TU Delft

Delft University of Technology

Document Version
Final published version

Licence
Dutch Copyright Act (Article 25fa)

Citation (APA)
Wijnands, R., Cavallo, A., Hunyadi, B., Isufi, E., & Leus, G. (2025). Scalable Higher-Order Topology Identification from Nodal Observations. In *Proceedings of the 2025 IEEE 10th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP 2025)* (pp. 101-105). IEEE.
https://doi.org/10.1109/CAMSAP66162.2025.11423970

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.

This work is downloaded from Delft University of Technology.

2025 IEEE 10th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP) | 979-8-3315-2669-6/25/$31.00 ©2025 IEEE | DOI: 10.1109/CAMSAP66162.2025.11423970

Scalable Higher-Order Topology Identification from Nodal Observations

Ruben Wijnands, Andrea Cavallo, Borbála Hunyadi, Elvin Isufi, Geert Leus
Faculty of EEMCS, Delft University of Technology, Delft, The Netherlands

***Abstract*—This paper proposes a scalable method for identifying interactions in higher-order networks from observations of nodal processes. Finding such dependencies is important in many disciplines, including neuroscience, social influence modeling, and beyond. However, current approaches are either limited to extracting pairwise dependencies or struggle with scalability, as estimating higher-order dependencies becomes computationally prohibitive. To overcome these challenges, we introduce a tensor-based graph Volterra model that leverages low-rank decomposition techniques to estimate higher-order interactions efficiently. Our approach not only reduces computational and storage complexity but also acts as an implicit regularizer, improving network estimation in ill-posed settings. We validate our method through simulations and real data experiments, demonstrating competitive performance and enhanced scalability compared to existing techniques.**

***Index Terms*—Higher-order interactions, graph Volterra models, tensor decompositions, structural equation models**

I. Introduction

Estimating interactions beyond graph-based pairwise connections is key to understanding the complex systems arising in applications such as neuroscience, where co-activation patterns of ensembles of brain areas reveal higher-order influence and deeper insights into the cerebral dynamics [1], [2], and social networks, where groups and communities represent beyond-pairwise influences that govern the system [3]. Various approaches estimate interactions from nodal observations relying on statistical moments [4], [5], information-theoretic metrics [6], [7] and model-based estimators [8]–[10], which define structured relationships among node signals. For example, the structural equation model (SEM) [8] defines pairwise dependencies through linear equations, whereas [9], [10] account for non-linear pairwise interactions and temporal memory. To estimate beyond-pairwise interactions, the autoregressive graph Volterra model (AGVM) [11], [12] merges the SEM with Volterra kernels, i.e., it defines dependencies between a target node and subsets of the other nodes.

However, the AGVM suffers from combinatorial growth in the number of model parameters as the number of nodes and interaction order increase, which consequently requires vast amounts of observations for reliable estimation and poses severe computational challenges. To reduce the number of observations needed for parameter estimation, several works propose sparsity-based regularizers [11]–[14], which, nevertheless, are still problematic when increasing the order of interactions and require explicit storage of the high-dimensional learned model parameters. Other methods apply kernel-based algorithms [9], [10], [15] to deal with high-dimensional data, but they do not scale well in the number of observations due to kernel matrix inversion.

To overcome these limitations, we propose the canonical polyadic Volterra model (CPVM), which reformulates the AGVM in tensor format and enforces low-rank constraints via the canonical polyadic decomposition (CPD). The corresponding estimation method, which we refer to as CPV, follows a similar approach to [16], which applied the CPD to kernel machines. CPV is significantly more scalable than AGVM estimation and kernel-based methods in terms of the number of observations, nodes, and order of interactions. Additionally, our simulations show that the CPD acts as a regularizer, enabling reliable topology estimation in low-sample regimes. In higher-order topology identification on a co-authorship dataset, CPV matches competing methods in performance while drastically reducing computation time.

