

Covariance Scattering Transforms

Andrea Cavallo*, Ayushman Raghuvanshi†, Sundeep Prabhakar Chepuri†, Elvin Isufi*

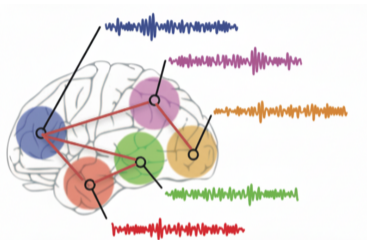
*Delft University of Technology, Delft, Netherlands

†Indian Institute of Science, Bangalore, India

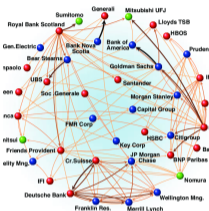
e.isufi-1@tudelft.nl

Covariance Relationships Between Data Points

- ▶ Consider observations $\mathbf{X} \in \mathbb{R}^{N \times T}$ and targets $\mathbf{y} \in \mathbb{R}^T$



Brain recordings



Financial time series

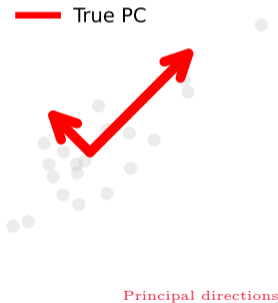


Motion sensors

- ▶ Data may contain **latent interconnections** that can help prediction
 - ⇒ Captured by the **covariance matrix** $\mathbf{C} = \mathbb{E}[(\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^T]$
 - ⇒ Sensitive to **statistical errors** and **outliers**

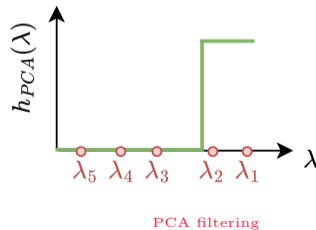
Principal Component Analysis (PCA)

- ▶ Project the data onto the **covariance eigenspace**
 - ⇒ $\mathbf{C} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$
 - ⇒ $\tilde{\mathbf{x}} = \mathbf{V}^T \mathbf{x}$
- ▶ Select (**filter**) the directions that **maximize the variance** of data points
 - ⇒ Used for **dimensionality reduction** by selecting only a few eigenvectors



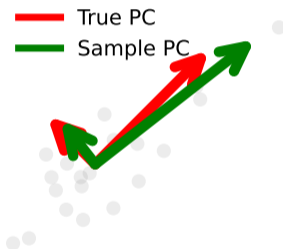
Principal Component Analysis (PCA)

- ▶ Project the data onto the **covariance eigenspace**
 - ⇒ $\mathbf{C} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$
 - ⇒ $\tilde{\mathbf{x}} = \mathbf{V}^T \mathbf{x}$
- ▶ Select (**filter**) the directions that **maximize the variance** of data points
 - ⇒ Used for **dimensionality reduction** by selecting only a few eigenvectors



Principal Component Analysis (PCA)

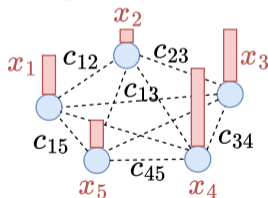
- ▶ However, PCA is
 - ⇒ sensitive to **statistical estimation errors**
 - ⇒ has **limited expressivity** (only performs high-pass filtering)



Covariance Filters

- ▶ Data sample $\mathbf{x} = [x_1, \dots, x_N]^T$ with covariance \mathbf{C}
 - ⇒ Build a **graph** where:
 - ⇒ the features are the node signals x_i
 - ⇒ the edges are the covariance values c_{ij} → fully-connected graph

