Quantifying the alignment of graph and features in deep learning

Yifan Qian, Paul Expert, Tom Rieu, Pietro Panzarasa, Mauricio Barahona

***Abstract*—We show that the classification performance of Graph Convolutional Networks is related to the alignment between features, graph and ground truth, which we quantify using a subspace alignment measure corresponding to the Frobenius norm of the matrix of pairwise chordal distances between three subspaces associated with features, graph and ground truth. The proposed measure is based on the principal angles between subspaces and has both spectral and geometrical interpretations. We showcase the relationship between the subspace alignment measure and the classification performance through the study of limiting cases of Graph Convolutional Networks as well as systematic randomizations of both features and graph structure applied to a constructive example and several examples of citation networks of different origin. The analysis also reveals the relative importance of the graph and features for classification purposes.**

***Index Terms*—Deep learning, Graph Convolutional Networks, Data alignment, Graph subspaces, Principal angles**

I. Introduction

Deep learning encompasses a broad class of machine learning methods that use multiple layers of nonlinear processing units in order to learn multi-level representations for detection or classification tasks [1, 2, 3, 4, 5]. The main realizations of deep multi-layer architectures are the so-called Deep Neural Networks (DNNs), which correspond to Artificial Neural Networks (ANNs) with multiple layers between input and output layers. DNNs have been shown to perform successfully in processing a variety of signals with an underlying Euclidean or grid-like structure, such as speech, images and videos. Signals with an underlying Euclidean structure usually come in the form of multiple arrays [1] and are known for their statistical properties such as locality, stationarity and hierarchical compositionality from local statistics [6, 7]. For instance, an image can be seen as a function on Euclidean space (the 2D plane) sampled from a grid. In this setting, locality is a consequence of local connections, stationarity results from shift-invariance, and compositionality stems from the intrinsic multi-resolution structure of many images [4]. It has been suggested that such statistical properties can be exploited by convolutional architectures via DNNs, namely (deep) Convolutional Neural Networks (CNNs) [8, 9, 10] which are based on four main ideas: local connections, shared weights, pooling, and multiple layers [1]. The role of the convolutional layer in a typical CNN architecture is to detect local features from the previous layer that are shared across the image domain, thus largely reducing the parameters compared with traditional fully connected feed-forward ANNs.

Although deep learning models, and in particular CNNs, have achieved highly improved performance on data characterized by an underlying Euclidean structure, many real-world data sets do not have a natural and direct connection with a Euclidean space. Recently there has been interest in extending deep learning techniques to non-Euclidean domains, such as graphs and manifolds [4]. An archetypal example is social networks, which can be represented as graphs with users as nodes and edges representing social ties between them. In biology, gene regulatory networks represent relationships between genes encoding proteins that can up- or down-regulate the expression of other genes. In this paper, we illustrate our results through examples stemming from another kind of relational data with no discernible Euclidean structure, yet with a clear graph formulation, namely citation networks, where nodes represent documents and an edge is established if one document cites the other [11].

To address the challenge of extending deep learning techniques to graph-structured data, a new class of deep learning algorithms, broadly named Graph Neural Networks (GNNs), has been recently proposed [4, 12, 13]. In this setting, each node of the graph represents a sample, which is described by a feature vector, and we are additionally provided with relational information between the samples that can be formalized as a graph. GNNs are well suited to node (i.e., sample) classification tasks. For a recent survey of this fast-growing field, see [14].

Generalizing convolutions to non-Euclidean domains is not straightforward [15]. Recently, Graph Convolutional Networks (GCNs) have been proposed [16] as a subclass of GNNs with convolutional properties. The GCN architecture combines the full relational information from the graph together with the node features to accomplish the classification task, using the ground truth class assignment of a small subset of nodes during the training phase. GCNs have shown improved performance for semi-supervised classification of documents (described by their text) into topic areas, outperforming methods that rely exclusively on text information without the use of any citation information, e.g., multilayer perceptron (MLP) [16].

