coVariance Neural Networks

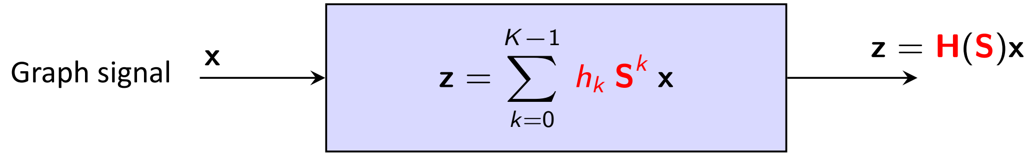
Saurabh Sihag[1], Gonzalo Mateos[2], Corey McMillan[1], Alejandro Ribeiro[1]

[1]University of Pennsylvania

[2]University of Rochester

Graph Filters and coVariance Filters

- Graph filter[a]



h_k: filter taps

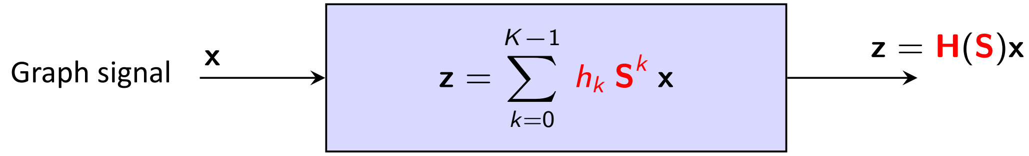
Graph filter of order K supported on undirected graph $\mathbf{S} = \mathbf{R}\mathbf{F}\mathbf{R}^\mathsf{T}$

- coVariance filter

[a] Ortega, Antonio, et al. "Graph signal processing: Overview, challenges, and applications." *Proceedings of the IEEE* 106.5 (2018): 808-828.

Graph Filters and coVariance Filters

- Graph filter[a]



h_k: filter taps

Graph filter of order K supported on undirected graph $\mathbf{S} = \mathbf{R}\mathbf{F}\mathbf{R}^{\mathsf{T}}$

Spectral representation of graph filter $\mathbf{H}(\mathbf{S})$

$$\mathbf{R}^{\mathsf{T}}\mathbf{H}(\mathbf{S})\, \mathbf{x} = \sum_{k=0}^{K-1} h_k \mathbf{F}^k\, \mathbf{R}^{\mathsf{T}}\mathbf{x}$$

$$= \underbrace{h(\mathbf{F})}_{\text{filter frequency response}}\, \mathbf{R}^{\mathsf{T}}\mathbf{x}$$

- coVariance filter

[a] Ortega, Antonio, et al. "Graph signal processing: Overview, challenges, and applications." *Proceedings of the IEEE* 106.5 (2018): 808-828.

Graph Filters and coVariance Filters

- **Graph filter**[a]

Graph signal $\mathbf{x}$ → $\mathbf{z} = \sum_{k=0}^{K-1} h_k \mathbf{S}^k \mathbf{x}$ → $\mathbf{z} = \mathbf{H}(\mathbf{S})\mathbf{x}$

h_k: filter taps

Graph filter of order K supported on undirected graph $\mathbf{S} = \mathbf{RFR}^\mathsf{T}$

Spectral representation of graph filter $\mathbf{H}(\mathbf{S})$

$$\mathbf{R}^\mathsf{T}\mathbf{H}(\mathbf{S})\,\mathbf{x} = \sum_{k=0}^{K-1} h_k \mathbf{F}^k\ \mathbf{R}^\mathsf{T}\mathbf{x}$$

$$= \underbrace{h(\mathbf{F})}_{\text{filter frequency response}}\ \mathbf{R}^\mathsf{T}\mathbf{x}$$

- **coVariance filter**

For an m-dimensional dataset of n samples, $\mathbf{x}_n \in \mathbb{R}^{m\times n}$,
sample covariance matrix $\mathbf{C}_n = \frac{1}{n}(\mathbf{x}_n - \bar{\mathbf{x}}_n)(\mathbf{x}_n - \bar{\mathbf{x}}_n)^\mathsf{T}$

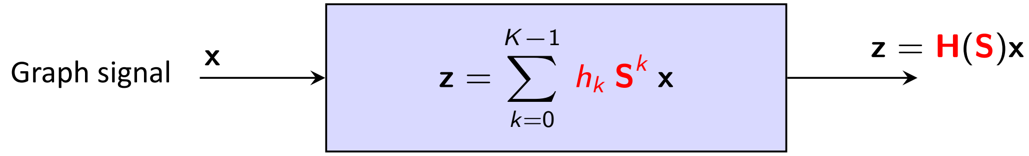
$\mathbf{x}$ → $\mathbf{z} = \sum_{k=0}^{K-1} h_k \mathbf{C}_n^k \mathbf{x}$ → $\mathbf{z} = \mathbf{H}(\mathbf{C}_n)\mathbf{x}$

coVariance filter of order K supported on sample covariance matrix $\mathbf{C}_n = \mathbf{UWU}^\mathsf{T}$

[a] Ortega, Antonio, et al. "Graph signal processing: Overview, challenges, and applications." *Proceedings of the IEEE* 106.5 (2018): 808-828.

Graph Filters and coVariance Filters

- Graph filter[a]



h_k: filter taps

Graph filter of order K supported on undirected graph $\mathbf{S} = \mathbf{R}\mathbf{F}\mathbf{R}^{\mathsf{T}}$

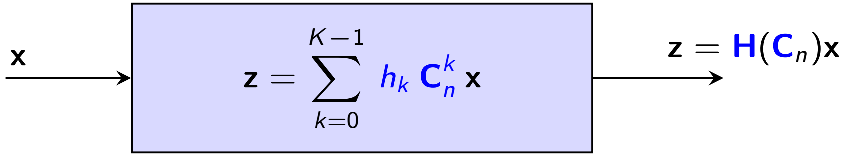
Spectral representation of graph filter $\mathbf{H}(\mathbf{S})$

$$\mathbf{R}^{\mathsf{T}}\mathbf{H}(\mathbf{S})\,\mathbf{x} = \sum_{k=0}^{K-1} h_k \mathbf{F}^k\, \mathbf{R}^{\mathsf{T}}\mathbf{x}$$

$$= \underbrace{h(\mathbf{F})}_{\text{filter frequency response}}\, \mathbf{R}^{\mathsf{T}}\mathbf{x}$$

- coVariance filter

For an m-dimensional dataset of n samples, $\mathbf{x}_n \in \mathbb{R}^{m \times n}$,
sample covariance matrix $\mathbf{C}_n = \frac{1}{n}(\mathbf{x}_n - \bar{\mathbf{x}}_n)(\mathbf{x}_n - \bar{\mathbf{x}}_n)^{\mathsf{T}}$



coVariance filter of order K supported on sample covariance matrix $\mathbf{C}_n = \mathbf{U}\mathbf{W}\mathbf{U}^{\mathsf{T}}$

Spectral representation of coVariance filter $\mathbf{H}(\mathbf{C}_n)$

$$\mathbf{U}^{\mathsf{T}}\mathbf{H}(\mathbf{C}_n)\,\mathbf{x} = \sum_{k=0}^{K-1} h_k \mathbf{W}^k\, \mathbf{U}^{\mathsf{T}}\mathbf{x}$$

$$= h(\mathbf{W})\, \mathbf{U}^{\mathsf{T}}\mathbf{x}$$

[a] Ortega, Antonio, et al. "Graph signal processing: Overview, challenges, and applications." *Proceedings of the IEEE* 106.5 (2018): 808-828.

Graph Filters and coVariance Filters

- **Graph filter**[a]

Graph signal $\mathbf{x}$ → $\mathbf{z} = \sum_{k=0}^{K-1} h_k \mathbf{S}^k \mathbf{x}$ → $\mathbf{z} = \mathbf{H}(\mathbf{S})\mathbf{x}$

h_k: filter taps

Graph filter of order K supported on undirected graph $\mathbf{S} = \mathbf{R}\mathbf{F}\mathbf{R}^\mathsf{T}$

Spectral representation of graph filter $\mathbf{H}(\mathbf{S})$

$$\mathbf{R}^\mathsf{T}\mathbf{H}(\mathbf{S})\,\mathbf{x} = \sum_{k=0}^{K-1} h_k \mathbf{F}^k\, \mathbf{R}^\mathsf{T}\mathbf{x}$$

$$= \underbrace{h(\mathbf{F})}_{\text{filter frequency response}}\, \mathbf{R}^\mathsf{T}\mathbf{x}$$

filter frequency response

- **coVariance filter**

For an m-dimensional dataset of n samples, $\mathbf{x}_n \in \mathbb{R}^{m\times n}$,
sample covariance matrix $\mathbf{C}_n = \frac{1}{n}(\mathbf{x}_n - \bar{\mathbf{x}}_n)(\mathbf{x}_n - \bar{\mathbf{x}}_n)^\mathsf{T}$

$\mathbf{x}$ → $\mathbf{z} = \sum_{k=0}^{K-1} h_k \mathbf{C}_n^k \mathbf{x}$ → $\mathbf{z} = \mathbf{H}(\mathbf{C}_n)\mathbf{x}$

coVariance filter of order K supported on
sample covariance matrix $\mathbf{C}_n = \mathbf{U}\mathbf{W}\mathbf{U}^\mathsf{T}$

Spectral representation of coVariance filter $\mathbf{H}(\mathbf{C}_n)$

$$\mathbf{U}^\mathsf{T}\mathbf{H}(\mathbf{C}_n)\,\mathbf{x} = \sum_{k=0}^{K-1} h_k \mathbf{W}^k\, \mathbf{U}^\mathsf{T}\mathbf{x}$$

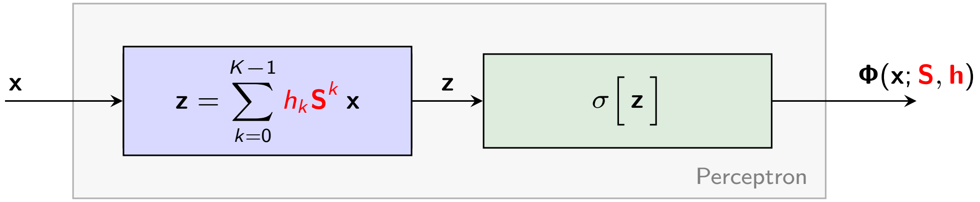
$$= h(\mathbf{W})\, \mathbf{U}^\mathsf{T}\mathbf{x}$$ → PCA!!

Advantages over PCA:
- **Stability** to perturbations
- **Transferability**

[a] Ortega, Antonio, et al. "Graph signal processing: Overview, challenges, and applications." *Proceedings of the IEEE* 106.5 (2018): 808-828.

Graph Neural Networks and coVariance Neural Networks

- Graph Neural Networks[b]

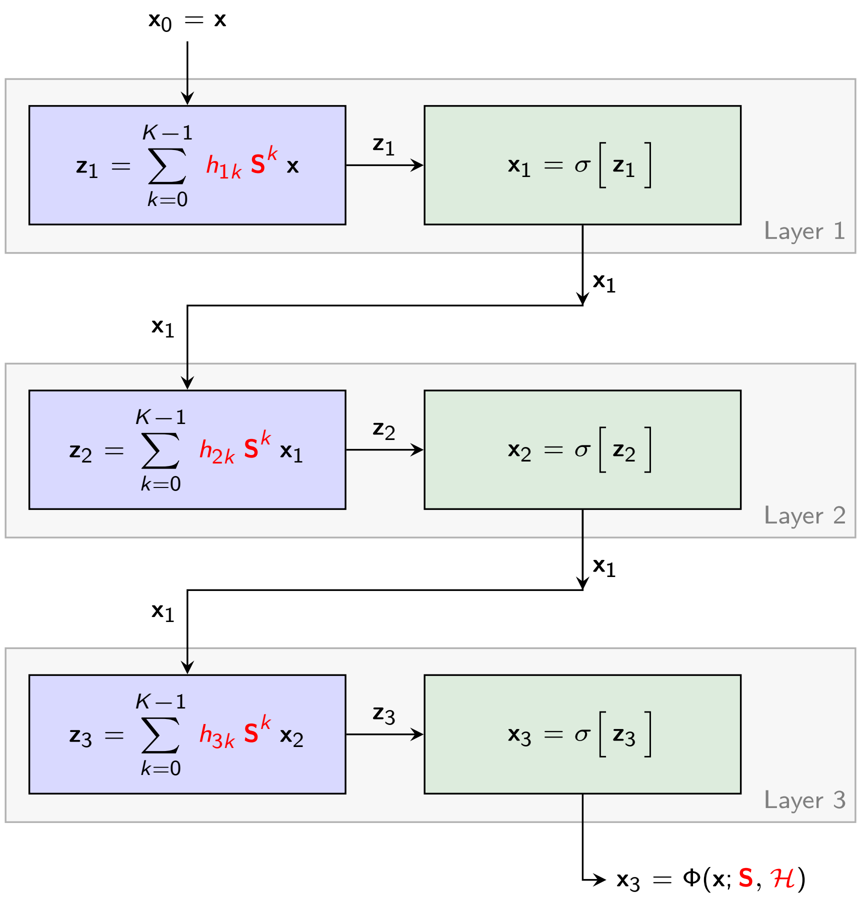


$\sigma(\cdot)$: pointwise non-linearity function (e.g. ReLU, $\tanh$)

[b] Ruiz, Luana, Fernando Gama, and Alejandro Ribeiro. "Graph neural networks: architectures, stability, and transferability." *Proceedings of the IEEE* 109.5 (2021): 660-682.

Graph Neural Networks and coVariance Neural Networks

- Graph Neural Networks[b]

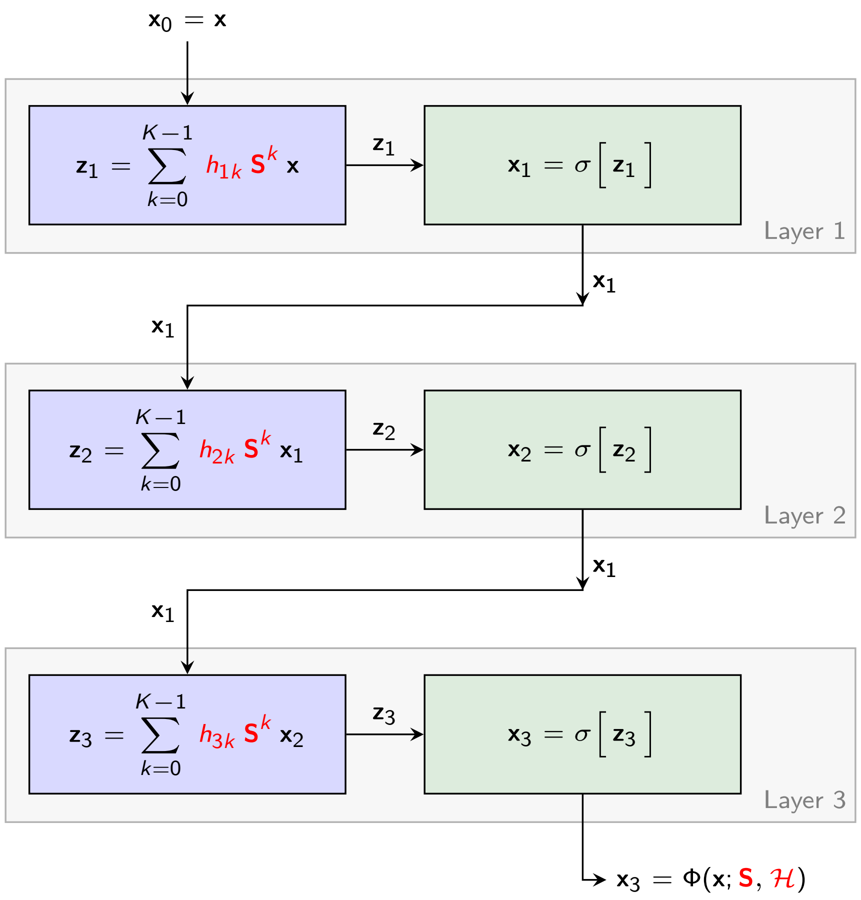


$\sigma(\cdot)$: pointwise non-linearity function (e.g. ReLU, $\tanh$)

[b] Ruiz, Luana, Fernando Gama, and Alejandro Ribeiro. "Graph neural networks: architectures, stability, and transferability." *Proceedings of the IEEE* 109.5 (2021): 660-682.

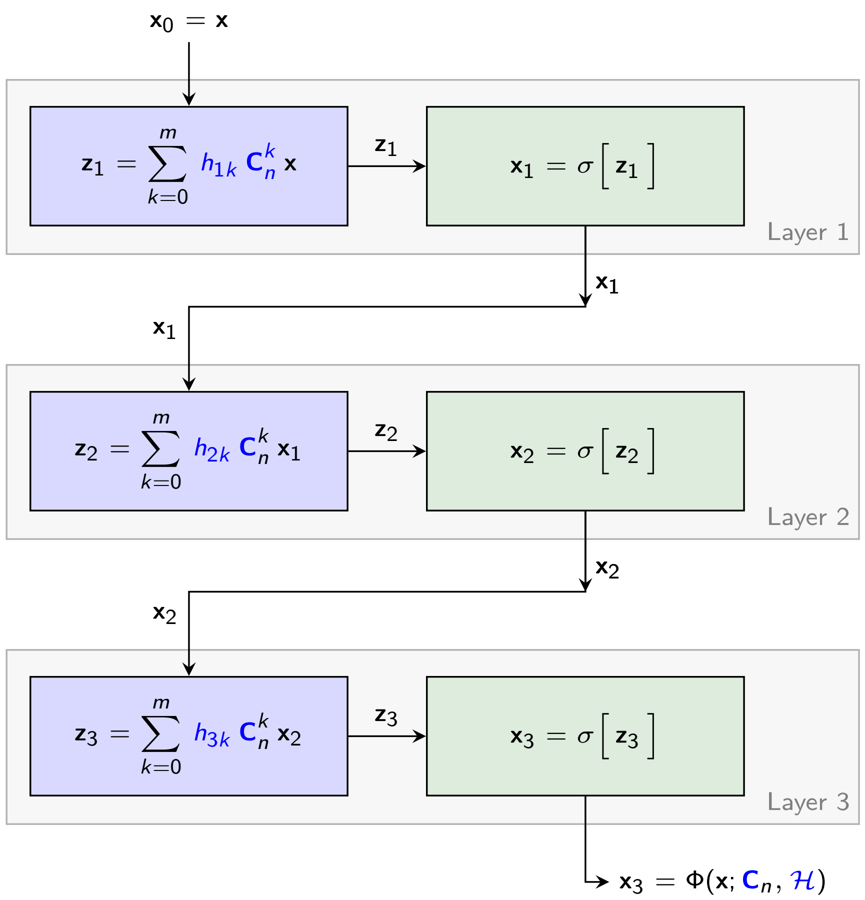
Graph Neural Networks and coVariance Neural Networks

- Graph Neural Networks[b]



$\sigma(\cdot)$: pointwise non-linearity function (e.g. ReLU, tanh)

- coVariance Neural Networks (VNN)



Stability and transferability extend to VNNs

[b] Ruiz, Luana, Fernando Gama, and Alejandro Ribeiro. "Graph neural networks: architectures, stability, and transferability." *Proceedings of the IEEE* 109.5 (2021): 660-682.

Perturbation Theory of Covariance Matrix

- Sample covariance matrix $\mathbf{C}_n$ is estimate of ensemble covariance matric $\mathbf{C}$

$$\mathbf{C}_n = \frac{1}{n}(\mathbf{x}_n - \bar{\mathbf{x}}_n)(\mathbf{x}_n - \bar{\mathbf{x}}_n)^\mathsf{T}$$

Eigenvalues: $w_1, \ldots, w_m$
Eigenvectors: $\mathbf{u}_1, \ldots, \mathbf{u}_m$

$$\mathbf{C} = \mathbb{E}[(\mathbf{x} - \mathbb{E}[\mathbf{x}])(\mathbf{x} - \mathbb{E}[\mathbf{x}])^\mathsf{T}]$$

Eigenvalues: $\lambda_1, \ldots, \lambda_m$
Eigenvectors: $\mathbf{p}_1, \ldots, \mathbf{p}_m$

Perturbation Theory of Covariance Matrix

- Sample covariance matrix $\mathbf{C}_n$ is estimate of ensemble covariance matric $\mathbf{C}$

$$\mathbf{C}_n = \frac{1}{n}(\mathbf{x}_n - \bar{\mathbf{x}}_n)(\mathbf{x}_n - \bar{\mathbf{x}}_n)^\mathsf{T} \qquad \mathbf{C} = \mathbb{E}[(\mathbf{x} - \mathbb{E}[\mathbf{x}])(\mathbf{x} - \mathbb{E}[\mathbf{x}])^\mathsf{T}]$$

Eigenvalues: $w_1, \ldots, w_m$
Eigenvectors: $\mathbf{u}_1, \ldots, \mathbf{u}_m$

Eigenvalues: $\lambda_1, \ldots, \lambda_m$
Eigenvectors: $\mathbf{p}_1, \ldots, \mathbf{p}_m$

- Perturbations in eigenvalues and eigenvectors scale with sample size, n[c]

1. $\mathbb{P}\left(\frac{|w_i - \lambda_i|}{\lambda_i} \leq t\right) \geq 1 - \frac{1}{n}\left(\frac{k_i}{\lambda_i t}\right)^2$

[c] Loukas, Andreas. "How close are the eigenvectors of the sample and actual covariance matrices?." *International Conference on Machine Learning*. PMLR, 2017.

Perturbation Theory of Covariance Matrix

- Sample covariance matrix $\mathbf{C}_n$ is estimate of ensemble covariance matric $\mathbf{C}$

$$\mathbf{C}_n = \frac{1}{n}(\mathbf{x}_n - \bar{\mathbf{x}}_n)(\mathbf{x}_n - \bar{\mathbf{x}}_n)^\mathsf{T} \qquad \mathbf{C} = \mathbb{E}[(\mathbf{x} - \mathbb{E}[\mathbf{x}])(\mathbf{x} - \mathbb{E}[\mathbf{x}])^\mathsf{T}]$$

Eigenvalues: $w_1, \ldots, w_m$
Eigenvectors: $\mathbf{u}_1, \ldots, \mathbf{u}_m$

Eigenvalues: $\lambda_1, \ldots, \lambda_m$
Eigenvectors: $\mathbf{p}_1, \ldots, \mathbf{p}_m$

- Perturbations in eigenvalues and eigenvectors scale with sample size, n[c]

1. $\mathbb{P}\left(\frac{|w_i - \lambda_i|}{\lambda_i} \leq t\right) \geq 1 - \frac{1}{n}\left(\frac{k_i}{\lambda_i t}\right)^2$
2. $\mathbb{P}\left(|\mathbf{u_i}^\mathsf{H}\mathbf{p_j}| \leq t\right) \geq 1 - \frac{1}{n}\left(\frac{2k_j}{t|\lambda_i - \lambda_j|}\right)^2$ ➞ Eigenvectors with close eigenvalues are more likely to be confused with each other for small changes (addition or removal of samples) in the dataset

[c] Loukas, Andreas. "How close are the eigenvectors of the sample and actual covariance matrices?." *International Conference on Machine Learning*. PMLR, 2017.

Stability of coVariance filters and VNN

Theorem 1 (Stability of coVariance filters)

Consider a random vector $\mathbf{X} \in \mathbb{R}^{m\times 1}$, such that, its corresponding covariance matrix is given by $\mathbf{C} = \mathbb{E}[(\mathbf{X} - \mathbb{E}[\mathbf{X}])(\mathbf{X} - \mathbb{E}[\mathbf{X}])^{\mathsf{T}}]$. For a sample covariance matrix $\mathbf{C}_n$ formed using n i.i.d instances of $\mathbf{X}$ and a random instance $\mathbf{x}$ of $\mathbf{X}$, such that, $\|\mathbf{x}\| \leq 1$ under appropriate assumptions, the following holds with probability at least $1 - n^{-2\varepsilon} - 2\kappa m/n$ for any $\varepsilon \in (0, 1/2]$:

$$\|\mathbf{H}(\mathbf{C}_n) - \mathbf{H}(\mathbf{C})\| = \frac{M}{n^{\frac{1}{2}-\varepsilon}} \cdot \mathcal{O}\left(\sqrt{m} + \frac{\|\mathbf{C}\|\sqrt{\log mn}}{k_{\mathsf{min}} n^{2\varepsilon}}\right) .$$

Stability of coVariance filters and VNN

Theorem 1 (Stability of coVariance filters)

Consider a random vector $\mathbf{X} \in \mathbb{R}^{m\times 1}$, such that, its corresponding covariance matrix is given by $\mathbf{C} = \mathbb{E}[(\mathbf{X} - \mathbb{E}[\mathbf{X}])(\mathbf{X} - \mathbb{E}[\mathbf{X}])^{\mathsf{T}}]$. For a sample covariance matrix $\mathbf{C}_n$ formed using n i.i.d instances of $\mathbf{X}$ and a random instance $\mathbf{x}$ of $\mathbf{X}$, such that, $\|\mathbf{x}\| \leq 1$ under appropriate assumptions, the following holds with probability at least $1 - n^{-2\varepsilon} - 2\kappa m/n$ for any $\varepsilon \in (0, 1/2]$:

$$\|\mathbf{H}(\mathbf{C}_n) - \mathbf{H}(\mathbf{C})\| = \frac{M}{n^{\frac{1}{2}-\varepsilon}} \cdot \mathcal{O}\left(\sqrt{m} + \frac{\|\mathbf{C}\|\sqrt{\log mn}}{k_{\mathsf{min}} n^{2\varepsilon}}\right) .$$

Theorem 2 (Stability of coVariance neural networks)

Consider a sample covariance matrix $\mathbf{C}_n$ and the ensemble covariance matrix $\mathbf{C}$. Given a bank of coVariance filters $\{\mathbf{H}_{fg}^{\ell}\}$, such that $|h_{fg}^{\ell}(\lambda)| \leq 1$ and a pointwise non-linearity $\sigma(\cdot)$, such that, $|\sigma(a) - \sigma(b)| \leq |a - b|$, if the covariance filters satisfy

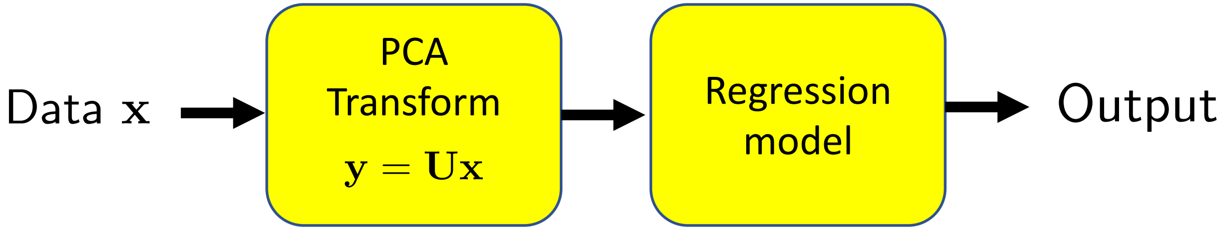
$$\|\mathbf{H}_{fg}^{\ell}(\mathbf{C}_n) - \mathbf{H}_{fg}^{\ell}(\mathbf{C})\| \leq \alpha_n ,$$

for some $\alpha_n > 0$, then, we have

$$\|\Phi(\mathbf{x}; \mathbf{C}_n, \mathcal{H}) - \Phi(\mathbf{x}; \mathbf{C}), \mathcal{H})\| \leq L F^{L-1} \alpha_n .$$

Stability of VNN: An Example

- Regression with PCA

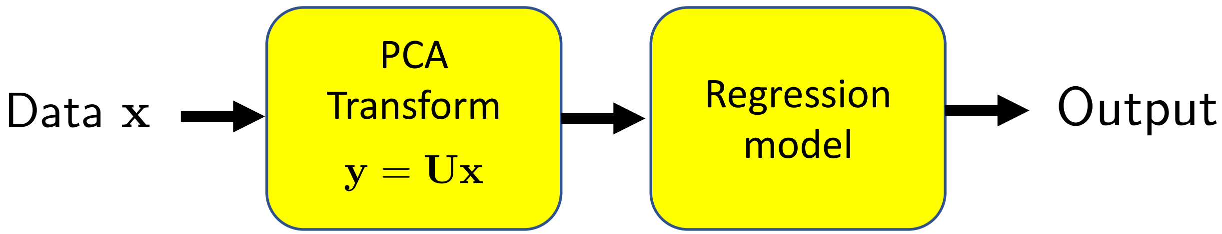


$\mathbf{U}$: Principal components from $\mathbf{C}_n$
($\mathbf{C}_n$: sample covariance matrix from n samples)

- Regression with VNN

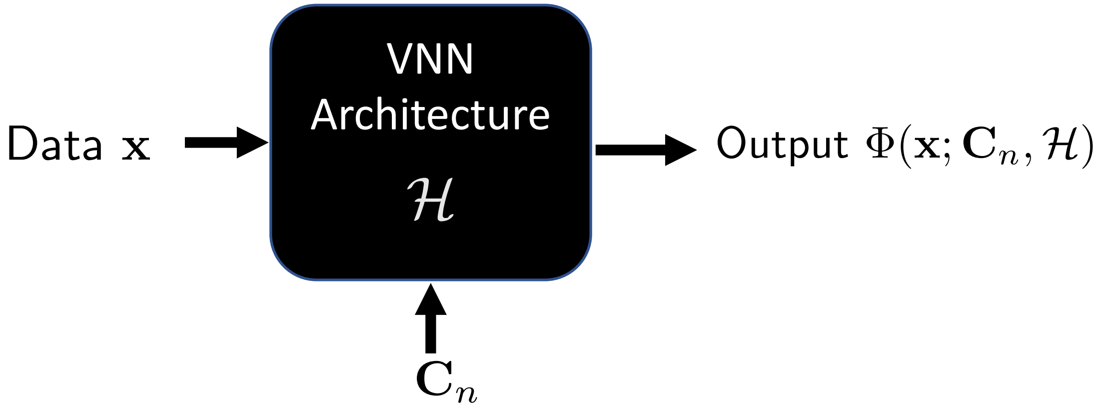
Stability of VNN: An Example

- Regression with PCA



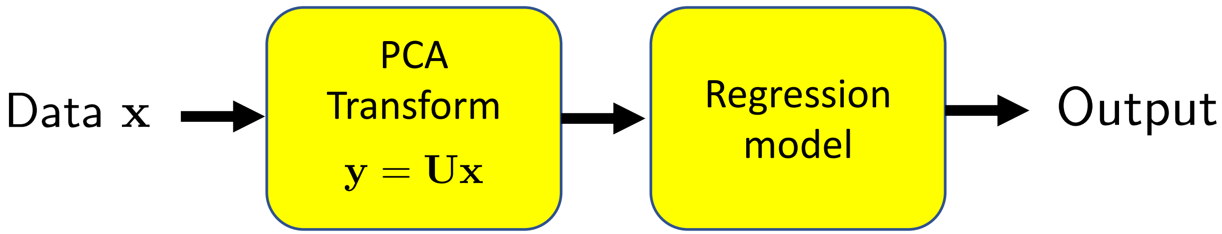
$\mathbf{U}$: Principal components from $\mathbf{C}_n$
($\mathbf{C}_n$: sample covariance matrix from n samples)

- Regression with VNN



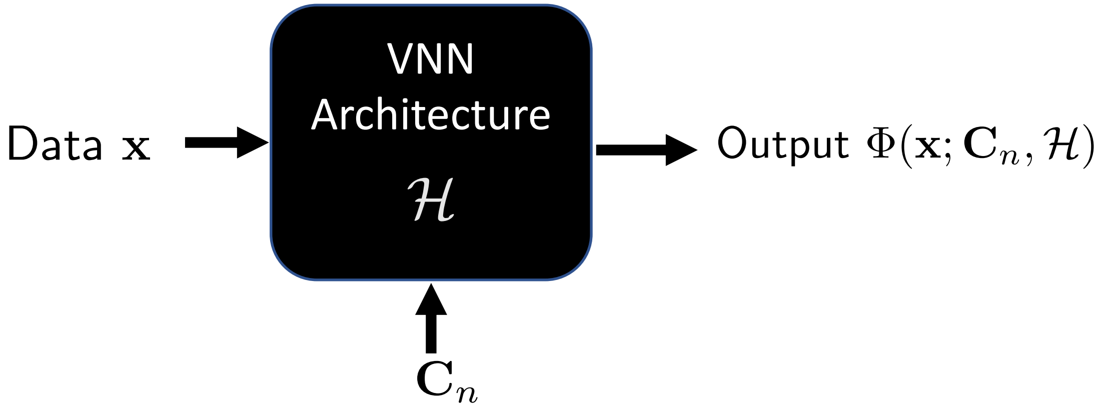
Stability of VNN: An Example

- Regression with PCA

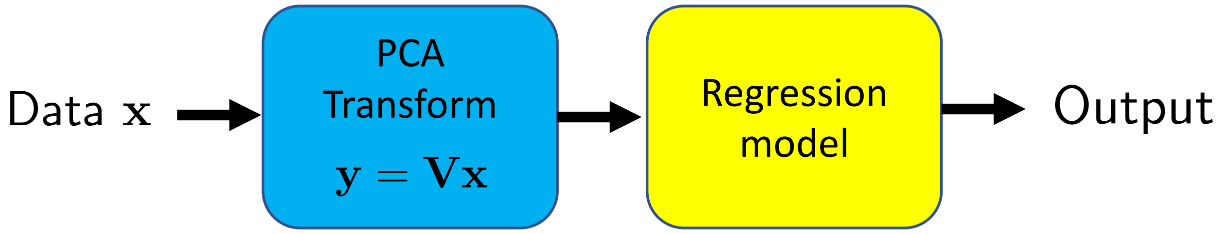


$\mathbf{U}$: Principal components from $\mathbf{C}_n$
($\mathbf{C}_n$: sample covariance matrix from n samples)

- Regression with VNN



Add k samples to dataset

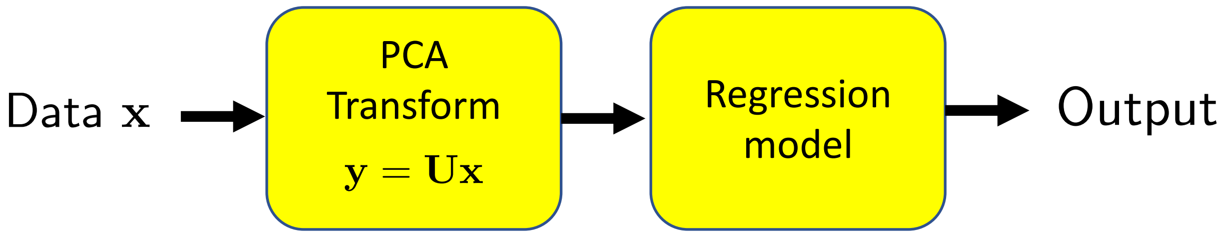


$\mathbf{V}$: Principal components from $\mathbf{C}_{n+k}$

Outputs are prone to instability: $\|\mathbf{C}_{n+k} - \mathbf{C}_n\| \ll \|\mathbf{V} - \mathbf{U}\|$

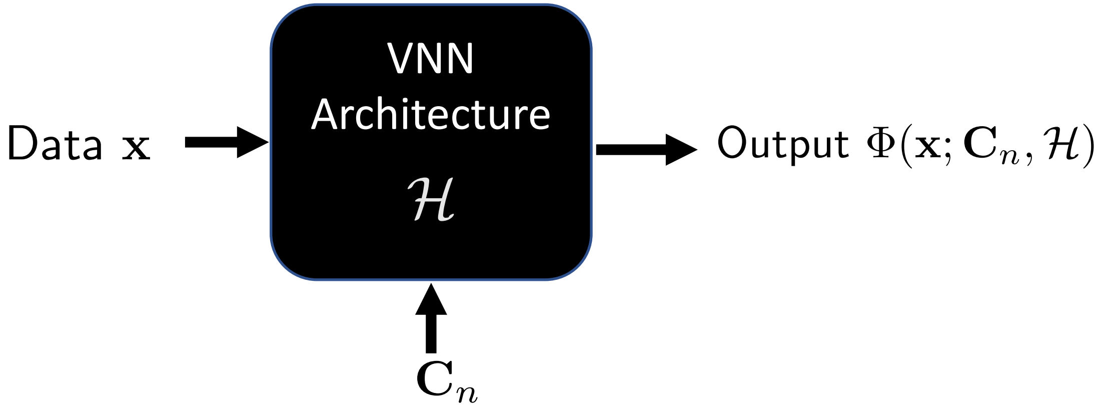
Stability of VNN: An Example

- Regression with PCA

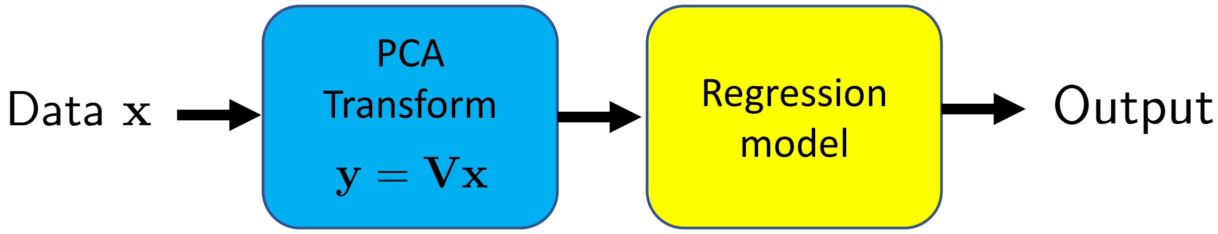


$\mathbf{U}$: Principal components from $\mathbf{C}_n$
($\mathbf{C}_n$: sample covariance matrix from n samples)

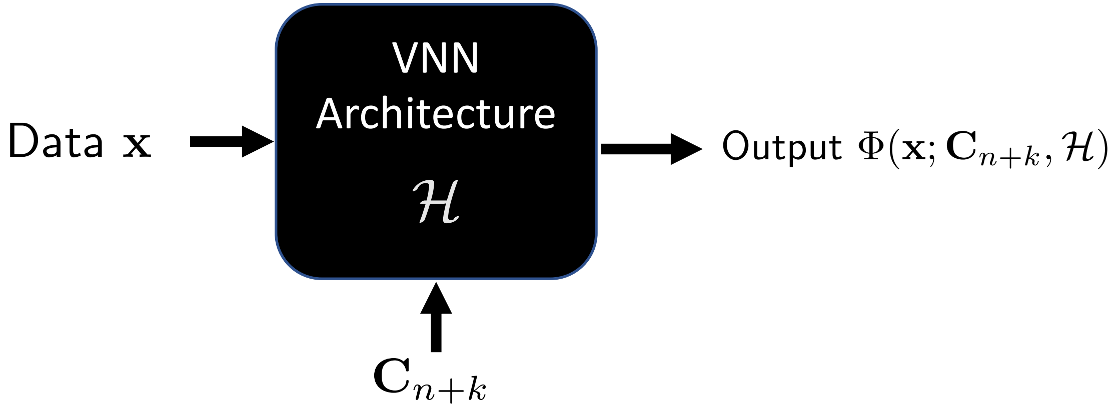
- Regression with VNN



Add k samples to dataset



$\mathbf{V}$: Principal components from $\mathbf{C}_{n+k}$



Outputs are prone to instability: $\|\mathbf{C}_{n+k} - \mathbf{C}_n\| \ll \|\mathbf{V} - \mathbf{U}\|$

Provably stable: $\|\Phi(\mathbf{x}; \mathbf{C}_n; \mathcal{H}) - \Phi(\mathbf{x}; \mathbf{C}_{n+k}; \mathcal{H})\| = \mathcal{O}\left(\frac{1}{\sqrt{n}} + \frac{1}{\sqrt{n+k}}\right)$

Stability of VNN: Experiments

- Comparison against PCA-regression cortical thickness dataset ($m = 104$) from (n = 341) human subjects

  Objective: Regression of cortical thickness against chronological age (307/34 training/test split)

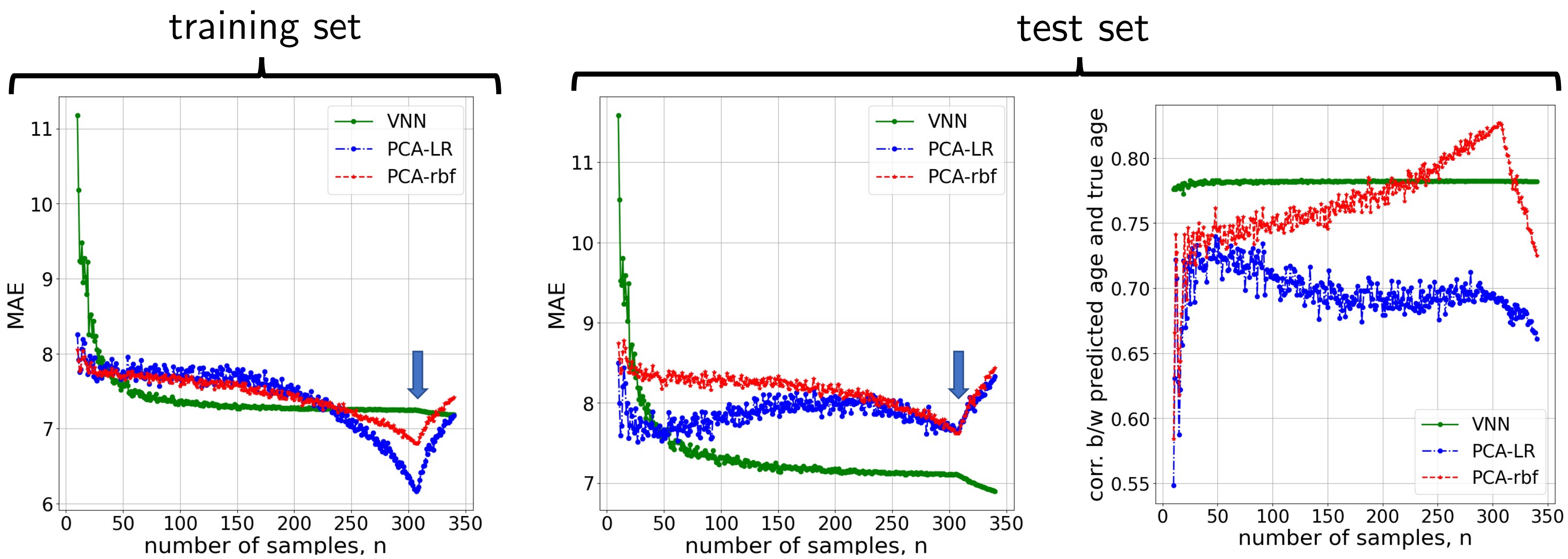
  Metrics: MAE (mean absolute error), correlation between predicted age and true age

Stability of VNN: Experiments

- Comparison against PCA-regression cortical thickness dataset ($m = 104$) from ($n = 341$) human subjects

Objective: Regression of cortical thickness against chronological age (307/34 training/test split)

Metrics: MAE (mean absolute error), correlation between predicted age and true age



MAE | MAE | correlation

VNN: coVariance Neural Network PCA-LR: PCA-regression with linear kernel PCA-rbf: PCA regression with radial basis function kernel

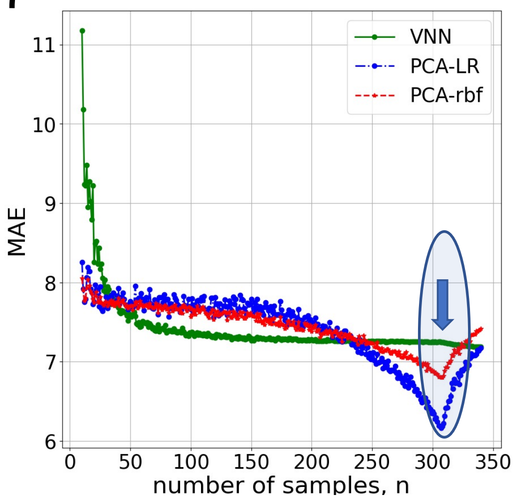
Stability of VNN: Experiments

- Comparison against PCA-regression cortical thickness dataset ($m = 104$) from ($n =$ 341) human subjects

Objective: Regression of cortical thickness against chronological age (307/34 training/test split)

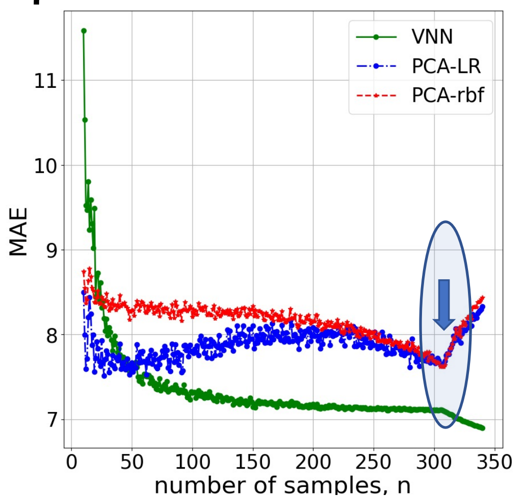
Metrics: MAE (mean absolute error), correlation between predicted age and true age

training set

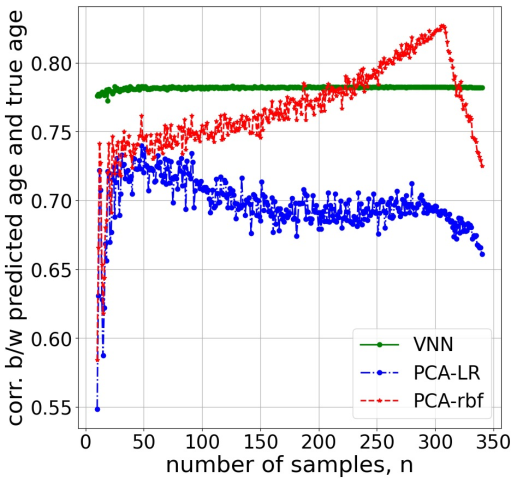


MAE

test set



MAE

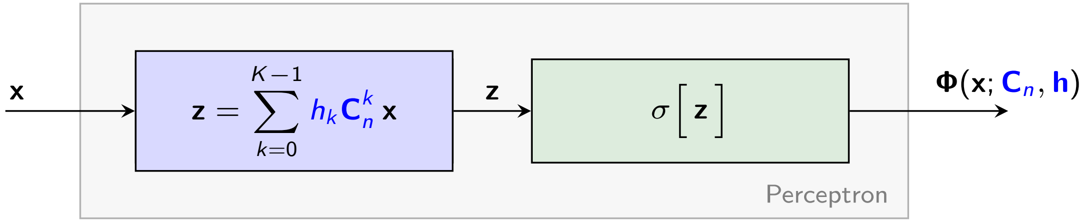


correlation

VNN: coVariance Neural Network PCA-LR: PCA-regression with linear kernel PCA-rbf: PCA regression with radial basis function kernel

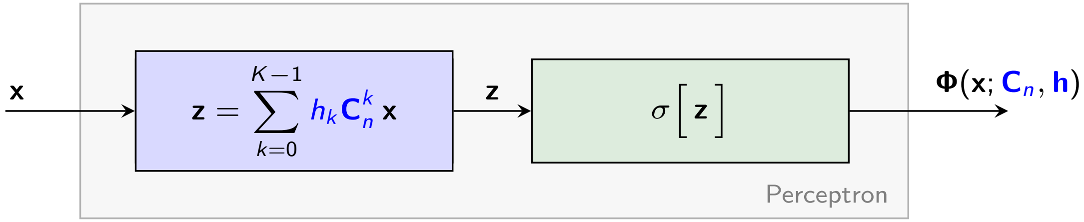
Transferability of VNN

- Parameters to be learnt (filter taps) are independent of covariance matrix dimension

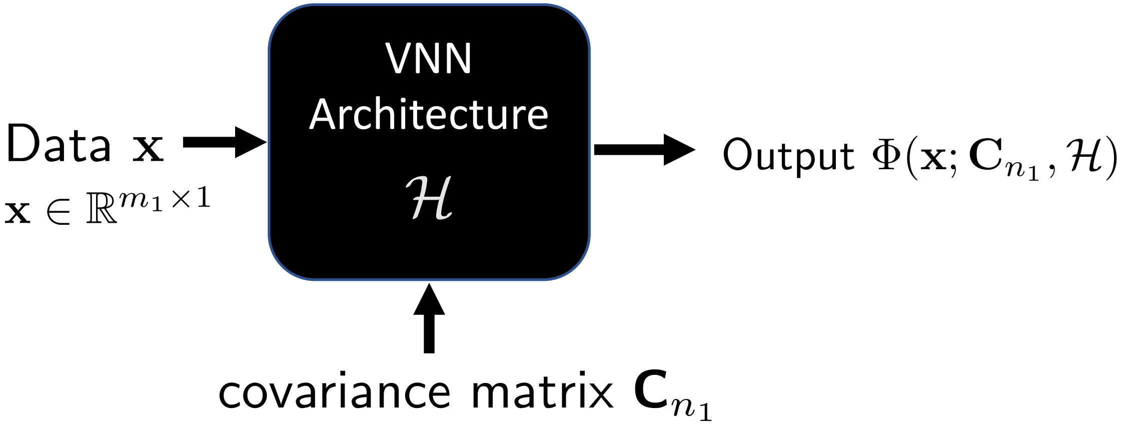


Transferability of VNN

- Parameters to be learnt (filter taps) are independent of covariance matrix dimension

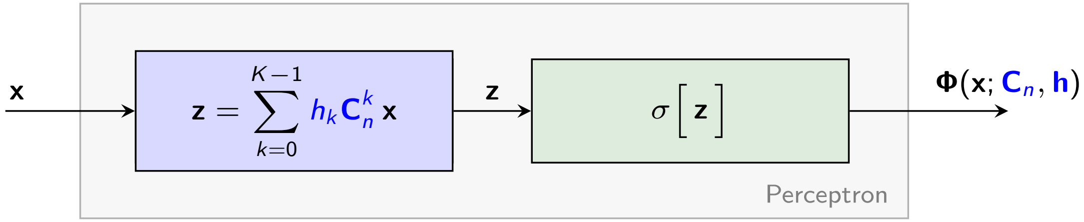


- Transferability of learnt parameters to datasets/ covariance matrix of different dimension

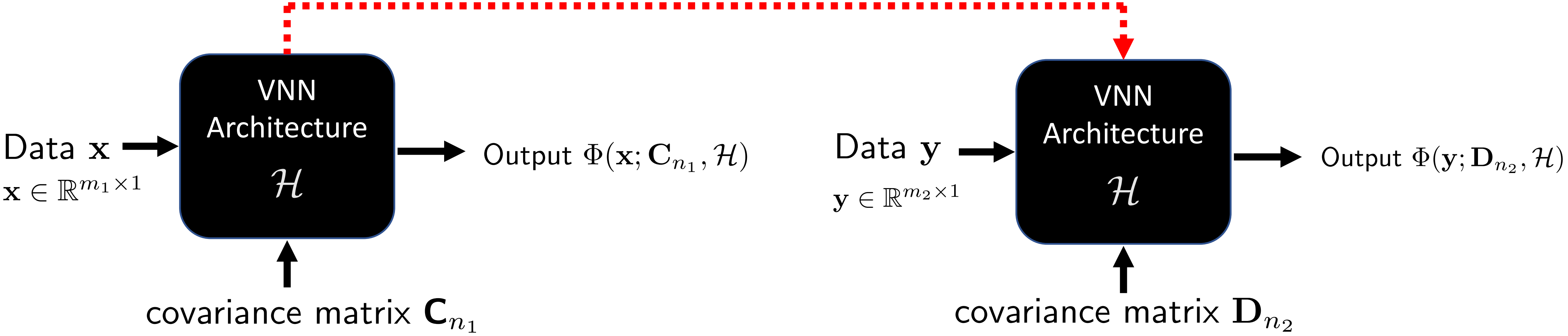


Transferability of VNN

- Parameters to be learnt (filter taps) are independent of covariance matrix dimension



- Transferability of learnt parameters to datasets/ covariance matrix of different dimension



Transferability of VNN: Experiments

- Multi-resolution cortical thickness datasets for 170 human subjects
  - FTDC100 (dimension = 100)
  - FTDC300 (dimension = 300)
  - FTDC500 (dimension = 500)

  Objective: Regression of cortical thickness against chronological age

Transferability of VNN: Experiments

- Multi-resolution cortical thickness datasets for 170 human subjects
  - FTDC100 (dimension = 100)
  - FTDC300 (dimension = 300)
  - FTDC500 (dimension = 500)

  Objective: Regression of cortical thickness against chronological age

MAE

Train \ Test	FTDC100	FTDC300	FTDC500
FTDC100	-	5.38 ± 0.044	5.47 ± 0.047
FTDC300	5.33 ± 0.28	-	5.57 ± 0.32
FTDC500	5.35 ± 0.05	5.38 ± 0.04	-

MAE: mean absolute error

Conclusions

- Study of VNNs as GNN operating on covariance matrices as graphs
- VNNs are stable to perturbations in datasets, implying reproducibility
- Transferability of VNNs shown empirically