$$\mathbf{x} = [x_1, x_2, x_3, x_4, x_5]$$

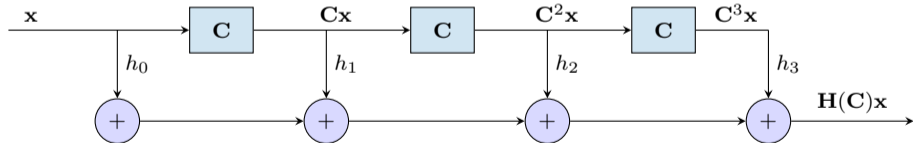
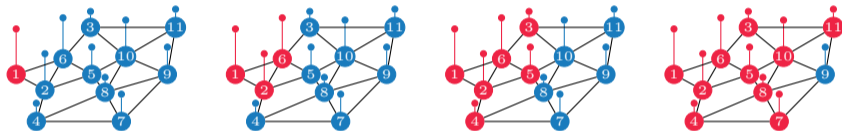


Covariance Filters

- **Definition:** Graph convolution covariance filters

$$\mathbf{z} = \mathbf{H}(\mathbf{C})\mathbf{x} = \sum_{k=0}^K h_k \mathbf{C}^k \mathbf{x}$$

⇒ learnable parameters: h_k

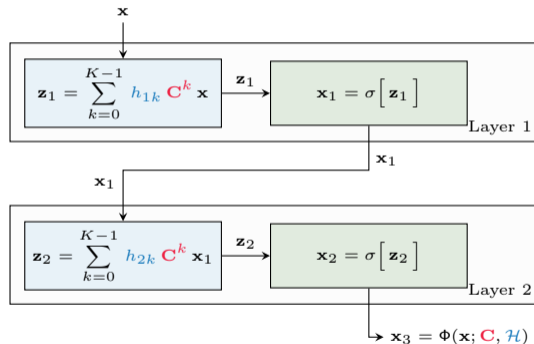


- $\mathbf{C}^k \mathbf{x}$ shifts signal \mathbf{x} k times over the covariance graph \mathbf{C}

Covariance Neural Networks (VNNs)

- **Definition:** Covariance filters followed by pointwise nonlinearities σ

$$\mathbf{x}^l = \sigma \left(\mathbf{H}^l(\mathbf{C}) \mathbf{x}^{l-1} \right) \quad l = 1, \dots, L.$$



Connections to PCA

- ▶ The covariance filter is

$$\mathbf{H}(\mathbf{C})\mathbf{x} = \sum_{k=0}^K h_k \mathbf{C}^k \mathbf{x}$$

- ▶ Taking the eigendecomposition $\mathbf{C} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top$ we get

$$\mathbf{H}(\mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top)\mathbf{x} = \sum_{k=0}^K h_k \mathbf{V}\mathbf{\Lambda}^k \mathbf{V}^\top \mathbf{x}$$

Connections to PCA

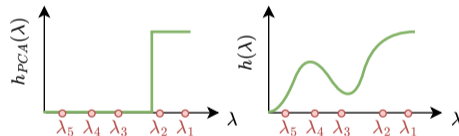
- ▶ The covariance filter is

$$\mathbf{H}(\mathbf{C})\mathbf{x} = \sum_{k=0}^K h_k \mathbf{C}^k \mathbf{x}$$

- ▶ Taking the eigendecomposition $\mathbf{C} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top$ we get

$$\mathbf{H}(\mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top)\mathbf{x} = \sum_{k=0}^K h_k \mathbf{V}\mathbf{\Lambda}^k \mathbf{V}^\top \mathbf{x}$$

⇒ The covariance filter processes the principal components $\mathbf{V}^\top \mathbf{x}$!



Connections to PCA

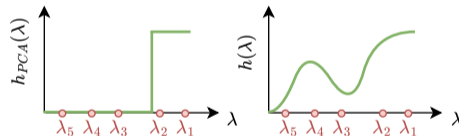
- ▶ The covariance filter is

$$\mathbf{H}(\mathbf{C})\mathbf{x} = \sum_{k=0}^K h_k \mathbf{C}^k \mathbf{x}$$

- ▶ Taking the eigendecomposition $\mathbf{C} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top$ we get

$$\mathbf{H}(\mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top)\mathbf{x} = \sum_{k=0}^K h_k \mathbf{V}\mathbf{\Lambda}^k \mathbf{V}^\top \mathbf{x}$$

⇒ The covariance filter processes the principal components $\mathbf{V}^\top \mathbf{x}$!