Notation. Scalars are denoted by y, vectors by $\mathbf{y}$, and matrices by $\mathbf{Y}$. Sets or tensors are denoted by $\mathcal{Y}$. We use $\text{vec}(\cdot)$ for the vectorization operator such that the indices of $\text{vec}(\mathcal{Y})$ follow reverse lexicographic ordering [17]. The Kronecker product is denoted by $\otimes$, the Khatri-Rao product (a column-wise Kronecker product) by $\odot$, the outer product by $\circ$, and the Hadamard product by $\circledast$. The repeated Kronecker product of vector $\mathbf{y}$, is given by $\mathbf{y}^{\otimes D} = \mathbf{y} \otimes \cdots \otimes \mathbf{y}$ (with D factors). We use similar notation for the repeated outer and Hadamard products. Finally, the Frobenius inner product between two tensors $\mathcal{Y}$ and $\mathcal{W}$ is given by $\langle \mathcal{Y}, \mathcal{W} \rangle_{\text{F}} = \text{vec}(\mathcal{Y})^{\mathsf{T}} \text{vec}(\mathcal{W})$.

II. Problem formulation

Given a set of N nodes and respective nodal observations $\mathbf{x}_t \in \mathbb{R}^N$ over a time period $t \in \{0, ..., T-1\}$, our goal is to infer relationships between the nodes from the nodal observations. For ease of model exposition, we ignore the time index t for now, as our model has no temporal memory and can be explained only from a single observation. Later, we will use a subscript t whenever the related variable depends on time. Now, consider a target node signal $x \in \mathbb{R}$, and $\tilde{\mathbf{x}} = [\tilde{x}_1, \ldots, \tilde{x}_{N-1}]^{\mathsf{T}} \in \mathbb{R}^{N-1}$ being all node signals excluding the target node, indexed from 1 to $N-1$ for simplicity.

This work is part of the GraSPA project (project 19497 within the TTW OTP programme), which is financed by the Netherlands Organization for Scientific Research (NWO). Part of this work was supported by the TU Delft AI Labs programme and the NWO VENI proposal 222.032. E-mail: g.j.t.leus@tudelft.nl. Code can be found at https://github.com/rubenwijnands999/CAMSAP2025-Higher-order-interactions

Authorized licensed use limited to: TU Delft Library. Downloaded on May 04,2026 at 07:29:44 UTC from IEEE Xplore. Restrictions apply.

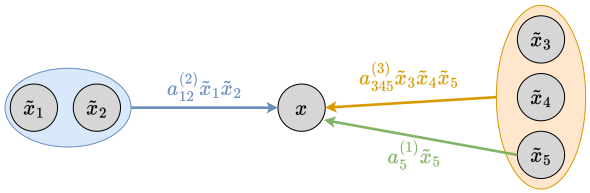


Fig. 1: Example higher-order interactions.

The linear SEM (without exogenous inputs) postulates that a target node signal, considered as an output variable, depends linearly on the other observed node signals [8]:

$$x = \sum_{j=1}^{N-1} a_j^{(1)} \tilde{x}_j + \epsilon, \tag{1}$$

where $a_j^{(1)}$ encodes the directed relationship from node j to the target node, and ϵ is a noise term.

Moving beyond pairwise interactions, the AGVM in [11], [12] extends (1) by accounting for nonlinear interactions between groups of nodes. In detail, considering interactions of any order up to order D, the AGVM is specified by

$$x = a^{(0)} + \sum_{j=1}^{N-1} a_j^{(1)} \tilde{x}_j + \sum_{d=2}^{D} A^{(d)}(\tilde{\mathbf{x}}) + \epsilon, \tag{2}$$

where $a^{(0)}$ is a bias term. The higher-order interaction module $A^{(d)}(\tilde{\mathbf{x}}) \in \mathbb{R}$ is given by

$$A^{(d)}(\tilde{\mathbf{x}}) = \sum_{j_1=1}^{N-1} \cdots \sum_{j_d=j_{d-1}}^{N-1} a_{j_1,j_2,\ldots,j_d}^{(d)} f(\{\tilde{x}_{j_q}\}_{q=1}^{d}). \tag{3}$$

Here, each non-zero Volterra coefficient $a_{j_1,j_2,\ldots,j_d}^{(d)}$ indicates the presence of an interaction of d nodes towards the target node. Moreover, $f(\cdot)$ is a permutation-invariant nonlinear function specifying the interaction type between the d nodes. The AGVM described in (2) allows all subsets of nodes with cardinality at most D to describe the output signal x nonlinearly in the observed node signals. In total, the AGVM has $P = \binom{N+D-1}{D}$ unique model parameters [13], which grows fast for larger D and N. From now on, we consider a multiplicative interaction function $f(\cdot)$, a typical choice for modeling nonlinear higher-order relationships [18]. Thereby, the AGVM coincides with a polynomial SEM [19].

