Multi-Source Domain Adaptation for Learning on Biosignals

Adaptation de Domaine Multi-Sources pour l'Apprentissage sur les Bio-Signaux

Théo Gnassounou

Jury:

- Maarten De Vos, *Professeur à Leuven University*
- Nicolas Thome, *Professeur à Sorbonne Université*
- Agnès Desolneux, *Directrice de Recherche à Centre Borelli*
- Romain Tavenard, *Professeur à Université de Rennes*
- Alain Rakotomamonjy, *Professeur à Université de Rouen*

Introduction

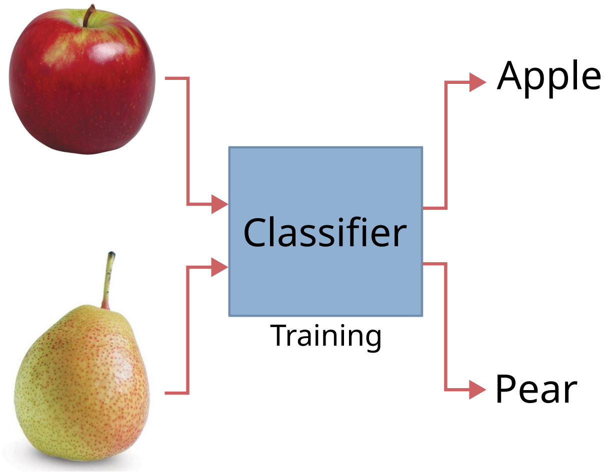
SKADA & Benchmark

Monge mapping for Biosignals

Introduction to Domain Adaptation

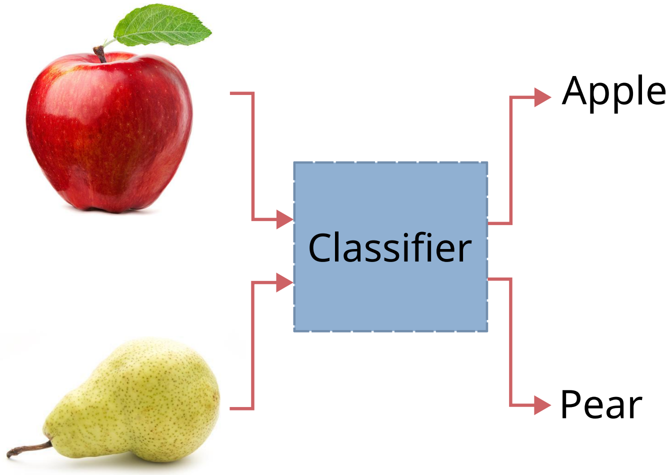
Machine Learning: a powerful tool

Le machine learning permet, par exemple, d'apprendre à un modèle a reconnaître des pommes et des poires.



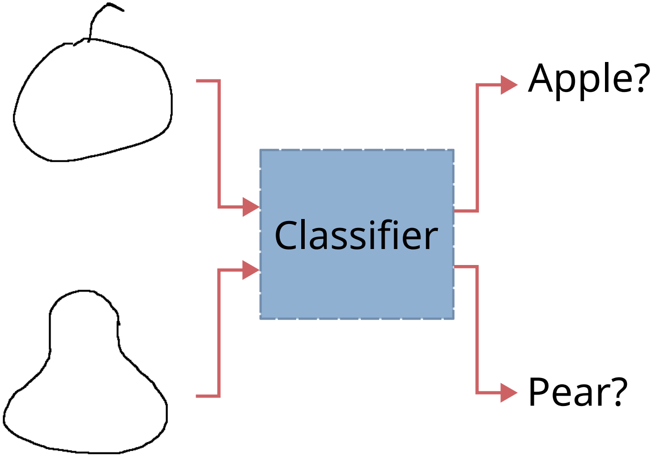
Machine Learning: a powerful tool

Le modèle peut ensuite reconnaitre de nouvelles images de pommes et poires.



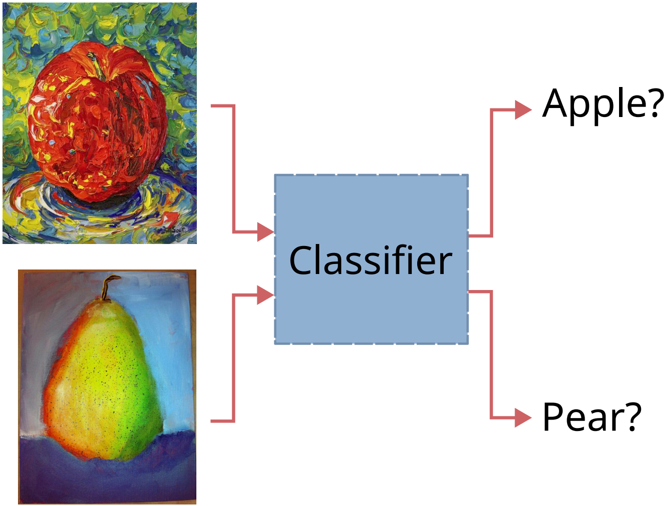
Machine Learning: but shift happens . . .

Cependant, le modèle peut avoir des difficultés à reconnaître des images de pommes et poires dessinées.



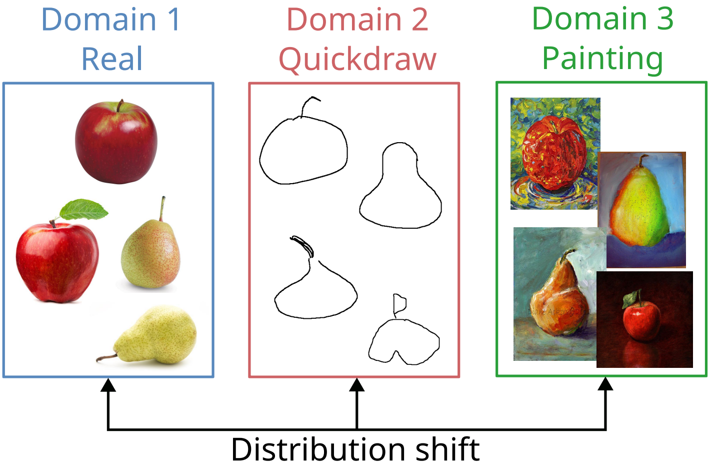
Machine Learning: but shift happens . . .

Ou bien peintes.



Distribution shift in Image classification[1]

En adaptation de domaine, on considère different domaines, qui decrivent la même chose (pomme et poire) mais de manière differentes.



[1] *Dataset: DomainNet (Peng et. al., 2019)*

Distribution shift in Autonomous Driving

On aurait ce problème la si on entraine une voiture à conduire seule pendant une beau temps et qu'ensuite il pleuve.

Domain 1
Clear Images[1]

Domain 2
Foggy Images[2]

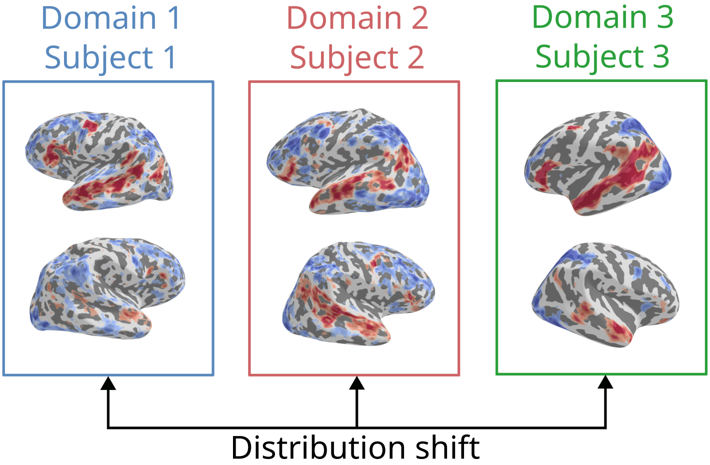
Domain 3
Rainy Images[3]

Distribution shift

Dataset: [1] Cityscapes (Cordts et. al., 2016) [2] Foggy Cityscapes (Sakaridis et. al., 2018) [3] Rainy Cityscapes (Li et. al., 2024)

Distribution shift in Functional MRI[1]

Ou bien si on veut généraliser un modèle entrainé sur un patient sur d'autre patient d'IRM avec une structure de cerveau different.



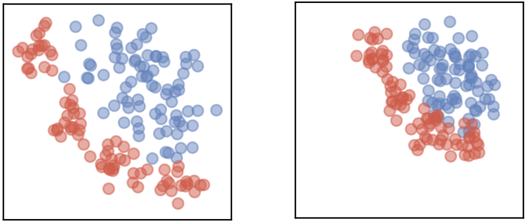
[1] *Dataset: IBC Project (Pinho et. al., 2018)*

Impact of Distribution Shift

Considerons maintenant un problème de classification binaire avec deux classes, la classe bleue et la classe rouge.

Source data
$(\mathbf{X}_s, \mathbf{y}_s)$

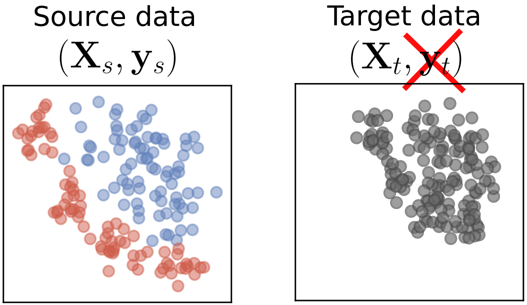
Target data
$(\mathbf{X}_t, \mathbf{y}_t)$



- Suppose classification problem with two classes (**blue** and **red**).
- Source domain (left): **labeled** data.

Impact of Distribution Shift

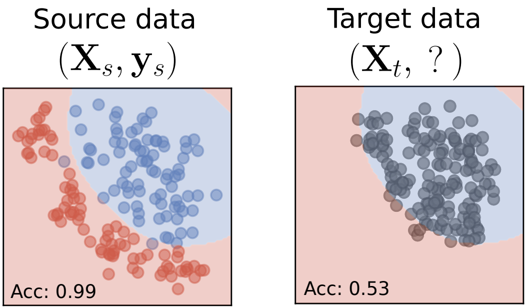
On dispose de données étiquetées dans le domaine source, à gauche, et de données non étiquetées dans le domaine cible, à droite.



- Suppose classification problem with two classes (**blue** and **red**).
- Source domain (left): **labeled** data.
- Target domain (right): **unlabeled** data.

Impact of Distribution Shift

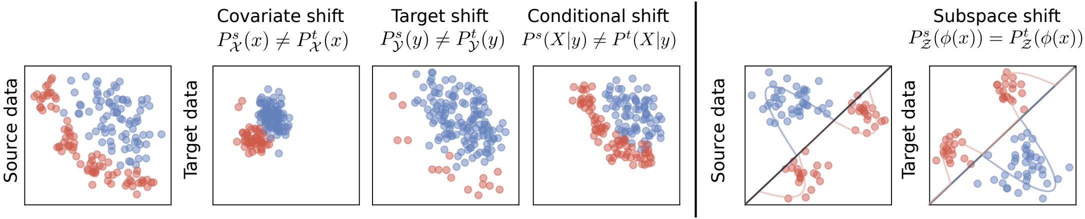
Lorsqu'on applique le classifieur entrainé sur le domaine source aux données du domaine cible, on observe une chute de performance importante.



- Suppose classification problem with two classes (**blue** and **red**).
- Source domain (left): **labeled** data.
- Target domain (right): **unlabeled** data.
- Drop in performance when applying source classifier to target data.

Existing Type of Distribution Shifts

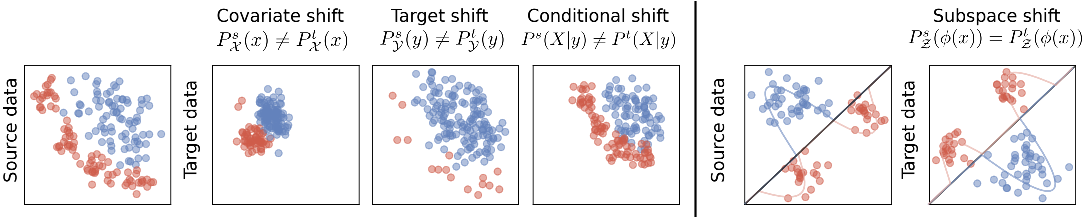
Plusieurs types de shifts existent en adaptation de domaine.



- **Covariate shift**: Distribution of **data** changes over domains.
- **Target shift**: Distribution of **labels** changes over domains.
- **Conditional shift**: Distribution of **data conditioned on labels** changes over domains.
- **Subspace Assumption**: Exist a **subspace that is invariant** over domains.

Tackling Shift with Domain Adaptation (DA)

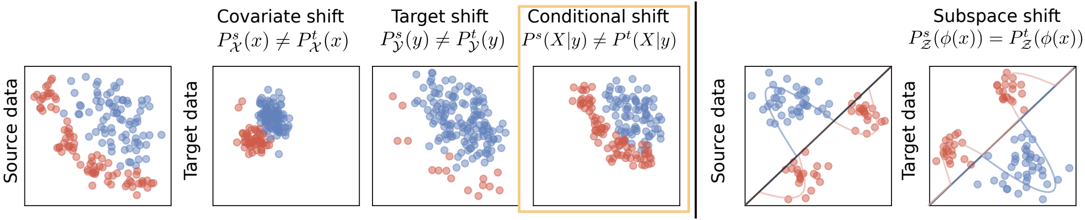
Et pour chacun des types de shifts, des méthodes existent pour y remédier.



- **Covariate shift** $\rightarrow$ Reweighting methods
- **Target shift** $\rightarrow$ Reweighting methods
- **Conditional shift** $\rightarrow$ Mapping methods
- **Subspace Assumption** $\rightarrow$ Subspace methods $+$ Deep Learning Methods

Tackling Shift with Domain Adaptation (DA)

Et pour chacun des types de shifts, des méthodes existent pour y remédier.

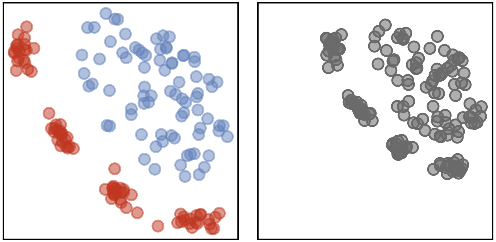


- **Covariate shift** → Reweighting methods
- **Target shift** → Reweighting methods
- **Conditional shift** → Mapping methods
- **Subspace Assumption** → Subspace methods + Deep Learning Methods

Example of Mapping DA method

Revenons au problème de classification binaire.

Source data $(\mathbf{X}_s, \mathbf{y}_s)$ | Target data $(\mathbf{X}_t, \ ?\ )$

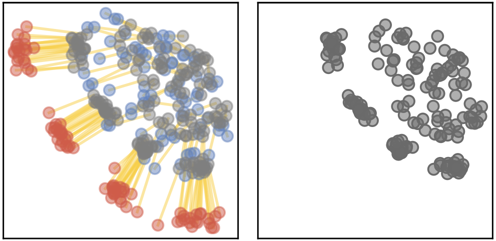


- Goal: Estimates a **mapping** from source to target domain.

Example of Mapping DA method

L'objectif de ces méthodes est d'estimer une projection entre le domaine source et le domaine cible.

Source data $(\mathbf{X}_s, \mathbf{y}_s)$ Target data $(\mathbf{X}_t, \; ? \;)$



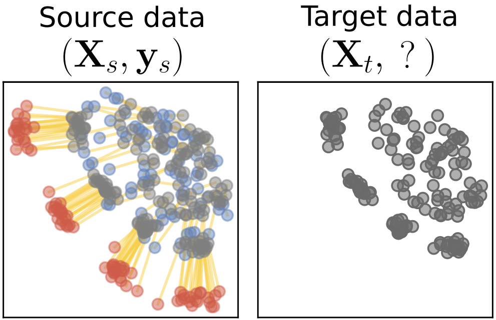
- Goal: Estimates a **mapping** from source to target domain.
- **Covariance mapping**[1]:

$$m(\mathbf{X}_s^i) = \mathbf{\Sigma}_t^{1/2}\mathbf{\Sigma}_s^{-1/2}\mathbf{X}_s^i$$

[1] *Sun et. al., 2017*

Example of Mapping DA method

L'objectif de ces méthodes est d'estimer une projection entre le domaine source et le domaine cible.



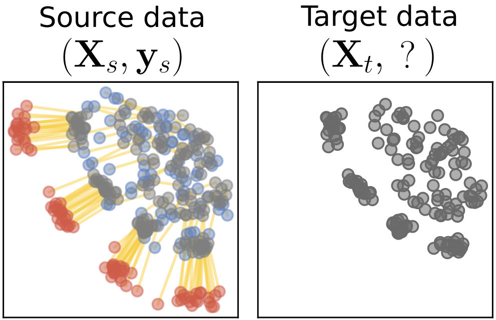
- Goal: Estimates a **mapping** from source to target domain.
- **Optimal Transport**[2]:

$$m(\mathbf{X}_s^i) = \sum_j \mathbf{T}_{ij}\mathbf{X}_t^j \quad \text{with} \quad \mathbf{T} = \underset{\mathbf{T}}{\operatorname{argmin}} \langle \mathbf{C}, \mathbf{T} \rangle_F$$

[2] *Courty et. al., 2016*

Example of Mapping DA method

L'objectif de ces méthodes est d'estimer une projection entre le domaine source et le domaine cible.



- Goal: Estimates a **mapping** from source to target domain.
- **Entropic Optimal Transport**[3]:

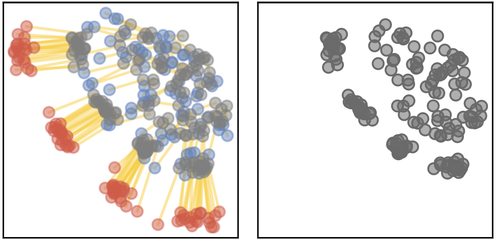
$$m(\mathbf{X}_s^i) = \sum_j \mathbf{T}_{ij}\mathbf{X}_t^j \quad \text{with} \quad \mathbf{T} = \underset{\mathbf{T}}{\operatorname{argmin}} \langle \mathbf{C}, \mathbf{T} \rangle_F + \lambda \sum_{i,j} \mathbf{T}_{ij} \log(\mathbf{T}_{ij} - 1)$$

[3] *Courty et. al., 2016*

Example of Mapping DA method

L'objectif de ces méthodes est d'estimer une projection entre le domaine source et le domaine cible.

Source data $(\mathbf{X}_s, \mathbf{y}_s)$ | Target data $(\mathbf{X}_t, \ ? \ )$



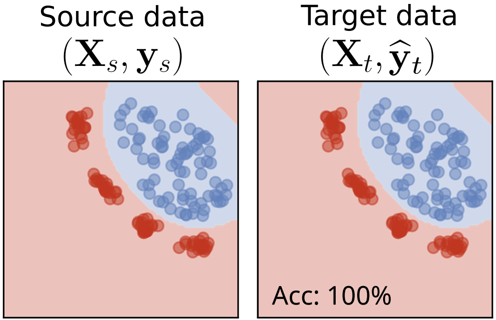
- Goal: Estimates a **mapping** from source to target domain.
- **Linear Optimal Transport**[4]:

$$m(\mathbf{X}_s^i) = \mathbf{A}\mathbf{X}_s^i \quad \text{with} \quad \mathbf{A} = \mathbf{\Sigma}_s^{-\frac{1}{2}} \left(\mathbf{\Sigma}_s^{\frac{1}{2}} \mathbf{\Sigma}_t \mathbf{\Sigma}_s^{\frac{1}{2}} \right)^{\frac{1}{2}} \mathbf{\Sigma}_s^{-\frac{1}{2}}$$

[4] *Flamary et. al., 2019*

Example of Mapping DA method

Une fois la projection estimée, on peut entrainer le classifieur sur les données projetées du domaine source et l'appliquer aux données du domaine cible.



- Goal: Estimates a **mapping** from source to target domain.
- **Linear Optimal Transport**[4]:

$$m(\mathbf{X}_s^i) = \mathbf{A}\mathbf{X}_s^i \quad \text{with} \quad \mathbf{A} = \mathbf{\Sigma}_s^{-\frac{1}{2}} \left(\mathbf{\Sigma}_s^{\frac{1}{2}} \mathbf{\Sigma}_t \mathbf{\Sigma}_s^{\frac{1}{2}} \right)^{\frac{1}{2}} \mathbf{\Sigma}_s^{-\frac{1}{2}}$$

- Train the classifier on mapped source data & apply it to target data.

[4] *Flamary et. al., 2019*

Challenges in Domain Adaptation

Lack of realistic and reproducible benchmarks in DA

	Method
Reweighting	Density Reweight
	Discriminative Reweight
	Gaussian Reweight
	KLIEP
	KMM
	NN Reweight
	MMDTarS
Mapping	Coral
	OT mapping
	Lin. OT mapping
	MMD-LS
Subsp.	SA
	TCA
	TSL
Other	JDOT
	OT label prop
	DASVM

Lack of realistic and reproducible benchmarks in DA

	Method
Reweighting	Density Reweight
	Discriminative Reweight
	Gaussian Reweight
	KLIEP
	KMM
	NN Reweight
	MMDTarS
Mapping	Coral
	OT mapping
	Lin. OT mapping
	MMD-LS
Subsp.	SA
	TCA
	TSL
Other	JDOT
	OT label prop
	DASVM

1. Many existing DA methods in the literature and few open-source implementations:
   - Introduce **SKADA**, a new open-source DA library[1].

[1] *Gnassounou et. al. 2024*

Lack of realistic and reproducible benchmarks in DA

	Method	Validation Procedure
Reweighting	Density Reweight	**None**
	Discriminative Reweight	NA
	Gaussian Reweight	**None**
	KLIEP	Integrated CV
	KMM	**None**
	NN Reweight	**None**
	MMDTarS	CV
Mapping	Coral	NA
	OT mapping	**CV target/CircCV**
	Lin. OT mapping	NA
	MMD-LS	**CV**
Subsp.	SA	**2-fold CV on source**
	TCA	**Validation on target**
	TSL	**None**
Other	JDOT	Reverse CV
	OT label prop	NA
	DASVM	Circular Validation

1. Many existing DA methods in the literature and few open-source implementations:
   - Introduce **SKADA**, a new open-source DA library[1].

[1] *Gnassounou et. al. 2024*

Lack of realistic and reproducible benchmarks in DA

	Method	Validation Procedure
Reweighting	Density Reweight	**None**
	Discriminative Reweight	NA
	Gaussian Reweight	**None**
	KLIEP	Integrated CV
	KMM	**None**
	NN Reweight	**None**
	MMDTarS	CV
Mapping	Coral	NA
	OT mapping	**CV target/CircCV**
	Lin. OT mapping	NA
	MMD-LS	**CV**
Subsp.	SA	**2-fold CV on source**
	TCA	**Validation on target**
	TSL	**None**
Other	JDOT	Reverse CV
	OT label prop	NA
	DASVM	Circular Validation

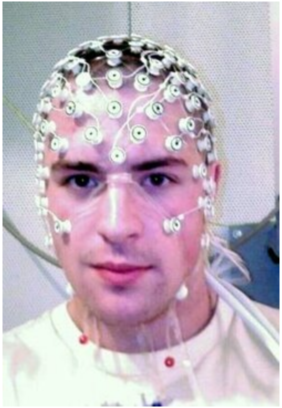
1. Many existing DA methods in the literature and few open-source implementations:
   - Introduce **SKADA**, a new open-source DA library[1].

2. Hard to validate DA methods in practice:
   - Introduce **SKADA**-bench, a new DA benchmark[2].

[1] *Gnassounou et. al. 2024*

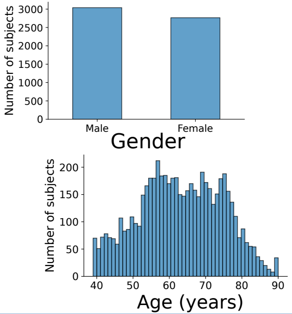
[2] *Lalou, Gnassounou, Collas, De Mathelin, Kachaiev, Odonnat, Gramfort, Moreau, Flamary, TMLR, 2025*

Introduction to Electroencephalography (EEG)

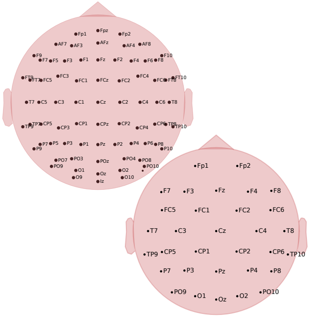


- Non-invasive technique to **record brain activity**.
- **Different electrodes** placed on the scalp measure voltage fluctuations.
- Used in various applications: **sleep staging**, epilepsy detection, **brain-computer interfaces**, cognitive load monitoring.

Variability in population



Variability in devices



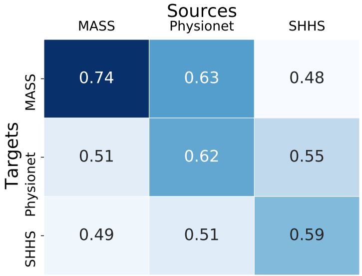
Variability in interpretation[1]

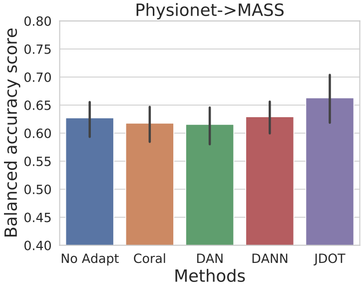
Scorer	Score
Expert 1	0.76 ± 0.11
Expert 2	0.78 ± 0.07
Expert 3	$\mathbf{0.79 \pm 0.07}$
Expert 4	0.72 ± 0.11
Expert 5	0.78 ± 0.08

Distribution shift

Table 3: Perslev et. al. 2021

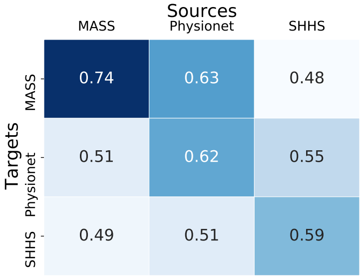
Tackle distribution shift in Biosignals

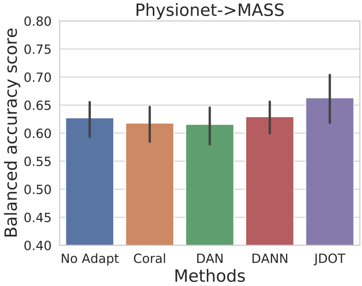




Shift happens . . .

Tackle distribution shift in Biosignals



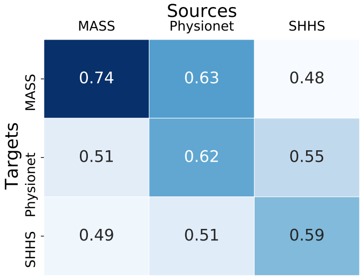


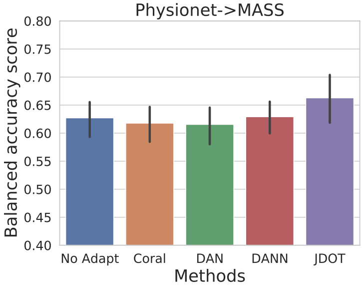
Shift happens . . .

1. How to reduce the drop in performance when testing on a new subject?
   - Introduce **Monge Mapping Normalisation** based on Optimal Transport[1].

[1] *Gnassounou, Gramfort, Flamary, Neurips, 2023*

Tackle distribution shift in Biosignals





Shift happens . . .

1. How to reduce the drop in performance when testing on a new subject?
   - Introduce **Monge Mapping Normalisation** based on Optimal Transport[1].
2. How to deal with multiple channels?
   - Extend Monge Mapping Normalisation to a **multi-channels** setting[2].
3. How to incorporate in deep learning architectures?
   - Introduce **PSDNorm**, a new layer for deep learning architectures[3].

[1] *Gnassounou, Gramfort, Flamary, Neurips, 2023*
[2] *Gnassounou, Collas, Gramfort, Lounici, Flamary, submitted at JMLR, 2024*
[3] *Gnassounou, Collas, Gramfort, Flamary, submitted at ICLR, 2025*

SKADA and SKADA-bench

SKADA: a Open-source Python library for DA methods.

- **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.

SKADA

SKADA: a Open-source Python library for DA methods.

- **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.

```python
from skada.datasets import make_shifted_datasets

X, y, sample_domain = make_shifted_datasets(
    shift='conditional_shift',
)
```

SKADA: a Open-source Python library for DA methods.

- **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.

```python
from skada import OTMapping

estimator = OTMapping()
```

SKADA: a Open-source Python library for DA methods.

- **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.

```python
from skada import EntropicOTMapping

estimator = EntropicOTMapping(reg_e=0.1)
```

SKADA: a Open-source Python library for DA methods.

- **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.

```python
from skada import EntropicOTMapping

estimator = EntropicOTMapping(reg_e=0.1)
estimator.fit(X, y, sample_domain=sample_domain)
```

Validation procedure in DA

Source data

$$(\mathbf{X}_s, \mathbf{y}_s)$$

Target data

$$(\mathbf{X}_t, \ ? \ )$$

Validation procedure in DA

Source data

$$(\mathbf{X}_s, \mathbf{y}_s)$$

Target data

$$(\mathbf{X}_t, \ ? \ )$$

1st solution: Use **only source data** to validate hyper-parameters.

$$(\mathbf{X}_s, \mathbf{y}_s) \qquad\qquad (\mathbf{X}_t, \ ? \ )$$

$$(\mathbf{X}_{\text{train}}, \mathbf{y}_{\text{train}}) \qquad\qquad (\mathbf{X}_{\text{val}}, \mathbf{y}_{\text{val}})$$

Validation procedure in DA

Source data

$$(\mathbf{X}_s, \mathbf{y}_s)$$

Target data

$$(\mathbf{X}_t, \ ? \ )$$

1st solution: Use **only source data** to validate hyper-parameters.

$$(\mathbf{X}_s, \mathbf{y}_s) \qquad (\mathbf{X}_t, \ ? \ )$$

$$(\mathbf{X}_{\text{train}}, \mathbf{y}_{\text{train}}) \qquad (\mathbf{X}_{\text{val}}, \mathbf{y}_{\text{val}})$$

Validation procedure in DA

Source data

$$(\mathbf{X}_s, \mathbf{y}_s)$$

Target data

$$(\mathbf{X}_t, \ ? \ )$$

2nd solution: Use **target data** to validate hyper-parameters.

$$(\mathbf{X}_s, \mathbf{y}_s) \quad\quad (\mathbf{X}_t, \ ? \ )$$

$$\downarrow \quad\quad\quad\quad \downarrow$$

$$(\mathbf{X}_{\text{train}}, \mathbf{y}_{\text{train}}) \quad\quad (\mathbf{X}_{\text{val}}, \mathbf{y}_{\text{val}})$$

Source data

$$(\mathbf{X}_s, \mathbf{y}_s)$$

Target data

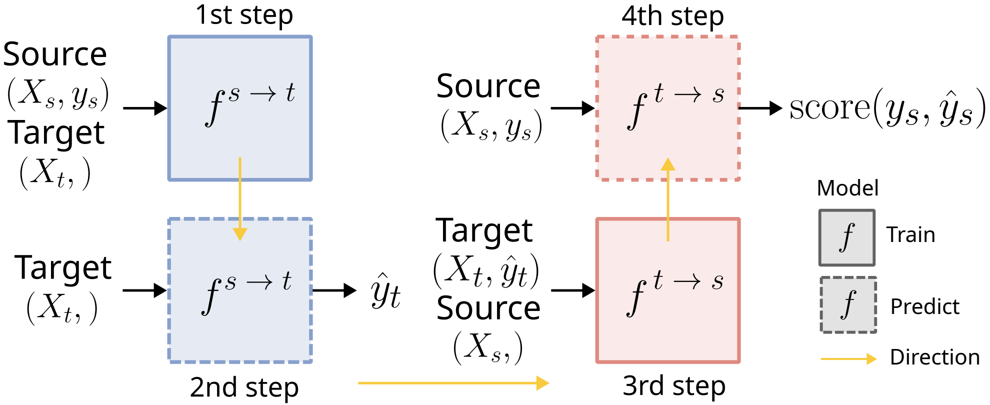
$$(\mathbf{X}_t, \ ? \ )$$

2nd solution: Use **target data** to validate hyper-parameters.

$$\cancel{(\mathbf{X}_s, \mathbf{y}_s)} \qquad \cancel{(\mathbf{X}_t, \ ? \ )}$$

$$\cancel{(\mathbf{X}_{\text{train}}, \mathbf{y}_{\text{train}})} \qquad \cancel{(\mathbf{X}_{\text{val}}, \mathbf{y}_{\text{val}})}$$

Circular Validation (CircV)[1]



[1] *Bruzonne et. al., 2009*

SKADA-bench: a new benchmark for DA

skada-bench Public

generated from benchopt/template_benchmark

Edit Pins | Watch 5 | Fork 5 | Starred 25

SKADA Methods

- 20 shallow methods
- 7 deep DA methods

SKADA Scorers

- Circular Validation
- 2 Reweighting scorers
- 2 Entropy Minimization scorers

Modalities

- 2 Computer Vision
- 2 Tabular
- 2 NLP
- 1 Biosignals (BCI)

SKADA-bench: a new benchmark for DA

SKADA Methods

- 20 shallow methods
- 7 deep DA methods

SKADA Scorers

- Circular Validation
- 2 Reweighting scorers
- 2 Entropy Minimization scorers

Modalities

- 2 Computer Vision
- 2 Tabular
- 2 NLP
- 1 Biosignals (BCI)

↓

Benchopt

Benchmark 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	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
	Gauss. RW	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	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	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
	MMDTarS	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	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	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	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
	LinOT	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	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
Sub.	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	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	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

Adapt. helps, Adapt. hurts,, not statistically significant

Benchmark 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	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
	Gauss. RW	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	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	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
	MMDTarS	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	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	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	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
	LinOT	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	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
Sub.	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	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	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

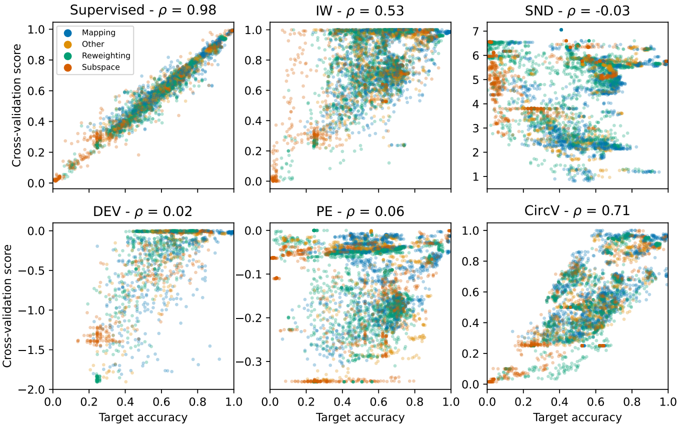
Adapt. helps, Adapt. hurts,, not statistically significant

Benchmark 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	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
	Gauss. RW	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	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	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
	MMDTarS	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	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	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	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
	LinOT	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	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
Sub.	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	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	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

Adapt. helps, Adapt. hurts,, not statistically significant

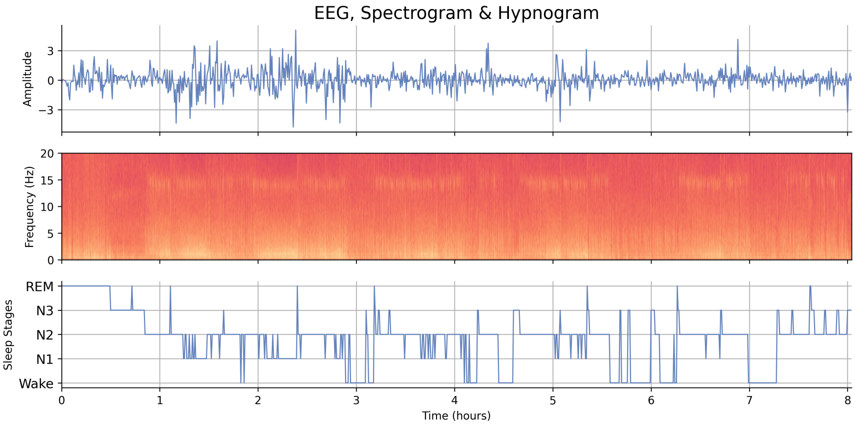
DA scorers



- **Supervised** scorers (cheating) correlate well with target accuracy.
- **Unsupervised** scorers are **less correlated** with target accuracy.
- Only **CircV** is correlated with target accuracy.

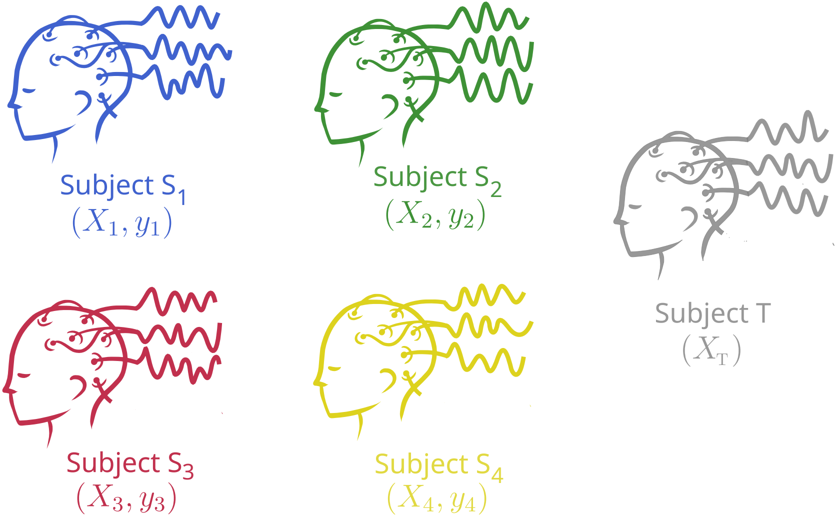
Monge Mapping for biosignals

Impact of distribution shift in Sleep Staging

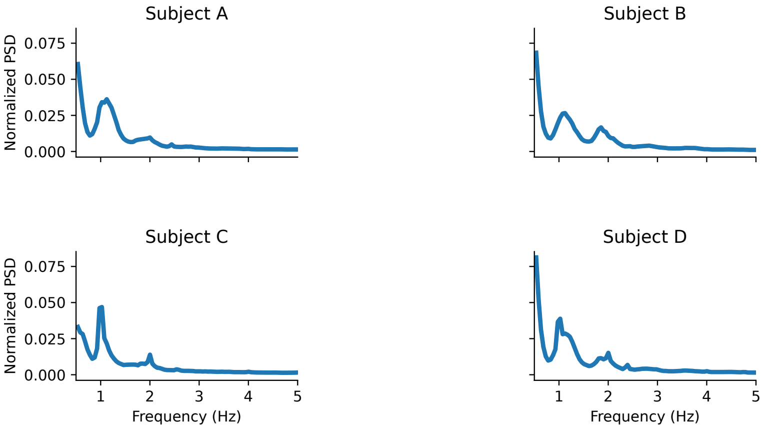


- **Classification** problem with **five** classes: Wake, N1, N2, N3, REM for each **30s** epoch.
- **Frequency** helps to classify sleep stage

Multi-source Domain Adaptation for Sleep Staging

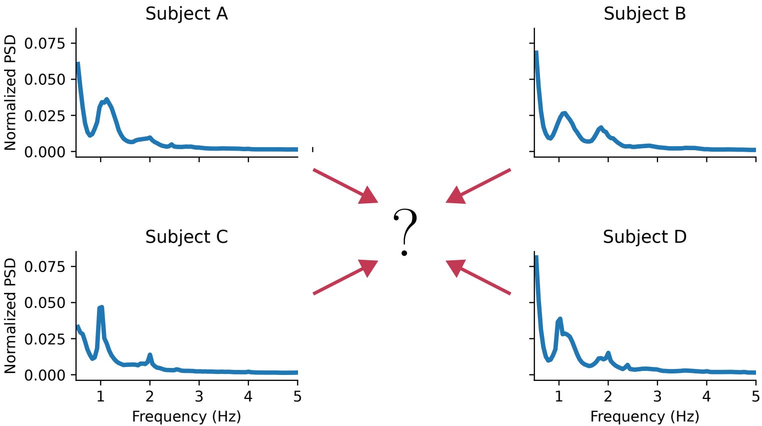


Shift in PSD for Sleep Staging



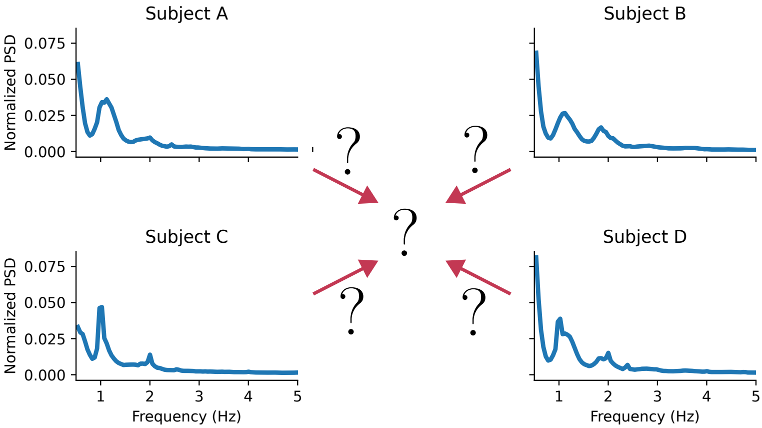
- **Power Spectral Density (PSD)**: representation of the signal in the frequency domain.

Shift in PSD for Sleep Staging



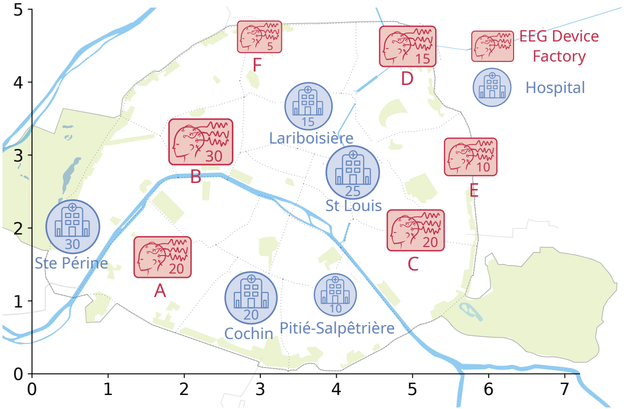
- **Power Spectral Density (PSD)**: representation of the signal in the frequency domain.
- Project each PSD to a common reference: **Which Reference ?**

Shift in PSD for Sleep Staging



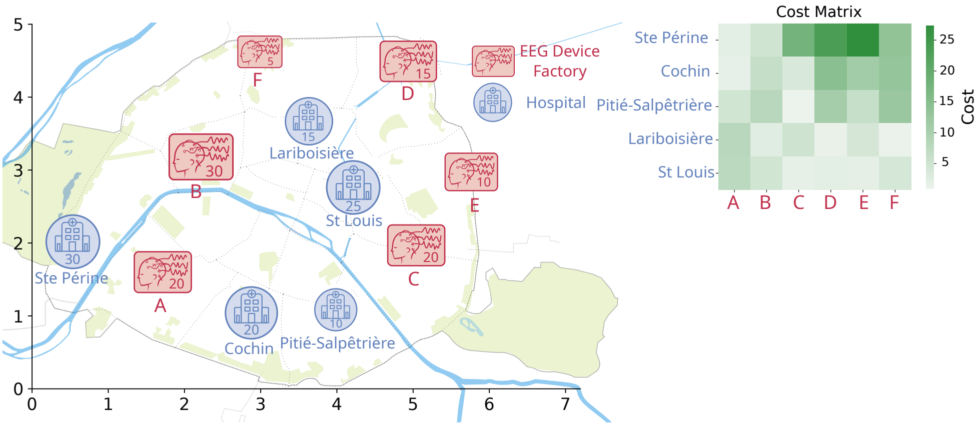
- **Power Spectral Density (PSD)**: representation of the signal in the frequency domain.
- Project each PSD to a common reference: **Which Reference ?** **How to project ?**

What is Optimal Transport?



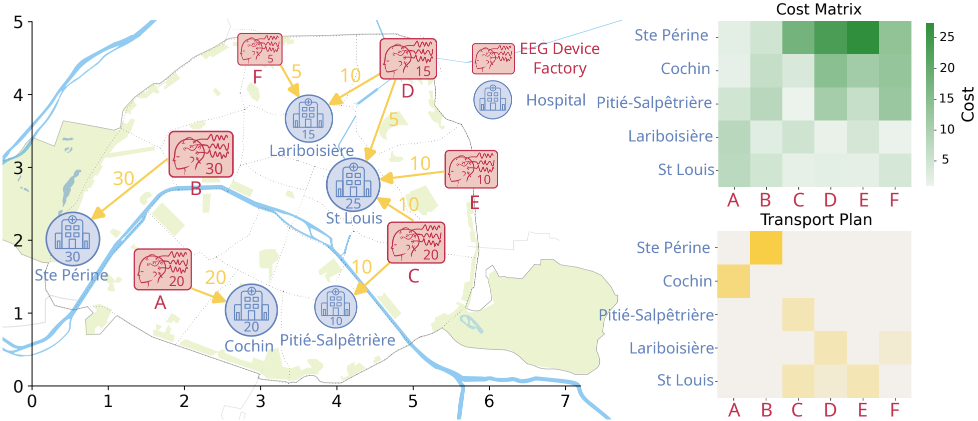
- Consider two distributions μ_s (**EEG device factory**) and μ_t (**hospital**).

What is Optimal Transport?



- A **ground metric** defines the cost of transporting mass between points.

What is Optimal Transport?



- **Optimal Transport** aim to find the **best way** to transport source distribution to target distribution.

Optimal Transport formulation

$\mu_s = \sum_i p_i^s \delta_{\mathbf{X}_i^s}$

Probability mass

$\mu_t = \sum_i p_i^t \delta_{\mathbf{X}_i^t}$

Location

- We can define a **Wasserstein distance** with a transport cost matrix $\mathbf{C}_{ij} = \|\mathbf{x}_s^i - \mathbf{x}_t^j\|^p$

$$\mathcal{W}_p^p(\mu_s, \mu_t) = \min_{\mathbf{T} \in \Pi(\mathbf{a}, \mathbf{b})} \langle \mathbf{C}, \mathbf{T} \rangle_F$$

  where $\Pi(\mathbf{a}, \mathbf{b}) = \{\mathbf{T} \in \mathbb{R}_+^{n_s \times n_t} \mid \mathbf{T}\mathbf{1}_{n_t} = \mu_s, \mathbf{T}^\top \mathbf{1}_{n_s} = \mu_t\}$ is the set of valid transport plans.

- And a **Monge Mapping** from source to target samples:

$$m(\mathbf{X}_t^s) = \frac{1}{a_i} \sum_{j=1}^{n_t} \mathbf{T}_{ij} \mathbf{X}_j^t$$

Monge mapping for Gaussian distributions

$$\mu_s = \mathcal{N}(\mathbf{m}_s, \boldsymbol{\Sigma}_s) \qquad \mu_t = \mathcal{N}(\mathbf{m}_t, \boldsymbol{\Sigma}_t)$$

Mean

Covariance

- We can define a **Wasserstein distance** or Bures-Wasserstein distance:

$$\mathcal{W}_2^2(\mu_s, \mu_t) = \mathrm{Tr}\left(\boldsymbol{\Sigma}_s + \boldsymbol{\Sigma}_t - 2\left(\boldsymbol{\Sigma}_t^{\frac{1}{2}}\boldsymbol{\Sigma}_s\boldsymbol{\Sigma}_t^{\frac{1}{2}}\right)^{\frac{1}{2}}\right)$$

- And a **Monge Mapping**:

$$m(\mathbf{x}) = \underbrace{\mathbf{A}}_{\text{Transport Map}}(\mathbf{x} - \mathbf{m}_s) + \mathbf{m}_t, \quad \text{with} \quad \mathbf{A} = \overbrace{\boldsymbol{\Sigma}_s}^{\text{Source Covariance}}{}^{-\frac{1}{2}}\left(\boldsymbol{\Sigma}_s^{\frac{1}{2}}\underbrace{\boldsymbol{\Sigma}_t}_{\text{Target Covariance}}\boldsymbol{\Sigma}_s^{\frac{1}{2}}\right)^{\frac{1}{2}}\boldsymbol{\Sigma}_s^{-\frac{1}{2}} = \mathbf{A}^T$$

Wasserstein barycenter between Gaussian distributions

Considering multiple Gaussian distributions μ_k. The barycenter $\bar{\mu}$ is expressed as

$$\bar{\mu} = \arg\min_{\mu} \frac{1}{K} \sum_{k=1}^{K} \mathcal{W}_2^2(\mu, \mu_k) \ . \tag{1}$$

Wasserstein barycenter between Gaussian distributions

Considering multiple Gaussian distributions μ_k. The barycenter $\bar{\mu}$ is expressed as

$$\bar{\mu} = \arg\min_{\mu} \frac{1}{K} \sum_{k=1}^{K} \mathcal{W}_2^2(\mu, \mu_k) \ . \tag{1}$$

The barycenter is still a Gaussian distribution $\bar{\mu} = \mathcal{N}(\bar{\mathbf{m}}, \bar{\boldsymbol{\Sigma}})$.

Wasserstein barycenter between Gaussian distributions

Considering multiple Gaussian distributions μ_k. The barycenter $\bar{\mu}$ is expressed as

$$\bar{\mu} = \arg\min_{\mu} \frac{1}{K} \sum_{k=1}^{K} \mathcal{W}_2^2(\mu, \mu_k) \ . \tag{1}$$

The barycenter is still a Gaussian distribution $\bar{\mu} = \mathcal{N}(\bar{\mathbf{m}}, \bar{\mathbf{\Sigma}})$.

$\Longrightarrow$ **No closed-form** for computing the covariance $\bar{\mathbf{\Sigma}}$.

Wasserstein barycenter between Gaussian distributions

Considering multiple Gaussian distributions μ_k. The barycenter $\bar{\mu}$ is expressed as

$$\bar{\mu} = \arg\min_{\mu} \frac{1}{K} \sum_{k=1}^{K} \mathcal{W}_2^2(\mu, \mu_k) \ . \tag{1}$$

The barycenter is still a Gaussian distribution $\bar{\mu} = \mathcal{N}(\bar{\mathbf{m}}, \bar{\boldsymbol{\Sigma}})$.

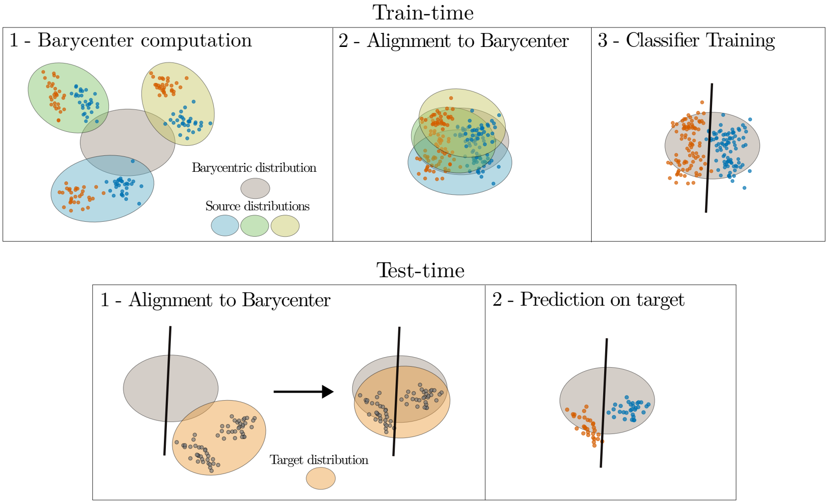
$\Longrightarrow$ **No closed-form** for computing the covariance $\bar{\boldsymbol{\Sigma}}$.

One uses the following optimality condition[1]:

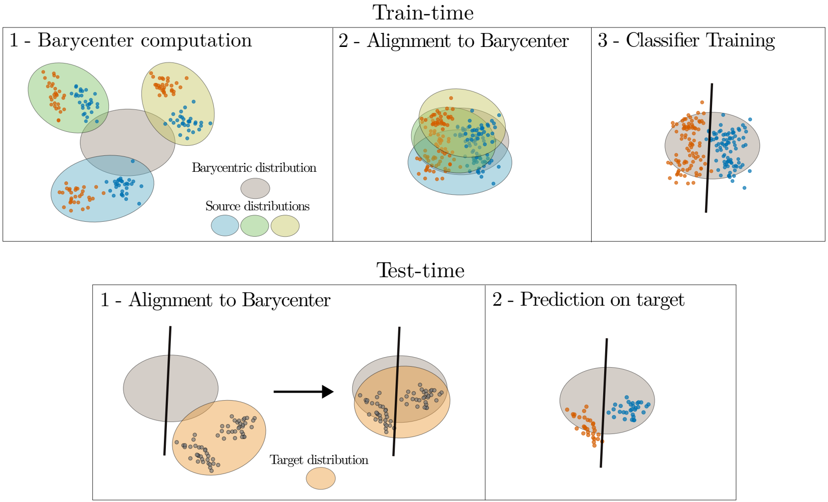
$$\bar{\boldsymbol{\Sigma}} = \frac{1}{K} \sum_{k=1}^{K} \left(\bar{\boldsymbol{\Sigma}}^{\frac{1}{2}} \boldsymbol{\Sigma}_k \bar{\boldsymbol{\Sigma}}^{\frac{1}{2}} \right)^{\frac{1}{2}} \ , \tag{2}$$

[1] *Agueh et. al., 2011*

Monge Mapping Normalisation



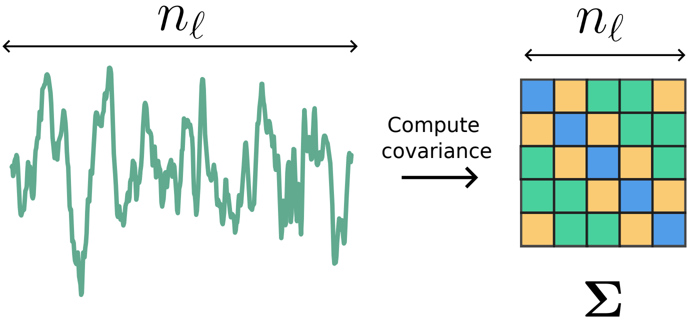
Monge Mapping Normalisation



→ Use Square root of the covariance: **High computational cost !**

Assumptions on the signals

- Centered **Gaussian** distributions $\rightarrow$ $\mathbf{X} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ with $\mathbf{\Sigma} \in \mathcal{S}_{n_\ell}^{++}$
- $\mathbf{\Sigma}$ is the "auto-covariance", computed with time-lagged. $\mathbf{\Sigma}_{i,j} = \mathbf{X}_i \mathbf{X}_j$
- **Stationarity+Periodicity** $\rightarrow$ Covariance matrices are **Toeplitz circulant** matrices.



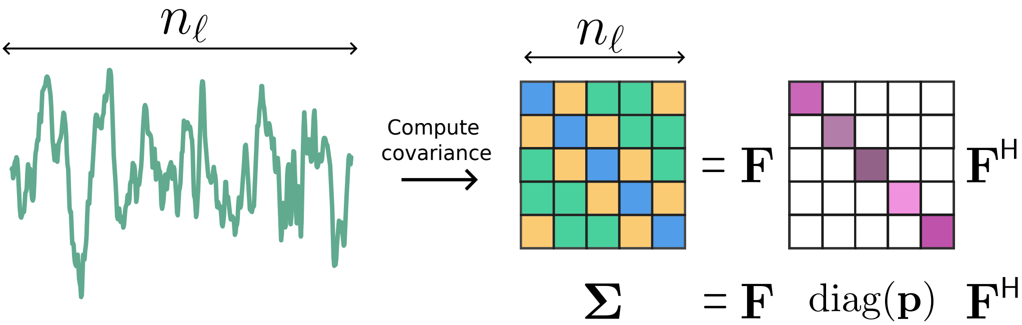
Assumptions on the signals

The Discrete Fourier Transform (DFT) can diagonalize the circulant matrix

$$\mathbf{\Sigma} = \mathbf{F}\mathrm{diag}(\mathbf{p})\mathbf{F}^* ,$$

Power Spectral Density (PSD)

with $\mathbf{F}$ and $\mathbf{F}^*$ the Fourier transform operator and its inverse, and $\mathbf{p}$ the Power Spectral Density (PSD) of the signal.



Convolutional Monge Mapping Normalization

Consider K centered stationary Gaussian signals of covariance $\mathbf{\Sigma}_k = \mathbf{F}\mathrm{\mathbf{diag}}(\mathbf{p}_k)\mathbf{F}^*$ and **PSD** $\mathbf{p}_k$ with $k \in [K]$.

- First step: **barycenter computation**:

$$\underbrace{\bar{\mathbf{\Sigma}}}_{\text{Barycenter Covariance}} = \frac{1}{K}\sum_{k=1}^{K}\left(\bar{\mathbf{\Sigma}}^{\frac{1}{2}}\underbrace{\mathbf{\Sigma}_k}_{\text{Domain } k \text{ Covariance}}\bar{\mathbf{\Sigma}}^{\frac{1}{2}}\right)^{\frac{1}{2}},$$

- Second step: **map** each **signal k** to barycenter:

$$m(\mathbf{x}) = \underbrace{\mathbf{A}}_{\text{Transport Map}}\mathbf{x}, \quad \text{with} \quad \mathbf{A} = \underbrace{\mathbf{\Sigma}_s}_{\text{Source Covariance}}^{-\frac{1}{2}}\left(\mathbf{\Sigma}_s^{\frac{1}{2}}\underbrace{\mathbf{\Sigma}_t}_{\text{Target Covariance}}\mathbf{\Sigma}_s^{\frac{1}{2}}\right)^{\frac{1}{2}}\mathbf{\Sigma}_s^{-\frac{1}{2}} = \mathbf{A}^{\top}.$$

Convolutional Monge Mapping Normalization

Consider K centered stationary Gaussian signals of covariance $\boldsymbol{\Sigma}_k = \mathbf{F}\mathrm{\mathbf{diag}}(\mathbf{p}_k)\mathbf{F}^*$ and **PSD** $\mathbf{p}_k$ with $k \in [K]$.

- First step: **barycenter computation**:

$$\underbrace{\bar{\mathbf{p}}}_{\text{Barycenter PSD}} = \left(\frac{1}{K} \sum_{k=1}^{K} \underbrace{\mathbf{p}_k}_{\text{Domain } k \text{ PSD}}^{\odot \frac{1}{2}} \right)^{\odot 2} .$$

- Second step: **map** each **signal k** to barycenter:

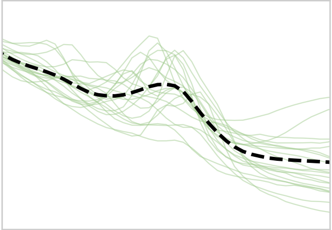
$$m(\mathbf{x}) = \underbrace{\mathbf{h}}_{\text{Filter}} * \mathbf{x} , \quad \text{with} \quad \mathbf{h} = \mathbf{F}^* \left(\underbrace{\bar{\mathbf{p}}}_{\text{Target PSD}}^{\odot \frac{1}{2}} \odot \underbrace{\mathbf{p}_k}_{\text{Source PSD}}^{\odot -\frac{1}{2}} \right) .$$

Convolutional Monge Mapping Normalization

Consider K centered stationary Gaussian signals of covariance $\mathbf{\Sigma}_k = \mathbf{F}\mathrm{\mathbf{diag}}(\mathbf{p}_k)\mathbf{F}^*$ and **PSD** $\mathbf{p}_k$ with $k \in [K]$.

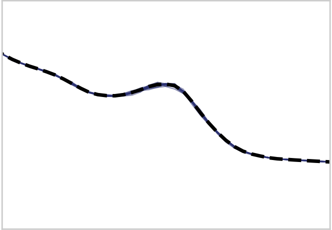
- First step: **barycenter computation**:

$$\underbrace{\bar{\mathbf{p}}}_{\text{Barycenter PSD}} = \left(\frac{1}{K}\sum_{k=1}^{K} \underbrace{\mathbf{p}_k}_{\text{Domain } k \text{ PSD}}^{\odot\frac{1}{2}}\right)^{\odot 2} .$$

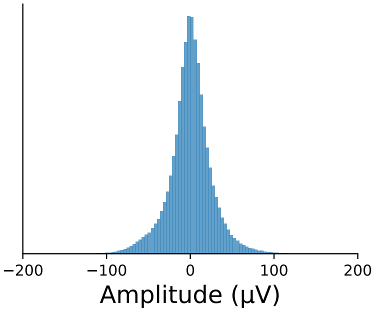


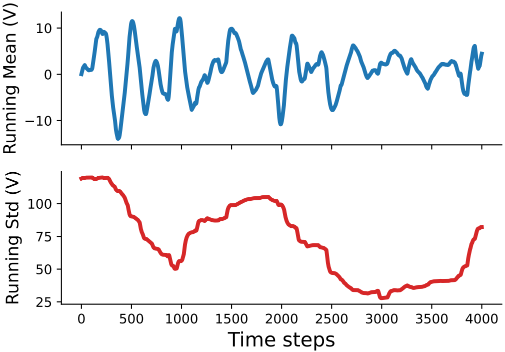
- Second step: **map** each **signal k** to barycenter:

$$m(\mathbf{x}) = \underbrace{\mathbf{h}}_{\text{Filter}} * \mathbf{x}\,, \quad \text{with} \quad \mathbf{h} = \mathbf{F}^*\left(\underbrace{\bar{\mathbf{p}}}_{\text{Target PSD}}^{\odot\frac{1}{2}} \odot \underbrace{\mathbf{p}_k}_{\text{Source PSD}}^{\odot-\frac{1}{2}}\right) .$$



Discussion on Stationarity and gaussiannity assumptions





- EEG signals are **non-stationary** and **non Gaussian**
- **CMMN** relies on these assumptions for effective learning
- The "real" assumption is: **The shift is comprised in the second order statistics.**
- Only keep **individual information** of each subject in the highest orders.

Limit of CMMN setup

- **Three** different datasets: **MASS**, **Physionet** and **SHHS**
- Only **200** subjects in total
- Use simple **CNN** architecture from[1]
- Only for **univariate** signals

Architecture	Chambon [6] No Adapt	Chambon [6] CMMN
MASS→MASS	75.1 ± 1.0	$\mathbf{76.2 \pm 2.2}$
Phys.→Phys.	69.2 ± 2.7	$\mathbf{71.7 \pm 2.4}$
SHHS→SHHS	61.2 ± 3.8	$\mathbf{64.3 \pm 2.7}$
MASS→Phys.	58.4 ± 2.4	$\mathbf{62.3 \pm 1.5}$
MASS→SHHS	41.8 ± 3.6	$\mathbf{47.6 \pm 4.0}$
Phys.→MASS	64.0 ± 2.7	$\mathbf{68.3 \pm 2.5}$
Phys.→SHHS	45.6 ± 2.1	$\mathbf{51.6 \pm 1.8}$
SHHS→MASS	57.0 ± 2.8	$\mathbf{64.5 \pm 2.8}$
SHHS→Phys.	55.0 ± 2.7	$\mathbf{58.3 \pm 1.7}$
Mean	58.6 ± 2.6	$\mathbf{62.7 \pm 2.4}$

→ **Problems:** Low **number of subjects**, not **SOTA network**, 2 steps **preprocess than train**.

[1] *Chambon et. al., 2018*

Extension of CMMN setup

Monge Mapping for Multi-channel signals:

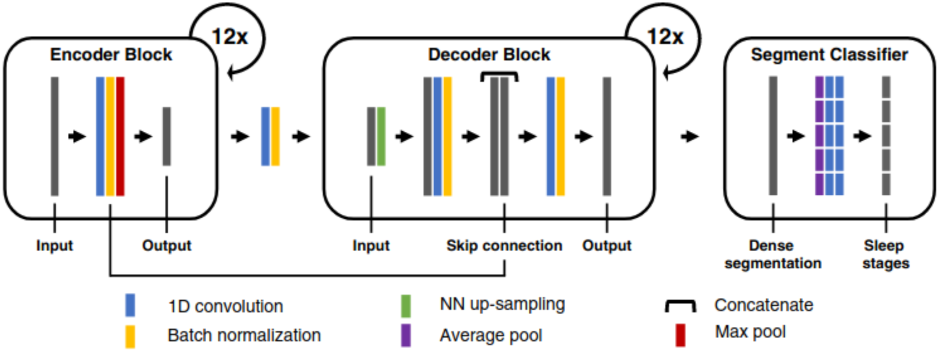
- Use **cross-PSD** to capture the **spatial** information
- Application on Sleep staging and BCI.
- **Statistical guarantees** on the estimations.

Go beyond pre-processing with PSDNorm:

- **PSDNorm** as a new normalization layer in deep learning
- Application on Sleep staging with **SOTA architecture**.

End-to-end training with PSDNorm

Better architecture

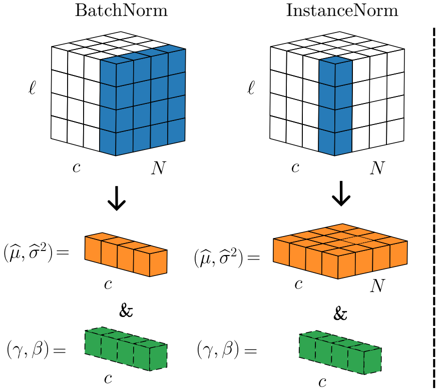


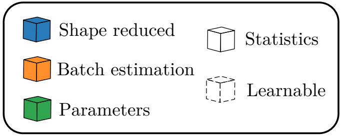
- **UNet**[1] architecture: encoder-decoder with skip connections
- Take as input **sequence of 17min of sleep** (one annotation every 30seconds)

  → How can we use the **CMMN** to improve the training of this architecture?

[1] *Perslev et. al., 2021*

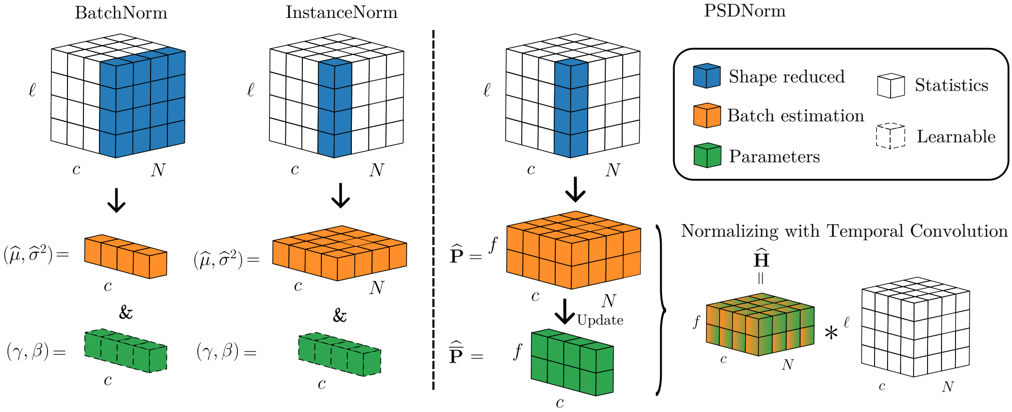
Classical Normalizations in Deep Learning





- **Batch Normalization** (BN): Normalize the input of each layer with the mean and variance of the batch
- **Instance Normalization** (IN):Normalize the input of each layer with the mean and variance of the instance. **Adapted to time series**

PSDNorm as a new normalization layer



- Compute **PSD over the sequence** of sleep (vs. one on night before)
- Compute the barycenter with a **geodesic update on the batch**.

Increase the number of subjects

Dataset	Subj.	Rec.	Age ± std	Sex (F/M)
ABC	44	117	48.8 ± 9.8	43%/57%
CCSHS	515	515	17.7 ± 0.4	50%/50%
CFS	681	681	41.7 ± 20.0	55%/45%
HPAP	166	166	46.5 ± 11.9	43%/57%
MROS	2101	2698	76.4 ± 5.5	0%/100%
PHYS	70	132	58.8 ± 22.0	33%/67%
SHHS	5730	8271	63.1 ± 11.2	52%/48%
MASS	61	61	42.5 ± 18.9	55%/45%
CHAT	1230	1635	6.6 ± 1.4	52%/48%
SOF	434	434	82.8 ± 3.1	100%/0%
Total	11032	14710	–	–

→**10** datasets, **10k** subjects, **10M** of samples

Results on LODO sleep staging

	Dataset	BatchNorm	InstanceNorm	CMMN	PSDNorm
All subjects	ABC	$78.49_{\pm 0.42}$	$\mathbf{78.83_{\pm 0.59}}$	$78.33_{\pm 0.12}$	$78.56_{\pm 0.67}$
	CCSHS	$\mathbf{88.79_{\pm 0.21}}$	$88.75_{\pm 0.04}$	$88.61_{\pm 0.10}$	$88.56_{\pm 0.36}$
	CFS	$84.97_{\pm 0.37}$	$\mathbf{85.73_{\pm 0.29}}$	$84.85_{\pm 0.13}$	$85.42_{\pm 0.09}$
	CHAT	$64.72_{\pm 3.94}$	$68.86_{\pm 2.49}$	$69.76_{\pm 1.62}$	$\mathbf{70.57_{\pm 1.24}}$
	HOMEPAP	$76.39_{\pm 0.29}$	$76.70_{\pm 0.35}$	$\mathbf{76.77_{\pm 0.66}}$	$76.72_{\pm 0.27}$
	MASS	$73.71_{\pm 0.62}$	$72.12_{\pm 0.70}$	$\mathbf{73.90_{\pm 0.69}}$	$72.51_{\pm 1.68}$
	MROS	$81.30_{\pm 0.25}$	$81.49_{\pm 0.18}$	$80.91_{\pm 0.42}$	$\mathbf{81.57_{\pm 0.34}}$
	PhysioNet	$76.13_{\pm 0.57}$	$76.15_{\pm 0.52}$	$\mathbf{76.48_{\pm 0.37}}$	$75.96_{\pm 1.02}$
	SHHS	$77.97_{\pm 1.46}$	$79.05_{\pm 0.89}$	$78.21_{\pm 0.39}$	$\mathbf{79.14_{\pm 1.01}}$
	SOF	$81.33_{\pm 0.54}$	$81.98_{\pm 0.22}$	$81.84_{\pm 0.49}$	$\mathbf{82.50_{\pm 0.34}}$
	Mean(Dataset)	$78.38_{\pm 0.47}$	$78.97_{\pm 0.11}$	$78.98_{\pm 0.14}$	$\mathbf{79.15_{\pm 0.14}}$
	Mean(Subject)	$78.14_{\pm 1.01}$	$79.26_{\pm 0.48}$	$78.77_{\pm 0.07}$	$\mathbf{79.51_{\pm 0.62}}$
Balanced@400	ABC	$78.26_{\pm 1.33}$	$\mathbf{78.73_{\pm 0.42}}$	$78.04_{\pm 0.51}$	$78.18_{\pm 0.68}$
	CCSHS	$87.42_{\pm 0.16}$	$\mathbf{87.62_{\pm 0.42}}$	$87.57_{\pm 0.20}$	$87.58_{\pm 0.30}$
	CFS	$84.32_{\pm 0.57}$	$\mathbf{84.72_{\pm 0.33}}$	$84.58_{\pm 0.20}$	$84.29_{\pm 0.36}$
	CHAT	$66.55_{\pm 0.88}$	$64.43_{\pm 4.41}$	$68.73_{\pm 2.48}$	$\mathbf{70.28_{\pm 1.70}}$
	HOMEPAP	$75.25_{\pm 0.50}$	$76.47_{\pm 0.63}$	$76.10_{\pm 0.32}$	$\mathbf{76.83_{\pm 0.61}}$
	MASS	$70.00_{\pm 1.91}$	$71.52_{\pm 1.13}$	$71.63_{\pm 1.92}$	$\mathbf{72.77_{\pm 1.09}}$
	MROS	$\mathbf{80.37_{\pm 0.20}}$	$80.28_{\pm 0.21}$	$80.09_{\pm 0.40}$	$80.26_{\pm 0.11}$
	PhysioNet	$\mathbf{75.81_{\pm 0.13}}$	$74.68_{\pm 0.55}$	$75.31_{\pm 1.54}$	$74.82_{\pm 2.11}$
	SHHS	$76.44_{\pm 0.92}$	$78.68_{\pm 0.37}$	$77.00_{\pm 0.39}$	$\mathbf{78.88_{\pm 0.68}}$
	SOF	$81.08_{\pm 1.14}$	$80.68_{\pm 1.38}$	$\mathbf{81.25_{\pm 0.71}}$	$79.49_{\pm 0.41}$
	Mean(Dataset)	$77.55_{\pm 0.34}$	$77.78_{\pm 0.46}$	$78.03_{\pm 0.35}$	$\mathbf{78.34_{\pm 0.42}}$
	Mean(Subject)	$77.22_{\pm 0.34}$	$78.17_{\pm 0.28}$	$77.74_{\pm 0.36}$	$\mathbf{78.85_{\pm 0.59}}$

- Balanced@400 have **10x less** subjects than all subjects
- PSDNorm reaches performances of BatchNorm with **10x less** data
- PSDNorm outperforms CMMN, **better adaptation with end-to-end training**

Conclusion and perspectives

Key takeaways

- SKADA makes easy to use DA:
  - Théo Gnassounou, Oleksii Kachaiev, Rémi Flamary, Antoine Collas, Yanis Lalou, Antoine Mathelin, Alexandre Gramfort, Ruben Bueno, Florent Michel, Apolline Mellot, Virginie Loison, Ambroise Odonnat, and Thomas Moreau. Skada : Scikit adaptation, July 2024b. URL `https://doi.org/10.5281/zenodo.12666838`.
- Extensive benchmark show the **limit of DA field**
  - Yanis Lalou, Théo Gnassounou, Antoine Collas, Antoine de Mathelin, Oleksii Kachaiev, Ambroise Odonnat, Alexandre Gramfort, Thomas Moreau, and Rémi Flamary. Skada-bench: Benchmarking unsupervised domain adaptation methods with realistic validation on diverse modalities. Transactions on Machine Learning Research, 2025.
- Domain adaptation can works when **specifically designed for each applications**.

Key takeaways

- Distribution shift is a major challenge in biosignal applications.
- **Monge mapping helps** to mitigate shift:
  - Théo Gnassounou, Rémi Flamary, and Alexandre Gramfort. Convolutional monge mapping normalization for learning on biosignals. In Neural Information Process- ing Systems (NeurIPS), 2023
- Extension to multivariate signals and **new theoretical guarantees**.
  - Théo Gnassounou, Antoine Collas, Rémi Flamary, Karim Lounici, and Alexandre Gramfort. Multi-source and test-time domain adaptation on multivariate signals using spatio-temporal monge alignment. submitted at JMLR, 2024a.
- **PSDNorm** improves end-to-end training with SOTA architectures with **limited data**
  - Théo Gnassounou, Antoine Collas, Rémi Flamary, and Alexandre Gramfort. PSDNorm: Test-time temporal normalization for deep learning in sleep staging, submitted at ICLR, 2025.

Open source contributions

- Creator of **SKADA** library for Domain Adaptation in Python
- Co-creator of **Skada-bench**, Benchmark for Domain Adaptation methods using benchopt
- Contributor to **Braindecode**, Deep learning for EEG in Python
- Contributor to **POT**, Python Optimal Transport library

Perspectives and Future Work

- General Domain Adaptation challenges:
  - **Detect the shift** to apply the right DA method
  - Propose **new DA scorers** better correlated with target accuracy
  - Extend SKADA-bench to **Deep DA** methods
- Domain Adaptation for biosignals:
  - Apply PSDNorm/CMMN to **other applications**: BCI, ECG classification, . . .
  - Comparison with **Foundational models** for biosignals

General Outlook

My key takeaways from 3-year PhD journey:

- **Understand the societal and environnemental impact of my work**:
  - **0.416T of CO_2** emitted for PSDNorm paper training (Paris - NYC = 1T)
- **Understanding the problem**:
  - **Characterize the data (and possible shifts)** to inform model design.
  - **Design experiments thoughtfully** to avoid unnecessary and resource-intensive computations.
- **Embracing Open Science**:
  - **Share code and data** to foster collaboration and reproducibility.
  - Collaborate on **open-source packages**.