Adapting learning models to distribution shifts with affine alignment

Antoine Collas - POPILSS 2025
Inria Saclay - Mind team

19/06/2025

About me

2019 - 2022:

PhD at CentraleSupélec, Sondra lab

Statistical estimation and learning on Riemannian manifolds

Supervisors: Jean-Philippe Ovarlez and Guillaume Ginolhac

2022 - Present:

Postdoc at Inria Saclay, Mind team

Domain adaptation and generative models for neuroscience applications

Supervisors: Bertrand Thirion, Alexandre Gramfort, and Rémi Flamary

1. Introduction to Domain Adaptation (DA)
2. Skada-Bench: Benchmarking Unsupervised DA methods
3. Joint shifts in (X, y) on manifolds: GOPSA
4. Temporal normalization: PSDNorm layer

1. Introduction to Domain Adaptation (DA)

Supervised learning

Independent and identically distributed data:

$$\{(x_n, y_n)\}_{n=1}^{N} \sim \mathbb{P}(X, Y)$$

$x_n \in \mathscr{X}$, e.g. $\mathbb{R}^d$ and $y_n \in \mathscr{Y}$, e.g. $\{-1,1\}$ for binary classification

Goal: find a predictor $f : \mathscr{X} \mapsto \mathscr{Y}$ by empirical risk minimization

$$\min_{f \in \mathscr{F}} \left\{ \hat{R}(f) = \frac{1}{N} \sum_{n=1}^{N} \ell(y_n, f(x_n)) \right\}$$

with ℓ a loss function

Domain Adaptation (DA) 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

2

8

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

2

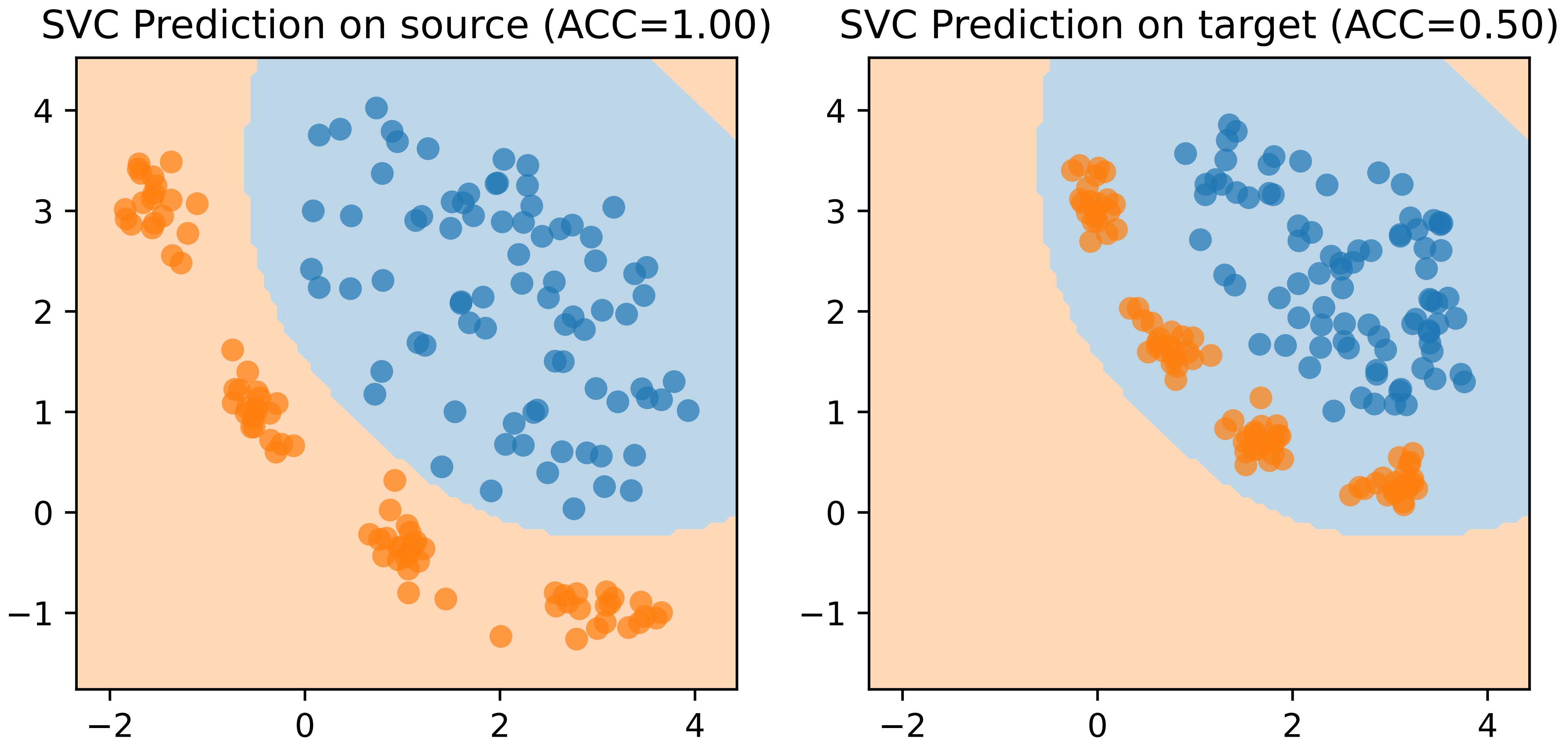
8

Domain Adaptation (DA) 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):



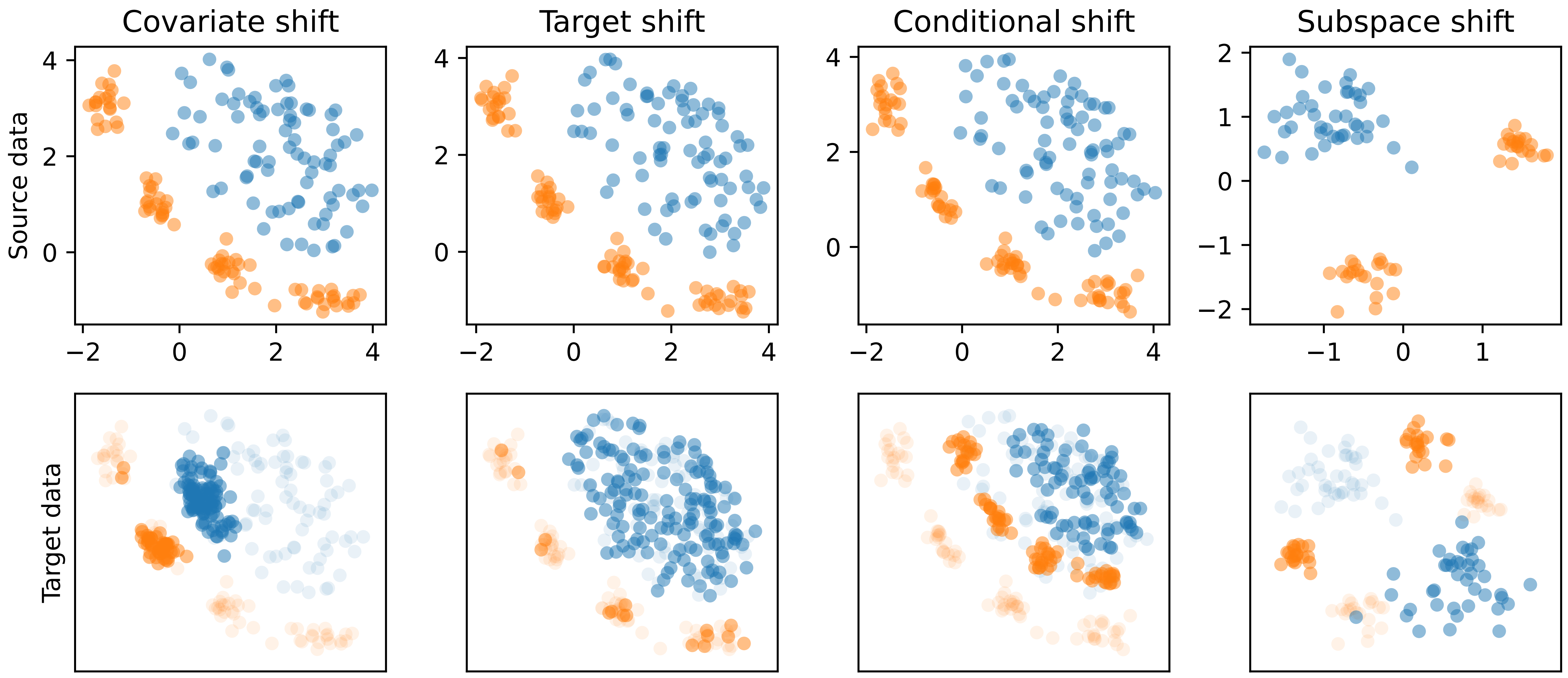
Simple shifts can lead to very bad classifications on the target domain!

Unsupervised DA Problem

Source domain is labeled: $\{(x_n, y_n)\}_{n=1}^{N^{\mathcal{S}}} \sim \mathbb{P}_{\mathcal{S}}(X, Y)$

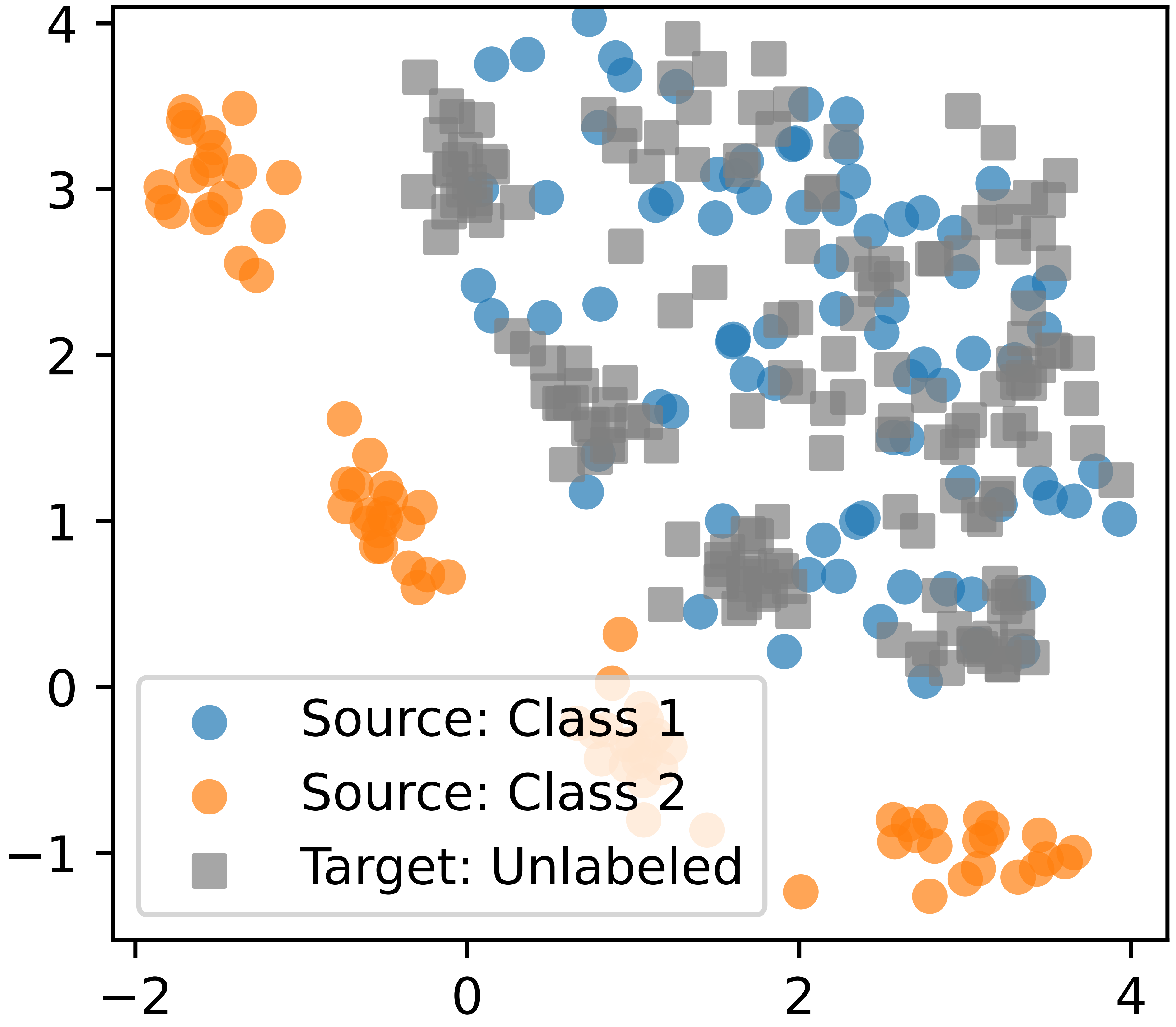
Target domain is unlabeled: $\{(x_n, \cdot\,)\}_{n=1}^{N^{\mathcal{T}}} \sim \mathbb{P}_{\mathcal{T}}(X, Y)$

Assumptions on the shift between $\mathbb{P}_{\mathcal{S}}(X, Y)$ and $\mathbb{P}_{\mathcal{T}}(X, Y)$:



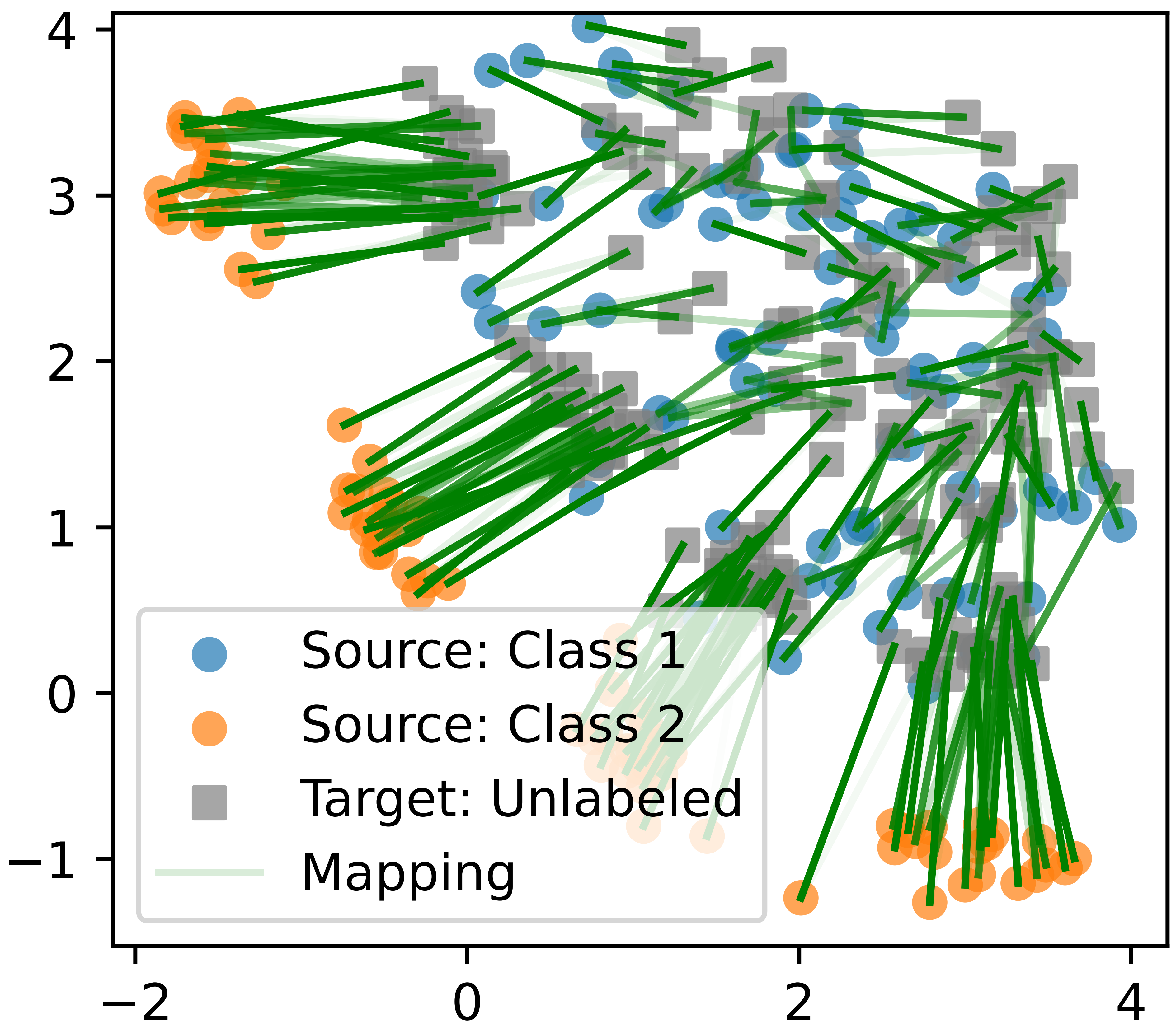
Toy Example: Mapping-Based UDA

Problem setting:



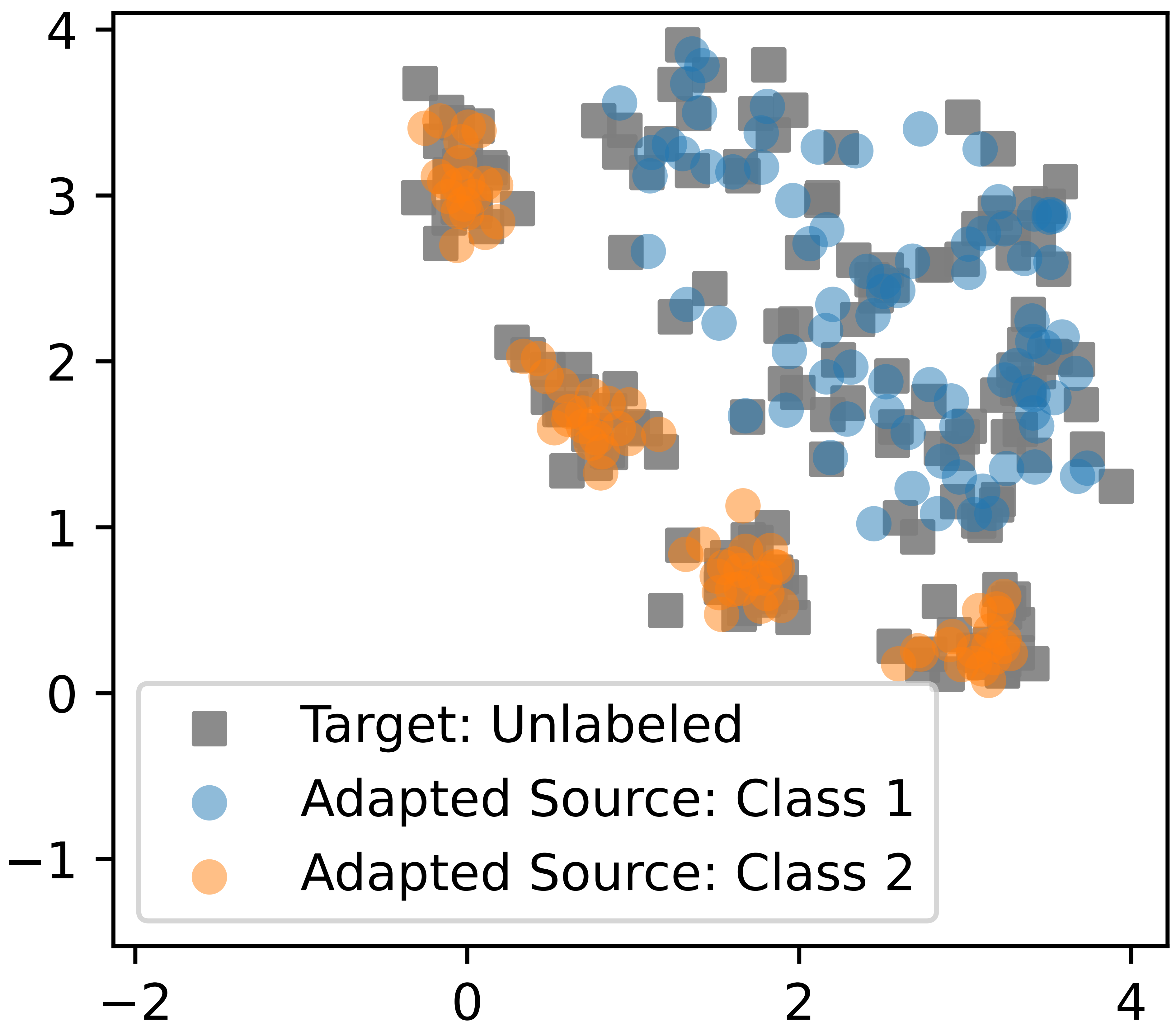
Toy Example: Mapping-Based UDA

Mapping computation:



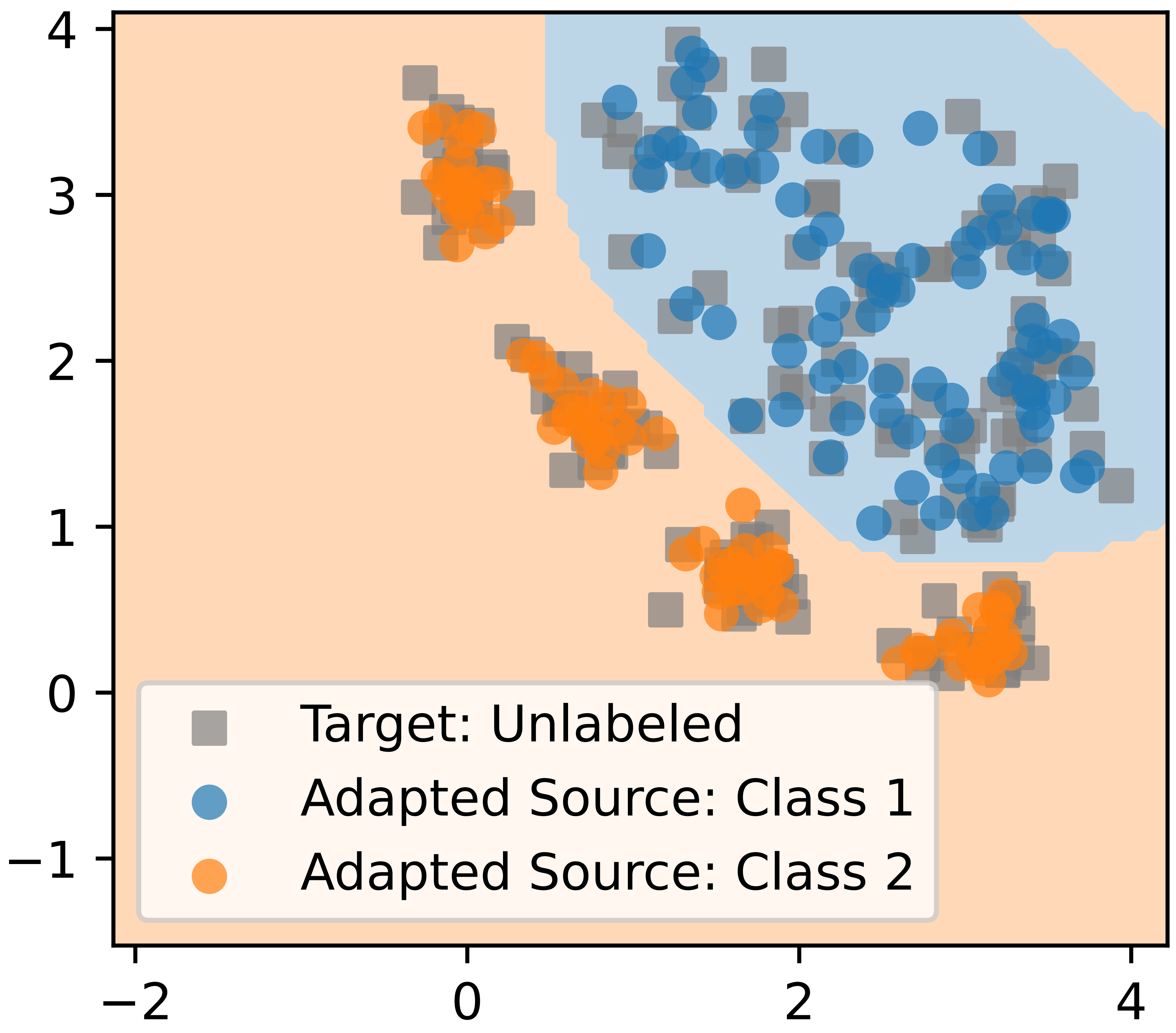
Toy Example: Mapping-Based UDA

Mapping application:



Toy Example: Mapping-Based UDA

Classifier fitting:



DA methods

Mapping:

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

where m aligns source and target domains

m can be a normalization, subspace projection, optimal transport mapping, ...

Reweighting:

$$f_{\mathrm{DA}} \in \arg\min_{f \in \mathcal{F}} \frac{1}{N_S} \sum_{n=1}^{N_S} w_i \ell \left(y_n, f(x_n) \right)$$

where w_i are importance weights depending on source and target domains

End-to-End Deep Learning:

Adversarial or discrepancy losses to align source and target distributions

Scikit-Adaptation: Skada

Python library for domain adaptation:

- Sklearn-like API with estimators (.fit, .predict, . . .), pipeline, grid search
- Shallow and Deep learning methods
- DA scorers to validate hyper-parameters without using target labels

SKADA

https://github.com/scikit-adaptation/skada

2. SKADA-Bench: Benchmarking Unsupervised DA methods

Yanis Lalou*, Théo Gnassounou*, AC*, Antoine de Mathelin*, Oleksii Kachaiev, Ambroise Odonnat, Alexandre Gramfort, Thomas Moreau, Rémi Flamary

*Equal contribution

https://arxiv.org/abs/2407.11676

Motivation and setup

Evaluate DA methods across diverse modalities:

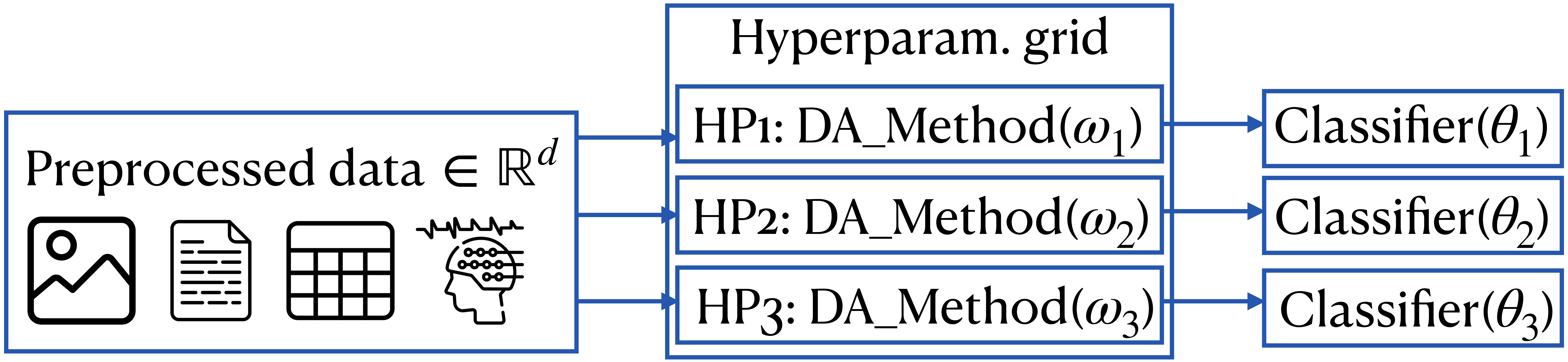
- 8 real-wold datasets of 4 modalities
- 4 synthetic datasets
- 51 adaptation pairs (source → target)
- 20 DA Methods
- 5-fold nested cross-validation
- built with Skada

https://github.com/scikit-adaptation/skada-bench

Dataset	Modality	Preprocessing
Office 31	CV	Decaff + PCA
Office Home	CV	ResNet + PCA
MNIST/USPS	CV	Vect + PCA
20 Newsgroup	NLP	LLM + PCA
Amazon Review	NLP	LLM + PCA
Mushrooms	Tabular	One Hot Encoding
Phishing	Tabular	None
BCI	Biosignals	Cov + TS

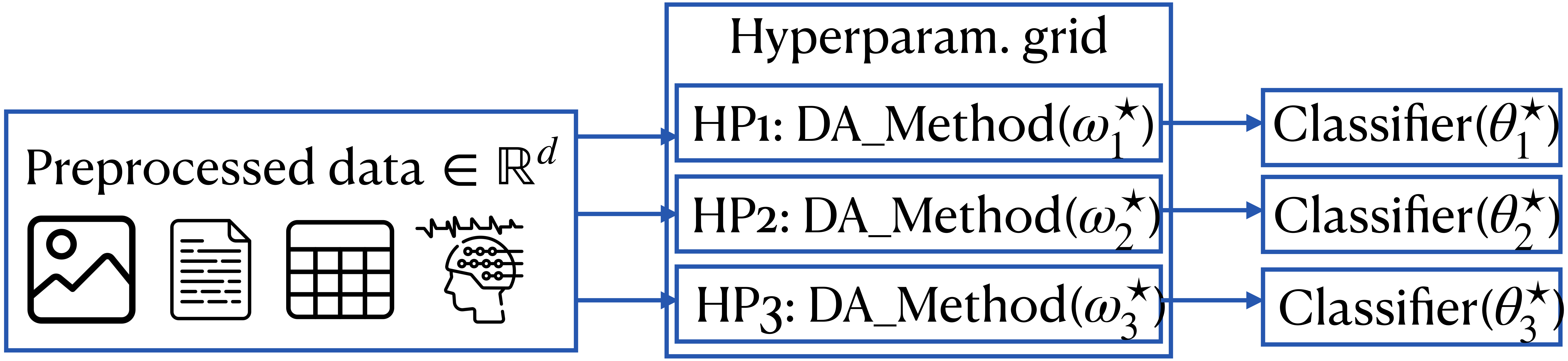
Pipeline

① Training: $\left(X^{\mathcal{S}}_{\text{train}}, y^{\mathcal{S}}_{\text{train}}, X^{\mathcal{T}}_{\text{train}}\right)$



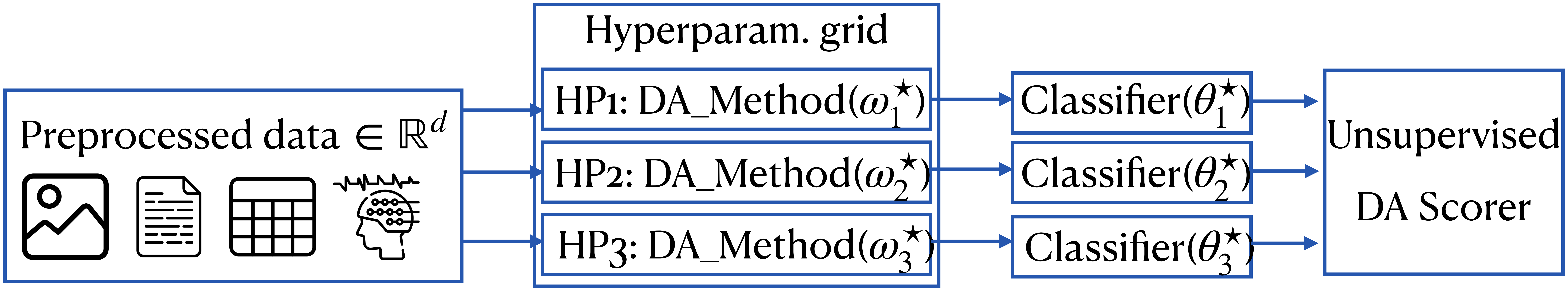
Pipeline

① Training: $\left(X^{\mathcal{S}}_{\text{train}}, y^{\mathcal{S}}_{\text{train}}, X^{\mathcal{T}}_{\text{train}}\right)$



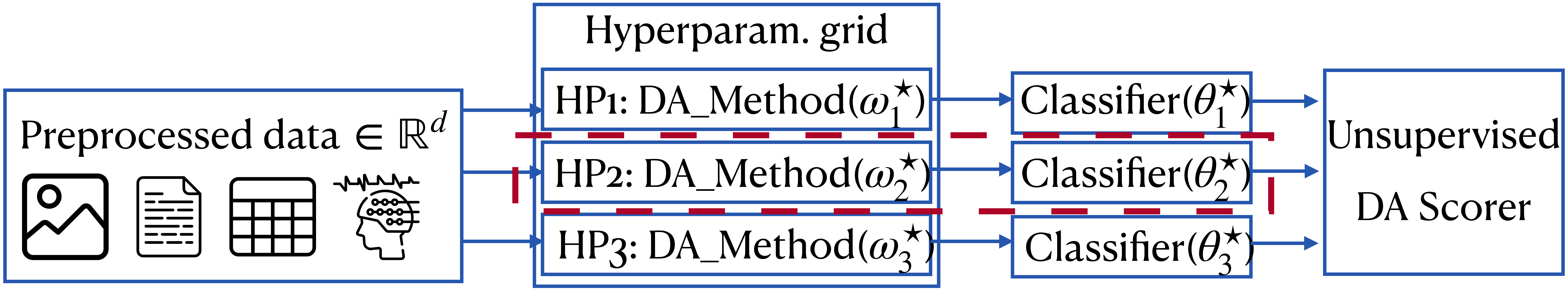
Pipeline

② Validation: $\left(X^{\mathcal{S}}_{\text{val}}, y^{\mathcal{S}}_{\text{val}}, X^{\mathcal{T}}_{\text{val}}\right)$