Example 1. In Fig. 1 we display an example of multiplicative interactions up to order $D = 3$ with corresponding Volterra coefficients in the AGVM. Specifically, a 1-st order interaction between node 5 and the target node is described by the coefficient $a_5^{(1)}$, the 2-nd order interaction between nodes 1, 2 and the target node by $a_{1,2}^{(2)}$, and the 3-rd order interaction between nodes 3, 4, 5 and the target node by $a_{3,4,5}^{(3)}$. The aggregated signal at the target node is the weighted sum of node signal products across interactions:

$$x = a_5^{(1)}\tilde{x}_5 + a_{1,2}^{(2)}\tilde{x}_1\tilde{x}_2 + a_{3,4,5}^{(3)}\tilde{x}_3\tilde{x}_4\tilde{x}_5. \tag{4}$$

In this work, we aim to estimate the Volterra coefficients in (2)-(3) from nodal observations. Increasing the interaction order D for a fixed N leads to combinatorial growth in the number of model parameters P. In turn, this requires many observations for accurate model estimation and drastically increases the computational and storage complexity of estimation methods. To overcome this issue, we reformulate the AGVM in tensor form to adopt the method from [16] for higher-order topology inference with high D.

A. Tensor-based formulation of the AGVM

To create a tensor-based formulation of the AGVM, we can conveniently write all multiplicative combinations of node signals up to order D as follows. Let us define an interaction tensor $\mathcal{Z} \in \mathbb{R}^{N\times\cdots\times N}$ (with D modes) as $\mathcal{Z} = \mathbf{z} \circ \cdots \circ \mathbf{z}$, where $\mathbf{z} \in \mathbb{R}^N$ is a feature map [20] of the node signals $\tilde{\mathbf{x}}$: $\mathbf{z} = [1, \tilde{\mathbf{x}}^\mathsf{T}]^\mathsf{T}$. This construction generates all possible N^D multiplicative interactions of order up to D for the set $\{\tilde{x}_j\}_{j=1}^N$, which include redundancies compared to the P unique interactions in $f(\cdot)$ from (2)-(3). To model x based on these interactions, we introduce a weight tensor $\mathcal{W} \in \mathbb{R}^{N\times\cdots\times N}$, which assigns weights to all multiplicative interactions, including redundant ones. This leads to a compact tensor-based formulation of the AGVM:

$$x = \langle \mathcal{W}, \mathcal{Z} \rangle_\mathrm{F}. \tag{5}$$

Note that many of the Volterra coefficients in (2)-(3) can be non-uniquely mapped to the weights in $\mathcal{W}$. We can retrieve a Volterra coefficient from $\mathcal{W}$, as we explain next.

Let $\mathcal{I} = \{j_q\}_{q=1}^d$ be the sorted index set of a d-th order interaction. Its zero-padded index set is $\mathcal{L} = \mathcal{I} \cup \{0\}^{D-d}$ if $d < D$, and $\mathcal{L} = \mathcal{I}$ if $d = D$. Also, $\Pi(\mathcal{L})$ is the set of sets that contains all unique permutations of $\mathcal{L}$, including the original one. Then, a Volterra coefficient $a_\mathcal{I}^{(d)}$ is obtained as

$$a_\mathcal{I}^{(d)} = \sum_{\pi\in\Pi(\mathcal{L})} w_\pi, \tag{6}$$

where π is an ordered index set taken from $\Pi(\mathcal{L})$. Clearly, a coefficient $a_\mathcal{I}^{(d)}$ could simply be located at only one entry in $\mathcal{W}$, or be split across multiple entries.

Example 1 (cont.). Our tensor-based AGVM leads to the following expression for the target node signal x:

$$\begin{aligned} x &= a_5^{(1)}\tilde{x}_5 + a_{1,2}^{(2)}\tilde{x}_1\tilde{x}_2 + a_{3,4,5}^{(3)}\tilde{x}_3\tilde{x}_4\tilde{x}_5 \\ &= \sum_{\pi\in\Pi(\{5,0,0\})} w_\pi \tilde{x}_5 + \sum_{\pi\in\Pi(\{1,2,0\})} w_\pi \tilde{x}_1\tilde{x}_2 + \\ &\quad \sum_{\pi\in\Pi(\{3,4,5\})} w_\pi \tilde{x}_3\tilde{x}_4\tilde{x}_5, \end{aligned} \tag{7}$$

where the weights w_π corresponding to lower-order $d < D$ interactions are located at the edges and faces of $\mathcal{W}$ due to the zero indexes.

