=["t"])
model = OTMapping()
model.fit(X, y, sample_domain=sample_domain)

X_target, y_target, sample_domain = dataset.pack_test(as_targets=["t"])
model.score(X_target, y_target)
```

```
y = [ 0  0  0  0  0  0  0  0  1  1  1  1  1  1  1  1 -1 -1 -1 -1 -1 -1 -1 -1
 -1 -1 -1 -1 -1 -1 -1 -1]
sample_domain = [ 1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1 -2 -2 -2 -2 -2 -2 -2 -2
 -2 -2 -2 -2 -2 -2 -2 -2]
```

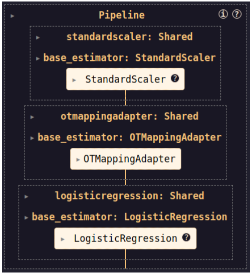
# What are Skada model: Sklearn pipeline

```python
from sklearn.svm import SVC

from skada import make_da_pipeline
from skada import OTMappingAdapter

model = make_da_pipeline(
    OTMappingAdapter(),
    SVC(),
)
```

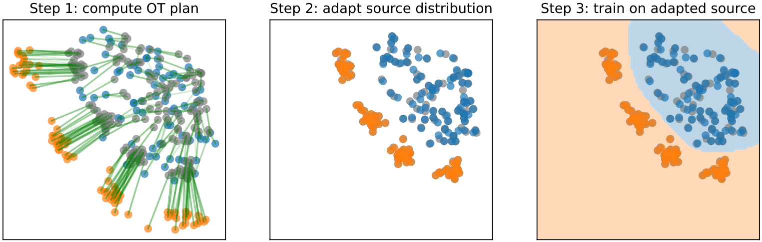


# What are Skada model: Sklearn pipeline

```python
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler

from skada import make_da_pipeline
from skada import OTMappingAdapter

model = make_da_pipeline(
    StandardScaler(),
    OTMappingAdapter(),
    LogisticRegression(),
)
```

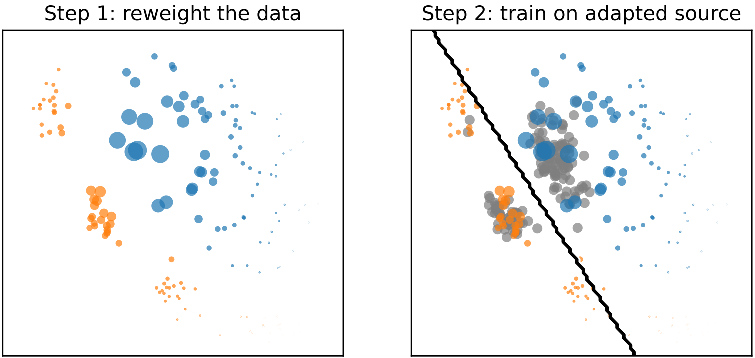


# Mapping methods



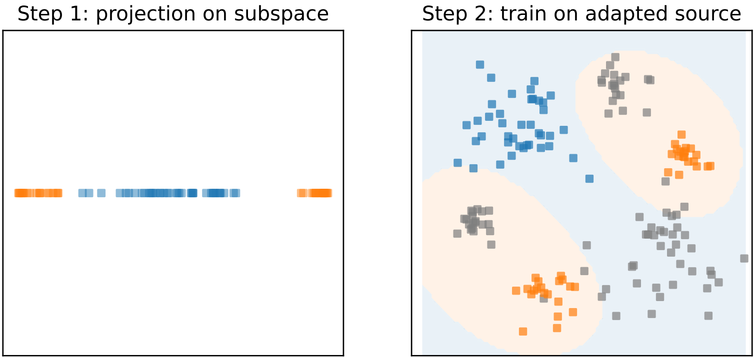

**6** available mapping methods in `Skada` : `CORAL`, `OTMapping`, `EntropicOTMapping`, `ClassRegularizerOTMapping`, `LinearOTMapping`, `MMDLSConSMapping`

# Reweighting methods




**6** available reweighting methods in `Skada` : `GaussianReweight`, `KLIEPReweight`, `KMMReweight`, `DiscriminatorReweight`, `NearestNeighborReweight`, `MMDTarSReweight`

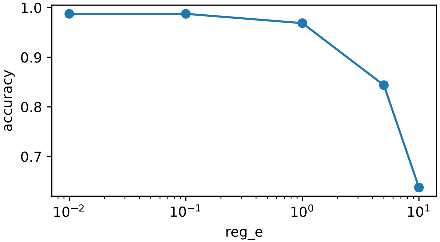
# Subspace methods




**4** available subspace methods in **Skada** : `SubspaceAlignment`, `TransferComponentAnalysis`, `TransferJointMatching`, `TransferSubspaceLearning`

# How to validate parameters in DA?

