LEARNING LOCAL HIGHER-ORDER INTERACTIONS WITH TOTAL CORRELATION

Thomas Kerby, Teresa White, Kevin R. Moon

Department of Mathematics and Statistics, Utah State University, Logan UT, USA

2024 IEEE 34th International Workshop on Machine Learning for Signal Processing (MLSP) | 979-8-3503-7225-0/24/$31.00 ©2024 IEEE | DOI: 10.1109/MLSP58920.2024.10734758

ABSTRACT

In domains such as ecological systems, collaborations, and the human brain the variables can interact in complex ways. Yet accurately characterizing higher-order variable interactions (HOIs) is a difficult problem that is further exacerbated when the HOIs vary locally. To solve this problem we propose a new method called Local Correlation Explanation (CorEx) to learn HOIs at a local scale by first clustering data points based on their proximity on the data manifold. We then use a multivariate version of the mutual information called the total correlation, to construct a latent factor representation of the data within each cluster to learn the local HOIs. We show that Local CorEx matches or outperforms global methods in effectively learning HOIs in synthetic data and demonstrate its suitability to explore and interpret the inner workings of trained neural networks.

***Index Terms*—** Higher-Order Interaction Detection, Interpretable Machine Learning, Total Correlation

1. INTRODUCTION

In complex systems, variables often interact together in complicated ways [1, 2]. In data measured from neural activity [3, 4], collaborations [5], and ecological systems [6, 7], higher-order interactions (HOIs) have been shown to play a key role. These complex systems can be effectively modeled as a graph where entities exist with connections between them indicating a shared relationship. Recent work on discovering HOIs has focused on taking a single graph and extracting HOIs in the form of hyper edges [8, 9, 10]. These methods all require a graph as input where the edges are binary. However, data measured from complex systems often do not directly include information about edges between nodes. For instance, in gene expression data, we may want to know the relationships between the application of a drug and specific g enes. But we can only measure gene expressions when the drug is present and when it is not. For neural networks, we might desire to understand how the nodes within a hidden layer interact, but we can only measure the activations of the nodes given different inputs. In both of these instances, we must infer the relationships from the data collected.

This work was supported in part by the NSF under Grant 2212325.

Variable interactions have long been studied using tools such as the Pearson correlation, the Spearman correlation, mutual information, and total correlation (multivariate mutual information) [11]. Such tools have proven useful in exploring and understanding datasets and the relationships between sets of variables. Unfortunately, these tools are either limited to only pairwise interactions or do not scale well computationally as the number of possible HOIs grows as $O(2^p)$ where p is the number of variables.

In response to these challenges we present Local CorEx, an unsupervised method for learning potential HOIs based on the correlation structure of data. Local CorEx is to the best of our knowledge the first method for learning locally varying HOIs from tabular data without relying on a pre-built graph or hypergraph. Local CorEx is built on the principle of Cor-relation Ex-planation (CorEx), which was introduced to construct informative representations that provide valuable information about relationships between variables in high-dimensional data. A particular variant, called Linear CorEx, estimates multivariate Gaussian distributions by identifying independent latent factors that explain correlations among observed variables [12]. It incorporates a modular inductive prior, favoring models where the covariance matrix is block-diagonal, indicating clusters governed by a few latent factors. Another variant is called Bio CorEx, which focuses primarily on handling challenges inherent in several biomedical problems: missing data, continuous variables, and severely under-sampled data [13].

While both Linear and Bio CorEx have been used successfully, they fail to consider that variable interactions may vary across the data manifold. For example, in a classification problem, interactions are often not shared across classes because the data lie on separate data manifolds. Thus Linear and Bio CorEx may only find the interactions that span classes while obscuring or distorting class-specific interactions. Local CorEx solves this problem by partitioning the data prior to estimating variable interactions.