Y. Qian and P. Panzarasa are with the School of Business and Management, Queen Mary University of London, London E1 4NS, UK (e-mail: y.qian@qmul.ac.uk, p.panzarasa@qmul.ac.uk).

P. Expert is with the School of Public Health, Imperial College London, London SW7 2AZ, UK (e-mail: paul.expert08@imperial.ac.uk).

T. Rieu was with the Department of Mathematics, Imperial College London, and is now with Criteo, Paris, France (e-mail: t.rieu@criteo.com).

M. Barahona is with the Department of Mathematics, Imperial College London, London SW7 2AZ, UK (e-mail: m.barahona@imperial.ac.uk).

However, we would not expect such an improvement to be universal. In some cases, the additional information provided by the graph (i.e., the edges) might not be consistent with the similarities between the features of the nodes. In particular, in the case of citation graphs, it is not always the case that documents cite other documents that are similar in content. As we will show below with some illustrative data sets, in those cases the conflicting information provided by the graph means that a graph-less MLP approach outperforms GCN. Here, we explore the relative importance of the graph with respect to the features for classification purposes, and propose a geometric measure based on subspace alignment to explain the relative performance of GCN against different limiting cases.

Our hypothesis is that a degree of alignment among the three layers of information available (i.e., the features, the graph and the ground truth) is needed for GCN to perform well, and that any degradation in the information content leads to an increased misalignment of the layers and worsened performance. We will first use randomization schemes to show that the systematic degradation of the information contained in the graph and the features leads to a progressive worsening of GCN performance. Second, we propose a simple spectral alignment measure, and show that this measure correlates with the classification performance in a number of data sets: (i) a constructive example built to illustrate our work; (ii) CORA, a well-known citation network benchmark; (iii) AMiner, a newly constructed citation network data set; and (iv) two subsets of Wikipedia: Wikipedia I, where GCN outperforms MLP, and Wikipedia II, where instead MLP outperforms GCN.

II. Related work

A. Neural networks on graphs

The first attempt to generalize neural networks on graphs can be traced back to Gori *et al.* (2005) [17], who proposed a scheme combining recurrent neural networks (RNNs) and random walk models. Their method requires the repeated application of contraction maps as propagation functions until the node representations reach a stable fixed point. This method, however, did not attract much attention when it was proposed. With the current surge of interest in deep learning, this work has been reappraised in a new and modern form: Ref. [18] introduced modern techniques for RNN training based on the original graph neural network framework, whereas Ref. [19] proposed a convolution-like propagation rule on graphs and methods for graph-level classification. Non-spectral methods have also been successfully proposed. For example, Ref. [20] shows how diffusion-based representations can be learned from graph-structured data and used as the basis for node classification by introducing a diffusion-convolution operation. Ref. [21] converts graphs locally into sequences fed into a conventional 1D CNN, which needs the definition of a node ordering in a pre-processing step.

The first formulation of convolutional neural networks on graphs (GCNNs) was proposed by Bruna *et al.* (2014) [22]. These researchers applied the definition of convolutions to the spectral domain of the graph Laplacian. While being theoretically salient, this method is unfortunately impractical due to its computational complexity. This drawback was addressed by subsequent studies [15]. In particular, Ref. [15] leveraged fast localized convolutions with Chebyshev polynomials. In [16], a GCN architecture was proposed via a first-order approximation of localized spectral filters on graphs. In that work, Kipf and Welling considered the task of semi-supervised transductive node classification where labels are only available for a small number of nodes. Starting with a feature matrix X and a network adjacency matrix A, they encoded the graph structure directly using a neural network model $f(X, A)$, and trained on a supervised target loss function $\mathcal{L}$ computed over the subset of nodes with known labels. Their proposed GCN was shown to achieve improved accuracy in classification tasks on several benchmark citation networks and on a knowledge graph data set. In our study, we examine how the properties of features and the graph interact in the model proposed by Kipf and Welling for semi-supervised transductive node classification in citation networks. The architecture and propagation rules of this method are detailed in Section II-C.