```python
from skada import EntropicOTMapping

model = EntropicOTMapping(reg_e=1e-1, base_estimator=SVC(probability=True))
```




How to validate the parameter `reg_e` ?

- Gridsearch on **source** $\rightarrow$ **Bad parameter**
- Gridsearch on **target** $\rightarrow$ **Cheating!** (no label available)

$\hookrightarrow$ Use specific **DA scorers** !

# Validation with DA scorers

```python
from sklearn.model_selection import GridSearchCV, ShuffleSplit

from skada import EntropicOTMapping
from skada.metrics import PredictionEntropyScorer

model = EntropicOTMapping(base_estimator=SVC(probability=True))

cv = ShuffleSplit(n_splits=5, test_size=0.3, random_state=42)
reg_e = [0.01, 0.03, 0.05, 0.08, 0.1]

grid_search = GridSearchCV(
    model,
    {"entropicotmappingadapter__reg_e": reg_e},
    cv=cv,
    scoring=PredictionEntropyScorer(),
)
grid_search.fit(X, y, sample_domain=sample_domain)
```

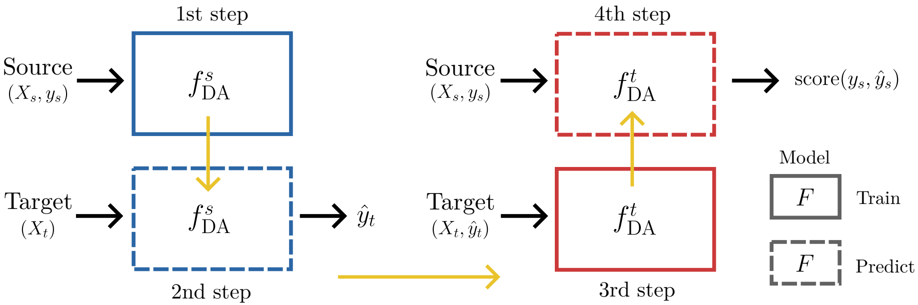
# Validation with DA scorers

```python
from sklearn.model_selection import GridSearchCV, ShuffleSplit

from skada import EntropicOTMapping
from skada.metrics import PredictionEntropyScorer

model = EntropicOTMapping(base_estimator=SVC(probability=True))

cv = ShuffleSplit(n_splits=5, test_size=0.3, random_state=42)
reg_e = [0.01, 0.03, 0.05, 0.08, 0.1]