Our contributions are: 1) we derive a novel method for estimating local variable interactions called Local CorEx (Section 2); 2) we show on synthetic data that Local CorEx is robust to hyperparameter selection and outperforms previous works when HOIs vary across the data manifold (Section 3.1); 3) as a demonstration we interpret the inner workings of a neural network classifier using Local CorEx and discover sets

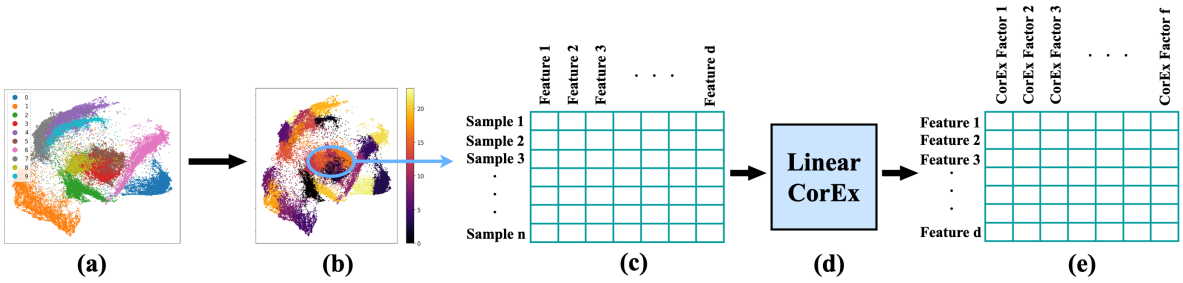

Fig. 1. Overview of the Local CorEx algorithm. *(a)* PHATE visualization of the MNIST dataset. *(b)* k-means clustering is applied to the PHATE embedding to generate the local clusters. *(c-d)* A cluster is chosen and passed through Linear CorEx. *(e)* We visualize the mutual information between the learned CorEx latent factors and the original features to identify HOIs.

of hidden nodes needed to accurately classify a local cluster while leaving accuracies for other clusters relatively unaffected (Section 3.2).

2. LOCAL COREX

Figure 1 gives an overview of Local CorEx. Local CorEx first uses PHATE [14] to create a low-dimensional embedding of the data that preserves the local and global structure of the data manifold(s). PHATE is a dimensionality reduction technique that is specialized for data visualization [14]. It learns the local structure via a specialized kernel function. The pairwise kernel matrix is row-normalized to create a Markov transition matrix, called the diffusion operator [15]. The global structure of the data is learned via diffusion by simulating t-step random walks between data points according to the normalized affinities. Finally, information distances are calculated between the diffused probabilities, and multidimensional scaling (MDS) [16] is used to preserve these distances in low dimensions.

To partition the data, we apply k-means clustering to the PHATE embedding where k is chosen to be large enough to ensure heterogeneity among the partitions according to a manual inspection of a 2D PHATE plot. This is akin to spectral clustering [17], which essentially applies k-means to the Laplacian Eigenmaps embedding. We choose this clustering approach over other methods because of PHATE's impressive visualization capabilities. After clustering we can easily visualize the data using PHATE and color by clusters to see how the clusters relate to each other and if the partition makes visual sense.

After the partition has been created, we next apply Linear CorEx to each cluster to learn latent factors that correspond to local HOIs. We now describe how Linear CorEx works. Let $X \equiv X_{1:p} \equiv \{X_1, X_2, \cdots, X_p\}$ denote a vector of p observable random variables and let $Z \equiv Z_{1:m} \equiv \{Z_1, Z_2, \cdots, Z_m\}$ denote a vector of m latent random variables. Instances of X and Z are denoted in lowercase with $x = (x_1, x_2, \cdots, x_p)$ and $z = (z_1, z_2, \cdots, z_m)$, respectively. We consider several information theoretic measures including differential entropy: $H(X) = -\mathbb{E}[\log p(x)]$ ($p(x)$ is the probability density of X), mutual information: $I(X;Y) = H(X) + H(Y) - H(X,Y)$, Total Correlation: $TC(X) = \sum_{i=1}^{p} H(X_i) - H(X)$, and their conditional variants such as $H(X|Z) = \mathbb{E}_z[H(X|Z=z)]$ and $TC(X|Z) = \mathbb{E}_z[TC(X|Z=z)]$. In particular, the total correlation measures the redundancy or dependency among a set of p random variables.