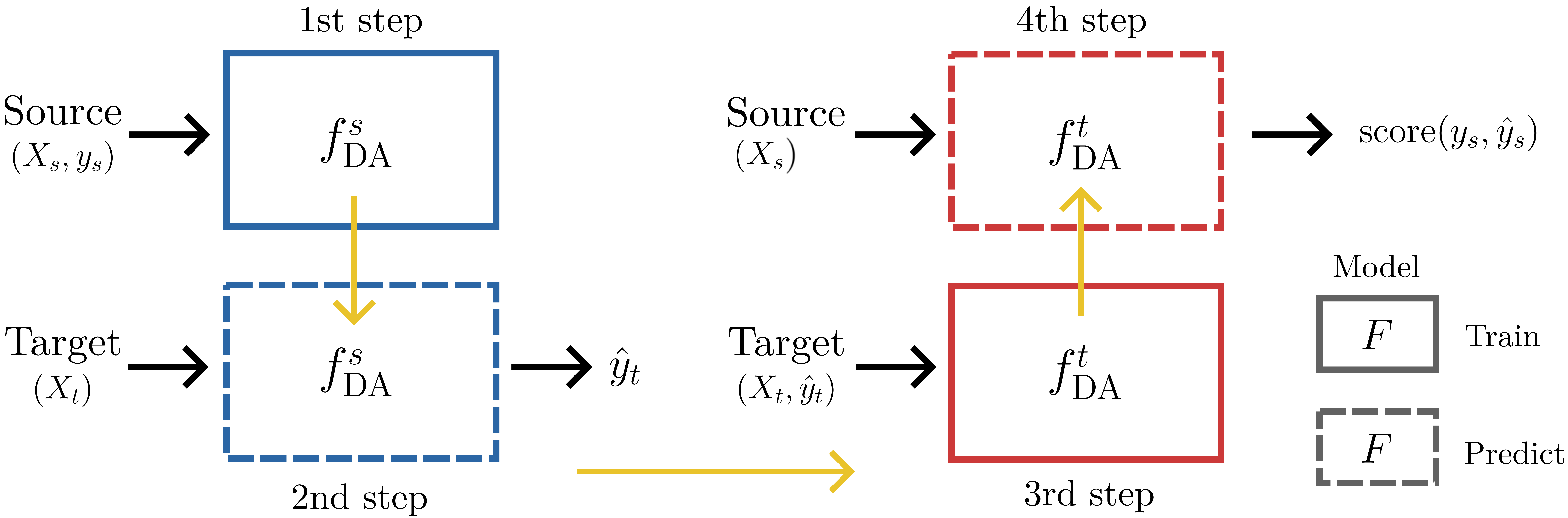


Pipeline

② Validation: $\left(X^{\mathcal{S}}_{\text{val}}, y^{\mathcal{S}}_{\text{val}}, X^{\mathcal{T}}_{\text{val}}\right)$



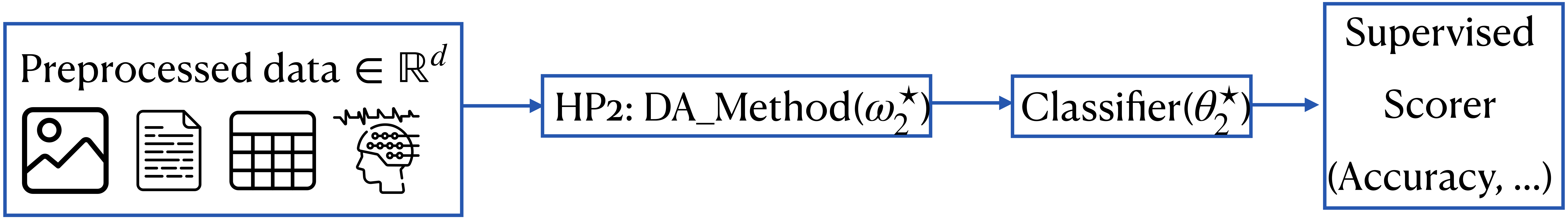
DA scorer: circular validation



Assumption: a good adapter can adapt source→target and then target→source

Pipeline

③ Evaluation: $\left(X^{\mathcal{T}}_{\text{test}}, y^{\mathcal{T}}_{\text{test}}\right)$



Realistic setup: no target labels for training, labels used only for benchmarking

Results

Adaptation

= gain

= loss

		Synthetic				Real									
		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	0.65	0.56	0.54	0.59	0.7	0.72	0.91	0.55		10.66
	Train Tgt	0.92	0.93	0.82	0.98	0.89	0.8	0.96	1.0	0.73	1.0	0.97	0.64		1.55
Reweighting	Dens. RW	0.88	0.86	0.66	0.18	0.62	0.56	0.54	0.58	0.7	0.71	0.91	0.55	IW	12.20
	Disc. RW	0.85	0.83	0.71	0.18	0.63	0.54	0.5	0.6	0.68	0.75	0.91	0.56	CircV	8.75
	Gauss. RW	0.89	0.86	0.65	0.21	0.22	0.44	0.11	0.54	0.55	0.51	0.46	0.25	CircV	16.45
	KLIEP	0.88	0.86	0.66	0.19	0.65	0.56	0.54	0.6	0.69	0.72	0.91	0.55	CircV	10.56
	KMM	0.89	0.85	0.64	0.16	0.64	0.54	0.52	0.7	0.57	0.74	0.91	0.52	CircV	11.74
	NN RW	0.89	0.86	0.67	0.15	0.65	0.55	0.54	0.59	0.66	0.71	0.91	0.54	CircV	9.15
	MMDTarS	0.88	0.86	0.64	0.2	0.6	0.56	0.54	0.59	0.7	0.74	0.91	0.55	IW	10.81
Mapping	CORAL	0.74	0.7	0.76	0.18	0.65	0.57	0.62	0.73	0.7	0.72	0.92	0.62	CircV	5.08
	MapOT	0.72	0.57	0.82	0.02	0.6	0.51	0.61	0.76	0.68	0.63	0.84	0.47	PE	10.21
	EntOT	0.71	0.6	0.82	0.12	0.64	0.58	0.6	0.83	0.62	0.75	0.86	0.54	CircV	9.40
	ClassRegOT	0.74	0.58	0.81	0.11	NA	0.53	0.62	0.97	0.68	0.82	0.89	0.52	IW	8.25
	LinOT	0.73	0.73	0.76	0.18	0.66	0.57	0.64	0.82	0.7	0.76	0.91	0.61	CircV	4.06
	MMD-LS	0.78	0.72	0.76	0.56	0.65	0.56	0.55	0.97	0.63	0.85	NA	0.5	MixVal	8.22
Subspace	JPCA	0.88	0.85	0.66	0.15	0.62	0.48	0.51	0.77	0.69	0.78	0.9	0.54	PE	8.98
	SA	0.74	0.68	0.8	0.11	0.65	0.57	0.56	0.88	0.67	0.78	0.89	0.53	CircV	7.80
	TCA	0.52	0.47	0.51	0.62	0.04	0.02	0.07	0.61	0.61	0.49	0.48	0.26	DEV	17.58
	TSL	0.88	0.85	0.66	0.2	0.63	0.48	0.45	0.63	0.69	0.45	0.89	0.26	IW	15.09
Other	JDOT	0.72	0.58	0.82	0.13	0.6	0.42	0.59	0.79	0.67	0.65	0.79	0.47	IW	11.42
	OTLabelProp	0.72	0.59	0.8	0.07	0.66	0.56	0.62	0.86	0.67	0.64	0.86	0.5	CircV	10.01
	DASVM	0.89	0.86	0.65	0.15	NA	NA	NA	0.87	NA	0.83	0.85	NA	MixVal	7.29

Results

Adaptation

= gain

= loss

		Synthetic				Real									
		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	0.65	0.56	0.54	0.59	0.7	0.72	0.91	0.55		10.66
	Train Tgt	0.92	0.93	0.82	0.98	0.89	0.8	0.96	1.0	0.73	1.0	0.97	0.64		1.55
Reweighting	Dens. RW	0.88	0.86	0.66	0.18	0.62	0.56	0.54	0.58	0.7	0.71	0.91	0.55	IW	12.20
	Disc. RW	0.85	0.83	0.71	0.18	0.63	0.54	0.5	0.6	0.68	0.75	0.91	0.56	CircV	8.75
	Gauss. RW	0.89	0.86	0.65	0.21	0.22	0.44	0.11	0.54	0.55	0.51	0.46	0.25	CircV	16.45
	KLIEP	0.88	0.86	0.66	0.19	0.65	0.56	0.54	0.6	0.69	0.72	0.91	0.55	CircV	10.56
	KMM	0.89	0.85	0.64	0.16	0.64	0.54	0.52	0.7	0.57	0.74	0.91	0.52	CircV	11.74
	NN RW	0.89	0.86	0.67	0.15	0.65	0.55	0.54	0.59	0.66	0.71	0.91	0.54	CircV	9.15
	MMDTarS	0.88	0.86	0.64	0.2	0.6	0.56	0.54	0.59	0.7	0.74	0.91	0.55	IW	10.81
Mapping	CORAL	0.74	0.7	0.76	0.18	0.65	0.57	0.62	0.73	0.7	0.72	0.92	0.62	CircV	5.08
	MapOT	0.72	0.57	0.82	0.02	0.6	0.51	0.61	0.76	0.68	0.63	0.84	0.47	PE	10.21
	EntOT	0.71	0.6	0.82	0.12	0.64	0.58	0.6	0.83	0.62	0.75	0.86	0.54	CircV	9.40
	ClassRegOT	0.74	0.58	0.81	0.11	NA	0.53	0.62	0.97	0.68	0.82	0.89	0.52	IW	8.25
	LinOT	0.73	0.73	0.76	0.18	0.66	0.57	0.64	0.82	0.7	0.76	0.91	0.61	CircV	4.06
	MMD-LS	0.78	0.72	0.76	0.56	0.65	0.56	0.55	0.97	0.63	0.85	NA	0.5	MixVal	8.22
Subspace	JPCA	0.88	0.85	0.66	0.15	0.62	0.48	0.51	0.77	0.69	0.78	0.9	0.54	PE	8.98
	SA	0.74	0.68	0.8	0.11	0.65	0.57	0.56	0.88	0.67	0.78	0.89	0.53	CircV	7.80
	TCA	0.52	0.47	0.51	0.62	0.04	0.02	0.07	0.61	0.61	0.49	0.48	0.26	DEV	17.58
	TSL	0.88	0.85	0.66	0.2	0.63	0.48	0.45	0.63	0.69	0.45	0.89	0.26	IW	15.09
Other	JDOT	0.72	0.58	0.82	0.13	0.6	0.42	0.59	0.79	0.67	0.65	0.79	0.47	IW	11.42
	OTLabelProp	0.72	0.59	0.8	0.07	0.66	0.56	0.62	0.86	0.67	0.64	0.86	0.5	CircV	10.01
	DASVM	0.89	0.86	0.65	0.15	NA	NA	NA	0.87	NA	0.83	0.85	NA	MixVal	7.29

Affine Distribution Alignment Across Domains

CORAL (Correlation Alignment) [Sun et al., 2017]

Aligns second-order moments between source and target:

$$m(x) = \Sigma_{\mathcal{T}}^{\frac{1}{2}} \Sigma_{\mathcal{S}}^{-\frac{1}{2}} x, \qquad \arg\min_{A} \|A \Sigma_{\mathcal{S}} A^{\top} - \Sigma_{\mathcal{T}}\|_F^2$$

LinOT (Linear Optimal Transport) [Flamary et al., 2019]

Affine map minimizing transport cost between $\mathbb{P}_{\mathcal{S}} = \mathcal{N}(\mu_{\mathcal{S}}, \Sigma_{\mathcal{S}})$ and $\mathbb{P}_{\mathcal{T}} = \mathcal{N}(\mu_{\mathcal{T}}, \Sigma_{\mathcal{T}})$:

$$m(x) = Ax + b \quad \textbf{with} \quad A = \Sigma_{\mathcal{S}}^{-\frac{1}{2}} \left(\Sigma_{\mathcal{S}}^{\frac{1}{2}} \Sigma_{\mathcal{T}} \Sigma_{\mathcal{S}}^{\frac{1}{2}} \right)^{\frac{1}{2}} \Sigma_{\mathcal{S}}^{-\frac{1}{2}}, \quad b = \mu_{\mathcal{T}} - A\mu_{\mathcal{S}}$$

Affine Distribution Alignment Across Domains

Many benefits:

- easy to code
- fast
- no need to access source and target data simultaneously
- no/few hyperparameters
- unsupervised: no target labels required

Affine Distribution Alignment Across Domains

How to deal with ...?

- ...target shifts, i.e. joint shifts (X, y)?
- ...several source domains, i.e. multi-source adaptation?
- ...$\mathscr{X} = \mathscr{M}$, i.e. data on manifolds?
- ...time-series, e.g. $\mathscr{X} = \mathbb{R}^d$ with d large?
- ...feature maps in neural network, i.e. end-to-end normalization?

3. Joint shifts in (X, y) on manifolds: GOPSA

Apolline Mellot*, AC*, Sylvain Chevallier,
Alexandre Gramfort, Denis A. Engemann

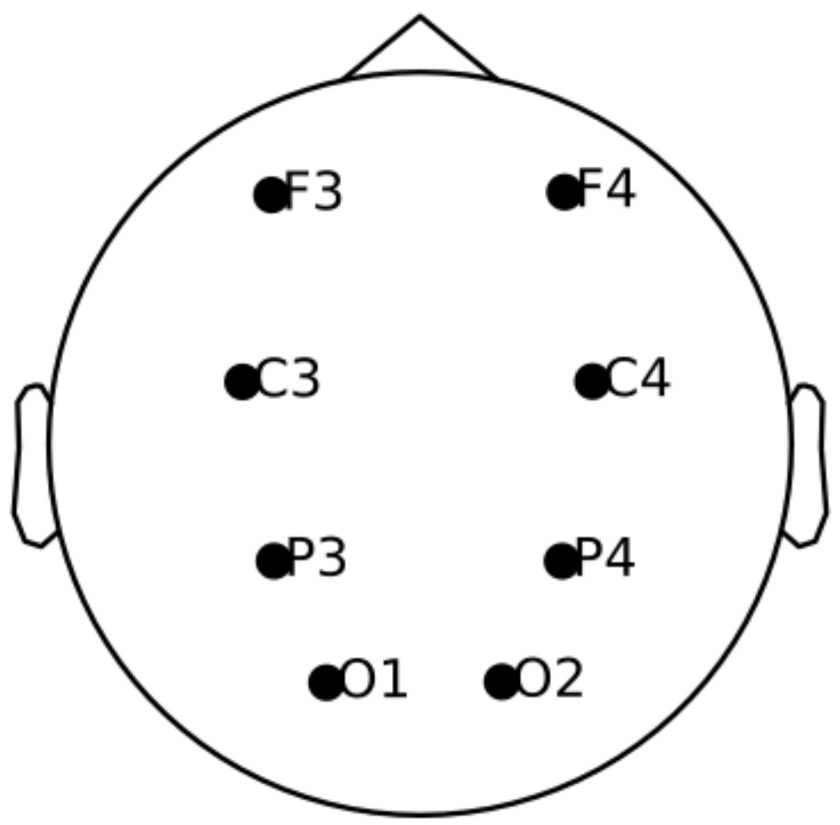
*Equal contribution

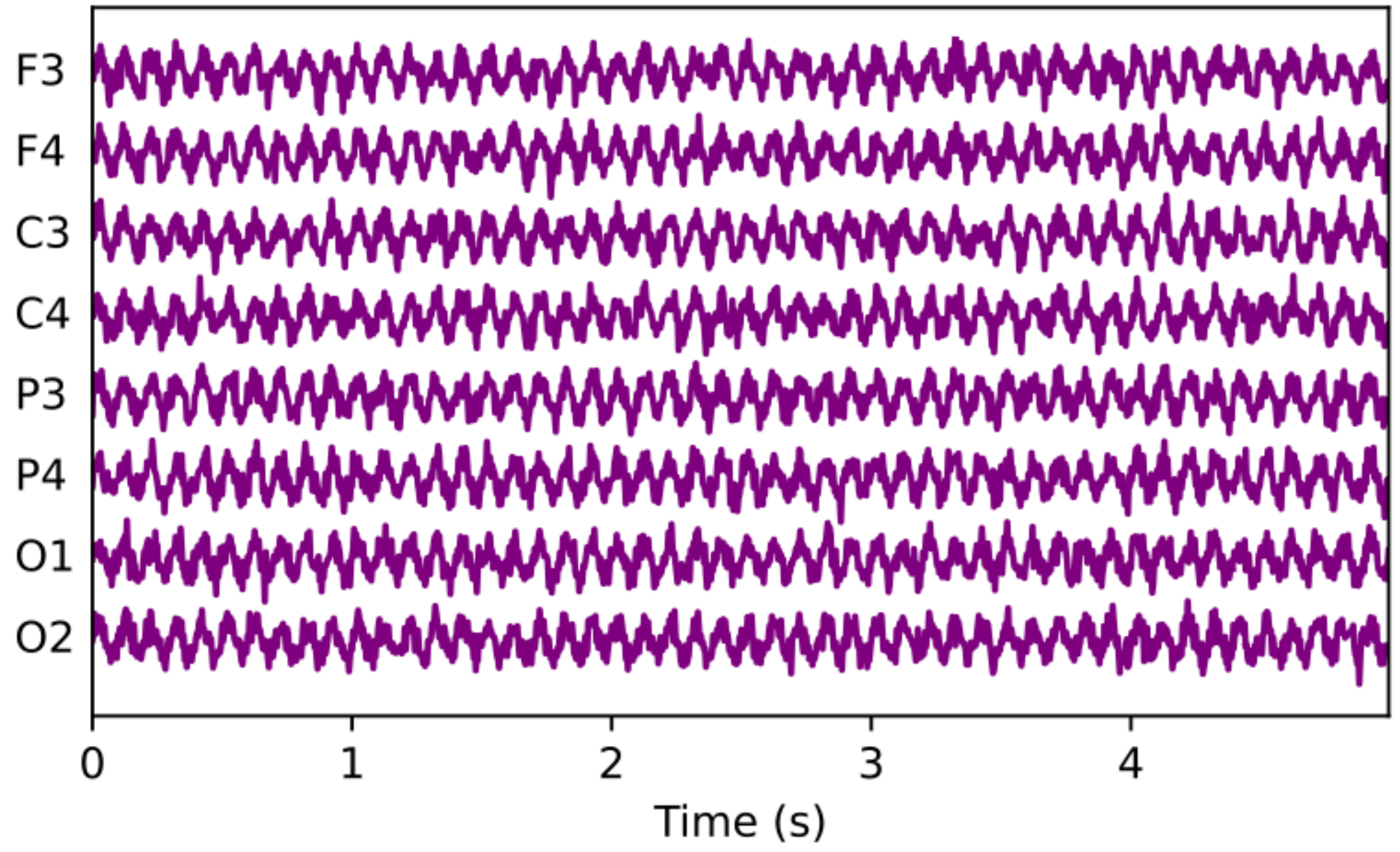
NeurIPS 2024, Spotlight

https://arxiv.org/abs/2407.03878

Machine learning on EEG data

Predicting an outcome from neural activity measured by EEG, a non-invasive technique that records brain signals from the scalp.