grid_search = GridSearchCV(
    model,
    {"entropicotmappingadapter__reg_e": reg_e},
    cv=cv,
    scoring=PredictionEntropyScorer(),
)
grid_search.fit(X, y, sample_domain=sample_domain)
```

$\hookrightarrow$ **6** DA scorer available in `Skada`

# Circular validation [BM10]

- Train a model on **source** domain and predict on **target** domain.
- Train a model on **target** domain with **predicted label**.
- Predict on **source** domain and compute **accuracy**.



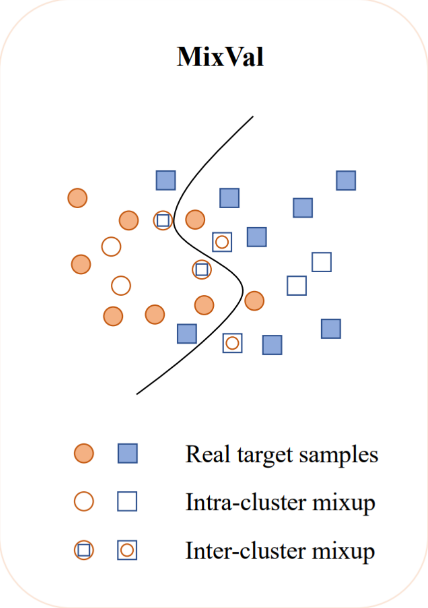
# Mixval scorer [HLL+23]

- Create **mixed sample** from source

$$\tilde{X} = \lambda X_i^s + (1-\lambda) X_j^s$$

$$\tilde{y} = \begin{cases} y_i^s & \text{if } \lambda \geq 0.5 \\ y_j^s & \text{otherwise} \end{cases}$$

- Distinguish between samples mixed from **same classes** and **different classes** .
- Validate on mixed samples.



# Other DA scorers

- Validate on **source** domain with **weighted samples** : `ImportanceWeightedScorer`, `DeepEmbeddedValidation`.
- Compute the **entropy** of the **target** predictions: `PredictionEntropyScorer`, `SoftNeighborhoodDensity`.

# Deep DA methods

Deep DA methods $\rightarrow$ Reduce **divergence** between source and target domains in the **embedding space** .

$$\ell = \sum_{i=1}^{n_{\mathcal{S}}} \overbrace{\ell_{\mathrm{CE}}}^{\text{Cross-entropy loss}} (f(g(\boldsymbol{x}_i^s)), y_i^s) + \underbrace{\mathrm{reg}}_{\text{Regularization term}} \sum_{i=1}^{n_{\mathcal{S}}} \sum_{j=1}^{n_{\mathcal{T}}} \underbrace{\ell_{\mathrm{DA}}}_{\text{DA loss}} (g(\boldsymbol{x}_i^s), g(\boldsymbol{x}_j^t), y_i^s)$$

- $f$ is the classifier, $g$ is the feature extractor.
- $\ell_{\mathrm{DA}}$ can compute **OT distance** , **MMD distance** , **adversarial loss** , ...

# Deep DA methods

- Wrapper of **Skorch**
- Param `layer_name` to know which **feature space** to consider
- Parameter `reg` is the **threshold** between classical and DA loss
- All the rest work the same as before: learning rate, batch size ...

```python
from skada.deep import DeepCoral
from skada.deep.modules import ToyCNN

model = DeepCoral(
    ToyCNN(),
    layer_name="feature_extractor",
    batch_size=128,
    max_epochs=5,
    reg=1,
    lr=1e-2,
)