Linear CorEx [12] estimates the latent factors z by optimizing a tractable lower bound for the following expression:

$$\min_{W} TC(X|Z) + TC(Z) + \sum_{i=1}^{p} Q_i, \tag{1}$$

where $z = Wx + \epsilon$, $W \in \mathbf{R}^{m \times p}$, $\epsilon \sim \mathcal{N}(0, \Sigma)$, Σ is a diagonal matrix, and the Q_i's are non-negative regularization terms that encourage modular solutions (i.e. solutions with small values of $TC(Z|X_i)$) and only equal 0 when the solution is modular. In essence, Linear CorEx attempts to identify latent factors that constitute a linear combination of the inputs, explain the total correlation in the data, remain independent of one another, and are modular. In practice, to solve equation 1, the data are used to estimate the distributions of X (assumed to be Gaussian in Linear CorEx) and stochastic gradient descent is used to optimize the weight matrix W that minimizes equation 1.

The total correlation explained by a latent factor is a measure of its importance with respect to how much of the correlation structure it explains. We can identify the predicted HOIs by examining the mutual information between each latent factor and the original set of features. Here the magnitude of the mutual information can be used as a proxy for the strength or importance of the feature to the predicted HOI.

Local CorEx hyperparameters include the number of dimensions for the PHATE embedding, clusters for k-means, CorEx latent factors, and thresholds for HOI inclusion. We suggest around 10 dimensions for the PHATE embedding unless features are scarce. If desired a more principled way would be to check the loss of the final MDS stage as a function of embedding dimension. For k-means we recommend visually inspecting a 2D PHATE plot to determine the baseline

Table 1. Ablation study results. Based on the metrics, Local CorEx outperforms the global methods when the data have mixed variable interactions. Similar results were obtained for different sample sizes in which Local CorEx also outperforms or approximately matches the global methods when interactions are not mixed for sufficiently large sample sizes.

			Cosine Dist		AUPRC	
Data	Size	Meth	α:0.0	α:1.0	α:0.0	α:1.0
Dis-joint	100	Lin	0.484	0.424	0.607	0.591
		Bio	**0.447**	0.473	0.607	0.509
		Loc	0.477	**0.190**	**0.610**	**0.894**
	1000	Lin	0.516	0.408	0.606	0.604
		Bio	**0.414**	0.474	**0.652**	0.507
		Loc	0.504	**0.146**	0.607	**0.902**
	10000	Lin	0.526	0.412	0.599	0.598
		Loc	**0.140**	**0.142**	**0.907**	**0.907**
Non-dis-joint	100	Lin	**0.207**	**0.197**	**0.883**	0.856
		Bio	0.236	0.289	0.842	0.754
		Loc	0.254	**0.197**	0.844	**0.890**
	1000	Lin	**0.191**	0.182	**0.886**	0.856
		Bio	0.203	0.276	0.869	0.773
		Loc	0.197	**0.141**	0.870	**0.908**
	10000	Lin	**0.189**	0.183	**0.887**	0.855
		Loc	0.190	**0.139**	0.883	**0.912**

number of clusters. For CorEx factors you can examine the total correlation explained by each factor and only keep factors where the contribution isn't nominal. For thresholding learned HOIs we recommend constructing a scree plot of the mutual information a factor explains to determine a threshold.

3. RESULTS

3.1. Ablation Study - Synthetic Data

To demonstrate the effectiveness of Local CorEx to learn HOIs we constructed a synthetic dataset containing two clusters where the variable interactions are known and consist of grouped pairwise interactions, since to the best of our knowledge no tabular dataset with known HOIs exists. We also compare to standard Linear and Bio CorEx. We note that we also ran a local version of Bio CorEx, but since it performed on par with Local CorEx and takes substantially longer to run we excluded it from the Table. In this ablation study, we show how all methods perform when we vary the number of chosen latent factors, the difference in interactions between partitions,

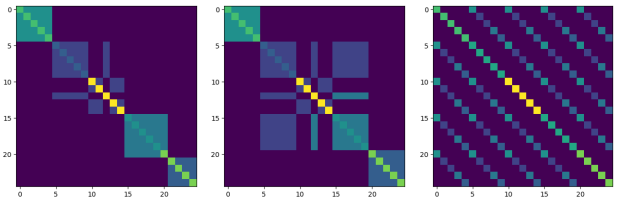

Fig. 2. The set of covariance matrices used in generating simulated data for the ablation study. The matrix on the left was the covariance matrix used in all simulations for Cluster 1. Cluster 2 uses either the middle or right matrix for its covariance matrix depending on whether we were simulating non-disjoint (middle) or disjoint (right) HOIs between the two clusters.

and the number of data samples. For each simulation setup, we ran 16 replicates.