B. The canonical polyadic Volterra model (CPVM)

The redundancies introduced in the tensor-based formulation of the AGVM are necessary to exploit the advantages of imposing a tensor decomposition on $\mathcal{W}$. Specifically, we consider the canonical polyadic decomposition (CPD):

$$\mathcal{W} = \sum_{r=1}^{R} \mathbf{w}_r^{(1)} \circ \cdots \circ \mathbf{w}_r^{(D)}, \tag{8}$$

which writes tensor $\mathcal{W}$ as a sum of R rank-1 tensors, with factor matrices given by $\mathbf{W}^{(d)} = [\mathbf{w}_1^{(d)}, \ldots, \mathbf{w}_R^{(d)}] \in \mathbb{R}^{N\times R}$. This results in the following CPVM, also visualized in Fig. 2:

$$\begin{aligned} x &= \sum_{r=1}^{R} (\mathbf{w}_r^{(D)} \otimes \cdots \otimes \mathbf{w}_r^{(1)})^\mathsf{T} \mathbf{z}^{\otimes D} \\ &= \mathbf{1}^\mathsf{T} \left(\mathbf{W}^{(1)\mathsf{T}} \mathbf{z} \circledast \cdots \circledast \mathbf{W}^{(D)\mathsf{T}} \mathbf{z} \right). \end{aligned} \tag{9}$$

Authorized licensed use limited to: TU Delft Library. Downloaded on May 04,2026 at 07:29:44 UTC from IEEE Xplore. Restrictions apply.

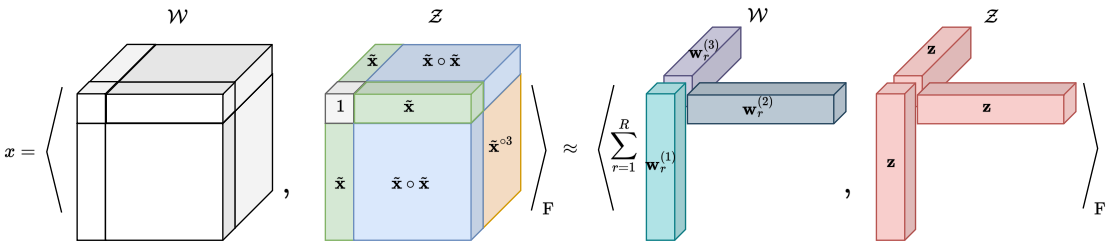


Fig. 2: Visualization of the AGVM in tensor format approximated by the CPVM. Left: the target node signal is a sum over the elements resulting from the point-wise multiplication of the weight tensor $\mathcal{W}$ and interaction tensor $\mathcal{Z}$. Right: $\mathcal{W}$ approximated as CPD and $\mathcal{Z}$ displayed as an outer product.

Crucially, the CPD significantly reduces the storage requirements as the number of model parameters decreases from N^D (or P) to RND. From (9), we also see that the model output can be computed efficiently with only $\mathcal{O}(RND)$ operations compared to $\mathcal{O}(N^D)$ (or $\mathcal{O}(P)$) operations.

In case only $M \ll P$ interactions occur, the interaction structure is sparse, and a rank $R = M$ suffices (though not always necessary) to represent the node signal x. Sparsity of polynomial expansions has been leveraged to improve estimation accuracy in [9], [13]. While our approach does not inherently yield a sparse representation, sparsity in the Volterra coefficients upper-bounds the rank R and thus reduces storage complexity. What remains is the estimation of the factor matrices of a rank-R CPD, which we describe next.