model.fit(X, y, sample_domain=sample_domain)
model.score(X_target, y_target)
```

**5** available deep DA methods in `Skada` : `DeepCORAL`, `DANN`, `CDAN`, `DAN`, `DeepJDOT`.

# 3. **Skada-Bench** : Benchmarking Unsupervised DA Methods with Realistic Validation

# Question in DA field

3 main questions emerge after implementing **Skada** :

- Does the DA methods have been compared properly?
- How the different **DA methods** **perform** on **diverse datasets** (other than Computer Vision)?
- Is there a **best DA scorers** for all methods? for each specific type of method?

# Question in DA field

3 main questions emerge after implementing **Skada** :

- Does the DA methods have been compared properly?
- How the different **DA methods perform** on **diverse datasets** (other than Computer Vision)?
- Is there a **best DA scorers** for all methods? for each specific type of method?

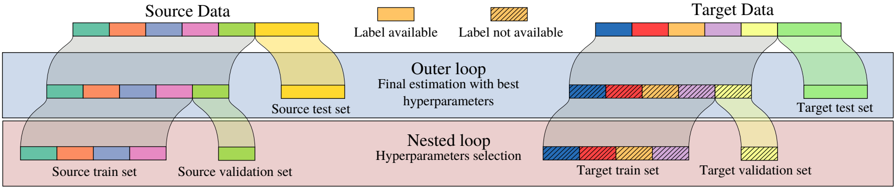
$\hookrightarrow$ **Benchmark** !

# Validation in the litterature

| | Method | Validation Procedure | Comment |
|---|---|---|---|
| Reweighting | Density Reweight [SM05] | None | Bandwidth fixed by Silverman method |
| | Discriminative Reweight [Shi00] | NA | No hyperparameters |
| | Gaussian Reweight [Shi00] | None | Not specified in [Shi00] |
| | KLIEP [SNK$^+$07] | Integrated CV | Likelihood CV [SNK$^+$07] on target |
| | KMM [HGB$^+$06] | None | Fixed data-dependent hyperparameters |
| | NN Reweight [Loo12] | None | Number of neighbors fixed to one |
| | MMDTarS [ZSMW13] | CV | Not specified if done on source or target |
| Mapping | Coral [SFS17] | NA | No hyperparameters |
| | OT mapping [CFTR17] | CV target/CircCV | Unclear in the text |
| | Lin. OT mapping [FLF20] | NA | No hyperparameters |
| | MMD-LS [ZSMW13] | CV | Not specified if done on source or target |
| Subsp. | SA [FHST13] | 2-fold CV on source | - |
| | TCA [PTKY11] | Validation on target | Target subset used to tune parameters |
| | TSL [STG10] | None | Not specified in [STG10] |
| Other | JDOT [CFHR17] | Reverse CV [ZFY$^+$10] | - |
| | OT label prop [SRGB14] | NA | No hyperparameters |
| | DASVM [BM10] | Circular Validation [BM10] | - |

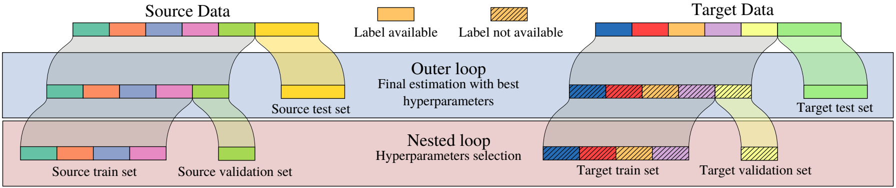
$\hookrightarrow$ **few realistic validation** procedure in the **literature** .

# Performance evaluation and Hyper-parameters selection




Nested cross-validation on source and target domains

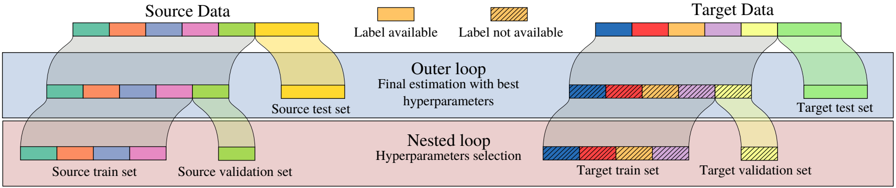
# Performance evaluation and Hyper-parameters selection




Nested cross-validation on source and target domains

- **Outer loop** : estimate generalization performance with unlabeled train target set but labeled test target set.

# Performance evaluation and Hyper-parameters selection




Nested cross-validation on source and target domains

- **Outer loop** : estimate generalization performance with unlabeled train target set but labeled test target set.