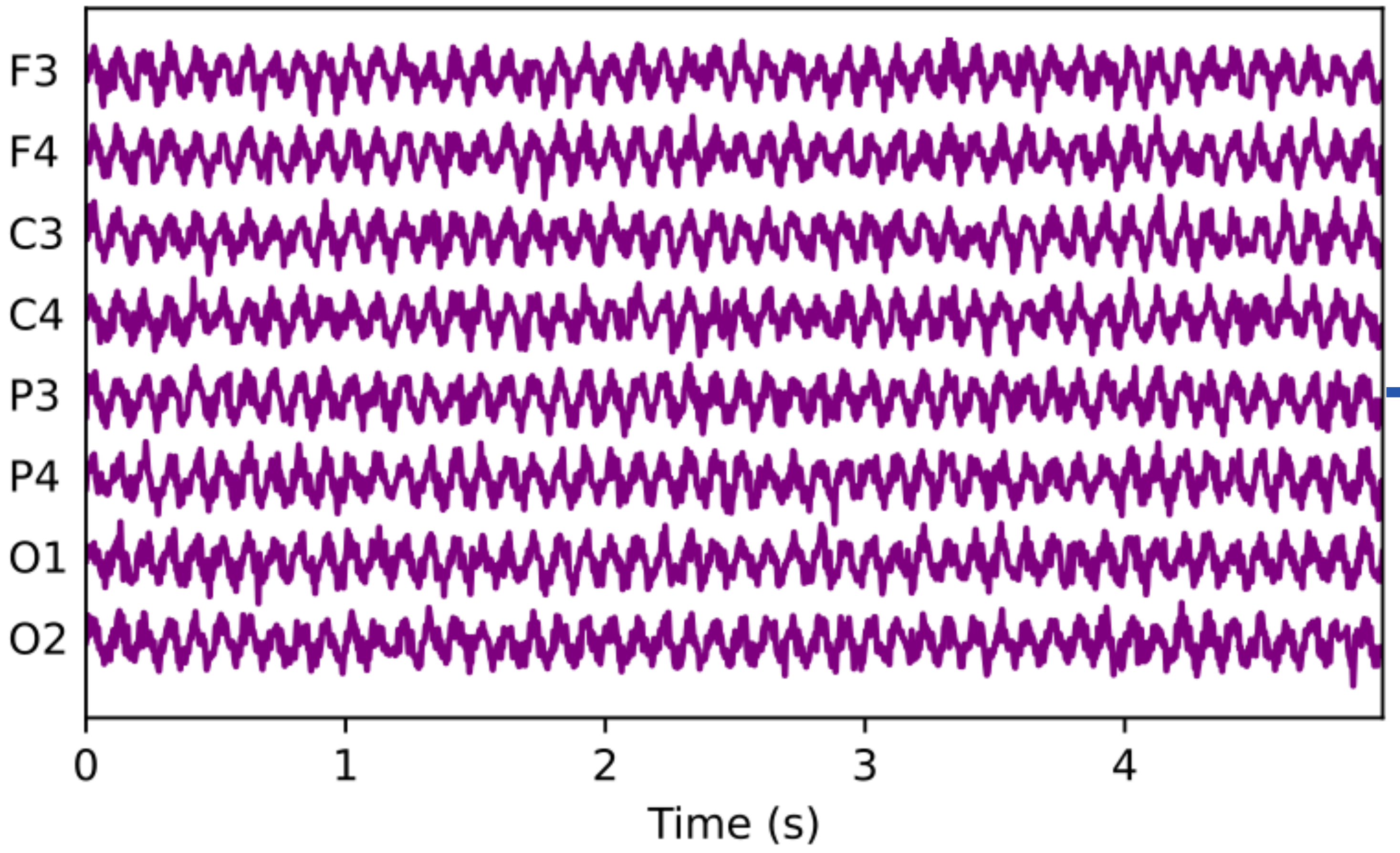


Applications:

sleep staging (awake, deep sleep, ...), brain-computer interface, biomarker regression, ...

Brain age: subject-level regression

1 time series = 1 subject



→ Co-spectrum matrix $\in \mathbb{S}_d^{++}$ → Tangent space features $\in \mathbb{R}^{\frac{d(d+1)}{2}}$ → Regression model → $\hat{y}$

Manifold-valued data

(Age estimation)

Brain age is a proxy for cognitive health and detects deviations from typical aging

Co-spectrum

Let $x_\ell \in \mathbb{R}^d$ be a stationary zero-mean time series

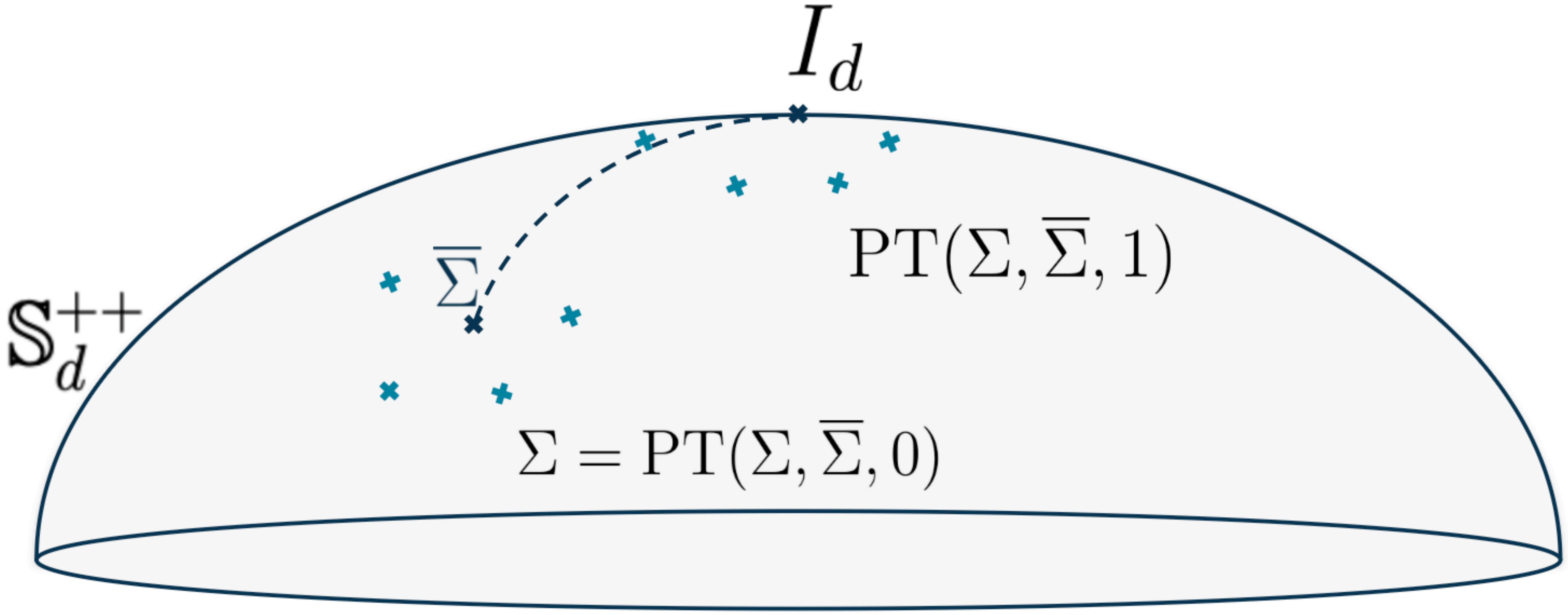
$$R(\tau) = \mathbb{E}\left[x_{\ell+\tau} x_\ell^\top\right] \in \mathbb{R}^{d\times d}$$

$$C(s) = \mathrm{Re}\left\{\sum_{\tau=-\infty}^{\infty} e^{-i2\pi s\tau} R(\tau)\right\} \in \mathbb{S}_d^{++}$$

$[C(s_0), \ldots, C(s_{F-1})]$ represents the time-series and belongs to the manifold $(\mathbb{S}_d^{++})^F$

Regression on $\mathcal{S}_d^{++}$

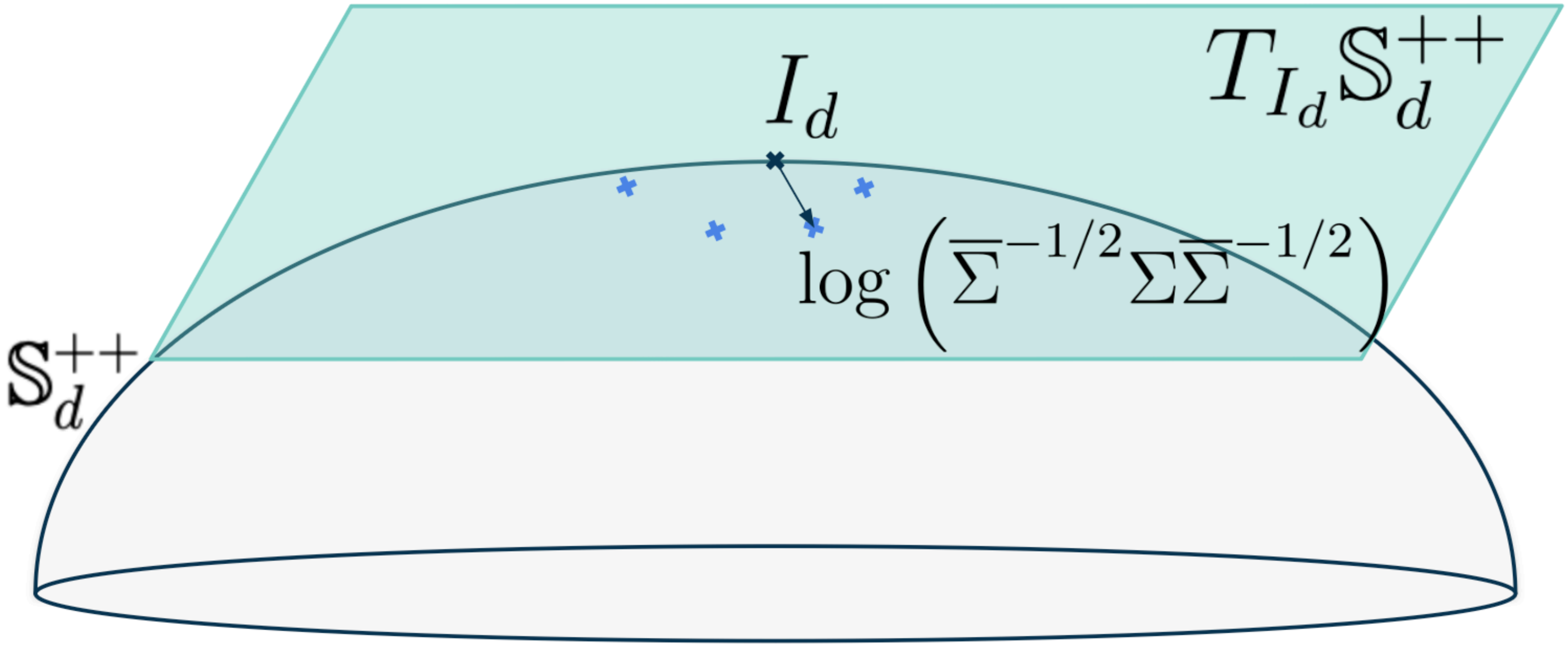
① Parallel transport moves Σ from $\bar{\Sigma}$ to I_d at $\alpha \in [0,1]$: $\mathrm{PT}(\Sigma, \bar{\Sigma}, \alpha) = \bar{\Sigma}^{-\alpha/2} \Sigma \bar{\Sigma}^{-\alpha/2}$



[Sabbagh, et al., 2019]

Regression on $\mathcal{S}_d^{++}$

② Riemannian logarithm of Σ at I_d



[Sabbagh, et al., 2019]

Regression on $\mathcal{S}_d^{++}$

③ Upper-triangular vectorization

$$\phi(\Sigma, \bar{\Sigma}) \triangleq \mathrm{uvec}\left(\log\left(\bar{\Sigma}^{-1/2}\Sigma\bar{\Sigma}^{-1/2}\right)\right) \in \mathbb{R}^{d(d+1)/2}$$

④ Regression

$$\hat{y} = \beta^\top \phi(\Sigma, \bar{\Sigma})$$

[Sabbagh, et al., 2019]

Data shifts in neuroscience

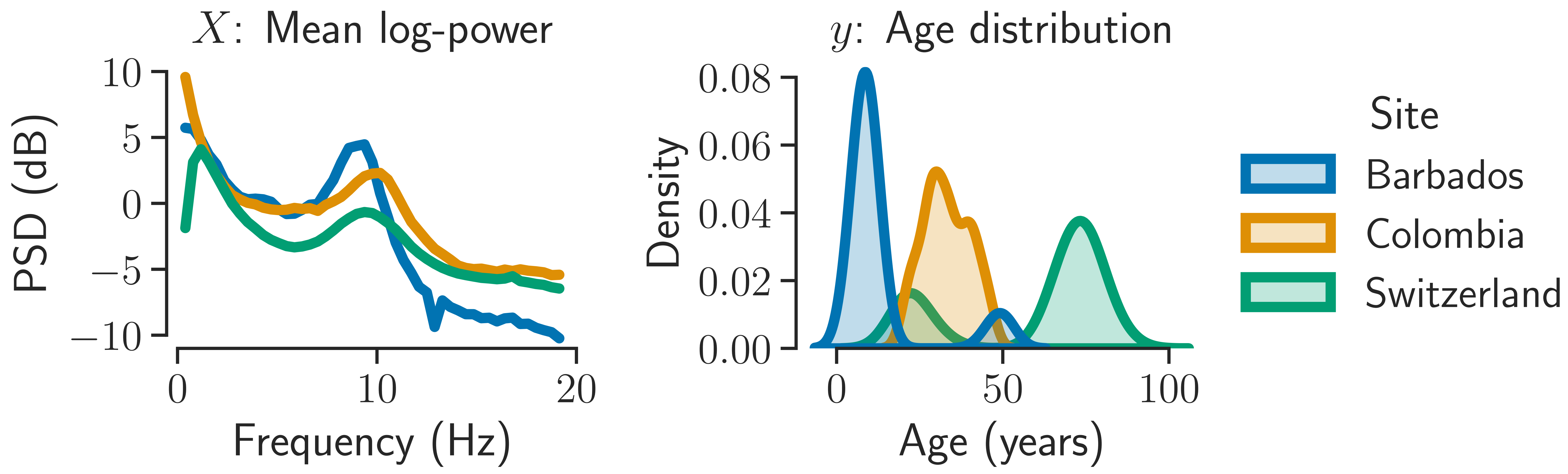
Sources of variability:

- population: age, gender, diseased or healthy, ...
- hardware and preprocessing: sensors, positions, sampling rates, ...
- task-related variability: tasks during recording, level of engagement, external stimuli, ...