III. Methods

To start with, we now consider multiple observations of the target node $\{x_t\}_{t=0}^{T-1}$ and multiple observations of the other nodes except the target node $\{\tilde{\mathbf{x}}_t\}_{t=0}^{T-1}$. Let us then consider the following optimization problem:

$$\begin{aligned} \min_{\mathcal{W}} \quad & \textstyle\sum_{t=0}^{T-1} (x_t - \langle \mathcal{W}, \mathcal{Z}_t \rangle_{\mathrm{F}})^2 + \lambda \mathcal{R}(\mathcal{W}) \\ \text{s.t.} \quad & \text{CPD-rank}(\mathcal{W}) = R, \end{aligned} \tag{10}$$

where $\mathcal{R}(\mathcal{W})$ is an optional regularization term, e.g., to combat ill-conditioning. Similar to [16], we take a block-coordinate descent approach by solving (10) for each d-th factor matrix $\mathbf{W}^{(d)} \in \mathbb{R}^{N \times R}$ separately, while keeping the other factor matrices fixed. Now, let us detail this block-coordinate descent approach. To start, note that we can rewrite the Frobenius inner product term to take $\mathbf{W}^{(d)}$ out (see [16] for the derivation):

$$\langle \mathcal{W}, \mathcal{Z}_t \rangle_{\mathrm{F}} = \mathrm{vec}(\mathbf{W}^{(d)})^{\mathsf{T}} \mathbf{g}^{(d)}(\tilde{\mathbf{x}}_t), \tag{11}$$

where $\mathbf{g}^{(d)}(\tilde{\mathbf{x}}_t) = \left(\mathbf{W}^{(1)\mathsf{T}}\mathbf{z} \circledast \cdots \circledast \mathbf{W}^{(D)\mathsf{T}}\mathbf{z}\right) \otimes \mathbf{z}$ excludes the $\mathbf{W}^{(d)\mathsf{T}}\mathbf{z}$ term. Now, we define matrix $\mathbf{G}^{(d)} \in \mathbb{R}^{NR \times T}$ as

$$\begin{aligned} \mathbf{G}^{(d)} &= [\mathbf{g}^{(d)}(\tilde{\mathbf{x}}_0), ..., \mathbf{g}^{(d)}(\tilde{\mathbf{x}}_{T-1})] \\ &= \left(\mathbf{W}^{(1)\mathsf{T}}\mathbf{Z} \circledast \cdots \circledast \mathbf{W}^{(D)\mathsf{T}}\mathbf{Z}\right) \odot \mathbf{Z}, \end{aligned} \tag{12}$$

where $\mathbf{Z} = [\mathbf{z}_0, ..., \mathbf{z}_{T-1}]$. Then, we can write a subproblem to include all observed time points collected in vector $\mathbf{m} = [x_0, ..., x_{T-1}]^{\mathsf{T}}$ as

$$\min_{\mathrm{vec}(\mathbf{W}^{(d)})} \quad ||\mathbf{m} - \mathbf{G}^{(d)\mathsf{T}} \mathrm{vec}(\mathbf{W}^{(d)})||_2^2 + \lambda \mathcal{R}(\mathbf{W}^{(d)}). \tag{13}$$

If the left-hand side of (13) is under-determined (when $NR > T$) or $\mathbf{G}^{(d)}$ is degenerate, additional regularization is necessary. If we use $\mathcal{R}(\mathcal{W}) = \langle \mathcal{W}, \mathcal{W} \rangle_{\mathrm{F}}$, we obtain the following regularization term:

$$\mathcal{R}(\mathbf{W}^{(d)}) = \mathrm{vec}(\mathbf{W}^{(d)})^{\mathsf{T}} \left(\mathbf{H}^{(d)\mathsf{T}} \otimes \mathbf{I}\right) \mathrm{vec}(\mathbf{W}^{(d)}), \tag{14}$$

where $\mathbf{H}^{(d)} = \left(\mathbf{W}^{(1)\mathsf{T}}\mathbf{W}^{(1)} \circledast \cdots \circledast \mathbf{W}^{(D)\mathsf{T}}\mathbf{W}^{(D)}\right)$ excludes the $\mathbf{W}^{(d)\mathsf{T}}\mathbf{W}^{(d)}$ term (see [16] for the derivation). Then, the closed-form solution for (13) is available and given by:

$$\mathrm{vec}(\hat{\mathbf{W}}^{(d)}) = \left(\mathbf{G}^{(d)}\mathbf{G}^{(d)\mathsf{T}} + \lambda(\mathbf{H}^{(d)\mathsf{T}} \otimes \mathbf{I})\right)^{-1} \mathbf{G}^{(d)}\mathbf{m}. \tag{15}$$

Complexity. Solving (15) D times per iteration leads to $\mathcal{O}(DTN^2R^2 + DN^3R^3)$ operations for computing the Gram matrix and performing matrix inversion, respectively. Fitting the AGVM using, e.g., ridge or lasso regression requires $\mathcal{O}(TP^2 + P^3)$ or $\mathcal{O}(TP^2 + TP)$ operations, respectively.