The synthetic dataset is composed of two multivariate normal distributions following two setups. In each setup, we varied the class means between the two distributions using the parameter α to control the degree of separation ($\alpha = 0$ when clusters share the same mean and $\alpha = 1$ when cluster means are maximally separated). This controlled how close the two clusters are in proximity and as a result how pure the clusters are after partitioning. In the first setup, which we call non-disjoint, the two clusters share the majority of their interactions (Left and middle covariance matrices in Figure 2). In the second setup, which we label as disjoint, none of the interactions match (Left and right covariance matrices in Figure 2). The interactions are identified by extracting each row from the covariance matrix and then examining all of the nonzero values in the row. These non-zero elements in each are grouped together as a HOI. To find the set of interactions present in the data we collect all of the unique rows and store them in a set.

To find each learned HOI we first ran either Linear CorEx or Bio CorEx on all of the simulation data points or Local CorEx on a cluster of datapoints (as shown in Figure 1), where the number of clusters was chosen to be 2. Then, for each CorEx latent factor, we examined the mutual information between the learned latent factor and the original covariates in the data. If a variable is highly associated with the latent factor it will share a high mutual information value with the latent factor, and if not, it will share a low mutual information value with the latent factor. To score how well the predicted HOI matches the ground truth interactions, we compute the following two metrics between a learned HOI and ground truth HOI pair: 1) the cosine distance and 2) the area under the precision-recall curve. For each metric, we take each ground truth HOI and find the learned latent factor that matches it best with respect to the metric being used. These scores are averaged for all ground truth HOIs. Additionally, because the effect of the number of latent factors used on the scores was

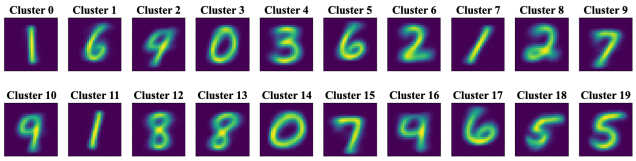

Fig. 3. Visualization of the average samples in each of the 20 clusters for the MNIST test dataset used in Section 3.2.

minimal, we further averaged metrics across different numbers of latent factors. The scores for a sample size of 100, 1000, and 10000 per cluster are shown in Table 1.

From Table 1, we observe that global methods, such as Linear CorEx and Bio CorEx, experience a decline in performance when data with mixed variable interactions (disjoint) are introduced. In contrast, Local CorEx, a local method, remains unaffected in this setting when the data is easily clustered ($\alpha = 1$). However, when the data is not easily clustered ($\alpha = 0$), the differences in scores between Local CorEx and the global methods are negligible, regardless of whether the variable interactions are disjoint or non-disjoint. Additionally, when the variable interactions are mostly consistent (non-disjoint), we find that Local CorEx still outperforms the global methods when the data are easily clustered, although by smaller margins compared to the disjoint setup.

3.2. Neural Network Model Interpretability

For our final case study, we use Local CorEx to explore the hidden representations and model weights for a neural network classifier trained on the MNIST dataset. All explorations are conducted on the test data which is partitioned into 20 clusters using k-means clustering on a 10-dimensional PHATE embedding of the data. We focus on cluster 16 for brevity and due to the heterogeneity of the class labels between 9s and 4s (see Figure 3 for a visualization of the average sample of each cluster).

Here we use Local CorEx to identify groups of hidden nodes and explore their impact on the model in two ways: 1) We used the first two layers of the classifier as an encoder with frozen weights and then trained a decoder that takes the output of the encoder as its input. This allows us to reconstruct hidden state representations back to inputs. By taking the average hidden state representation and perturbing a group of nodes in proportion to their mutual information with the hidden nodes, we can map the perturbed hidden state representation to the input space in an attempt to visualize the information encoded in the grouped nodes. 2) We delete groups of hidden nodes from the classifier model and compare the reduced classifier's accuracies to the unaltered model's accuracies across clusters to see if the impact is global or local. Local CorEx is used in both analyses to select which groups of nodes to perturb or delete.

We first apply Local CorEx to the concatenation of a hidden layer output of the classifier with their respective logits. The logits are included to aid with interpretability and are excluded when determining which nodes to perturb or delete. Using this we can find sets of related hidden nodes and get an intuition for which class they're associated with by using the logits.