- ⇒ Improved **stability** to statistical errors
- ⇒ **Expressive** covariance processing
- ⇒ Require **labeled data** and **resources** for training

Covariance Scattering Transforms

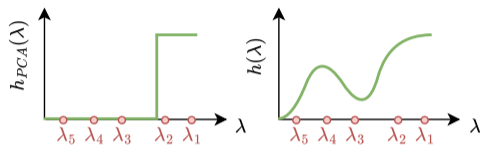
Desiderata:

- ▶ Process data using their **covariance** in an **expressive manner**
- ▶ Be **stable** to finite-sample estimation errors
- ▶ Remove the need for **labeled data** and **training**



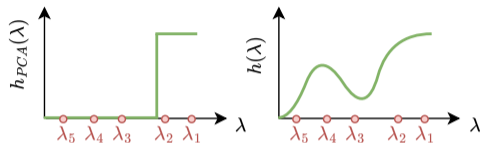
Covariance Wavelets

- Covariance filters learn expressive spectral functions, but **require training**

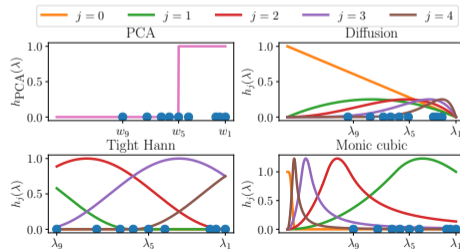


Covariance Wavelets

- Covariance filters learn expressive spectral functions, but **require training**



- **Key idea:** replace learnable functions with **pre-defined sets (frames)** of **expressive functions (wavelets)**



Covariance Wavelets

- ▶ Covariance wavelet operator matrix \mathbf{T} :

⇒ $\mathbf{C}_N = \gamma \mathbf{C} / w_1$ (normalized covariance)

⇒ $\mathbf{C}_I = \gamma (\mathbf{I} - \mathbf{C} / w_1)$ (swap eigenvalue order)

⇒ γ controls maximum eigenvalue of \mathbf{T}

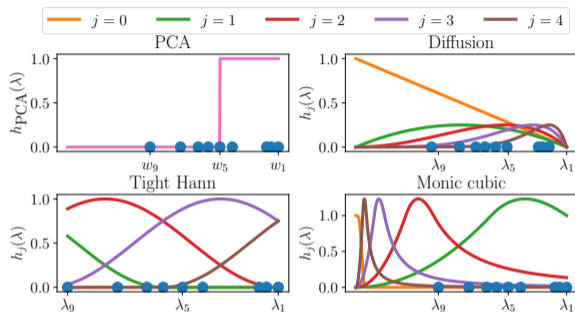
- ▶ Covariance wavelets are localized functions in the spectrum of \mathbf{T} instantiated at different scales j :

$$\mathbf{H}_j(\mathbf{T}) = \mathbf{V} \text{diag}(h_j(\lambda_1), \dots, h_j(\lambda_N)) \mathbf{V}^T,$$

⇒ e.g., diffusion wavelets:

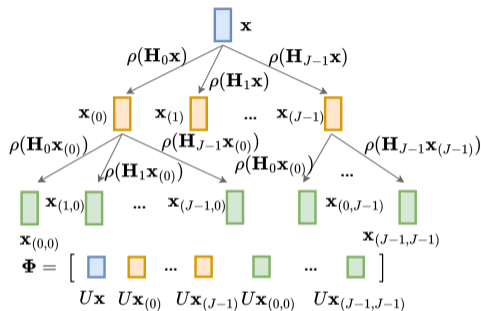
$$h_j(\lambda) = \lambda^{2^j - 1} - \lambda^{2^j}, \quad j \geq 1$$

⇒ Monic cubic and tight Hann are other possible definitions



Covariance Scattering Transforms

- Covariance scattering transforms (CSTs):
 - ⇒ deep hierarchical cascades of covariance wavelets
 - ⇒ expressivity via nonlinearities ρ
 - ⇒ dime. reduction via aggregation U (e.g., mean)
 - ⇒ final embedding Φ concatenates latent representations



Properties of Covariance Wavelets

► Frame

⇒ Covariance wavelets **conform a frame**: $A^2\|\mathbf{x}\|^2 \leq \sum_{j=1}^J \|\mathbf{H}_j\mathbf{x}\|^2 \leq B^2\|\mathbf{x}\|^2$ with $0 < A \leq B < \infty$.