Convergence. The algorithm may not converge to a global minimum due to the non-convexity of (10). However, for non-degenerate matrices $\mathbf{G}^{(d)}$ (and, optionally, positive definite matrices $\mathbf{H}^{(d)}$), each subproblem in (13) is strictly convex, ensuring monotonically decreasing cost and convergence to a stationary point under mild conditions [21]. In practice, we run the algorithm with multiple random initializations and select the solution with the lowest cost.

Remark. As noted in [16], imposing a CPD on $\mathcal{W}$ regularizes the model by restricting its flexibility. We demonstrate this effect in Section IV. However, for a given CPD rank R, the required number of observations T for proper estimation is unclear. Moreover, contrary to our approach, one could consider a symmetric CPD, which requires estimating and storing only RN parameters. However, imposing symmetry may increase the rank R and is only preferable if the rank increase factor is smaller than D, which is data-dependent.

IV. Results

We evaluate our method in two settings: (i) on synthetic data, we examine how the CPD-rank R and the number of observations T influence the estimation accuracy; (ii) on a real co-authorship dataset, we assess our method's ability to identify existing higher-order interactions.

We compare our CPV method to an AGVM fitted using ridge (RV), lasso (LV), and elastic net (EV) regression, implemented with the `Ridge`, `Lasso`, and `ElasticNet` functions from the Python scikit-learn [22] library, respectively.

Authorized licensed use limited to: TU Delft Library. Downloaded on May 04,2026 at 07:29:44 UTC from IEEE Xplore. Restrictions apply.

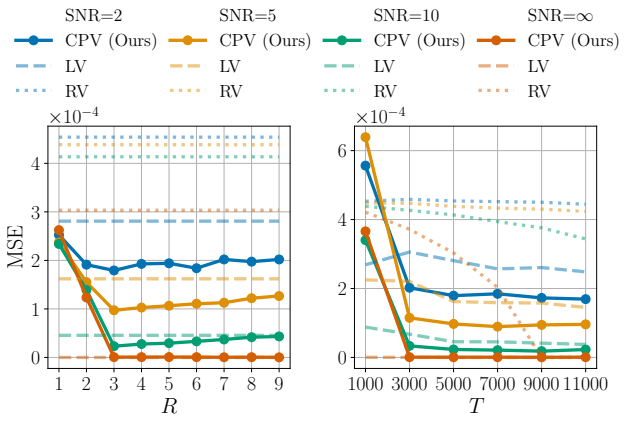


Fig. 3: Quality of estimated topology measured by the mean squared error (MSE) for varying CPD rank R (left) and increasing number of observations T (right) on simulated data.

Related works use RV, LV, and EV or slight adaptations [12]–[14] to fit the AGVM. We do not compare against the kernel method in [9], as it does not inherently estimate higher-order interactions, and a kernel-based extension of the AGVM would scale poorly in T, which is problematic in our setting. For all models, we select the best hyperparameters –the regularization parameters and the rank R for CPV– using simulation data for synthetic experiments and a validation set for real data experiments. All reported results are averaged over five runs.

Simulations. We generate a synthetic dataset following the AGVM model in (2) with $N = 20$ nodes, $\tilde{\mathbf{x}}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and multiplicative interactions. We add noise $\epsilon_t \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$ and define the signal-to-noise ratio as $\text{SNR} = \frac{1}{NT\sigma^2}\sum_{t=0}^{T-1}\mathbf{x}_t^\mathsf{T}\mathbf{x}_t$. The ground truth topology consists of a single target node with 4 particular incoming unit-weighted higher-order edges of order up to $D = 4$, resulting in $P = 8855$ model parameters. For the considered scenario, the minimum rank for perfect estimation is $R = 3$. We vary the rank R while fixing $T = 5000$ and the number of observations T while fixing $R = 3$ to demonstrate the effect of rank overestimation and the regularization effect of the CPD, respectively. Here, we do not use the optional regularization in the CPV algorithm. For clarity of presentation, we omit the EV method.