We demonstrate our two methods of analysis for the first two Local CorEx factors of cluster 16 as shown in part (a) of Figure 4 and parts (a) and (c) in Figure 5. From examining these figures we can see that the first Local CorEx factor is associated with the model predicting a 4 and perturbing the values of the hidden nodes associated with the first latent factor alters the curvature of the digit in the top left portion of the number changing a four to look more like a nine. When we delete the 50 nodes with the highest mutual information associated with this factor and recompute the accuracies across clusters, it has a large impact on classification accuracy for clusters 2, 10, and 16, as shown in Figure 5. All of these clusters have large quantities of 4s and 9s present as shown in Figure 3. Remarkably, almost all other clusters are relatively unaffected by this despite deleting 25% of the hidden nodes in the first layer.

When we repeat this analysis on the second Local CorEx factor we see that this factor is associated with the probability of being classified as a zero, two, or a six. Perturbing the hidden nodes associated with the second Local CorEx factor alters the pixels on the base of the four, the height of the base of the horizontal line in the center, and density of the top left arm of the four. When 25% of the nodes associated with this factor are deleted the effect on accuracy is more widespread. The clusters most impacted are 1, 2, 5, 10, 16, and 17. Clusters 2, 10, and 16 contain 4s and 9s so it isn't surprising that removing a feature calculated on a group of 4s and 9s affects their classification. Groups 1, 5, and 17 are composed almost entirely of 6s. If you were to mask out the middle portion of a six it could look like either an unfinished 0 or 6.

Moving on to the second hidden layer we repeat the same analysis. When we examine the first Local CorEx factor we see that we get largely the same results as seen for the first Local CorEx factor for the first hidden state representation. Perturbing these hidden nodes and reconstructing them seems to accentuate the change that we saw in the first hidden layer. We also see similar effects when deleting the 80 nodes (40% of the nodes in the hidden layer) most associated with the factor. It is surprising that we can delete almost half of the hidden nodes in a layer and only affect the classification accuracy of a local partition of our data which accounts for two out of ten classes.

We see this pattern continued when we examine the second Local CorEx factor in the second hidden layer and see it also largely follows what we saw from the second Local CorEx factor in the first hidden layer. Perturbing these nodes seems to accentuate the change seen from the first layer with the raising

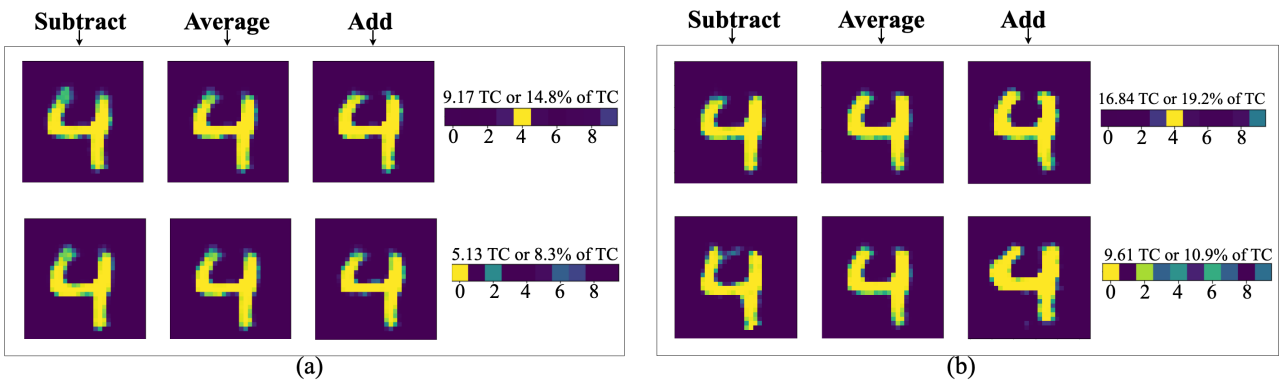

Fig. 4. Visualizing the effect of perturbing the average neural network hidden state representations of cluster 16 in the MNIST test dataset. *(a)* The plots are associated with perturbing the first hidden layer (H1) representation. This first row is associated with the first Local CorEx factor and the second row is associated with the second Local CorEx factor. *(b)* Same as in *(a)* but for the second hidden layer (H2). For each group of plots, the leftmost column image is generated by subtracting the mutual information between the Local CorEx factor and the hidden nodes from the average representation. The second column image gives the average hidden state representation. The third column image is generated by adding the mutual information between the Local CorEx factor and the hidden nodes from the average representation. Finally, the rightmost column plots the mutual information between the Local CorEx factor and the model logits. This analysis gives us a visual intuition for what role the grouped hidden nodes play.