- **Nested loop** : select hyper-parameters with the unlabeled train target domain and a DA scorer.

# Dataset description

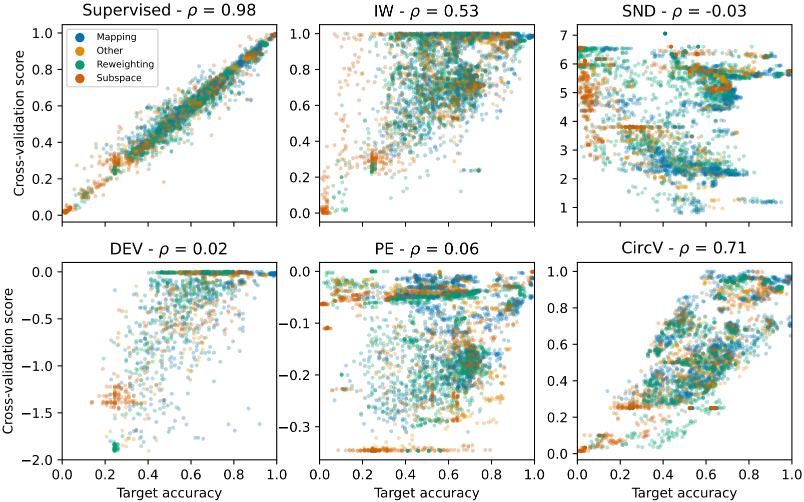
| Dataset | Modality | Preprocessing | # adapt | # classes | # samples | # features |
|---|---|---|---|---|---|---|
| `Office 31` [KTP17] | CV | Decaff [DJV+14] + PCA | 6 | 31 | $470 \pm 350$ | 100 |
| `Office Home` [VECP17] | CV | ResNet [HZRS16] + PCA | 12 | 65 | $3897 \pm 850$ | 100 |
| `MNIST/USPS` [LC15] | CV | Vect + PCA | 2 | 10 | 3000 / 10000 | 50 |
| `20 Newsgroup` [Lan95] | NLP | LLM [RG19, XLZM23] + PCA | 6 | 2 | $3728 \pm 174$ | 50 |
| `Amazon Review` [ML13, MTSvdH15] | NLP | LLM [RG19, XLZM23] + PCA | 12 | 4 | 2000 | 50 |
| `Mushrooms` [DYXY07] | Tabular | One Hot Encoding | 2 | 2 | $4062 \pm 546$ | 117 |
| `Phishing` [MTM12] | Tabular | NA | 2 | 2 | $5527 \pm 1734$ | 30 |
| `BCI` [TMA+12] | Biosignals | Cov+TS [BBCJ12] | 9 | 4 | 288 | 253 |

$\hookrightarrow$ **8** datasets from different modalities: Computer Vision, NLP, Tabular, Biosignals.

# Table results

| | | Cov. shift | Tar. shift | Cond. shift | Sub. shift | Office31 | OfficeHome | MNIST/USPS | 20NewsGroups | AmazonReview | Mushrooms | Phishing | BCI | Selected Scorer | Rank |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Train Src | 0.88 | 0.85 | 0.66 | 0.19 | 0.59 | 0.56 | 0.54 | 0.59 | 0.7 | 0.72 | 0.91 | 0.55 | | 9.75 |
| | Train Tgt | 0.92 | 0.93 | 0.82 | 0.98 | 0.88 | 0.8 | 0.96 | 1.0 | 0.73 | 1.0 | 0.97 | 0.64 | | 1.06 |
| Reweighting | Dens. RW [SM05] | 0.88 | 0.86 | 0.66 | 0.18 | 0.57 | 0.55 | 0.54 | 0.58 | 0.7 | 0.71 | 0.91 | 0.55 | IW | 10.76 |
| | Disc. RW [Shi00] | 0.85 | 0.83 | 0.71 | 0.18 | 0.58 | 0.53 | 0.5 | 0.6 | 0.68 | 0.75 | 0.91 | 0.56 | CircV | 11.12 |
| | Gauss. RW [Shi00] | 0.89 | 0.86 | 0.65 | 0.21 | 0.2 | 0.44 | 0.11 | 0.54 | 0.6 | 0.51 | 0.46 | 0.25 | CircV | 19.42 |
| | KLIEP [SNK+07] | 0.88 | 0.86 | 0.66 | 0.19 | 0.59 | 0.56 | 0.54 | 0.6 | 0.69 | 0.72 | 0.91 | 0.55 | CircV | 10.36 |
| | KMM [HGB+06] | 0.89 | 0.87 | 0.64 | 0.15 | 0.58 | 0.55 | 0.52 | 0.7 | 0.57 | 0.74 | 0.91 | 0.52 | CircV | 12.11 |
| | NN RW [Loo12] | 0.89 | 0.86 | 0.67 | 0.15 | 0.58 | 0.55 | 0.54 | 0.59 | 0.66 | 0.71 | 0.91 | 0.54 | CircV | 11.91 |
| | MMDTarS [ZSMW13] | 0.88 | 0.86 | 0.64 | 0.2 | 0.56 | 0.55 | 0.54 | 0.59 | 0.7 | 0.74 | 0.91 | 0.55 | IW | 9.51 |
| Mapping | CORAL [SFS17] | 0.66 | 0.84 | 0.66 | 0.19 | 0.59 | 0.57 | 0.62 | 0.73 | 0.69 | 0.72 | 0.92 | 0.62 | CircV | 7.10 |
| | MapOT [CFTR17] | 0.72 | 0.57 | 0.82 | 0.02 | 0.55 | 0.51 | 0.61 | 0.76 | 0.67 | 0.63 | 0.84 | 0.47 | PE | 10.98 |
| | EntOT [CFTR17] | 0.71 | 0.6 | 0.82 | 0.12 | 0.58 | 0.58 | 0.6 | 0.83 | 0.62 | 0.75 | 0.86 | 0.54 | CircV | 9.75 |
| | ClassRegOT [CFTR17] | 0.74 | 0.58 | 0.81 | 0.11 | 0.61 | 0.53 | 0.62 | 0.96 | 0.68 | 0.82 | 0.88 | 0.52 | IW | 8.71 |
| | LinOT [FLF20] | 0.73 | 0.73 | 0.76 | 0.18 | 0.59 | 0.57 | 0.64 | 0.82 | 0.7 | 0.76 | 0.91 | 0.61 | CircV | 5.33 |