B. Spectral graph convolutions

We now present briefly the key insights introduced by Bruna *et al.* [22] to extend CNNs to the non-Euclidean domain. For an extensive recent review, the reader should refer to [4].

We study GCNs in the context of a classification task for N samples. Each sample is described by a C^0-dimensional feature vector, which is conveniently arranged into the feature matrix $X \in R^{N \times C^0}$. Each sample is also associated with the node of a given graph $\mathcal{G}$ with N nodes, with edges representing additional relational (symmetric) information. This undirected graph is described by the adjacency matrix $A \in R^{N \times N}$. The ground truth assignment of each node to one of F classes is encoded into a 0-1 membership matrix $Y \in R^{N \times F}$.

The main hurdle is the definition of a convolution operation on a graph between a filter g_w and the node features X. This can be achieved by expressing g_w onto a basis encoding information about the graph, e.g., the adjacency matrix A or the Laplacian $L = D - A$, where $D = \mathrm{diag}(A\mathbf{1})$. This real symmetric matrix has an eigendecomposition $L = U\Lambda U'$, where U is the matrix of column eigenvectors with associated eigenvalues collected in the diagonal matrix Λ. The filters can then be expressed in the eigenbasis U of L:

$$g_w = U g_w(\Lambda) U^T, \tag{1}$$

with the convolution between filter and signal given by:

$$g_w \star X = U g_w(\Lambda) U^T X. \tag{2}$$

The signal is thus projected onto the space of the graph, filtered in the frequency domain, and projected back onto the nodes.

C. Graph Convolutional Networks

A GCN is a semi-supervised method, in which a small subset of the node ground truth labels are used in the training phase to infer the class of unlabeled nodes. This type of learning

paradigm, where only a small amount of labeled data is available, therefore lies between supervised and unsupervised learning.

Furthermore, the model architecture, and thus the learning, depend *explicitly* on the structure of the network. Hence the addition of any new data point (i.e., a new node in the network) will require a retraining of the model. GCNs are therefore an example of a transductive learning paradigm, where the classifier cannot be generalized to data it has not already seen. Node classification using a GCN can be seen as a label propagation task: given a set of seed nodes with known labels, the task is to predict which label will be assigned to the unlabeled nodes given a certain topology and attributes.

Layer-wise propagation rule and multi-layer architecture: Our study uses the multi-layer GCN proposed in [16]. Given the matrix X with sample features and the (undirected) adjacency matrix A of the graph $\mathcal{G}$ encoding relational information between the samples, the propagation rule between layers ℓ and $\ell+1$ (of size C^ℓ and $C^{\ell+1}$, respectively) is given by:

$$H^{\ell+1} = \sigma^\ell\left(\widehat{A}H^\ell W^\ell\right), \tag{3}$$

where $H^\ell \in R^{N\times C^\ell}$ and $H^{\ell+1} \in R^{N\times C^{\ell+1}}$ are matrices of activation in the ℓ^{th} and $(\ell+1)^{th}$ layers, respectively; $\sigma^\ell(\cdot)$ is the threshold activation function for layer ℓ; and the weights connecting layers ℓ and $\ell+1$ are stored in the matrix $W^\ell \in R^{C^\ell\times C^{\ell+1}}$. Note that the input layer contains the feature matrix $H^0 \equiv X$.

The graph is encoded in $\widehat{A} = \tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}$, where $\tilde{A} = A + I_N$ is the adjacency matrix of a graph with added self-loops, I_N is the identity matrix, and $\tilde{D} = \mathrm{diag}(\tilde{A}\mathbf{1})$ is a diagonal matrix containing the degrees of $\tilde{A}$. In the remainder of this work (and to ensure comparability with the results in [16]), we use $\widehat{A}$ as the descriptor of the graph $\mathcal{G}$.