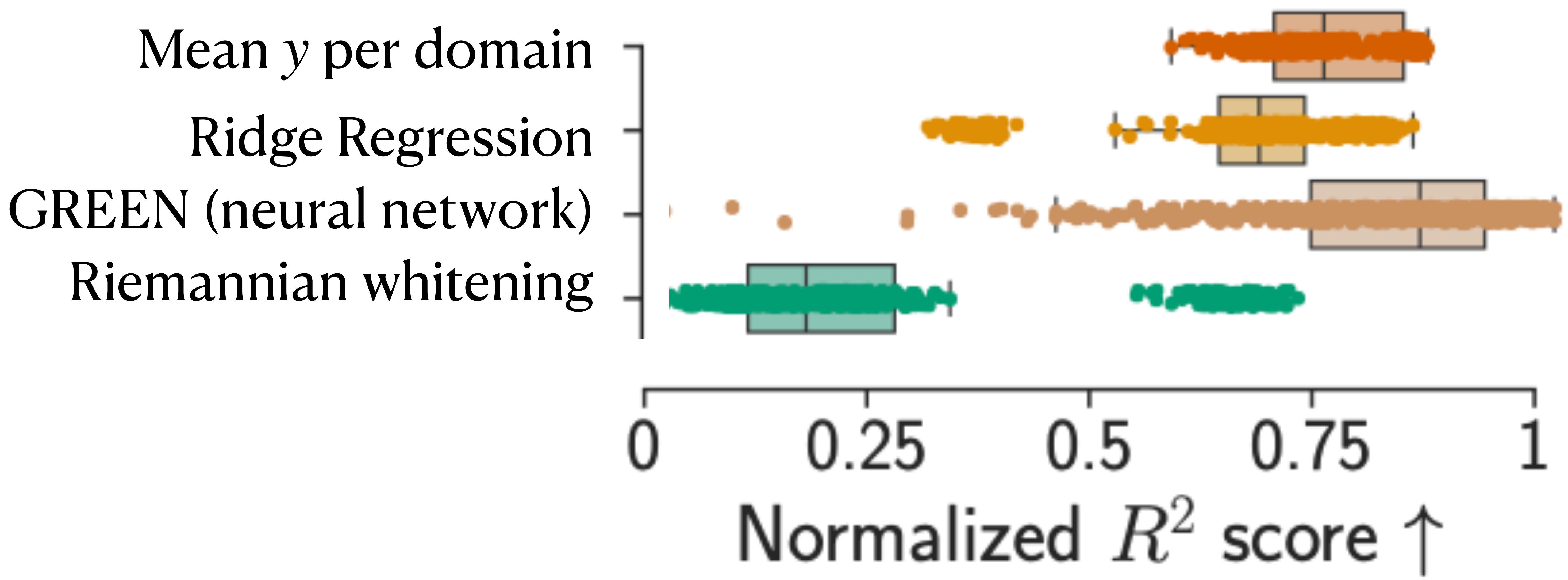
HarMNqEEG

dataset

[Li et al., 2022]



Data shifts in neuroscience



GOPSA: Geodesic Optimization for Predictive Shift Adaptation

Setup:

K source domains: $\left\{(\Sigma_{k,n}, y_{k,n})\right\}_{n=1}^{N_k}$ for $k \in \{1,\ldots,K\}$ and $N_{\mathcal{S}} = \sum_{k=1}^{K} N_k$

One target domain: $\left\{\Sigma_{\mathcal{T},n}\right\}_{n=1}^{N_{\mathcal{T}}}$, $\bar{y}_{\mathcal{T}}$ (mean value)

GOPSA: Geodesic Optimization for Predictive Shift Adaptation

Setup:

K source domains: $\left\{(\Sigma_{k,n}, y_{k,n})\right\}_{n=1}^{N_k}$ for $k \in \{1,\ldots,K\}$ and $N_{\mathcal{S}} = \sum_{k=1}^{K} N_k$

One target domain: $\left\{\Sigma_{\mathcal{T},n}\right\}_{n=1}^{N_{\mathcal{T}}}, \bar{y}_{\mathcal{T}}$ (mean value)

Learn the parallel transport $\alpha \in [0,1]$ per domain:

$$\phi(\Sigma, \bar{\Sigma}, \alpha) \triangleq \operatorname{uvec}\left(\log_{I_d}\left(\operatorname{PT}(\Sigma, \bar{\Sigma}, \alpha)\right)\right) = \operatorname{uvec}\left(\log\left(\bar{\Sigma}^{-\alpha/2}\Sigma\bar{\Sigma}^{-\alpha/2}\right)\right)$$

GOPSA: Geodesic Optimization for Predictive Shift Adaptation

Learnable source and target data matrices w.r.t. $\alpha_{\mathcal{S}} \in [0,1]^K$ and $\alpha_{\mathcal{T}}$:

$$Z_{\mathcal{S}}(\alpha_{\mathcal{S}}) \triangleq \left[\phi(\Sigma_{1,1}, \bar{\Sigma}_1, \alpha_1),\ \ldots, \phi(\Sigma_{K,N_K}, \bar{\Sigma}_K, \alpha_K)\right]^\top \in \mathbb{R}^{N_{\mathcal{S}} \times d(d+1)/2}$$

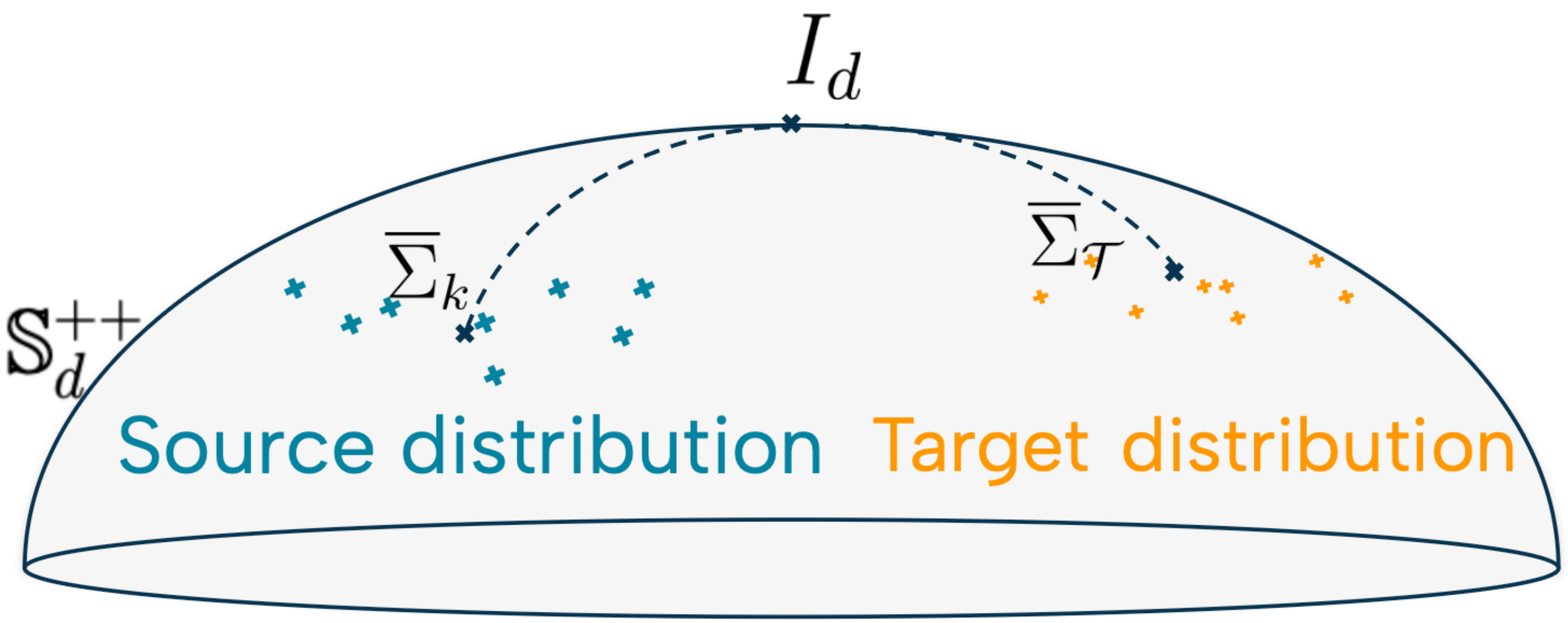
$$Z_{\mathcal{T}}(\alpha_{\mathcal{T}}) \triangleq \left[\phi(\Sigma_{\mathcal{T},1}, \bar{\Sigma}_{\mathcal{T}}, \alpha_{\mathcal{T}}),\ \ldots, \phi(\Sigma_{\mathcal{T},1}, \bar{\Sigma}_{\mathcal{T}}, \alpha_{\mathcal{T}})\right]^\top \in \mathbb{R}^{N_{\mathcal{T}} \times d(d+1)/2}$$

GOPSA: Geodesic Optimization for Predictive Shift Adaptation

Learnable source and target data matrices w.r.t. $\alpha_{\mathcal{S}} \in [0,1]^K$ and $\alpha_{\mathcal{T}}$:

$$Z_{\mathcal{S}}(\alpha_{\mathcal{S}}) \triangleq \left[\phi(\Sigma_{1,1}, \bar{\Sigma}_1, \alpha_1), \ \dots, \phi(\Sigma_{K,N_K}, \bar{\Sigma}_K, \alpha_K)\right]^\top \in \mathbb{R}^{N_{\mathcal{S}} \times d(d+1)/2}$$

$$Z_{\mathcal{T}}(\alpha_{\mathcal{T}}) \triangleq \left[\phi(\Sigma_{\mathcal{T},1}, \bar{\Sigma}_{\mathcal{T}}, \alpha_{\mathcal{T}}), \ \dots, \phi(\Sigma_{\mathcal{T},1}, \bar{\Sigma}_{\mathcal{T}}, \alpha_{\mathcal{T}})\right]^\top \in \mathbb{R}^{N_{\mathcal{T}} \times d(d+1)/2}$$

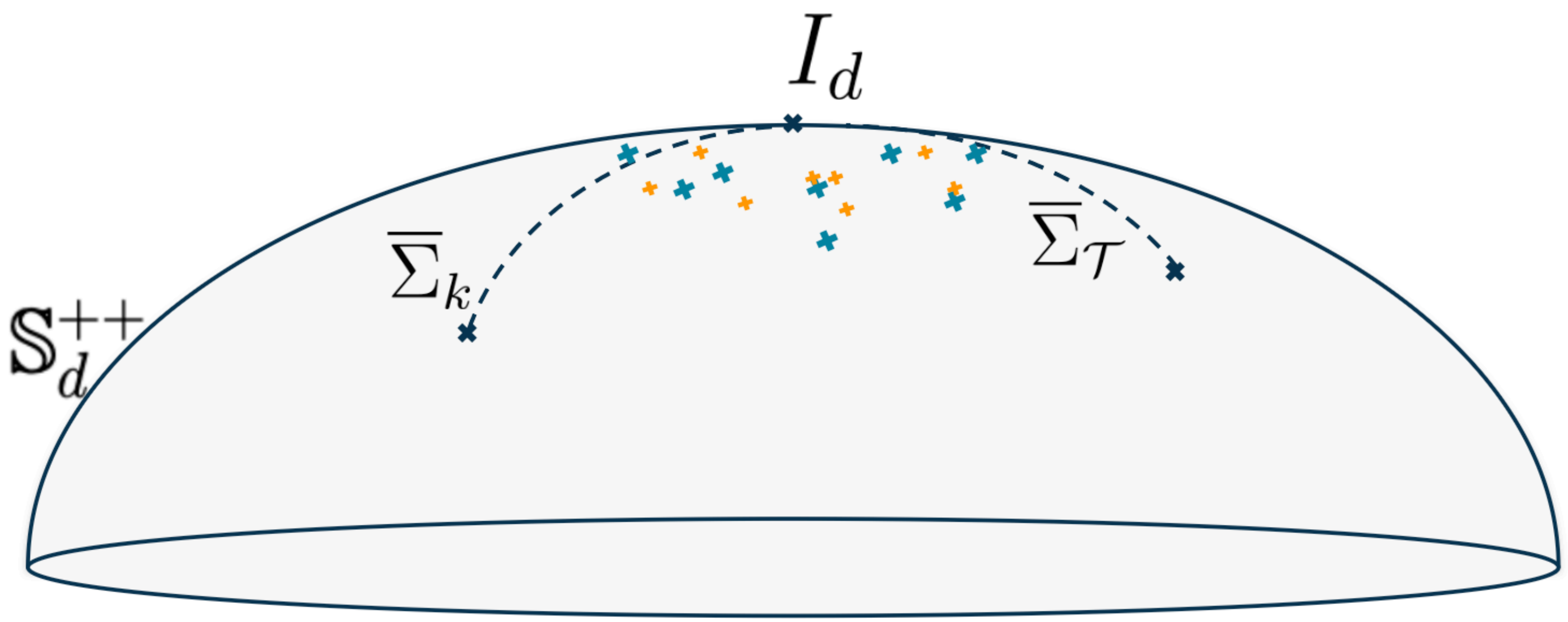


GOPSA: Geodesic Optimization for Predictive Shift Adaptation

Learnable source and target data matrices w.r.t. $\alpha_{\mathcal{S}} \in [0,1]^K$ and $\alpha_{\mathcal{T}}$:

$$Z_{\mathcal{S}}(\alpha_{\mathcal{S}}) \triangleq \left[\phi(\Sigma_{1,1}, \bar{\Sigma}_1, \alpha_1), \ \ldots, \phi(\Sigma_{K,N_K}, \bar{\Sigma}_K, \alpha_K)\right]^\top \in \mathbb{R}^{N_{\mathcal{S}} \times d(d+1)/2}$$

$$Z_{\mathcal{T}}(\alpha_{\mathcal{T}}) \triangleq \left[\phi(\Sigma_{\mathcal{T},1}, \bar{\Sigma}_{\mathcal{T}}, \alpha_{\mathcal{T}}), \ \ldots, \phi(\Sigma_{\mathcal{T},1}, \bar{\Sigma}_{\mathcal{T}}, \alpha_{\mathcal{T}})\right]^\top \in \mathbb{R}^{N_{\mathcal{T}} \times d(d+1)/2}$$



GOPSA: Geodesic Optimization for Predictive Shift Adaptation

Train-time:

$$\alpha^*_{\mathcal{S}} \triangleq \arg \min_{\alpha \in [0,1]^K} \left\| y_{\mathcal{S}} - Z_{\mathcal{S}}(\alpha)\, \beta^*_{\mathcal{S}}(\alpha) \right\|_2^2$$

subject to $\beta^*_{\mathcal{S}}(\alpha)$ is the ridge estimator

Test-time:

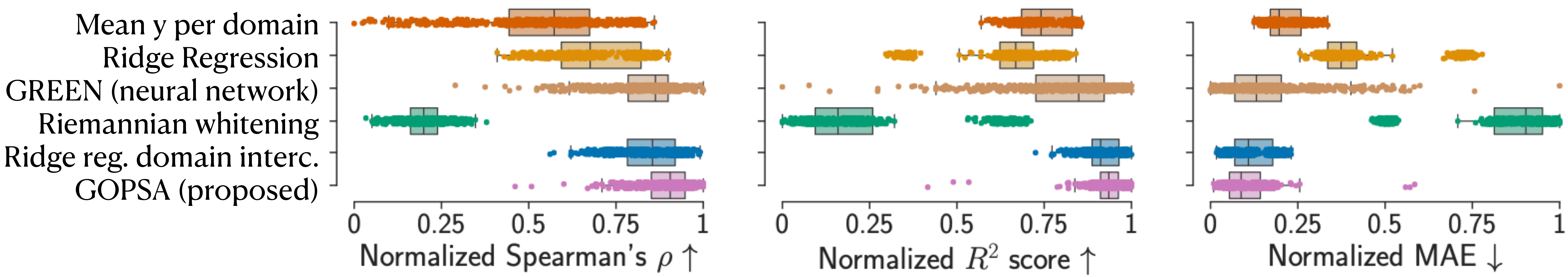
(Optimization) $$\alpha^*_{\mathcal{T}} \triangleq \arg \min_{\alpha \in [0,1]} \left(\bar{y}_{\mathcal{T}} - \text{mean}\left(Z_{\mathcal{T}}(\alpha) \beta^*_{\mathcal{S}}(\alpha^*_{\mathcal{S}}) \right) \right)^2$$

(Prediction) $$\hat{y}_{\mathcal{T}} \triangleq Z_{\mathcal{T}}(\alpha^*_{\mathcal{T}})\, \beta^*_{\mathcal{S}}(\alpha^*_{\mathcal{S}})$$

Evaluation

HarMNqEEG dataset [Li et al., 2022]

14 recording sites, 1500 human participants, random combination of source sites, single random target site