Fig. 3 shows that underestimating the rank (i.e., $R < 3$) leads to worse estimation quality for CPV, as expected. For $R \geq 3$, CPV matches the performance of LV for $\text{SNR} = \infty$ and outperforms both LV and RV in all other cases, showing the effectiveness of the CPV in retrieving ground-truth interaction weights. Overestimating the rank slightly worsens topology estimation due to noise fitting, yet our method still outperforms others. Interestingly, when the problem is ill-posed (i.e., for $T < P$), CPV performs better than LV and RV, except for very small T. Clearly, RV is less suited here and is only accurate for $T > P$ without noise. This demonstrates the regularization effect of the CPD.

Real data. We evaluate our model on a topology identification task for the ACM co-authorship network [23] dataset, containing 12,499 papers with 1,903 keywords and 17,431 authors.

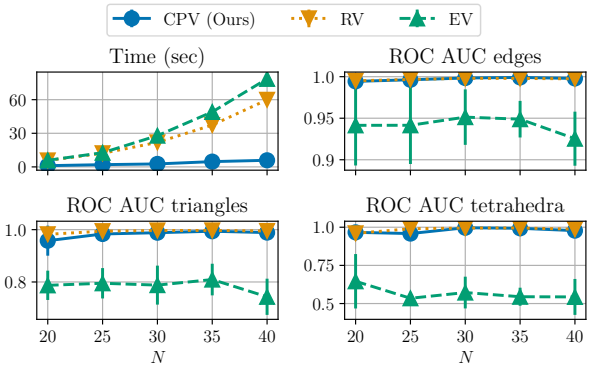


Fig. 4: Time (top-left) and ROC AUC for the identification of interactions of size 2 (edges), 3 (triangles), and 4 (tetrahedra) on the co-authorship network for increasing N, with $R = 4$.

We sample subsets of varying sizes from the authors as node sets, and we define ground-truth interactions between 2 or more authors (up to 4) if they collaborated on the same paper. We use the frequencies with which authors use keywords as observations. To define a global topology from the estimated target node-specific Volterra coefficients obtained using (6), we consider for each interaction group all the coefficients of the nodes involved, and we take their median as the score for the global interaction. This removes potential outlier nodes since higher-order interactions might not be equally important for all nodes involved. We aim to obtain higher scores for ground-truth interactions than non-existing ones, which we evaluate through the ROC AUC. Here, we do not report comparisons with LV since its performance appeared significantly worse.

Fig. 4 shows that CPV performs comparably to RV and better than EV for detecting interactions while significantly reducing the computation time when increasing the number of nodes. This corroborates the effectiveness of the CPVM in modeling higher-order interactions and retrieving ground-truth interactions with CPV while reducing the number of parameters to estimate and, ultimately, significantly improving the model efficiency and scalability.

V. Conclusion

In this work, we proposed a scalable method for learning higher-order interactions from data. We significantly reduced the computational and storage complexity typically associated with higher-order topology estimation by leveraging a tensor-based representation of the AGVM combined with a CPD representation of the Volterra kernels. Our approach scales linearly in the interaction order and the number of observations, making it well-suited for large-scale applications. Furthermore, the CPD allows for direct control over the number of model parameters, acting as an implicit regularizer that improves estimation in ill-posed settings. Experimental results demonstrate that our method achieves performance on par with other computationally expensive approaches while offering substantial speed-up and improved robustness, highlighting the potential of tensor-based techniques for efficient and scalable higher-order network analysis.

Authorized licensed use limited to: TU Delft Library. Downloaded on May 04,2026 at 07:29:44 UTC from IEEE Xplore. Restrictions apply.

REFERENCES

[1] T. Varley, M. Pope, J. Faskowitz, and O. Sporns, "Multivariate Information Theory Uncovers Synergistic Subsystems of the Human Cerebral Cortex," *arXiv:2206.06477*, 2022.

[2] F. Santos, P. Tewarie, P. Baudot, A. Luchicchi, D. Barros De Souza *et al.*, "Emergence of High-Order Functional Hubs in the Human Brain," *BioRxiv*, 2023.

[3] U. Alvarez-Rodriguez, F. Battiston, G. De Arruda, Y. Moreno, M. Perc *et al.*, "Evolutionary dynamics of higher-order interactions in social networks," *Nat. Hum. Behav.*, vol. 5, no. 5, pp. 586–595, 2021.

[4] R. Hindriks, T. Broeders, M. Schoonheim, L. Douw, F. Santos *et al.*, "Higher-order functional connectivity analysis of resting-state functional magnetic resonance imaging data using multivariate cumulants," *Hum. Brain Mapp.*, vol. 45, no. 5, p. e26663, 2024.

[5] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*, vol. 9, no. 3, pp. 432–441, 2008.

[6] F. Rosas, P. Mediano, M. Gastpar, and H. Jensen, "Quantifying high-order interdependencies via multivariate extensions of the mutual information," *Phys. Rev. E.*, vol. 100, no. 3, Sep. 2019.

[7] P. Williams and R. Beer, "Nonnegative decomposition of multivariate information," *arXiv:1004.2515*, 2010.

[8] J. Hox and T. Bechger, "An introduction to structural equation modeling," *Fam. Sci. Rev.*, vol. 11, pp. 354–373, 1998.

[9] Y. Shen, B. Baingana, and G. Giannakis, "Kernel-Based Structural Equation Models for Topology Identification of Directed Networks," *IEEE Trans. Signal Process.*, vol. 65, no. 10, pp. 2503–2516, 2017.

[10] Y. Shen, B. Baingana, and G. B. Giannakis, "Topology inference of directed graphs using nonlinear structural vector autoregressive models," *ICASSP 2017*, pp. 6513–6517, 2017.

[11] M. Coutino, G. Karanikolas, G. Leus, and G. Giannakis, "Self-Driven Graph Volterra Models for Higher-Order Link Prediction," *ICASSP 2020*, pp. 3887–3891, 2020.

[12] Q. Yang, M. Coutino, G. Leus, and G. Giannakis, "Autoregressive graph Volterra models and applications," *EURASIP J. Adv. Signal Process.*, vol. 2023, no. 1, p. 4, 2023.

[13] V. Kekatos and G. Giannakis, "Sparse Volterra and Polynomial Regression Models: Recoverability and Estimation," *IEEE Trans. Signal Process.*, vol. 59, no. 12, pp. 5907–5920, 2011.

[14] A. Buciulea, E. Isufi, G. Leus, and A. Marques, "Learning Graphs and Simplicial Complexes from Data," *ICASSP 2024*, pp. 9861–9865, 2024.

[15] R. Money, J. Krishnan, and B. Beferull-Lozano, "Online non-linear topology identification from graph-connected time series," *2021 IEEE Data Science and Learning Workshop (DSLW)*, pp. 1–6, 2021.

[16] F. Wesel and K. Batselier, "Large-Scale Learning with Fourier Features and Tensor Decompositions," *NeurIPS*, vol. 34, pp. 17 543–17 554, 2021.

[17] A. Cichocki, N. Lee, I. Oseledets, A. Phan, Q. Zhao *et al.*, "Tensor Networks for Dimensionality Reduction and Large-scale Optimization: Part 1 Low-Rank Tensor Decompositions," *Found. Trends Mach. Learn.*, vol. 9, no. 4-5, pp. 249–429, 2016.

[18] L. Aiken and S. West, *Multiple Regression: Testing and Interpreting Interactions*. SAGE Publications, 1991.

[19] M. M. Wall and Y. Amemiya, "Estimation for Polynomial Structural Equation Models," *J. Am. Stat. Assoc.*, vol. 95, no. 451, pp. 929–940, 2000.

[20] K. Batselier, Z. Chen, and N. Wong, "Tensor Network alternating linear scheme for MIMO Volterra system identification," *Automatica*, vol. 84, pp. 26–35, 2017.

[21] P. Tseng, "Convergence of a Block Coordinate Descent Method for Nondifferentiable Minimization," *J. Optim. Theory Appl.*, vol. 109, no. 3, pp. 475–494, 2001.

[22] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[23] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye *et al.*, "Heterogeneous graph attention network," in *WWW*, 2019, pp. 2022–2032.

Authorized licensed use limited to: TU Delft Library. Downloaded on May 04,2026 at 07:29:44 UTC from IEEE Xplore. Restrictions apply.