| | MMD-LS [ZSMW13] | 0.65 | 0.68 | 0.81 | 0.52 | 0.55 | 0.54 | 0.52 | 0.97 | 0.68 | 0.86 | 0.88 | 0.56 | IW | 9.66 |
| Subspace | JPCA | 0.88 | 0.85 | 0.66 | 0.15 | 0.55 | 0.47 | 0.51 | 0.77 | 0.69 | 0.78 | 0.9 | 0.54 | PE | 8.77 |
| | SA [FHST13] | 0.74 | 0.68 | 0.8 | 0.11 | 0.59 | 0.57 | 0.56 | 0.88 | 0.66 | 0.88 | 0.89 | 0.53 | CircV | 8.53 |
| | TCA [PTKY11] | 0.46 | 0.48 | 0.55 | 0.56 | 0.04 | NA | 0.11 | 0.57 | 0.6 | 0.45 | NA | 0.27 | CircV | 19.57 |
| | TSL [STG10] | 0.88 | 0.85 | 0.66 | 0.19 | 0.59 | 0.2 | 0.25 | 0.68 | 0.7 | 0.56 | 0.86 | 0.25 | IW | 14.65 |
| Other | JDOT [CFHR17] | 0.72 | 0.57 | 0.82 | 0.14 | 0.6 | 0.51 | 0.63 | 0.77 | 0.67 | 0.63 | 0.8 | 0.46 | DEV | 10.12 |
| | OTLabelProp [SRGB14] | 0.72 | 0.59 | 0.81 | 0.05 | 0.61 | 0.56 | 0.62 | 0.86 | 0.67 | 0.64 | 0.86 | 0.5 | CircV | 10.49 |
| | DASVM [BM10] | 0.89 | 0.86 | 0.65 | 0.14 | NA | NA | NA | 0.68 | NA | 0.78 | 0.88 | NA | CircV | 11.00 |

# Results



Supervised - $\rho$ = 0.98
IW - $\rho$ = 0.53
SND - $\rho$ = -0.03
DEV - $\rho$ = 0.02
PE - $\rho$ = 0.06
CircV - $\rho$ = 0.71
Mapping
Other
Reweighting
Subspace
Cross-validation score
Target accuracy

# Conclusion

**`Skada`**

- First release few month ago
- Still a lot of work: examples, documentation of `skada.deep` ...

**`Skada-bench`**

- Benchmark only on **shallow** DA methods
- Currently adding **deep** DA methods

# References I

Alexandre Barachant, Stéphane Bonnet, Marco Congedo, and Christian Jutten, *Multiclass brain–computer interface classification by riemannian geometry*, IEEE Transactions on Biomedical Engineering **59** (2012), no. 4, 920–928.

Lorenzo Bruzzone and Mattia Marconcini, *Domain adaptation problems: A dasvm classification technique and a circular validation strategy*, IEEE Transactions on Pattern Analysis and Machine Intelligence **32** (2010), no. 5, 770–787.

Nicolas Courty, Rémi Flamary, Amaury Habrard, and Alain Rakotomamonjy, *Joint distribution optimal transportation for domain adaptation*, Advances in Neural Information Processing Systems (I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, Curran Associates, Inc., 2017.

# References II


- Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy, *Optimal transport for domain adaptation*, IEEE Transactions on Pattern Analysis and Machine Intelligence **39** (2017), no. 9, 1853–1865.

- Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell, *Decaf: A deep convolutional activation feature for generic visual recognition*, Proceedings of the 31st International Conference on Machine Learning (Bejing, China) (Eric P. Xing and Tony Jebara, eds.), Proceedings of Machine Learning Research, vol. 32, PMLR, 22–24 Jun 2014, pp. 647–655.

- Wenyuan Dai, Qiang Yang, Gui-Rong Xue, and Yong Yu, *Boosting for transfer learning*, Proceedings of the 24th International Conference on Machine Learning (New York, NY, USA), ICML '07, Association for Computing Machinery, 2007, p. 193–200.