4. Temporal Normalization: PSDNorm layer

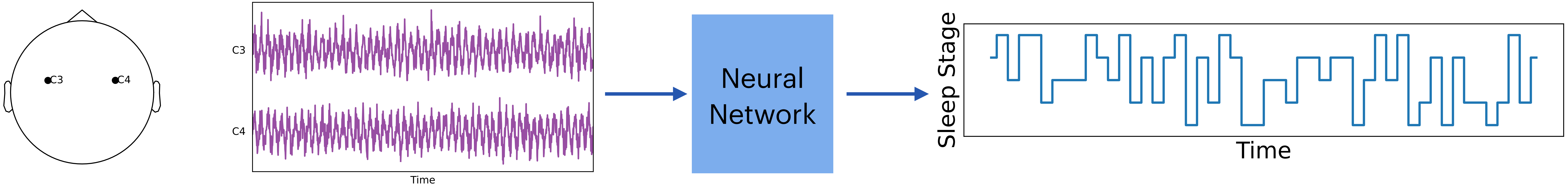
Théo Gnassounou, AC,
Rémi Flamary, Alexandre Gramfort

https://arxiv.org/abs/2503.04582

Sleep staging: epoch-level classification

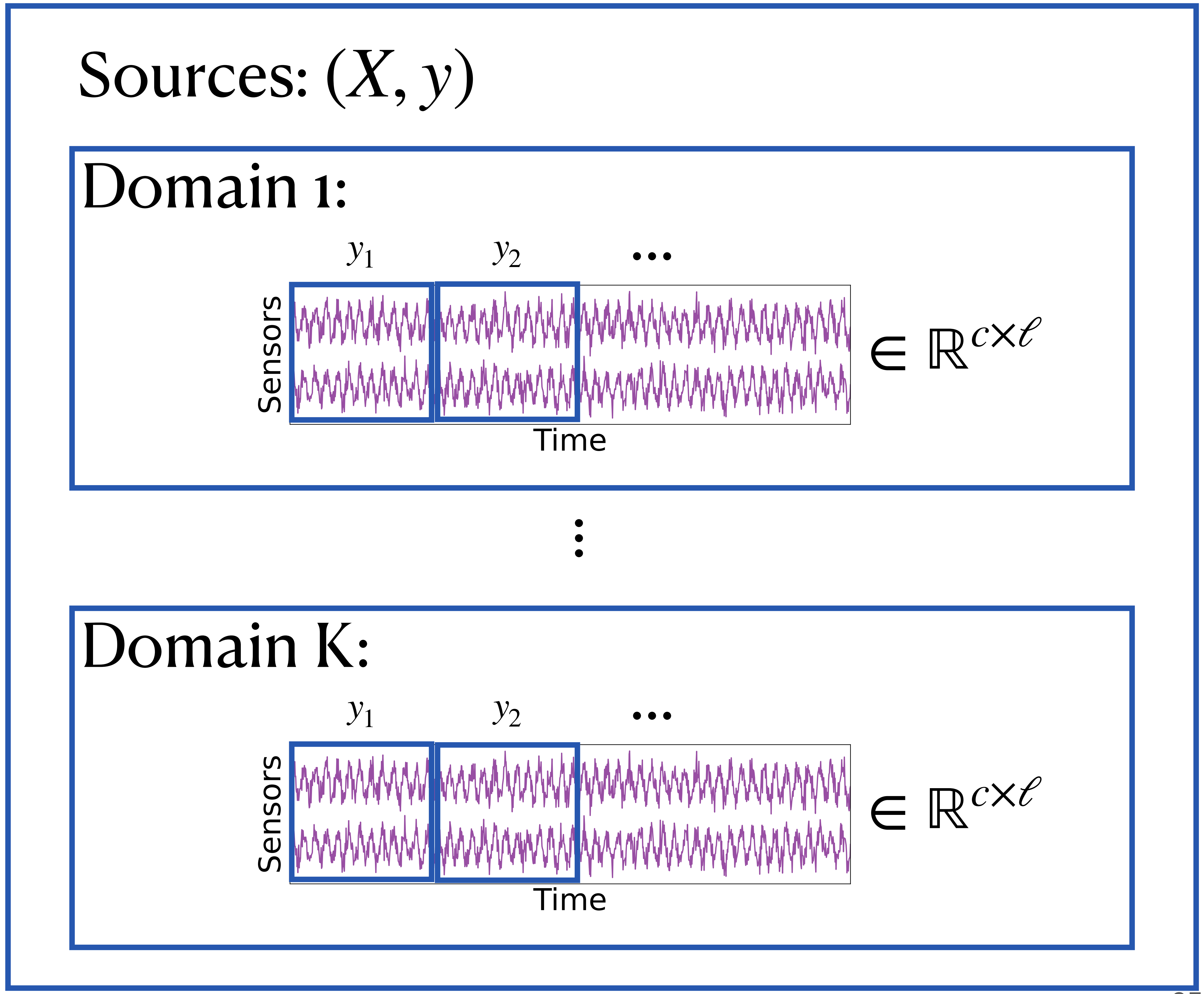
1 time series = 1 night = 8 hours

1 prediction every epoch (30s.)



Multi-source sleep staging

Classification setup: (e.g. 1 domain = 1 hospital)

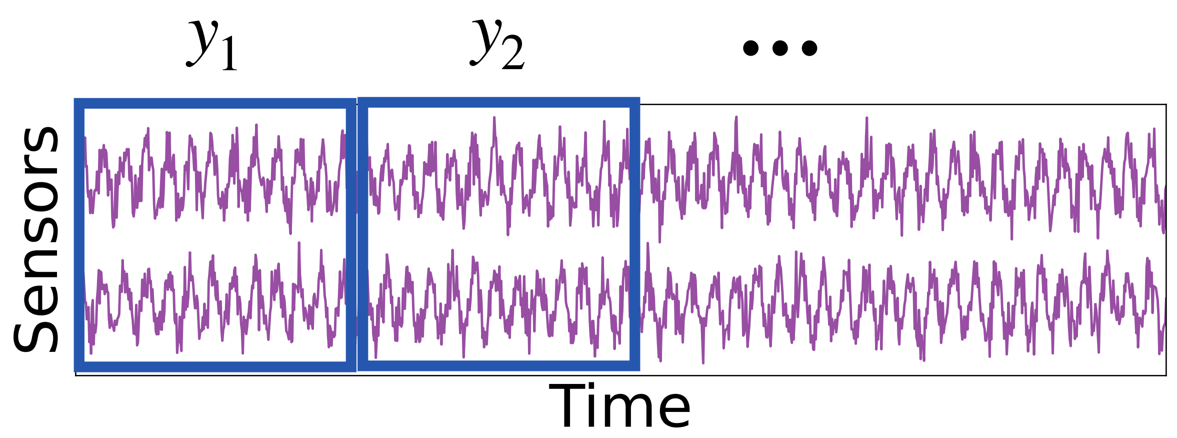


Multi-source sleep staging

Classification setup: (e.g. 1 domain = 1 hospital)

Sources: (X, y)

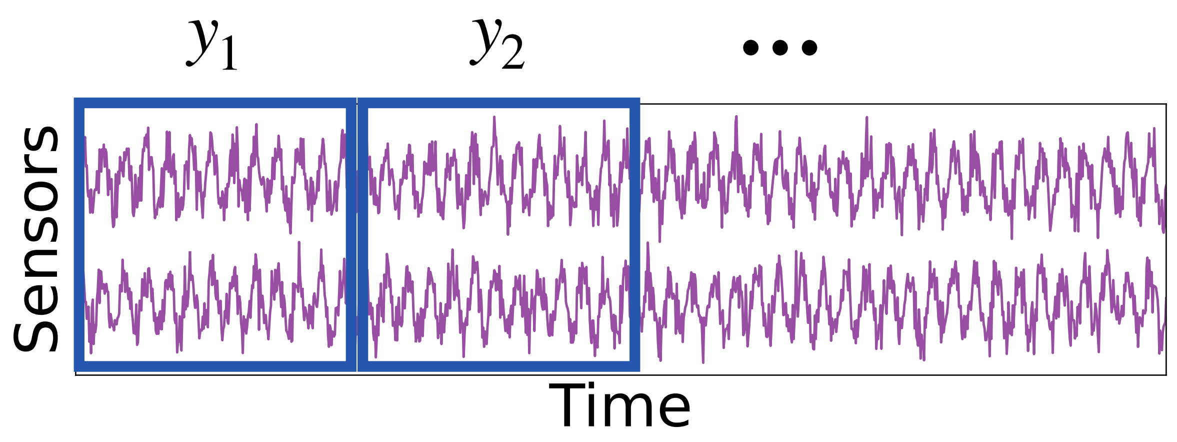
Domain 1:



$\in \mathbb{R}^{c \times \ell}$

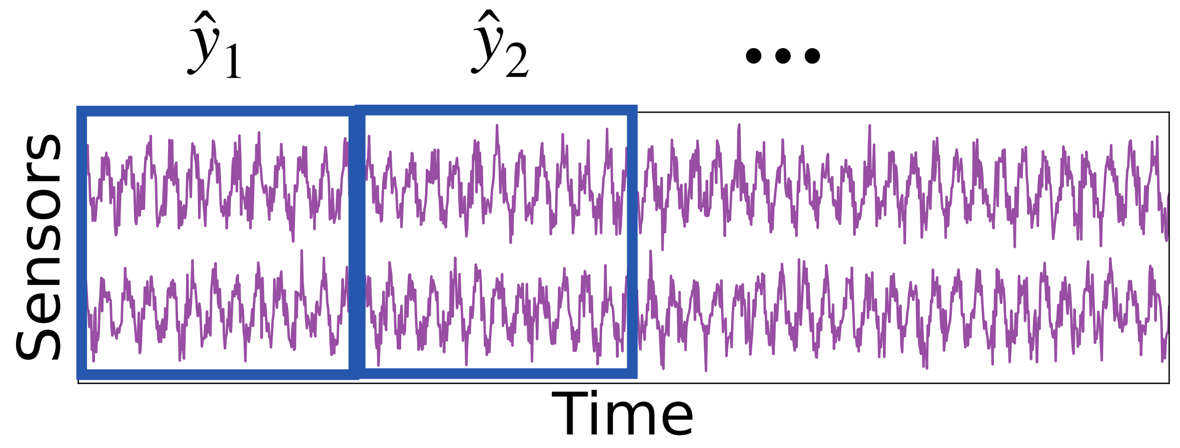
⋮

Domain K:



$\in \mathbb{R}^{c \times \ell}$

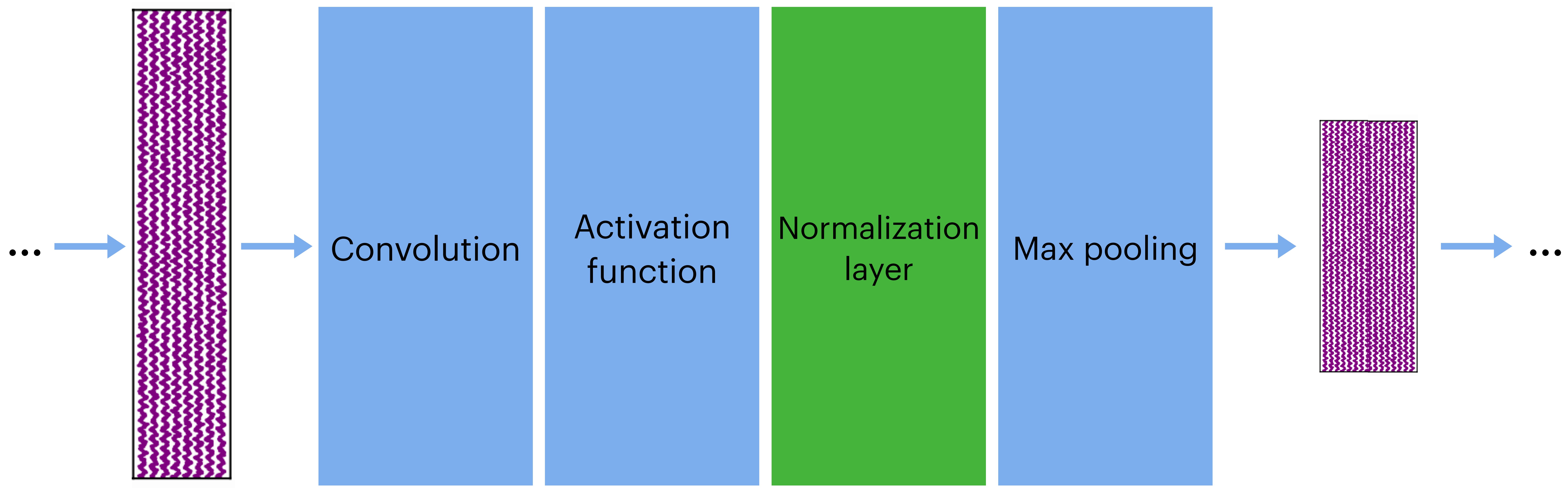
Target: X



$\in \mathbb{R}^{c \times \ell}$

Neural network achitecture

Encoder block: transforms a feature map into a new one



Normalization layers: BatchNorm, InstanceNorm, LayerNorm...

InstanceNorm layer

Normalizes a feature map $G \in \mathbb{R}^{c \times \ell}$ to have zero mean and unit variance

$$\widetilde{G}_{m,l} \triangleq \frac{G_{m,l} - \widehat{\mu}_m}{\sqrt{\widehat{\sigma}_m^2 + \varepsilon}}$$

InstanceNorm

$$\widehat{\mu}_m \triangleq \frac{1}{\ell} \sum_{l=1}^{\ell} G_{m,l}$$

Mean across time

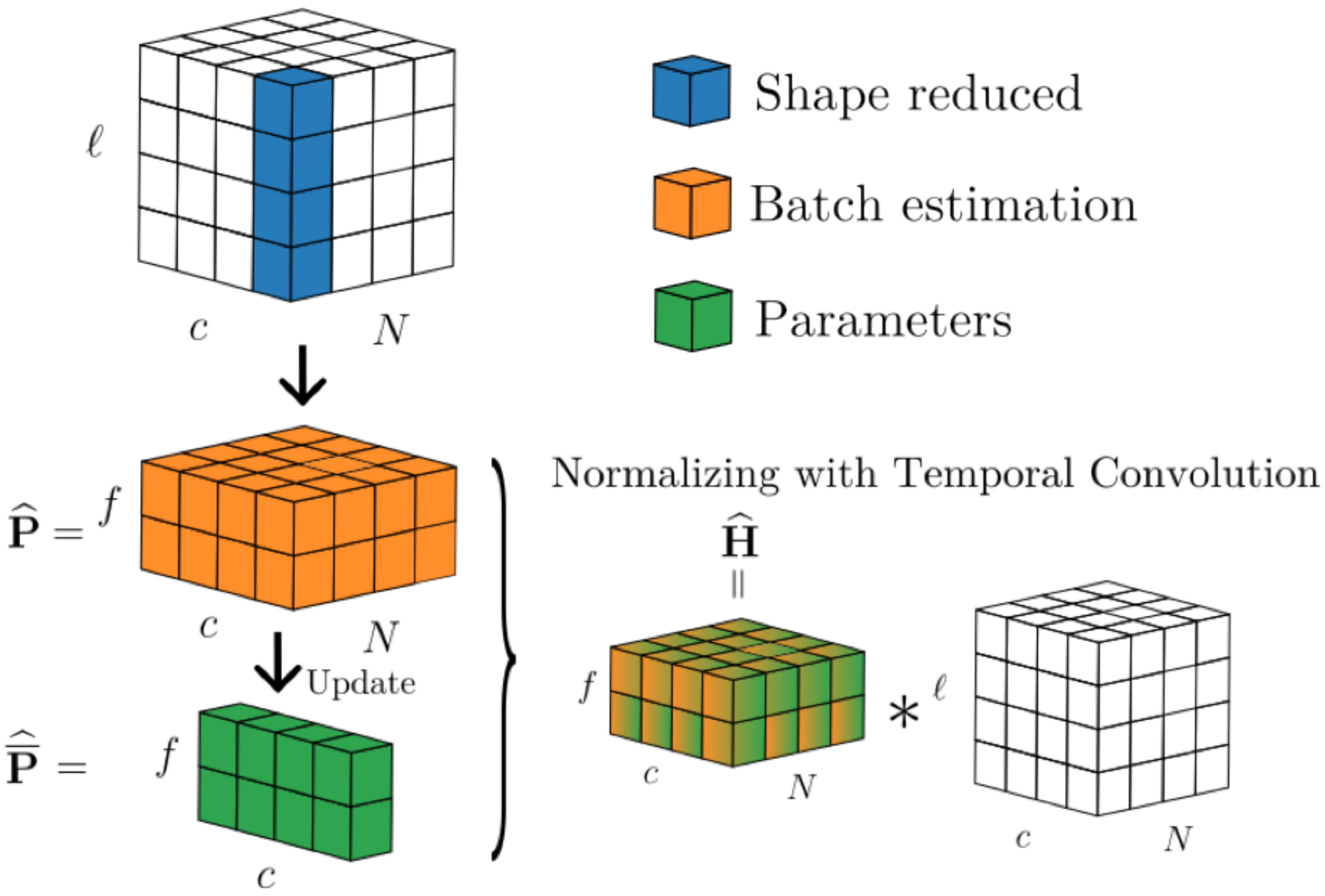
$$\widehat{\sigma}_m^2 \triangleq \frac{1}{\ell} \sum_{l=1}^{\ell} \left(G_{m,l} - \widehat{\mu}_m\right)^2$$

Variance across time

[Ulyanov et al., 2016]