of the middle bar, changing the base of the four, and changing the angle or density of the top left arm of the four. This along with what was seen with the first Local CorEx factors suggests that the correlation structure from one hidden layer to another is preserved between layers. However, deleting the top 50 nodes associated with the second Local CorEx factor on the second hidden state representation seems to have far less of an effect on classification accuracy than the second Local CorEx factor on the first hidden state representation. The clusters most affected though are 1, 2, 10, and 16, which contain mostly digits with a flat horizontal line in the center of the digit.

4. CONCLUSION

We have shown that Local CorEx is superior to global methods at capturing variable interactions when these interactions vary across the data manifold. In Table 1, we demonstrated that even when interactions largely overlap, Local CorEx outperforms global methods with mostly pure clusters when the sample size exceeds 100. Furthermore, despite impure clusters, Local CorEx performs comparably to global methods once the sample size reaches 1000. The more the variable interactions vary, the greater the benefit of using Local CorEx over other global methods for extracting HOIs in the data.

We then effectively used Local CorEx to explore several different data types to extract meaningful variable interactions, including tabular synthetic data and internal neural network representations. With the neural network, we used Local CorEx in an unsupervised manner to determine the hidden

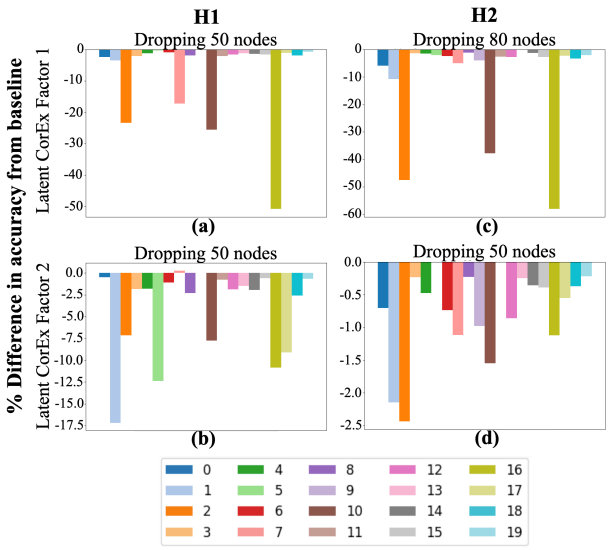

Fig. 5. *(a)* Difference in classification accuracy between the unaltered model and the model after deleting 50 hidden nodes in the first hidden layer with the highest mutual information with the first Local CorEx factor associated with cluster 16. (b) Same as *(a)* except when using the second CorEx factor to determine the 50 hidden nodes to delete. (c) Same as in *(a)* except when deleting 80 hidden nodes in the second hidden layer. (d) Same as in *(b)* except in the second hidden layer. Note that the y-axis scale differs for each plot.

nodes the model leveraged for predicting a specific class and showed that by removing them that only the main class was affected. This demonstrates that despite the interconnectedness of neural networks we can isolate clusters of nodes that perform a particular task of interest.

We believe that this approach can be used to further explore and interpret the inner workings of neural networks. For example, Local CorEx could be used to study robustness in neural networks by identifying features the network associates with a class and use them to create adversarial data points to improve model performance in a manner conceptually similar to work done by [18, 19]. Additionally, Local CorEx can be used as part of exploratory data analysis to detect variable interactions in different regions of the data manifold such as in ecological systems [6, 7], collaborations [5], the human brain [3, 4], and any network-based data including biological networks. For future work, we believe this method can be further adapted as a visualization tool to aid in summarizing complex datasets.

5. REFERENCES

[1] Y. Zhang, M. Lucas, and F. Battiston, "Higher-order interactions shape collective dynamics differently in hypergraphs and simplicial complexes," *Nature Communications*, vol. 14, no. 1, pp. 1605, 2023.