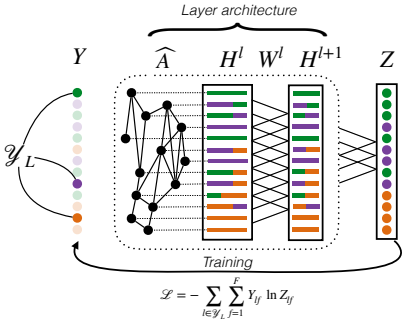


Figure 1: **Schematic illustration of the Graph Convolutional Network used.** The graph $\widehat{A}$ is applied to the input of each layer ℓ before it is funneled into the input of layer $\ell+1$. The process is repeated until the output has dimension $N\times F$ and produces a predicted class assignment. During the training phase, the predicted assignments are compared against a subset of values $\mathcal{Y}_L$ of the ground truth.

Following [16], we implement a two-layer GCN with propagation rule (3) and different activation functions for each layer, i.e., a rectified linear unit for the first layer and a softmax unit for the output layer:

$$\sigma^0 : \mathrm{ReLU}(x_i) = \max(x_i, 0) \tag{4}$$

$$\sigma^1 : \mathrm{softmax}(x)_i = \frac{\exp(x_i)}{\sum_i \exp(x_i)}, \tag{5}$$

where x is a vector. The model then takes the simple form:

$$Z = f(X, A) = \mathrm{softmax}(\widehat{A}\ \mathrm{ReLU}(\widehat{A}XW^0)\ W^1), \tag{6}$$

where the softmax function is applied row-wise and the ReLU is applied element-wise. Note there is only one hidden layer with C^1 units. Hence $W^0 \in R^{C^0\times C^1}$ maps the input with C^0 features to the hidden layer and $W^1 \in R^{C^1\times C^2}$ maps these hidden units to the output layer with $C^2 = F$ units, corresponding to the number of classes of the ground truth. In this semi-supervised multi-class classification, the cross-entropy error over all labeled instances is evaluated as follows:

$$\mathcal{L} = -\sum_{l\in\mathcal{Y}_L}\sum_{f=1}^{F} Y_{lf}\ \ln Z_{lf}, \tag{7}$$

where $\mathcal{Y}_L$ is the set of nodes that have labels. The weights of the neural network (W^0 and W^1) are trained using gradient descent to minimize the loss $\mathcal{L}$. A visual summary of the GCN architecture is shown in Fig. 1. The reader is referred to [16] for details and in-depth analysis. Although GCN was introduced as a simplified form of spectral-based GNNs, it shows a natural connection with spatial-based GNNs, in which graph convolutions are defined by information propagation. Hence our study of the alignment of graph and features is related more widely to graph-feature correlations, such as spatial autocorrelations measured with, e,g,, Moran's and Geary's indices [23, 24] that capture how the features of nodes influence each other via network structure.

III. Methods

A. Randomization strategies

To test the hypothesis that a degree of alignment across information layers is crucial for a good classification performance of GCN, we gradually randomize the node features, the node connectivity, or both. For the randomization to give a meaningful notion of alignment, at least one ingredient needs to be kept constant. Since we focus on the alignment of graph and features, we keep the ground truth constant.

1) Randomization of the graph: The edges of the graph are randomized by rewiring a percentage $p_{\widehat{A}}$ of edge stubs (i.e., 'half-edges') under the constraint that the degree distribution remains unchanged. This randomization strategy is described in Algorithm 1 which is based on the configuration model [25]. Once a randomized realization of the graph is produced, the corresponding $\widehat{A}$ is computed.

Algorithm 1: Randomization of the graph

Input: A graph $G(V, E)$, where V is the set of nodes and E is the set of edges, and a randomization percentage $0 \leq p_{\widehat{A}} \leq 100$.

Output: A randomized graph $G_{p_{\widehat{A}}}(V, E')$

1.Choose a random subset of edges E_r from E with $|E_r| = \lfloor |E| \times p_{\widehat{A}}/100 \rfloor$, and denote the unrandomized edges in E as E_u.