# References III


Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars, *Unsupervised visual domain adaptation using subspace alignment*, 2013 IEEE International Conference on Computer Vision, 2013, pp. 2960–2967.

Rémi Flamary, Karim Lounici, and André Ferrari, *Concentration bounds for linear monge mapping estimation and optimal transport domain adaptation*, 2020.

Jiayuan Huang, Arthur Gretton, Karsten Borgwardt, Bernhard Schölkopf, and Alex Smola, *Correcting sample selection bias by unlabeled data*, Advances in Neural Information Processing Systems (B. Schölkopf, J. Platt, and T. Hoffman, eds.), vol. 19, MIT Press, 2006.

# References IV


- Dapeng Hu, Jian Liang, Jun Hao Liew, Chuhui Xue, Song Bai, and Xinchao Wang, *Mixed samples as probes for unsupervised model selection in domain adaptation*, Advances in Neural Information Processing Systems (A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, eds.), vol. 36, Curran Associates, Inc., 2023, pp. 37923–37941.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, *Deep residual learning for image recognition*, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.
- Piotr Koniusz, Yusuf Tas, and Fatih Porikli, *Domain adaptation by mixture of alignments of second-or higher-order scatter tensors*, 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 7139–7148.

# References V


- Ken Lang, *Newsweeder: Learning to filter netnews*, Proceedings of the Twelfth International Conference on Machine Learning, 1995, pp. 331–339.
- Zhibin Liao and Gustavo Carneiro, *Competitive multi-scale convolution*, 2015.
- M Loog, *Nearest neighbor-based importance weighting*, The 2012 IEEE workshop on machine learning for signal processing (MLSP) proceedings (United States) (SN, ed.), IEEE, 2012, The 2012 IEEE workshop on machine learning for signal processing (MLSP) ; Conference date: 23-09-2012 Through 26-09-2012, pp. 1–6 (English).
- Julian McAuley and Jure Leskovec, *Hidden factors and hidden topics: understanding rating dimensions with review text*, Proceedings of the 7th ACM conference on Recommender systems, 2013, pp. 165–172.

# References VI


- Rami M. Mohammad, Fadi Thabtah, and Lee McCluskey, *An assessment of features related to phishing websites using an automated technique*, 2012 International Conference for Internet Technology and Secured Transactions, 2012, pp. 492–497.

- Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel, *Image-based recommendations on styles and substitutes*, Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval (New York, NY, USA), SIGIR '15, Association for Computing Machinery, 2015, p. 43–52.

- Sinno Jialin Pan, Ivor W. Tsang, James T. Kwok, and Qiang Yang, *Domain adaptation via transfer component analysis*, IEEE Transactions on Neural Networks **22** (2011), no. 2, 199–210.

# References VII


- Nils Reimers and Iryna Gurevych, *Sentence-bert: Sentence embeddings using siamese bert-networks*, EMNLP/IJCNLP (1) (Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, eds.), Association for Computational Linguistics, 2019, pp. 3980–3990.