⇒ A, B control the spread of energy when applying the frame



Properties of Covariance Wavelets

► Frame

⇒ Covariance wavelets **conform a frame**: $A^2 \|\mathbf{x}\|^2 \leq \sum_{j=1}^J \|\mathbf{H}_j \mathbf{x}\|^2 \leq B^2 \|\mathbf{x}\|^2$ with $0 < A \leq B < \infty$.

⇒ A, B control the spread of energy when applying the frame

► Lipschitz

⇒ Covariance wavelets $h_j(\lambda)$ **are Lipschitz**: $|h_j(\lambda_k) - h_j(\lambda_l)| / |\lambda_k - \lambda_l| \leq P$.

⇒ P measures the variability of the wavelet



Properties of Covariance Wavelets

► Frame

⇒ Covariance wavelets **conform a frame**: $A^2 \|\mathbf{x}\|^2 \leq \sum_{j=1}^J \|\mathbf{H}_j \mathbf{x}\|^2 \leq B^2 \|\mathbf{x}\|^2$ with $0 < A \leq B < \infty$.

⇒ A, B control the spread of energy when applying the frame

► Lipschitz

⇒ Covariance wavelets $h_j(\lambda)$ **are Lipschitz**: $|h_j(\lambda_k) - h_j(\lambda_l)| / |\lambda_k - \lambda_l| \leq P$.

⇒ P measures the variability of the wavelet

► Locality of covariance wavelets

⇒ **spectral localization** by definition

⇒ **spatial localization**: wavelets centered on one feature decay on features that are further away

⇒ **feature distance** depends on their covariance



Stability of Covariance Wavelets

- ▶ CSTs operate on the **sample operator** $\hat{\mathbf{T}}$ instead of the true one \mathbf{T} .
 - ⇒ we bound the effect on the **wavelet output**

Theorem

The output difference of the wavelet computed on the true and perturbed covariance wavelet operators \mathbf{T} and $\hat{\mathbf{T}}$, respectively, is bounded as

$$\|\mathbf{H}_j(\hat{\mathbf{T}}) - \mathbf{H}_j(\mathbf{T})\| \leq \mathcal{O}_P \left(\frac{PN\sqrt{\log N}\|\mathbf{C}\|\gamma}{\sqrt{T}w_1} \right) := \Delta,$$

where P is the Lipschitz constant of the wavelet function.

- ▶ Covariance wavelets are **stable** to covariance estimation errors.
 - ⇒ **More stable** for smaller P , but less expressive
 - ⇒ **Convergence to the ideal case** with rate $\mathcal{O}(T^{-1/2})$

Stability of Covariance Scattering Transforms

- ▶ CSTs operate on the **sample operator** $\hat{\mathbf{T}}$ instead of the true one \mathbf{T} .
 - ⇒ covariance wavelets are **stable** to these perturbations
 - ⇒ the stability **extends to CSTs**

Theorem

Consider a CST $\Phi(\cdot)$ with L layers and J scales. Let Δ and B be the largest stability and frame bounds among the wavelets in the CST. The stability to finite-sample perturbations is

$$\|\Phi(\mathbf{T}, \mathbf{x}) - \Phi(\hat{\mathbf{T}}, \mathbf{x})\| \leq \Delta \|\mathbf{x}\| \sqrt{\sum_{\ell=1}^{L-1} \ell^2 B^{2\ell-2} F_{\ell}}.$$

where $F_{\ell} \leq J^{\ell}$ is the number of selected scattering features at layer ℓ .