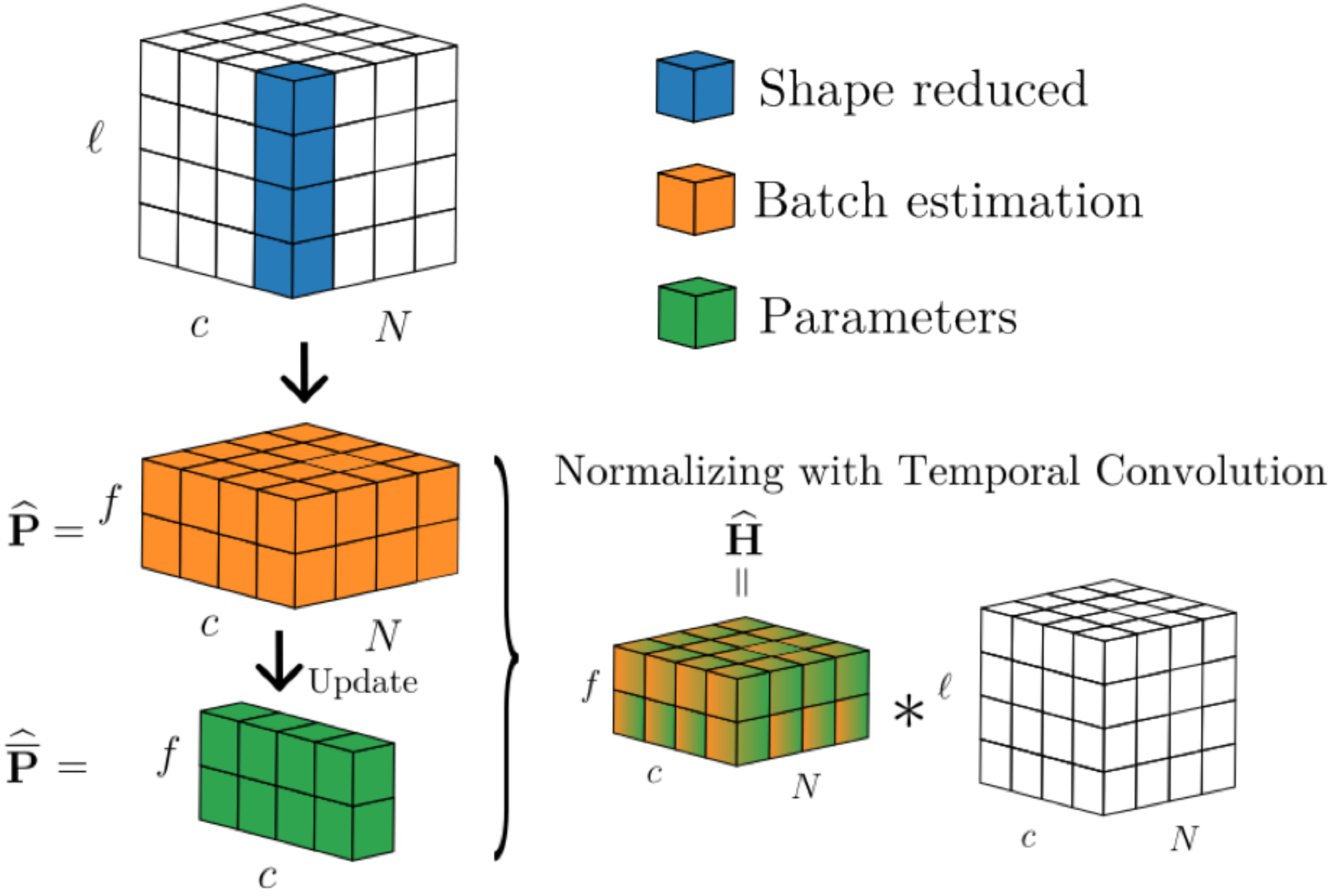
PSDNorm: Power Spectral Density Normalization

N : batch size, c : channel number, f : filter size (hyperparameter)



PSDNorm: Power Spectral Density Normalization

N : batch size, c : channel number, f : filter size (hyperparameter)



Remarks:

- normalization of the temporal autocorrelation
- $f = 1$ and $\hat{\bar{P}} = 1 \implies$ InstanceNorm
- the bigger f the stronger the normalization
- structured LinOT for multi-source DA

Algorithms

Train time:

$\mathscr{B} = \{G^{(1)}, \dots, G^{(N)}\} \subset \mathbb{R}^{c \times \ell}$

PSD estimation $\quad \hat{P}^{(j)} = \mathrm{Welch}\left(G^{(j)} - \hat{\mu}^{(j)} 1_c^\top\right)$

Batch barycentre $\quad \hat{\bar{P}}_{\mathscr{B}} = \left(\frac{1}{N} \sum_{n=1}^{N} (\hat{P}^{(j)})^{\odot \frac{1}{2}}\right)^{\odot 2}$

Running mean $\quad \hat{\bar{P}} \leftarrow \left((1-\alpha)\hat{\bar{P}}^{\odot \frac{1}{2}} + \alpha \hat{\bar{P}}_{\mathscr{B}}^{\odot \frac{1}{2}}\right)^{\odot 2}$

Filter computation $\quad \hat{H}^{(j)} = \frac{1}{\sqrt{f}} \left(\hat{\bar{P}} \oslash \hat{P}^{(j)}\right)^{\odot \frac{1}{2}} F_f^*$

f-Monge mapping $\quad \tilde{G}^{(j)} = \left(G^{(j)} - \hat{\mu}^{(j)} 1_\ell^\top\right) * \hat{H}^{(j)}$

Algorithms

Train time:

$\mathscr{B} = \{G^{(1)}, \ldots, G^{(N)}\} \subset \mathbb{R}^{c \times \ell}$

PSD estimation $\hat{P}^{(j)} = \text{Welch}\left(G^{(j)} - \hat{\mu}^{(j)} 1_c^\top\right)$

Batch barycentre $\hat{\bar{P}}_{\mathscr{B}} = \left(\frac{1}{N} \sum_{n=1}^{N} (\hat{P}^{(j)})^{\odot \frac{1}{2}}\right)^{\odot 2}$

Running mean $\hat{\bar{P}} \leftarrow \left((1-\alpha)\hat{\bar{P}}^{\odot \frac{1}{2}} + \alpha \hat{\bar{P}}_{\mathscr{B}}^{\odot \frac{1}{2}}\right)^{\odot 2}$

Filter computation $\hat{H}^{(j)} = \frac{1}{\sqrt{f}} \left(\hat{\bar{P}} \oslash \hat{P}^{(j)}\right)^{\odot \frac{1}{2}} F_f^*$

f-Monge mapping $\tilde{G}^{(j)} = \left(G^{(j)} - \hat{\mu}^{(j)} 1_\ell^\top\right) * \hat{H}^{(j)}$

Test time:

feature map: $G \in \mathbb{R}^{c \times \ell}$

$\hat{P} = \text{Welch}\left(G - \hat{\mu} 1_c^\top\right)$

$\hat{H} = \frac{1}{\sqrt{f}} \left(\hat{\bar{P}} \oslash \hat{P}\right)^{\odot \frac{1}{2}} F_f^*$

$\tilde{G} = \left(G - \hat{\mu} 1_\ell^\top\right) * \hat{H}$

Evaluation

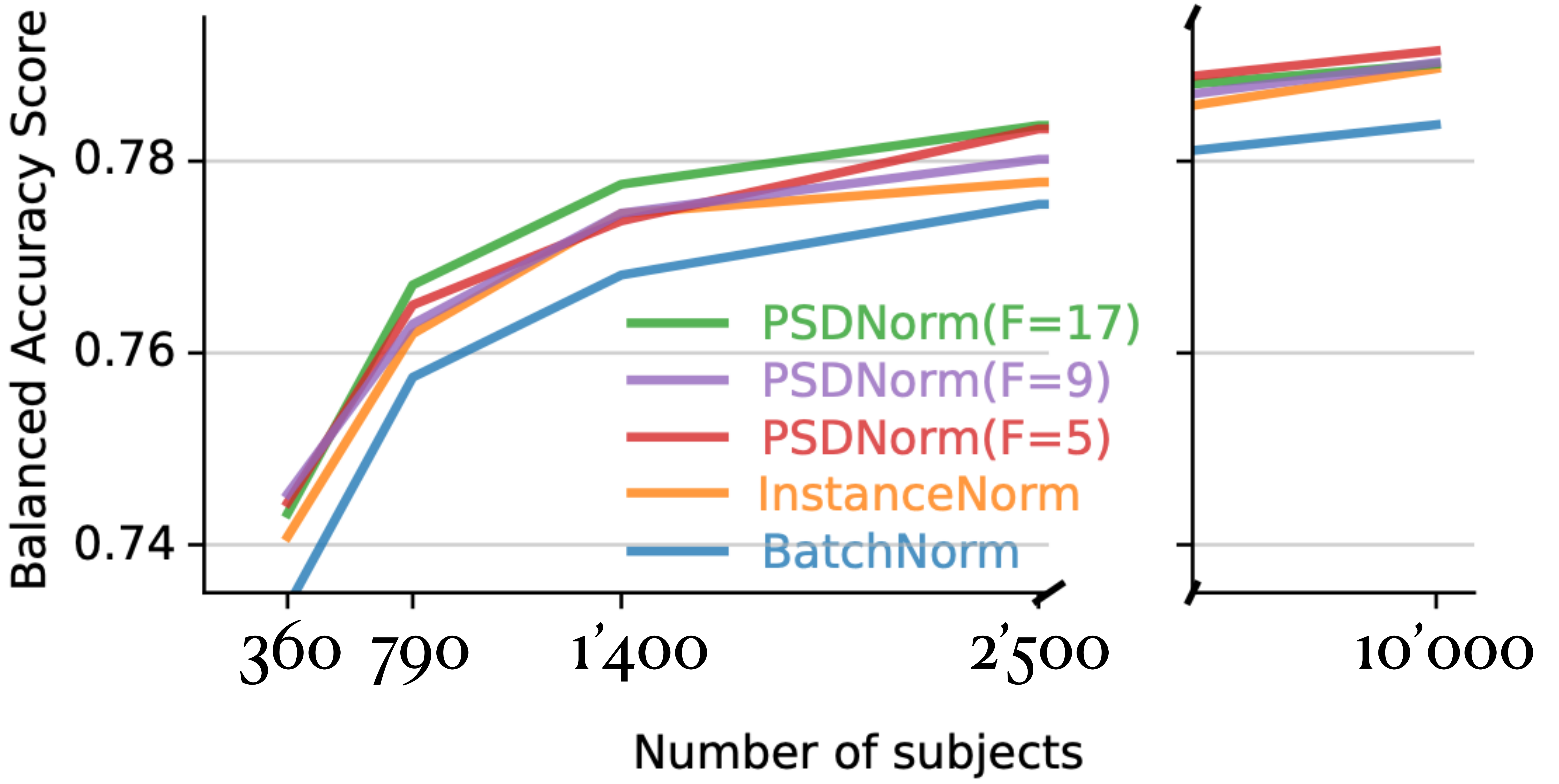
Setup:

- 10 datasets (11K subjects, >100K hours, >10M labels, 360 gb)
- leave-one-dataset-out
- averaged on 3 random seeds

Evaluation

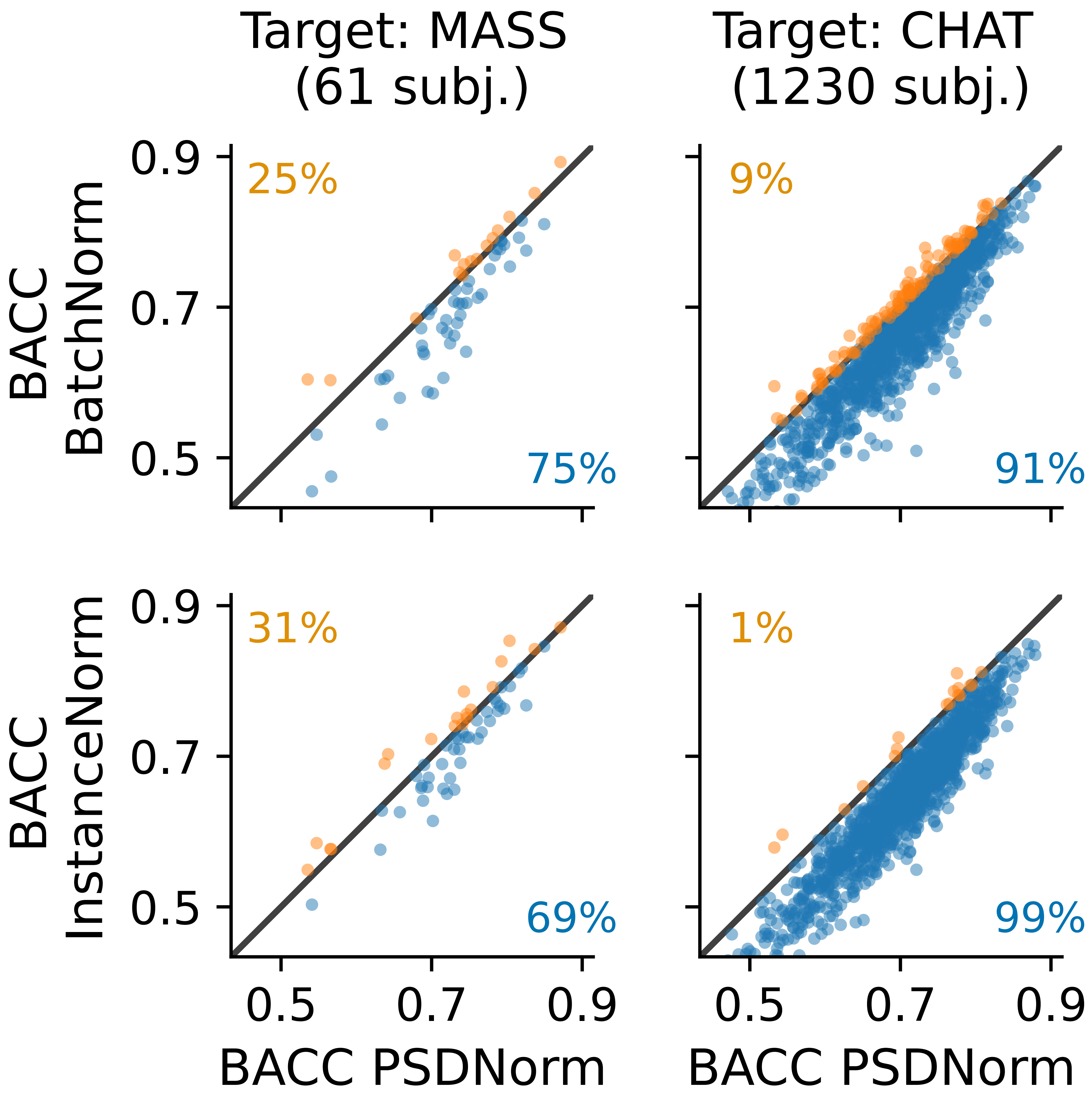
Setup:

- 10 datasets (11K subjects, >100K hours, >10M labels, 360 gb)
- leave-one-dataset-out
- averaged on 3 random seeds



Evaluation

Subject-wise balanced accuracy on MASS and CHAT datasets for different normalization layers in USleep trained on 2500 subjects



Conclusions

Benchmarks matter

→ Skada-Bench provides the first cross-modal, realistically-validated UDA benchmark

Back to basics: affine alignments as a unifying thread

→ simple moment-matching methods help reduce cross-domain error without labels

Joint shifts in (X, y) on manifolds

→ EEG data live on manifolds, yet we can design simple and efficient methods—such as GOPSA—that outperform more complex alternatives

Deep time-frequency normalization

→ PSDNorm aligns spectral representations to improve cross-dataset sleep staging across 10 corpora

References

[Lalou, et al. 2024] Lalou, Y., Gnassounou, T., Collas, A., de Mathelin, A., Kachaiev, O., Odonnat, A., Gramfort, A., Moreau, T. and Flamary, R., 2024. SKADA-Bench: Benchmarking Unsupervised Domain Adaptation Methods with Realistic Validation. arXiv preprint arXiv:2407.11676.

[Mellot, et al. 2024] Mellot, A., Collas, A., Chevallier, S., Gramfort, A. and Engemann, D., 2024, December. Geodesic optimization for predictive shift adaptation on EEG data. In *NeurIPS 2024-38th Conference on Neural Information Processing Systems*.

[Sun et al., 2017] Sun, B., Feng, J. and Saenko, K., 2017. Correlation alignment for unsupervised domain adaptation. *Domain adaptation in computer vision applications*, pp.153-171.

[Flamary et al., 2019] Flamary, R., Lounici, K. and Ferrari, A., 2019. Concentration bounds for linear monge mapping estimation and optimal transport domain adaptation. arXiv preprint arXiv:1905.10155.

[Sabbagh et al., 2019] Sabbagh, D., Ablin, P., Varoquaux, G., Gramfort, A. and Engemann, D.A., 2019. Manifold-regression to predict from MEG/EEG brain signals without source modeling. Advances in Neural Information Processing Systems, 32.

References

[Li et al., 2022] Li, M., Wang, Y., Lopez-Naranjo, C., Hu, S., Reyes, R.C.G., Paz-Linares, D., Areces-Gonzalez, A., Abd Hamid, A.I., Evans, A.C., Savostyanov, A.N. and Calzada-Reyes, A., 2022. Harmonized-multinational qEEG norms (HarMNqEEG). NeuroImage, 256, p.119190.

[Ulyanov et al., 2016] Ulyanov, D., Vedaldi, A. and Lempitsky, V., 2016. Instance normalization: The missing ingredient for fast stylization. arXiv preprint arXiv:1607.08022.

[Paillard et al., 2025] Paillard, J., Hipp, J.F. and Engemann, D.A., 2025. GREEN: A lightweight architecture using learnable wavelets and Riemannian geometry for biomarker exploration with EEG signals. Patterns.