- Baochen Sun, Jiashi Feng, and Kate Saenko, *Correlation alignment for unsupervised domain adaptation*, pp. 153–171, Springer International Publishing, Cham, 2017.
- Hidetoshi Shimodaira, *Improving predictive inference under covariate shift by weighting the log-likelihood function*, Journal of Statistical Planning and Inference **90** (2000), no. 2, 227–244.
- Masashi Sugiyama and Klaus-Robert Müller, *Input-dependent estimation of generalization error under covariate shift*, Statistics & Risk Modeling **23** (2005), no. 4, 249–279.

# References VIII


- Masashi Sugiyama, Shinichi Nakajima, Hisashi Kashima, Paul Buenau, and Motoaki Kawanabe, *Direct importance estimation with model selection and its application to covariate shift adaptation*, Advances in Neural Information Processing Systems (J. Platt, D. Koller, Y. Singer, and S. Roweis, eds.), vol. 20, Curran Associates, Inc., 2007.
- Justin Solomon, Raif Rustamov, Leonidas Guibas, and Adrian Butscher, *Wasserstein propagation for semi-supervised learning*, Proceedings of the 31st International Conference on Machine Learning (Bejing, China) (Eric P. Xing and Tony Jebara, eds.), Proceedings of Machine Learning Research, vol. 32, PMLR, 22–24 Jun 2014, pp. 306–314.
- Si Si, Dacheng Tao, and Bo Geng, *Bregman divergence-based regularization for transfer subspace learning*, IEEE Transactions on Knowledge and Data Engineering **22** (2010), no. 7, 929–942.

# References IX


Michael Tangermann, Klaus-Robert Müller, Ad Aertsen, Niels Birbaumer, Christoph Braun, Clemens Brunner, Robert Leeb, Carsten Mehring, Kai Miller, Gernot Mueller-Putz, Guido Nolte, Gert Pfurtscheller, Hubert Preissl, Gerwin Schalk, Alois Schlögl, Carmen Vidaurre, Stephan Waldert, and Benjamin Blankertz, *Review of the BCI Competition IV*, Frontiers in Neuroscience **6** (2012).

Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan, *Deep hashing network for unsupervised domain adaptation*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff, *C-pack: Packaged resources to advance general chinese embedding*, 2023.

# References X

Erheng Zhong, Wei Fan, Qiang Yang, Olivier Verscheure, and Jiangtao Ren, *Cross validation framework to choose amongst models and datasets for transfer learning*, Machine Learning and Knowledge Discovery in Databases (Berlin, Heidelberg) (José Luis Balcázar, Francesco Bonchi, Aristides Gionis, and Michèle Sebag, eds.), Springer Berlin Heidelberg, 2010, pp. 547–562.

Kun Zhang, Bernhard Schölkopf, Krikamol Muandet, and Zhikun Wang, *Domain adaptation under target and conditional shift*, Proceedings of the 30th International Conference on Machine Learning (Atlanta, Georgia, USA) (Sanjoy Dasgupta and David McAllester, eds.), Proceedings of Machine Learning Research, vol. 28, PMLR, 17–19 Jun 2013, pp. 819–827.