- ▶ CSTs stability is **proportional to wavelets' stability** Δ
 - ⇒ stability decreases for increasing layers and feature size (**stability vs expressivity tradeoff**)

Experimental Setup

▶ Datasets:

⇒ Cortical thickness measures from brain MRI, one vector per patient

⇒ 4 datasets: ADNI1, ADNI2, PPMI, REALDISP

▶ Task:

⇒ Patient's age prediction from cortical thickness measures

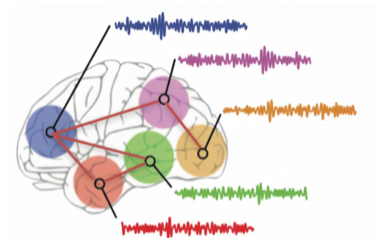
⇒ Relevant for early neurodegenerative disease detection

▶ Models:

⇒ 3 CSTs (Diffusion, Hann, Monic)+SVM regressor

⇒ PCA+SVM regressor (unstable, limited expressivity, untrained)

⇒ VNN (stable, expressive, needs training)

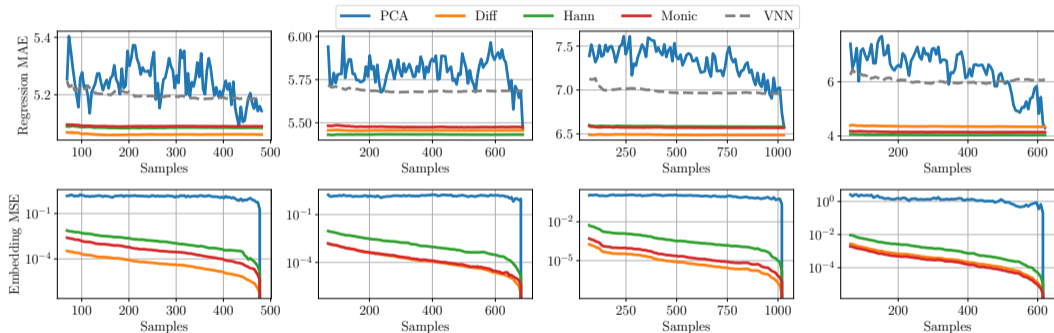


Stability

► Setup:

⇒ Train on full training dataset

⇒ During test, remove training samples and recompute covariances (finite-sample perturbations)



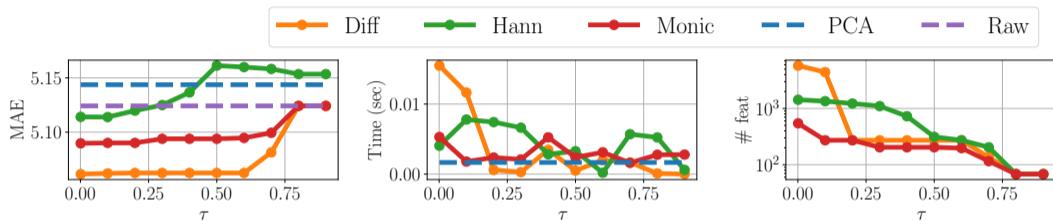
► Results:

⇒ CSTs are more stable than PCA

⇒ VNN is stable but performs worse with low-data

Pruning

- **Setup:** apply pruning for increasing threshold τ . Results for ADNI1.



- **Results:**

- ⇒ Performance **stays consistent** for several values of τ , then decreases
- ⇒ Even small τ yields **significant time and size improvements**

Conclusions

- ▶ Processing data via **covariance information**:
 - ⇒ **PCA** is sensitive to finite-samples and limited in expressivity
 - ⇒ **VNNs** require training
- ▶ **Covariance Scattering Transforms**
 - ⇒ Expressive spectral filtering via **covariance wavelets**
 - ⇒ Deep non-linear functions via **scattering transforms framework**
 - ⇒ **Stable** to finite-sample estimation errors
- ▶ **Thank you for the attention!**
 - ⇒ a.cavallo@tudelft.nl