[2] S. Boccaletti, P. De Lellis, CI del Genio, K. Alfaro-Bittner, R. Criado, S. Jalan, and M. Romance, "The structure and dynamics of networks with higher order interactions," *Physics Reports*, vol. 1018, pp. 1–64, 2023.

[3] G. Petri, P. Expert, F. Turkheimer, R. Carhart-Harris, D. Nutt, P.J. Hellyer, and F. Vaccarino, "Homological scaffolds of brain functional networks," *J. R. Soc. Interface*, vol. 11, no. 101, pp. 20140873, 2014.

[4] C. Giusti, R. Ghrist, and D.S. Bassett, "Two's company, three (or more) is a simplex: Algebraic-topological tools for understanding higher-order structure in neural data," *J. Comput. Neurosci.*, vol. 41, pp. 1–14, 2016.

[5] A. Civilini, O. Sadekar, F. Battiston, J. Gómez-Gardeñes, and V. Latora, "Explosive cooperation in social dilemmas on higher-order networks," *arXiv preprint arXiv:2303.11475*, 2023.

[6] J. Grilli, G. Barabás, M.J. Michalska-Smith, and S. Allesina, "Higher-order interactions stabilize dynamics in competitive network models," *Nature*, vol. 548, no. 7666, pp. 210–213, 2017.

[7] A.R. Kleinhesselink, N.J. Kraft, S.W. Pacala, and J.M. Levine, "Detecting and interpreting higher-order interactions in ecological communities," *Ecology Letters*, vol. 25, no. 7, pp. 1604–1617, 2022.

[8] Federico Musciotto, Federico Battiston, and Rosario N Mantegna, "Detecting informative higher-order interactions in statistically validated hypergraphs," *Communications Physics*, vol. 4, no. 1, pp. 218, 2021.

[9] Mehmet Emin Aktas, Thu Nguyen, Sidra Jawaid, Rakin Riza, and Esra Akbas, "Identifying critical higher-order interactions in complex networks," *Scientific reports*, vol. 11, no. 1, pp. 21288, 2021.

[10] Federico Musciotto, Federico Battiston, and Rosario N Mantegna, "Identifying maximal sets of significantly interacting nodes in higher-order networks," *arXiv preprint arXiv:2209.12712*, 2022.

[11] K. Bai, P. Cheng, W. Hao, R. Henao, and L. Carin, "Estimating total correlation with mutual information estimators," in *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent, Eds. 25–27 Apr 2023, vol. 206 of *Proceedings of Machine Learning Research*, pp. 2147–2164, PMLR.

[12] G. Ver Steeg, H. Harutyunyan, D. Moyer, and A. Galstyan, "Fast structure learning with modular regularization," in *NeurIPS*, 2019, vol. 32.

[13] S. Pepke and G. Ver Steeg, "Comprehensive discovery of subsample gene expression components by information explanation: therapeutic implications in cancer," *BMC Medical Genomics*, vol. 10, no. 1, pp. 12, Mar 2017.

[14] K.R. Moon, D. van Dijk, Z. Wang, S. Gigante, D.B. Burkhardt, W.S. Chen, K. Yim, A. Elzen, M.J. Hirn, R.R. Coifman, N.B. Ivanova, G. Wolf, and S. Krishnaswamy, "Visualizing structure and transitions in high-dimensional biological data," *Nature Biotechnology*, vol. 37, no. 12, pp. 1482–1492, Dec 2019.

[15] R.R. Coifman and S. Lafon, "Diffusion maps," *Appl. Comput. Harmon.*, vol. 21, no. 1, pp. 5–30, 2006.

[16] J.B. Kruskal, "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis," *Psychometrika*, vol. 29, no. 1, pp. 1–27, Mar 1964.

[17] D. Uminsky, M. Banuelos, L. González-Albino, R. Garza, and S.A. Nwakanma, "Detecting higher order genomic variant interactions with spectral analysis," in *Eur. Signal Process. Conf.*, 2019, pp. 1–5.

[18] S. Casper, M. Nadeau, D. Hadfield-Menell, and G. Kreiman, "Robust feature-level adversaries are interpretability tools," in *NeurIPS*, 2022, vol. 35, pp. 33093–33106.

[19] S. Casper, K. Hariharan, and D. Hadfield-Menell, "Diagnostics for deep neural networks with automated copy/paste attacks," *arXiv:2211.10024*, 2023.