2. SKADA-Bench: Benchmarking Unsupervised Domain Adaptation Methods with Realistic Validation

Yanis Lalou*, Théo Gnassounou*, Antoine Collas*, Antoine de Mathelin*, Oleksii Kachaiev, Ambroise Odonnat, Alexandre Gramfort, Thomas Moreau, Rémi Flamary

*Equal contribution

https://arxiv.org/abs/2407.11676

Datasets

Dataset	Modality	Preprocessing	Pairs of Adaptation	Classes	Samples	Features
Office 31	CV	Decaff + PCA	6	31	470 ± 350	100
Office Home	CV	ResNet + PCA	12	65	3897 ± 850	100
MNIST/USPS	CV	Vect + PCA	2	10	3000 / 10000	50
20 Newsgroup	NLP	LLM + PCA	6	2	3728 ± 174	50
Amazon Review	NLP	LLM + PCA	12	4	2000	50
Mushrooms	Tabular	One Hot Encoding	2	2	4062 ± 546	117
Phishing	Tabular	None	2	2	5527 ± 1734	30
BCI	Biosignals	Cov + TS	9	4	288	253

3. Normalization for joint shifts in (X, y): GOPSA

Apolline Mellot*, Antoine Collas*, Sylvain Chevallier,
Alexandre Gramfort, Denis A. Engemann

*Equal contribution

NeurIPS 2024, Spotlight

https://arxiv.org/abs/2407.03878

GOPSA: Geodesic Optimization for Predictive Shift Adaptation

Setup:

K source domains: $\left\{(\Sigma_{k,n}, y_{k,n})\right\}_{n=1}^{N_k}$ for $k \in \{1,\ldots,K\}$ and $N_{\mathcal{S}} = \sum_{k=1}^{K} N_k$

One target domain: $\left\{\Sigma_{\mathcal{T},n}\right\}_{n=1}^{N_{\mathcal{T}}}, \bar{y}_{\mathcal{T}}$ (mean value)

Learn the parallel transport $\alpha \in [0,1]$ per domain:

$$\phi(\Sigma, \bar{\Sigma}, \alpha) \triangleq \operatorname{uvec}\left(\log_{I_d}\left(\mathrm{PT}(\Sigma, \bar{\Sigma}, \alpha)\right)\right) = \operatorname{uvec}\left(\log\left(\bar{\Sigma}^{-\alpha/2}\Sigma\bar{\Sigma}^{-\alpha/2}\right)\right)$$

GOPSA: Geodesic Optimization for Predictive Shift Adaptation

Train-time:

$$\alpha^*_{\mathcal{S}} \triangleq \arg \min_{\alpha \in [0,1]^K} \left\| y_{\mathcal{S}} - Z_{\mathcal{S}}(\alpha)\, \beta^*_{\mathcal{S}}(\alpha) \right\|_2^2$$

$$\text{subject to} \quad \beta^*_{\mathcal{S}}(\alpha) \triangleq Z_{\mathcal{S}}(\alpha)^\top \left(\lambda I_N + Z_{\mathcal{S}}(\alpha) Z_{\mathcal{S}}(\alpha)^\top \right)^{-1} y_{\mathcal{S}}$$

Test-time:

$$\text{(Optimization)} \quad \alpha^*_{\mathcal{T}} \triangleq \arg \min_{\alpha \in [0,1]} \left(\bar{y}_{\mathcal{T}} - \frac{1}{N_{\mathcal{T}}} 1_{N_{\mathcal{T}}}^\top Z_{\mathcal{T}}(\alpha)\, \beta^*_{\mathcal{S}}(\alpha^*_{\mathcal{S}}) \right)^2$$

$$\text{(Prediction)} \quad \hat{y}_{\mathcal{T}} \triangleq Z_{\mathcal{T}}(\alpha^*_{\mathcal{T}})\, \beta^*_{\mathcal{S}}(\alpha^*_{\mathcal{S}})$$

GOPSA: Geodesic Optimization for Predictive Shift Adaptation

Train-time:

$$\alpha_{\mathcal{S}}^* \triangleq \arg \min_{\alpha \in [0,1]^K} \left\| y_{\mathcal{S}} - Z_{\mathcal{S}}(\alpha)\, \beta_{\mathcal{S}}^*(\alpha) \right\|_2^2$$

$$\text{subject to} \quad \beta_{\mathcal{S}}^*(\alpha) \triangleq Z_{\mathcal{S}}(\alpha)^\top \left(\lambda I_N + Z_{\mathcal{S}}(\alpha) Z_{\mathcal{S}}(\alpha)^\top\right)^{-1} y_{\mathcal{S}}$$

Test-time:

Assumption: $\bar{\Sigma}_{\mathcal{T}}, \bar{y}_{\mathcal{T}}$ are known

$$\text{(Optimization)} \quad \alpha_{\mathcal{T}}^* \triangleq \arg \min_{\alpha \in [0,1]} \left(\bar{y}_{\mathcal{T}} - \frac{1}{N_{\mathcal{T}}} 1_{N_{\mathcal{T}}}^\top Z_{\mathcal{T}}(\alpha)\, \beta_{\mathcal{S}}^*(\alpha_{\mathcal{S}}^*) \right)^2$$

$$\text{(Prediction)} \quad \hat{y}_{\mathcal{T}} \triangleq Z_{\mathcal{T}}(\alpha_{\mathcal{T}}^*)\, \beta_{\mathcal{S}}^*(\alpha_{\mathcal{S}}^*)$$

4. PSDNorm: Test-Time Temporal Normalization for Deep Learning in Sleep Staging

Théo Gnassounou, Antoine Collas,
Rémi Flamary, Alexandre Gramfort

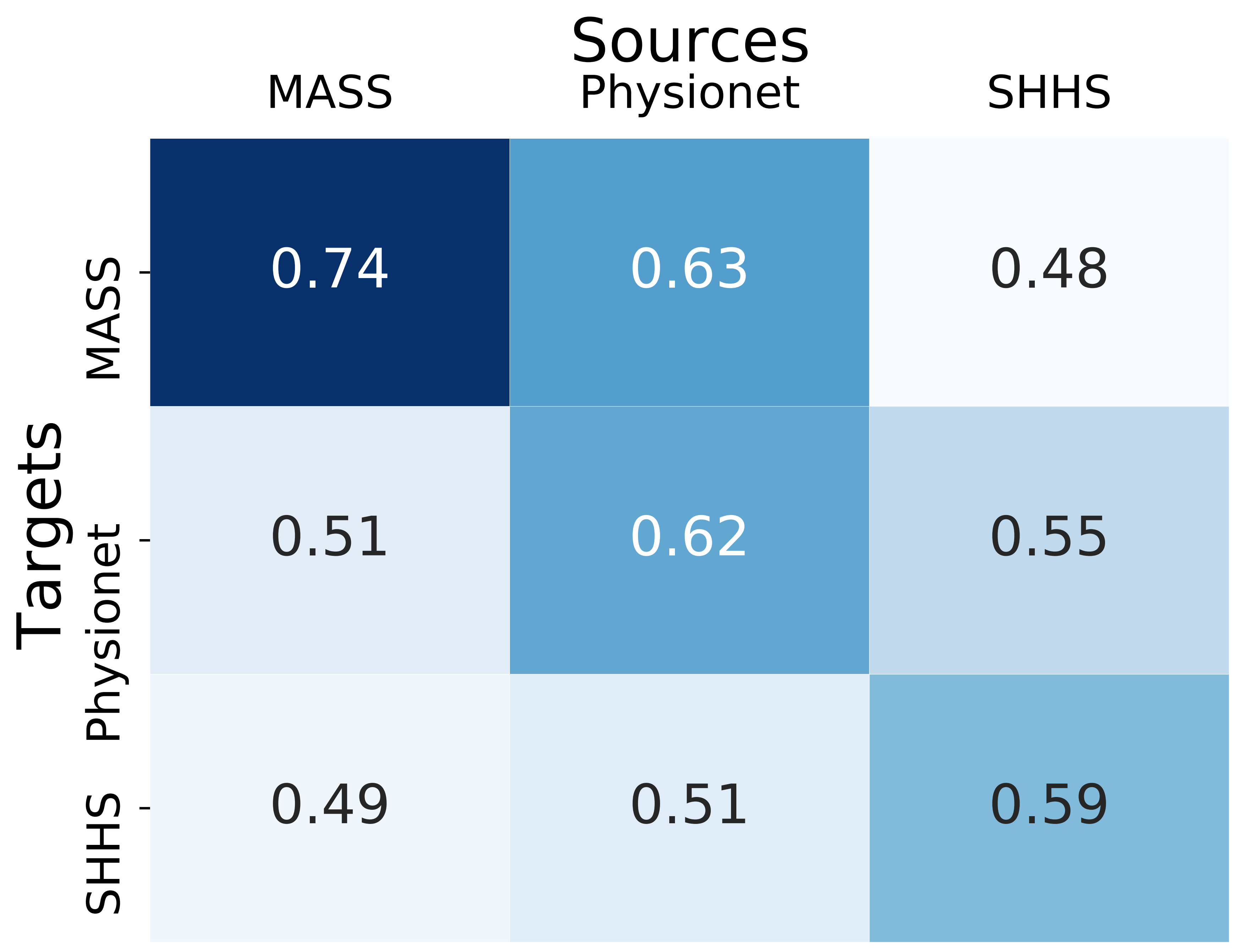
https://arxiv.org/abs/2503.04582

Data shifts in sleep staging

Example with 3 classification datasets:
MASS, Physionet, SHHS

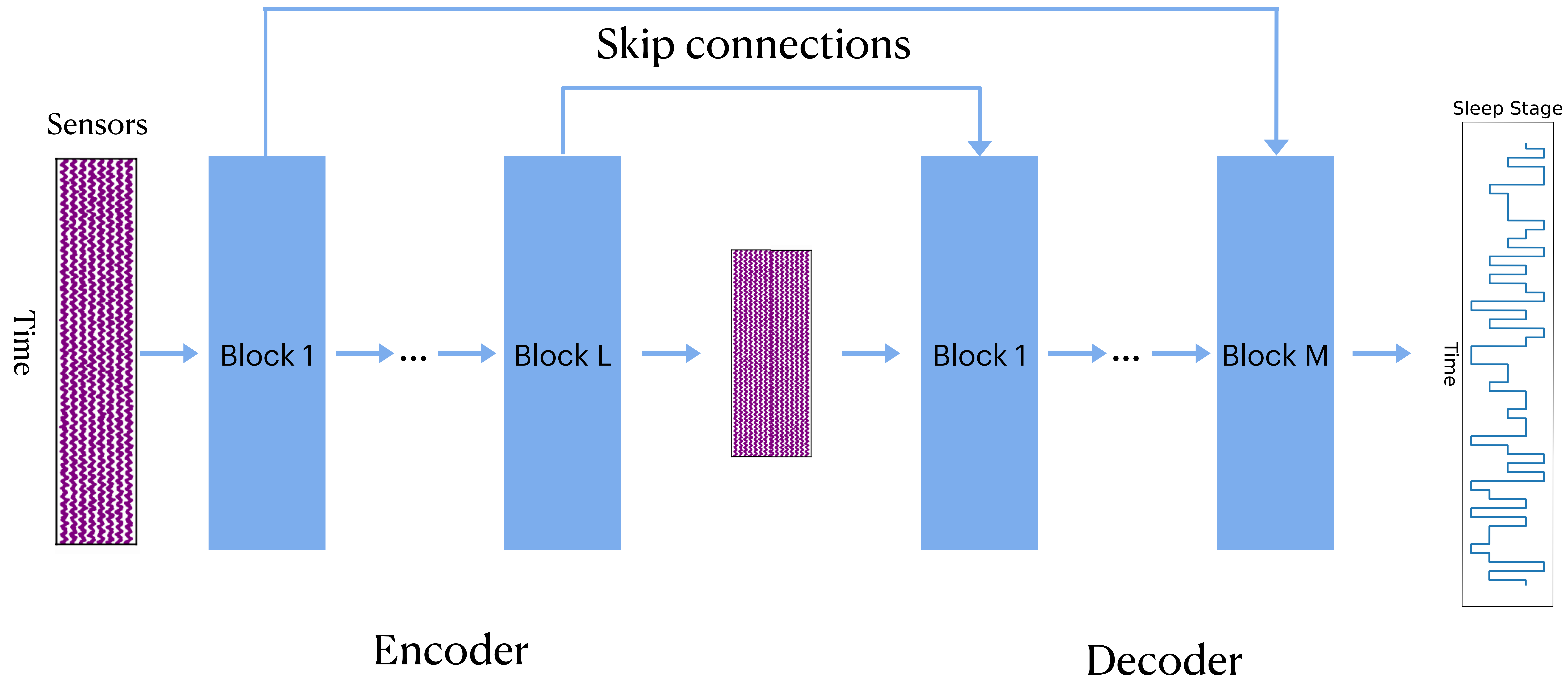
- Intra-dataset (cross-val.): good performance
- Inter-dataset: performance drops

Goal: improve generalization performance with DA



Balanced accuracies

U-Sleep architecture



BatchNorm layer

Normalizes a feature map $G \in \mathbb{R}^{c \times \ell}$ to have zero mean and unit variance

Train time:

batch of feature maps: $\{G^{(1)}, \ldots, G^{(N)}\} \subset \mathbb{R}^{c \times \ell}$

$$\hat{\mu}_m \triangleq \frac{1}{N\ell} \sum_{n=1}^{N} \sum_{l=1}^{\ell} G_{m,l}^{(n)}$$

Mean across batch and time

$$\hat{\sigma}_m^2 \triangleq \frac{1}{N\ell} \sum_{n=1}^{N} \sum_{l=1}^{\ell} \left(G_{m,l}^{(n)} - \hat{\mu}_m \right)^2$$

Variance across batch and time

BatchNorm layer

Train time:

batch of feature maps: $\{G^{(1)}, \ldots, G^{(N)}\} \subset \mathbb{R}^{c\times\ell}$

$$\widetilde{G}^{(j)}_{m,l} = \gamma_m \frac{G^{(j)}_{m,l} - \widehat{\mu}_m}{\sqrt{\widehat{\sigma}^2_m + \varepsilon}} + \beta_m$$

Normalization + affine transform

where $\gamma, \beta \in \mathbb{R}^c$ are learnable parameters

BatchNorm layer

Test time:

single feature map: $G \in \mathbb{R}^{c \times \ell}$

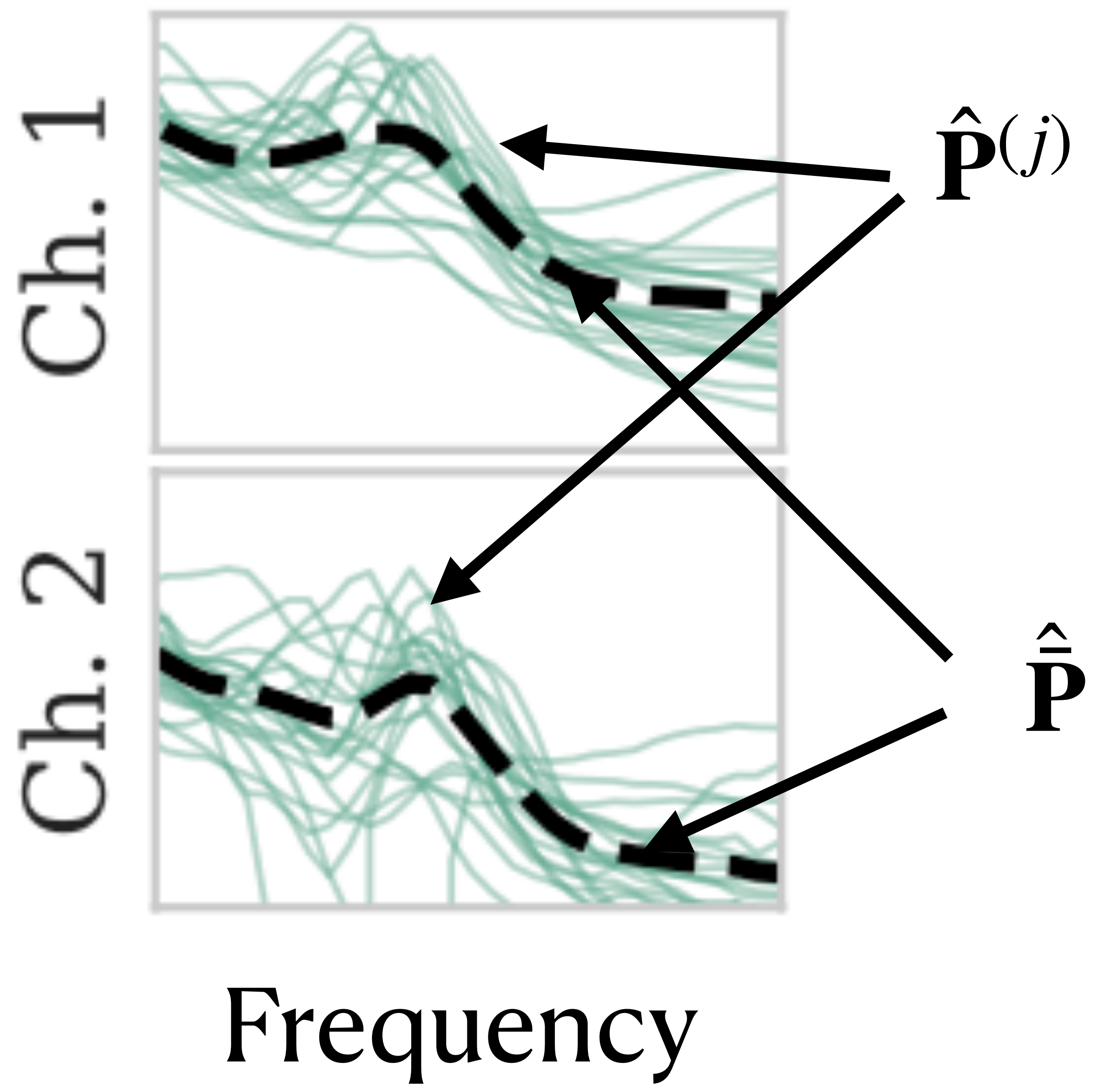
$$\widetilde{G}_{m,l} = \gamma_m \frac{G_{m,l} - \hat{\mu}_m}{\sqrt{\hat{\sigma}_m^2 + \varepsilon}} + \beta_m$$

Normalization + affine transform

where $\hat{\mu}$ and $\hat{\sigma}$ are the running mean and variance estimated during training.

PSDNorm: Power Spectral Density Normalization

Before normalization:



After normalization: