

**Document Version**

Final published version

**Licence**

Dutch Copyright Act (Article 25fa)

**Citation (APA)**

Cavallo, A., Raghuvanshi, A., Chepuri, S. P., & Isufi, E. (2025). Robust Covariance Neural Networks. In M. B. Matthews (Ed.), *Proceedings of the 2025 59th Asilomar Conference on Signals, Systems, and Computers* (pp. 1677-1681). (Conference Record - Asilomar Conference on Signals, Systems and Computers). IEEE.  
<https://doi.org/10.1109/IEEECONF67917.2025.11443734>

**Important note**

To cite this publication, please use the final published version (if applicable).  
Please check the document version above.

**Copyright**

In case the licence states "Dutch Copyright Act (Article 25fa)", this publication was made available Green Open Access via the TU Delft Institutional Repository pursuant to Dutch Copyright Act (Article 25fa, the Taverne amendment). This provision does not affect copyright ownership.  
Unless copyright is transferred by contract or statute, it remains with the copyright holder.

**Sharing and reuse**

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

**Takedown policy**

Please contact us and provide details if you believe this document breaches copyrights.  
We will remove access to the work immediately and investigate your claim.

# Robust Covariance Neural Networks

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

\*Delft University of Technology, Delft, Netherlands

†Indian Institute of Science, Bangalore, India

**Abstract**—Learning deep representations from covariance information via coVariance Neural Networks (VNNs) has shown an improved performance and insights with respect to Principal Component Analysis (PCA)-based alternatives and better stability in finite-sample regimes. VNNs extend the PCA transform by learning end-to-end the spectral processing function on the principal directions of the data in each layer. However, VNNs operate on the pre-computed sample covariance matrix, which is prone to estimation errors, sensitive to outliers, and not adapted to the task at hand. To overcome this limitation, we propose Robust coVariance Neural Networks (RVNNs), a framework that simultaneously learns a robust estimator of the covariance matrix and the VNN parameters in an end-to-end manner, leading to a fully task-aware pipeline. We prove that RVNNs combine the robustness to outliers with the finite-sample stability of VNNs, and we show that their end-to-end robust covariance learning leads to better prediction performance compared to robust PCA-based approaches on simulated and real-world data from brain recordings and human motion sensor measurements.

**Index Terms**—Covariance Neural Networks, Robust Principal Component Analysis.

## I. INTRODUCTION

Processing data via covariance information is widely adopted and effective in applications where the data observations are related and interconnected by unknown interactions, such as fMRI brain data processing [1], [2], financial data modeling [3], and human action recognition [4], [5], among others. Prominent data analysis techniques relying on covariance matrix are Principal Component Analysis (PCA) [6], where the data are projected onto the eigenspace of the sample covariance matrix to maximize their variance, and stationary graph signal processing [7], [8], where the covariance graph is used to encode signal stationarity. However, PCA and other covariance-based techniques are subject to instabilities related to the finite-sample estimation of the covariance (and, consequently, principal directions), which may differ significantly from the true one, especially in sparse low-data settings [6], [9]. In this context, coVariance Neural Networks (VNNs) [10] allow to process the principal directions of the data while achieving a better stability to covariance estimation errors. Essentially, VNNs are graph convolutional neural networks operating on the sample covariance matrix as a graph shift operator, an operation that extends PCA via processing the principal component of the data in a deep, hierarchical, and stable way even in low-data regimes [10] and achieving

transferability across datasets of different resolutions [11]. This makes VNNs successful also in temporal, sparse, and biased data settings [12]–[14].

However, data observations often contain missing values or outliers – e.g., a sensor might fail to send measurements for a time interval, data transmission can introduce noise, etc. – and this makes the covariance estimation problematic and data processing unreliable. VNNs, which precompute the sample covariance matrix or its sparse/fair variants and later use it for convolutional processing in a two-stage fashion, are prone to large inconsistencies in the case of perturbed samples, which ultimately hinder their performance in noisy data settings.

Robust covariance estimation and PCA in the presence of perturbed data are widely explored, with several approaches relying on M-estimators [15], [16], projection pursuit [17], [18], and low-rank/sparse estimates [19] to reduce sensitivity to outliers. More recently, Robust PCA (RPCA) has been integrated with deep learning via deep unrolling [20], [21] and autoencoder-based architectures [22], which exploit the expressivity and flexibility of neural networks to improve robustness to noise. However, these approaches estimate cleaned principal directions in an unsupervised or self-supervised manner, which is suboptimal when these directions are then used to perform a (semi-)supervised downstream task.

**Contributions.** To counteract the presence of outliers and missing values in the data, we propose Robust coVariance Neural Networks (RVNNs), a framework that incorporates a robust covariance estimator inspired by robust PCA [19] into the VNN and learns the covariance and neural network's parameters jointly in an end-to-end fashion, thus estimating a task-aware robust covariance to process data. We summarize our contributions as follows.

(C1) We put forth RVNN, which jointly estimates a covariance matrix that is robust to outliers and missing values and the neural network parameters in an end-to-end fashion that allows for task-aware learning.

(C2) We theoretically show that RVNNs are stable to both covariance estimation errors due to finite-sample estimation and to outliers and missing values.

(C3) We experimentally show that the stability of RVNNs to finite-sample estimations results in consistent performance under covariance perturbations, improving over RPCA-based approaches, and we show that the end-to-end robust covariance learning scheme achieves a better performance than nominal VNNs and RPCA on synthetic and real datasets from brain and human motion recognition under a varying range of missing values and outliers.

Part of this work was supported by the TU Delft AI Labs programme, the NWO OTP GraSPA proposal #19497, and the NWO VENI proposal 222.032. Emails: a.cavallo@tudelft.nl, ayushmanr@iisc.ac.in, spchehuri@iisc.ac.in, e.isufi-1@tudelft.nl.

## II. BACKGROUND

Consider a data matrix  $\mathbf{X} \in \mathbb{R}^{N \times T}$ , where each column  $\mathbf{x}_i \in \mathbb{R}^N$  is an observation of a random variable  $\mathbf{x}$  with finite covariance  $\mathbf{C} = \mathbb{E}[(\mathbf{x} - \mathbb{E}[\mathbf{x}])(\mathbf{x} - \mathbb{E}[\mathbf{x}])^\top]$ . In real-world data, observations are often corrupted, noisy, or partially unavailable (i.e., there are missing values). In this work, we assume that the data matrix  $\mathbf{X}$  contains sparse and non-structured corruptions, such that it can be modeled as

$$\mathbf{X} = \mathbf{L} + \mathbf{S} \quad (1)$$

where  $\mathbf{L} \in \mathbb{R}^{N \times T}$  is a low-rank matrix containing the clean data observation and  $\mathbf{S} \in \mathbb{R}^{N \times T}$  is sparse and contains the corruptions. This framework is the basis of Robust PCA (RPCA) [19] and applies to a variety of real-world settings such as video surveillance, face recognition, and collaborative filtering. We assume that the covariance matrix estimated from the corrupted observed data matrix (i.e.,  $\hat{\mathbf{C}} = \mathbf{X}\mathbf{X}^\top/T$ , assuming w.l.o.g. that  $\mathbf{X}$  is zero mean) is a perturbed version of the covariance matrix of the clean data, denoted as  $\tilde{\mathbf{C}}$ .  $\tilde{\mathbf{C}}$ , in turn, is a finite-sample approximation of the true covariance  $\mathbf{C}$ , such that  $\tilde{\mathbf{C}} \approx \mathbf{C}$  for  $T \rightarrow \infty$ . We model the perturbation due to the outliers via an error matrix  $\mathbf{E}$  such that  $\tilde{\mathbf{C}} = \hat{\mathbf{C}} + \mathbf{E}$ .

To process the data and their covariance, Covariance Neural Networks (VNNs) [10] are a powerful tool. A VNN is a graph neural network that operates on the sample covariance matrix of the data  $\hat{\mathbf{C}}$  as a shift operator. In detail, a covariance filter  $\mathbf{H}(\hat{\mathbf{C}})$  produces a shifted signal  $\mathbf{u}$  from an input signal  $\mathbf{x}$  as

$$\mathbf{u} = \mathbf{H}(\hat{\mathbf{C}})\mathbf{x} = \sum_{k=0}^K h_k \hat{\mathbf{C}}^k \mathbf{x}, \quad (2)$$

where  $K$  is the order of the filter and  $h_k$  are learnable parameters. By observing the behavior of a covariance filter in the spectral domain, i.e., by projecting  $\mathbf{u}$  on the eigenvectors of the covariance  $\hat{\mathbf{V}}$  (where  $\hat{\mathbf{C}} = \hat{\mathbf{V}}\hat{\Lambda}\hat{\mathbf{V}}^\top$  is the eigendecomposition of the sample covariance), we get  $\tilde{\mathbf{u}} = \hat{\mathbf{V}}^\top \mathbf{u} = \hat{\mathbf{V}}^\top \sum_{k=0}^K h_k [\hat{\mathbf{V}}\hat{\Lambda}\hat{\mathbf{V}}^\top]^k \mathbf{x} = \sum_{k=0}^K h_k \hat{\Lambda}^k \hat{\mathbf{V}}^\top \mathbf{x}$ , where  $\tilde{\mathbf{x}} = \hat{\mathbf{V}}^\top \mathbf{x}$  is the same operation as PCA [6]. Focusing on the  $i$ -th entry, we have  $\tilde{u}_i = \sum_{k=0}^K h_k \hat{\lambda}_i^k \tilde{x}_i = h(\hat{\lambda}_i) \tilde{x}_i$ , i.e., the covariance filter in the spectral domain is characterized by a polynomial  $h(\lambda) = \sum_{k=0}^K h_k \lambda^k$  with frequency variable  $\lambda$  specified on the covariance eigenvalues  $\hat{\lambda}_i$ . In this perspective, a covariance filter extends PCA as it processes the eigenvectors of the covariance matrix via the polynomial  $h(\lambda)$  and it can retrieve exactly the PCA transform for a proper choice of parameters  $h_k$  [10, Theorem 1].

The  $l$ -th VNN layer is composed of a filterbank of  $F_{l-1} \times F_l$  filters (where  $F_{l-1}, F_l$  are the feature sizes of the  $l-1$ -th and  $l$ -th layer, respectively) followed by a nonlinearity  $\sigma(\cdot)$ . For the  $f$ -th feature of the  $l$ -th layer, we have

$$\mathbf{u}_f^l = \sigma \left( \sum_{j=1}^{F_{l-1}} h_k^l \hat{\mathbf{C}}^k \mathbf{u}_j^{l-1} \right), \quad (3)$$

where  $\mathbf{u}^0 = \mathbf{x}$  is the input signal. Finally, a VNN  $\Phi(\mathcal{H}, \hat{\mathbf{C}})$ , where  $\mathcal{H}$  collects all parameters  $h_k^l$ , is a sequence of  $L$

VNN layers. The parameters  $\mathcal{H}$  are generally optimized to minimize a task-specific loss  $\mathcal{L}$  (e.g., negative log-likelihood for classification tasks, mean squared error for regression tasks) over a training set  $(\mathbf{X}_{\text{tr}}, \mathbf{y}_{\text{tr}})$ , i.e.,

$$\min_{\mathcal{H}} \mathcal{L}(\Phi(\mathcal{H}, \hat{\mathbf{C}}), \mathbf{X}_{\text{tr}}, \mathbf{y}_{\text{tr}}). \quad (4)$$

VNNs are generally applied on the sample covariance matrix or its sparse/fair variants, which are unreliable if the observed data contains outliers or missing values, ultimately leading to worse performance [19]. In this work, we overcome this limitation via a robust estimate of the covariance matrix that is task-aware and learned end-to-end.

## III. ROBUST COVARIANCE NEURAL NETWORKS

To exploit correlation information in data processing while gaining robustness to outliers and data corruption, we put forth Robust VNNs (RVNNs).

**Robust covariance matrix structure.** Following the model in (1), we derive that the estimated covariance is

$$\begin{aligned} \hat{\mathbf{C}} &= \mathbf{X}\mathbf{X}^\top/T = (\mathbf{L} + \mathbf{S})(\mathbf{L} + \mathbf{S})^\top/T \\ &= (\mathbf{L}\mathbf{L}^\top + \mathbf{S}\mathbf{L}^\top + \mathbf{L}\mathbf{S}^\top + \mathbf{S}\mathbf{S}^\top)/T. \end{aligned} \quad (5)$$

The term  $\mathbf{L}\mathbf{L}^\top/T$  is the covariance matrix of the clean data  $\tilde{\mathbf{C}}$ , thus we get  $\mathbf{E} = \hat{\mathbf{C}} - \tilde{\mathbf{C}} = (\mathbf{S}\mathbf{L}^\top + \mathbf{L}\mathbf{S}^\top + \mathbf{S}\mathbf{S}^\top)/T$ . The term  $\mathbf{S}\mathbf{S}^\top$  is the product of two sparse matrices; therefore, it is also sparse. The terms  $\mathbf{S}\mathbf{L}^\top$  and  $\mathbf{L}\mathbf{S}^\top$ , instead, have at most rank equal to the rank of  $\mathbf{L}$ , and their sum has rank  $2 \cdot \text{rank}(\mathbf{L})$ . If  $\text{rank}(\mathbf{L}) \ll N$ , then  $\mathbf{S}\mathbf{L}^\top + \mathbf{L}\mathbf{S}^\top$  is low-rank.

**RVNN loss.** To retrieve the covariance of the clean data  $\tilde{\mathbf{C}}$  from the observed  $\hat{\mathbf{C}}$ , we aim to estimate two correction matrices, one low-rank ( $\mathbf{E}_l$ ) and one sparse ( $\mathbf{E}_s$ ), such that  $\mathbf{E} \approx \mathbf{E}_s + \mathbf{E}_l$ . Thus, we train the RVNN with the objective:

$$\min_{\mathcal{H}, \mathbf{E}_s, \mathbf{E}_l} \mathcal{L}(\Phi(\mathcal{H}, \hat{\mathbf{C}} + \mathbf{E}_s + \mathbf{E}_l), \mathbf{X}_{\text{tr}}, \mathbf{y}_{\text{tr}}) + \gamma_s \|\mathbf{E}_s\|_1 + \gamma_l \|\mathbf{E}_l\|_* \quad (6)$$

with  $\mathcal{L}(\cdot)$  a task-specific loss,  $\|\mathbf{E}\|_1 = \sum_{i,j=1}^N |\mathbf{E}_{ij}|$  as in [19],  $\|\cdot\|_*$  the nuclear norm and hyperparameters  $\gamma_s, \gamma_l$ .

**Stability.** Nominal VNNs show relevant advantages in terms of stability to covariance estimation errors due to finite-sample size. In our setting, we consider in addition the presence of outliers or missing values in the observed data, and we characterize the stability of RVNN to both these sources of error. In particular, we provide a bound for the spectral norm of a robust covariance filter operating on the sample covariance  $\hat{\mathbf{C}}$  and on the true covariance of the clean data  $\mathbf{C}$ , i.e., the quantity  $\|\mathbf{H}(\hat{\mathbf{C}} + \mathbf{E}_s + \mathbf{E}_l) - \mathbf{H}(\mathbf{C})\|$ , in the following theorem. As a preliminary, we introduce the notion of a Lipschitz covariance filter.

**Definition 1** Consider a covariance filter  $\mathbf{H}(\mathbf{C})$  with frequency response  $h(\lambda)$ . The filter  $\mathbf{H}(\mathbf{C})$  is Lipschitz with constant  $P$  if  $|h(\lambda_i) - h(\lambda_j)| \leq P|\lambda_i - \lambda_j|$  for every non-zero eigenvalue pair  $\lambda_i, \lambda_j, i \neq j$ .

We now provide the result for the robust covariance filter stability.

**Theorem 1** Consider a data matrix  $\mathbf{X}$  with outliers and missing values and let  $\mathbf{C}$  be the true covariance of the clean data samples,  $\tilde{\mathbf{C}}$  be the sample covariance estimated from  $T$  clean samples and  $\hat{\mathbf{C}}$  be the sample covariance estimated from  $T$  perturbed samples (i.e., the observed ones). Consider also a covariance filter  $\mathbf{H}$  with fixed parameters  $\mathcal{H}$  that is Lipschitz with constant  $P$  (cf. Definition 1). The stability of the covariance filter is bounded with high probability as

$$\|\mathbf{H}(\hat{\mathbf{C}} + \mathbf{E}_s + \mathbf{E}_l) - \mathbf{H}(\mathbf{C})\| \leq P\sqrt{N}(1 + \sqrt{N})(\mathcal{O}(T^{-1/2}) + \delta) \quad (7)$$

where  $\delta = \|\tilde{\mathbf{C}} - \hat{\mathbf{C}} - \mathbf{E}_s - \mathbf{E}_l\|$  is a measure of the quality of reconstruction of the clean covariance matrix.

**Proof:** We begin by adding and subtracting the quantity  $\mathbf{H}(\tilde{\mathbf{C}})$  in the bound and using the triangle inequality:

$$\|\mathbf{H}(\hat{\mathbf{C}} + \mathbf{E}_s + \mathbf{E}_l) - \mathbf{H}(\mathbf{C})\| = \quad (8)$$

$$\|\mathbf{H}(\hat{\mathbf{C}} + \mathbf{E}_s + \mathbf{E}_l) - \mathbf{H}(\tilde{\mathbf{C}}) - \mathbf{H}(\mathbf{C}) + \mathbf{H}(\tilde{\mathbf{C}})\| \leq \quad (9)$$

$$\|\mathbf{H}(\hat{\mathbf{C}} + \mathbf{E}_s + \mathbf{E}_l) - \mathbf{H}(\tilde{\mathbf{C}})\| + \|\mathbf{H}(\mathbf{C}) - \mathbf{H}(\tilde{\mathbf{C}})\|. \quad (10)$$

The term  $\|\mathbf{H}(\mathbf{C}) - \mathbf{H}(\tilde{\mathbf{C}})\|$  measures the performance discrepancy of RVNN relatively to the errors in estimation of the covariance matrix due to finite data samples and can be bounded under Lipschitz assumption with high probability by using the results in [14, Proposition 2] and in [23, Theorem 5.6.1] as  $\|\mathbf{H}(\mathbf{C}) - \mathbf{H}(\tilde{\mathbf{C}})\| \leq P\sqrt{N}(1 + \sqrt{N})\mathcal{O}(T^{-1/2})$ .

The term  $\|\mathbf{H}(\hat{\mathbf{C}} + \mathbf{E}_s + \mathbf{E}_l) - \mathbf{H}(\tilde{\mathbf{C}})\|$ , instead, measures the discrepancy of the filter output due to the outliers in the sample covariance estimate. Using again [14, Proposition 2], we get  $\|\mathbf{H}(\hat{\mathbf{C}} + \mathbf{E}_s + \mathbf{E}_l) - \mathbf{H}(\tilde{\mathbf{C}})\| \leq P\sqrt{N}(1 + \sqrt{N})\delta$ . ■

We now extend the bound in Theorem 1 to an RVNN.

**Theorem 2** Consider an RVNN  $\Phi$  of  $L$  layers with  $F_l = F$  for  $l = 1, \dots, L$  and nonlinearities  $\sigma(\cdot)$  such that  $|\sigma(a) - \sigma(b)| \leq |a - b|$ . Assume that all the robust covariance filters in the RVNN are bounded by at most  $\beta$  following Theorem 1. Given a data sample  $\mathbf{x}$  with covariance  $\mathbf{C}$  and  $\|\mathbf{x}\| \leq 1$  w.l.o.g., with high probability it holds that

$$\|\Phi(\mathcal{H}, \mathbf{C}) - \Phi(\mathcal{H}, \hat{\mathbf{C}} + \mathbf{E}_s + \mathbf{E}_l)\| \leq LF^{L-1}\beta.$$

**Proof:** The proof follows from [24, Appendix E]. ■

Theorems 1, 2 show that the stability of RVNNs is affected by the error on the sample covariance estimation due to finite-sample size in a way that decreases with the number of samples as  $\mathcal{O}(T^{-1/2})$ . This represents an improvement over RPCA, which is more sensitive to covariance estimation errors in low-data settings similarly to PCA [6]. Furthermore, the impact of the outliers results in the term  $\delta$  in the bound, which measures the quality of the covariance reconstruction of RVNN. From [19, Theorem 1.1], it follows that the model  $\hat{\mathbf{C}} + \mathbf{E}_s + \mathbf{E}_l$  can reconstruct the clean sample covariance  $\mathbf{C}$  perfectly under the assumption that  $\mathbf{L}$  in (1) is low-rank and not sparse and  $\mathbf{S}$  is sparse and not low-rank by solving a reconstruction problem under the sparsity and low-rank regularization in (6). While this intuitively suggests that RVNN

can achieve good covariance reconstruction, (6) also includes a task-aware term that might either allow for covariance reconstruction even if the assumptions of [19, Theorem 1.1] do not hold, or decrease the quality of the covariance reconstruction if it is in contrast with the task-aware loss. Nevertheless, this improves the nominal VNN, which does not consider the presence of outliers or missing values in the data, resulting in potentially unstable behaviors.

## IV. EXPERIMENTS

### A. Synthetic data

We begin by evaluating RVNN on simulated data. Following the RPCA model, we generate a low-rank data matrix  $\mathbf{X}$  with  $T = 200$ ,  $N = 10$  and rank  $R = 3$  and create regression targets for each observation as  $\mathbf{y} = \mathbf{w}^T \mathbf{X}$ , where  $\mathbf{w} \in \mathbb{R}^N$  is a randomly generated vector of coefficients. We divide the datasets into train/validation/test splits of size 80%/10%/10%. We introduce missing values by setting a percentage  $p$  of the entries of the data matrix  $\mathbf{X}$  to zero.

**Stability.** First, we analyze the stability of RVNN discussed in Theorem 1. We set the percentage of missing values  $p = 0.03$  and, to generate perturbed covariances due to finite-sample errors, we first train RVNN and baselines using the sample covariance from the complete training set, and, during test, we replace it with a sample covariance computed on a subset of the training set of varying size. We compare RVNN and a VNN operating on the data matrix and covariance reconstructed via RPCA on the regression task. For both models, we use 4 layers of size 32, a 2-layer readout MLP with hidden size 16,  $\gamma_s = 0.1$ ,  $\gamma_l = 0.1$ ,  $K = 1$ , dropout of 0.1, a learning rate of 0.01 with Adam optimizer, a single batch and we train for 2000 epochs (we select these hyperparameters via a search). During testing, we recompute RPCA for each subset of the training set. Figure 1 (left) shows that RVNN maintains a consistent performance when the sample covariance is subject to perturbations due to the finite-sample size, whereas RPCA provides unstable results and leads to inconsistent performance. This shows that the stability discussed in Theorem 1 has practical advantages on a downstream task when the sample covariance is perturbed due to a small number of samples.

**Increasing number of missing values.** We further evaluate RVNN on simulated data with an increasing percentage of missing values  $p$ . We compare RVNN, a VNN on the data reconstructed via RPCA, a VNN on the clean data (VNN-C), and a VNN on the perturbed data (VNN-P). For all models, we use the sample covariance from the complete training set during test, and we employ the same hyperparameters mentioned above. We report the average results over 5 repetitions of each experiment with different initializations. Figure 1 (center) shows that introducing missing values does not affect VNN performance significantly for  $p \leq 0.03$ , likely due to the small amount of perturbation introduced, whereas for larger  $p$  it leads to an increase in the MSE for VNN-P compared to VNN-C. Preprocessing the data matrix and covariance with RPCA does not help to fill this gap, resulting

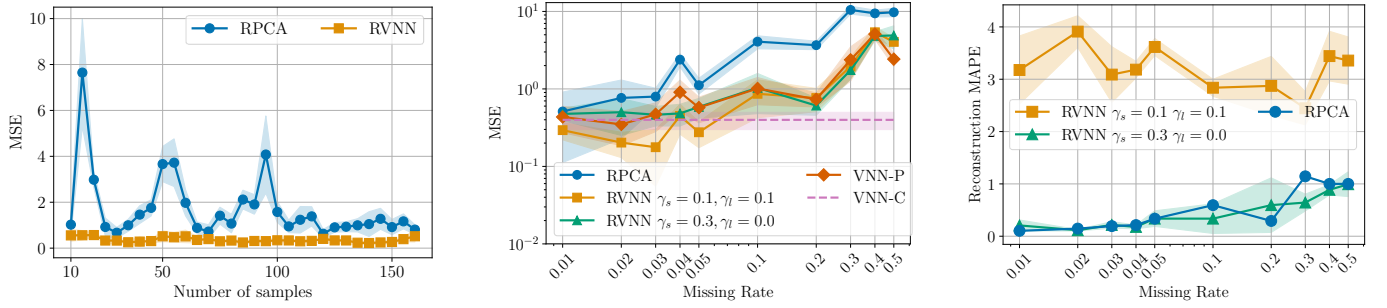


Fig. 1: Simulation results. (left) Regression Mean Squared Error (MSE) for finite-sample perturbed covariance due to decreasing number of samples; (center) Regression error for RVNN and baselines with increasing percentage of missing values; (right) Covariance matrix reconstruction Mean Absolute Percentage Error (MAPE) for RPCA and RVNN.  $\gamma_l = 0$  corresponds to removing  $\mathbf{E}_l$  (i.e., only using a sparse covariance correction term).

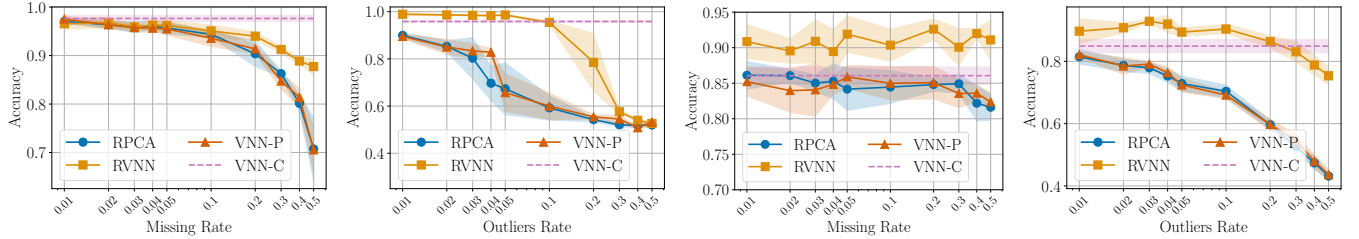


Fig. 2: Classification results on real datasets for increasing percentage of missing samples/outliers  $p$ . From left to right, Epilepsy-missing values, Epilepsy-outliers, Mhealth-missing values, Mhealth-outliers.

in a worse performance for all values of  $p$  (even for  $p \leq 0.05$ , which is its normal operating range [19]). RVNN, instead, matches the performance of the VNN on the clean data for  $p \leq 0.05$  for  $\gamma_s = 0.3$  (without  $\mathbf{E}_l$ ) and even improves the performance for  $\gamma_s = 0.1, \gamma_l = 0.1$ . This shows the effectiveness of estimating a task-aware robust covariance end-to-end, which allows for improvement even on the clean sample covariance when that is not optimal for the task at hand. This is confirmed by Figure 1 (right), which shows the quality of reconstruction of the covariance matrix of clean data for RPCA and RVNN. RPCA and RVNN with  $\gamma_s = 0.3$  (without  $\mathbf{E}_l$ ) provide similar covariance reconstruction quality, since larger  $\gamma_s, \gamma_l$  correspond to more focus on the covariance rather than on the downstream task loss for RVNN. RVNN with  $\gamma_s = 0.1, \gamma_l = 0.1$ , instead, reconstructs the covariance very poorly, as its training dynamics are mostly governed by the task-aware loss, and the learned covariance correction term in this case does not reflect the original covariance but an even better matrix for regression performance on the downstream task. Therefore, the parameters  $\gamma_s, \gamma_l$  allow for a tradeoff between covariance reconstruction quality and downstream task performance based on the application needs.

### B. Real data

Finally, we assess RVNN on two real datasets. **Epilepsy** [25] collects 8000 samples of size 76 from an electrocorticography time series equally divided before and after an epileptic seizure event. We preprocess the data following [26] and we perform

a classification task to identify whether a sample is recorded before or after the seizure event. **Mhealth** [27] collects recordings from wearable devices placed on different body locations of 10 subjects performing 12 different actions. We follow the same preprocessing in [14], leading to 5229 samples of 128 observations in each of the body locations, and we perform a classification task to predict the action performed by the subject in the recorded sample. To evaluate the robustness of RVNN, we introduce into both datasets a varying percentage  $p$  of missing values or outlier values  $\epsilon \sim \mathcal{N}(0, 5)$  in the observed samples and we compare the same models in Section IV-A. For all models on Epilepsy, we use 3 layers of size 32, a 2-layer readout MLP with hidden size 16,  $\lambda = 1$ ,  $K = 1$ ,  $\gamma_s = \gamma_l = 0.1$ , dropout of 0.1, a learning rate of 0.01 with Adam optimizer, a batch size of 100, and we train for 300 epochs. On Mhealth, instead, we use 2 layers of size 32, a 2-layer readout MLP with hidden size 16,  $\lambda = 1$ ,  $K = 1$ ,  $\gamma_s = \gamma_l = 0.1$  no dropout, a learning rate of 0.01 with Adam optimizer, a batch size of 100, and we train for 200 epochs. We report the average results over 5 repetitions of each experiment with different initializations. Figure 2 (left) shows that, on Epilepsy, RVNN performs comparably to the other baselines for small number of missing values ( $p \leq 0.05$ ), maintaining a close accuracy to the VNN-C, whereas for  $p \geq 0.05$ , RVNN performs significantly better than RPCA and VNN-P. With outliers, RVNN improves more significantly over the baselines even for  $p \leq 0.05$ . This confirms that the task-aware learning of a robust covariance brings advantages

in terms of downstream performance compared to RPCA-based methods, and this extends even to perturbations that are larger than those that RPCA is designed to correct (i.e.,  $p \geq 0.05$ ). On Realdisp, instead, the effect of missing values is in general less evident, as the decrease in performance for VNN-P becomes evident only for  $p > 0.3$ . This might indicate that the information is redundantly spread in the data features, making the addition of missing values less impactful. Nevertheless, RVNN outperforms RPCA, VNN-P and even VNN-C for all values of  $p$ . Outliers, instead, decrease baselines' performance more strongly, making the improvement of RVNN more evident (cf. Figure 2 (right)). This indicates that the covariance correction term learned by RVNN on this dataset not only compensates for the injected missing values but also estimates a shift operator that is better than the clean sample covariance for the task at hand. This is consistent with the findings in [14], which shows that on Realdisp the VNN on the sample covariance performs worse than a VNN on a sparse covariance, highlighting the suboptimality of the sample covariance for the task. In more general terms, this experiment demonstrates the flexibility of RVNN to go beyond the reconstruction of the clean covariance depending on the application requirements, thanks to the end-to-end task-aware learning of a robust covariance.

## V. CONCLUSION

In this work, we introduced Robust Covariance Neural Network (RVNN), a graph convolutional neural network that employs a robust covariance estimate as a shift operator to account for outliers and missing values. RVNN learns a robust covariance end-to-end by optimizing a loss on a downstream task jointly with a covariance correction term, which leads to task-aware covariance estimation and, ultimately, better performance compared to alternatives that separate the covariance estimation from the task learning. Furthermore, RVNNs are stable to finite-sample covariance estimation errors, extending nominal VNNs to the case of missing values and outliers, and they empirically provide good regression and classification results on simulated and real data. Future work will explore other robust covariance estimators for larger or more structured data perturbations and will analyze this framework in other application scenarios such as recommendation systems.

## REFERENCES

- [1] M. Dai, Z. Zhang, and A. Srivastava, "Analyzing dynamical brain functional connectivity as trajectories on space of covariance matrices," *IEEE Transactions on Medical Imaging*, vol. 39, no. 3, pp. 611–620, 2020.
- [2] L. Qiao, H. Zhang, M. Kim, S. Teng, L. Zhang, and D. Shen, "Estimating functional brain networks by incorporating a modularity prior," *NeuroImage*, vol. 141, pp. 399–407, 2016. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S105381191630372X>
- [3] J. V. de Miranda Cardoso, J. Ying, and D. P. Palomar, "Algorithms for learning graphs in financial markets," 2020.
- [4] T. Liao, J. Zhao, Y. Liu, K. Ivanov, J. Xiong, and Y. Yan, "Deep transfer learning with graph neural network for sensor-based human activity recognition," in *2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2022, pp. 2445–2452.
- [5] Y. Wang, X. Wang, and H. Yang et al., "Mhagnn: A novel framework for wearable sensor-based human activity recognition combining multi-head attention and graph neural networks," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–14, 2023.
- [6] I. Jolliffe, *Principal component analysis*. New York: Springer Verlag, 2002.
- [7] A. G. Marques, S. Segarra, G. Leus, and A. Ribeiro, "Stationary graph processes and spectral estimation," *IEEE Transactions on Signal Processing*, vol. 65, no. 22, pp. 5911–5926, 2017.
- [8] E. Isufi, F. Gama, D. I. Shuman, and S. Segarra, "Graph filters for signal processing and machine learning on graphs," *IEEE Transactions on Signal Processing*, pp. 1–32, 2024.
- [9] I. T. Jolliffe and J. Cadima, "Principal component analysis: a review and recent developments," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 374, 2016.
- [10] S. Sihag, G. Mateos, C. McMillan, and A. Ribeiro, "covariance neural networks," *Advances in neural information processing systems*, vol. 35, pp. 17 003–17 016, 2022.
- [11] S. Sihag, G. Mateos, C. T. McMillan, and A. Ribeiro, "Transferability of covariance neural networks and application to interpretable brain age prediction using anatomical features," *arXiv preprint arXiv:2305.01807*, 2023.
- [12] A. Cavallo, M. Sabbagi, and E. Isufi, "Spatiotemporal covariance neural networks," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2024, pp. 18–34.
- [13] A. Cavallo, M. Navarro, S. Segarra, and E. Isufi, "Fair covariance neural networks," in *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2025, pp. 1–5.
- [14] A. Cavallo, Z. Gao, and E. Isufi, "Sparse covariance neural networks," *arXiv preprint arXiv:2410.01669*, 2024.
- [15] R. A. Maronna, "Robust m-estimators of multivariate location and scatter," *The annals of statistics*, pp. 51–67, 1976.
- [16] M. Salibián-Barrera, S. Van Aelst, and G. Willems, "Principal components analysis based on multivariate mm estimators with fast and robust bootstrap," *Journal of the American Statistical Association*, vol. 101, no. 475, pp. 1198–1211, 2006.
- [17] G. Li and Z. Chen, "Projection-pursuit approach to robust dispersion matrices and principal components: primary theory and monte carlo," *Journal of the American Statistical Association*, vol. 80, no. 391, pp. 759–766, 1985.
- [18] R. Maronna, "Principal components and orthogonal regression based on robust scales," *Technometrics*, vol. 47, no. 3, pp. 264–273, 2005.
- [19] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *Journal of the ACM (JACM)*, vol. 58, no. 3, pp. 1–37, 2011.
- [20] H. Cai, J. Liu, and W. Yin, "Learned robust pca: A scalable deep unfolding approach for high-dimensional outlier detection," *Advances in Neural Information Processing Systems*, vol. 34, pp. 16 977–16 989, 2021.
- [21] H. Dong, M. Shah, S. Donegan, and Y. Chi, "Deep unfolded tensor robust pca with self-supervised learning," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.
- [22] J. Pu, Y. Panagakis, and M. Pantic, "Learning low rank and sparse models via robust autoencoders," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 3192–3196.
- [23] R. Vershynin, *High-Dimensional Probability: An Introduction with Applications in Data Science*, ser. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2018.
- [24] F. Gama, E. Isufi, G. Leus, and A. Ribeiro, "Graphs, convolutions, and neural networks: From graph filters to graph neural networks," *IEEE Signal Processing Magazine*, vol. 37, no. 6, p. 128–138, Nov. 2020. [Online]. Available: <http://dx.doi.org/10.1109/MSP.2020.3016143>
- [25] M. A. Kramer, E. D. Kolaczyk, and H. E. Kirsch, "Emergent network topology at seizure onset in humans," *Epilepsy Research*, vol. 79, no. 2, pp. 173–186, 2008. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0920121108000430>
- [26] A. Natali, E. Isufi, M. Coutino, and G. Leus, "Learning time-varying graphs from online data," *IEEE Open Journal of Signal Processing*, vol. 3, pp. 212–228, 2022.
- [27] O. Banos, R. García, and J. Holgado-Terriza et al., "mhealthdroid: A novel framework for agile development of mobile health applications," vol. 8868, 12 2014, pp. 91–98.