2. Obtain the degree sequence of nodes from E_r, and build a stub list l_s based on the degree sequence.

3. Obtain a randomized stub list l'_s by shuffling l_s, and randomized edges E'_r by connecting the stubs in the corresponding positions of the two stub lists l_s and l'_s.

4. Compute $E_u \cup E'_r$, remove multiedges and self-loops, and obtain the final edge set E'.

5. Generate randomized graph $G_{p_{\widehat{A}}}(V, E')$ from node set V and edge set E'.

Algorithm 2: Randomization of the features

Input: A feature matrix $X \in R^{N \times C^0}$, and a randomization percentage $0 \leq p_X \leq 100$.

Output: A randomized feature matrix $X_{p_X} \in R^{N \times C^0}$

1. Choose at random N_r rows from X, where $N_r = \lfloor N\, p_X/100 \rfloor$.

2. Swap randomly the N_r rows to obtain X_{p_X}.

2) Randomization of the features: The features were randomized by swapping feature vectors between a percentage p_X of randomly chosen nodes following the procedure described in Algorithm 2.

A fundamental difference between the two randomization schemes is that the graph randomization alters its spectral properties as it gradually destroys the graph structure, whereas the randomization of the features preserves its spectral properties in the principal component analysis (PCA) sense, i.e., the principal values are the same but the loadings on the components are swapped. Hence the feature randomization still alters the classification performance because the features are re-assigned to nodes that have a different environment, thereby changing the result of the convolution operation defined by the H^ℓ activation matrices (3).

B. Limiting cases

To interrogate the role that the graph plays in the classification performance of a GCN, it is instructive to consider three limiting cases:

- *No graph:* $A = \mathbf{0}\mathbf{0}^T$. If we remove all the edges in the graph, the classifier becomes equivalent to an MLP, a classic feed-forward ANN. The classification is based solely on the information contained in the features, as no graph structure is present to guide the label propagation.
- *Complete graph:* $A = \mathbf{1}\mathbf{1}^T - I_N$. In this case, the mixing of features is immediate and homogeneous, corresponding to a mean field approximation of the information contained in the features.
- *No features:* $X = I_N$. In this case, the label propagation and assignment are purely based on graph topology.

An illustration of these limiting cases can be found in the top row of Table II.

C. Spectral alignment measure

In order to quantify the alignment between the features, the graph and the ground truth, we propose a measure based on the chordal distance between subspaces, as follows.

1) Chordal distance between two subspaces: Recent work by Ye and Lim [26] has shown that the distance between two subspaces of different dimension in $\mathbb{R}^n$ is necessarily defined in terms of their principal angles.

Let $\mathcal{A}$ and $\mathcal{B}$ be two subspaces of the ambient space $\mathbb{R}^n$ with dimensions α and β, respectively, with $\alpha \leq \beta < n$. The principal angles between $\mathcal{A}$ and $\mathcal{B}$ denoted $0 \leq \theta_1 \leq \theta_2 \leq ... \leq \theta_\alpha \leq \frac{\pi}{2}$ are defined recursively as follows [27, 28]:

$$\theta_1 = \min_{a_1 \in \mathcal{A}, b_1 \in \mathcal{B}} \arccos\left(\frac{|a_1^T b_1|}{\|a_1\|\|b_1\|}\right),$$

$$\theta_j = \min_{\substack{a_j \in \mathcal{A}, b_j \in \mathcal{B} \\ a_j \perp a_1, ..., a_{j-1} \\ b_j \perp b_1, ..., b_{j-1}}} \arccos\left(\frac{|a_j^T b_j|}{\|a_j\|\|b_j\|}\right), \quad j = 2, ..., \alpha,$$

If the *minimal* principal angle is small, then the two subspaces are nearly linearly dependent, i.e., almost perfectly aligned. A numerically stable algorithm that computes the canonical correlations, (i.e., the cosine of the principal angles) between subspaces is given in Algorithm 3.

Algorithm 3: Principal angles [27, 28]

Input: matrices $A_{n\times\alpha}$ and $B_{n\times\beta}$ with $\alpha \leq \beta < n$.

Output: cosines of the principal angles $\theta_1 \leq \theta_2 \leq ... \leq \theta_\alpha$ between $\mathcal{R}(A)$ and $\mathcal{R}(B)$, the column spaces of A and B.

1. Find orthonormal bases $\mathcal{Q}_A$ and $\mathcal{Q}_B$ for A and B using the QR decomposition: $\mathcal{Q}_A^T \mathcal{Q}_A = \mathcal{Q}_B^T \mathcal{Q}_B = I$; $\mathcal{R}(\mathcal{Q}_A) = \mathcal{R}(A)$, $\mathcal{R}(\mathcal{Q}_B) = \mathcal{R}(B)$.

2. Compute the singular value decomposition (SVD): $\mathcal{Q}_A^T \mathcal{Q}_B$ = UCV^T.

3. Extract the diagonal elements of C: $C_{ii} = \cos\theta_i$, to obtain the canonical correlations $\{\cos\theta_1, ..., \cos\theta_\alpha\}$.

The principal angles are the basic ingredient of a number of well defined Grassmannian distances between subspaces [26]. Here we use the chordal distance given by:

$$d(\mathcal{A}, \mathcal{B}) = \sqrt{\sum_{j=1}^{\alpha} \sin^2\theta_j}. \tag{8}$$

The larger the chordal distance $d(\mathcal{A},\mathcal{B})$ is, the worse the alignment between the subspaces $\mathcal{A}$ and $\mathcal{B}$.

We remark that the last inequality in $\alpha \leq \beta < n$ is strict. If a subspace spans the whole ambient space (i.e., $\beta = n$), then its distance to all other strict subspaces of $\mathbb{R}^n$ is trivially zero, as it is always possible to find a rotation that aligns the strict subspace with the whole space.

2) Alignment metric: Our task involves establishing the alignment between *three* subspaces associated with the features X, the graph $\widehat{A}$, and the ground truth Y. To do so, we consider the distance matrix containing all the pairwise chordal distances:

$$D(X,\widehat{A},Y) = \begin{bmatrix} 0 & d(X,\widehat{A}) & d(X,Y) \\ d(X,\widehat{A}) & 0 & d(\widehat{A},Y) \\ d(X,Y) & d(\widehat{A},Y) & 0 \end{bmatrix}, \quad (9)$$

and we take the Frobenius norm [28] of this matrix D as our *subspace alignment measure* (SAM):

$$\mathcal{S}(X,\widehat{A},Y) = \|D(X,\widehat{A},Y)\|_{\mathrm{F}} = \sqrt{\sum_{i=1}^{3}\sum_{j=1}^{3} D_{ij}^2}. \quad (10)$$

The larger $\|D\|_{\mathrm{F}}$ is, the worse the alignment between the three subspaces. This alignment measure has a geometric interpretation related to the area of the triangle with sides $d(X,\widehat{A}), d(X,Y), d(\widehat{A},Y)$ (blue triangle in Fig. 2).

3) Determining the dimension of the subspaces: The feature, graph and ground truth matrices $(X,\widehat{A},Y)$ are associated with subspaces of the ambient space $\mathbb{R}^N$, where N is the number of nodes (or samples). These subspaces are spanned by: the eigenvectors of $\widehat{A}$, the principal components of the feature matrix X, and the principal components of the ground truth matrix Y, respectively [29]. The dimension of the graph subspace is N; the dimension of the feature subspace is the number of features $C^0 < N$ (in our examples); and the dimension of the ground truth subspace is the number of classes $F < C^0 < N$.

The pairwise chordal distances D_{ij} in (9) are computed from a number of minimal angles, corresponding to the smaller of the two dimensions of the subspaces being compared. Hence the dimensions of the subspaces $(k_X, k_{\widehat{A}}, k_Y)$ need to be defined to compute the distance matrix D. Here, we are interested in finding low dimensional subspaces of features, graph and ground truth with dimensions $(k_X^*, k_{\widehat{A}}^*, k_Y^*)$ such that they provide maximum discriminatory power between the original problem and the fully randomized (null) model. To do this, we propose the following criterion:

$$k_Y^* = F \quad (11)$$

$$(k_X^*, k_{\widehat{A}}^*) = \max_{k_X, k_{\widehat{A}}} \left(\|D(X_{100}, \widehat{A}_{100}, Y)\|_{\mathrm{F}} - \|D(X,\widehat{A},Y)\|_{\mathrm{F}} \right).$$

We choose k_Y^* equal to the number of ground truth classes since they are non-overlapping [29]. Our optimization selects k_X^* and $k_{\widehat{A}}^*$ such that the difference in alignment between the original problem with no randomization ($p_X = p_{\widehat{A}} = 0$) and an ensemble of 100 fully randomized (feature and graph, $p_X = p_{\widehat{A}} = 100$) problems is maximized (see SI for details on the optimization scheme). This criterion maximizes the range of values that $||D||_{\mathrm{F}}$ can take, thus augmenting the discriminatory power of the alignment measure when finding the alignment between *both* data sources and the ground truth, beyond what is expected purely at random. Importantly, the reduced dimension of features and graph are found simultaneously, since our objective is to quantify the alignment (or amount of shared information) contained in the three subspaces. Our criterion effectively amounts to finding the dimensions of the subspaces that maximize a difference in the surfaces of the blue and red triangles in Fig. 2.

We provide the code to compute our proposed alignment measure at https://github.com/haczqyf/gcn-data-alignment.

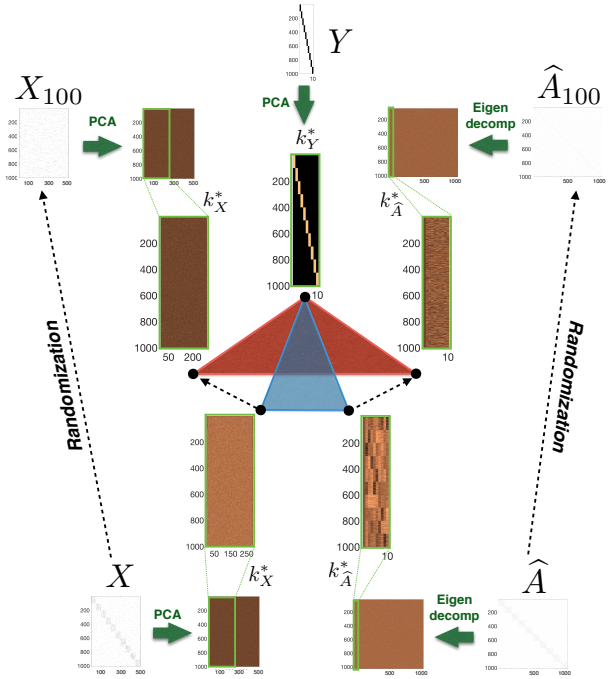


Figure 2: **Method to determine relevant subspaces (Eq. 11).** Using the constructive example, we illustrate the subspaces representing features, graph and ground truth. The feature and ground truth matrices are decomposed via PCA and the graph matrix is similarly eigendecomposed. Fixing $k_Y^* = F$, we optimize (11) to find the dimensions k_X^* and $k_{\widehat{A}}^*$ that maximize the difference between the area of the blue triangle, which reflects the alignment of the three subspaces $(X,\widehat{A},Y)$ of the original data, and the area of the red triangle, which corresponds to the alignment of the subspaces $(X_{100},\widehat{A}_{100},Y)$ of the fully randomized data. The edges of the triangles correspond to the pairwise chordal distances (e.g., the base of the blue triangle corresponds to $d(X,\widehat{A})$).

IV. Experiments

A. Data sets

Relevant statistics of the data sets, including number of nodes and edges, dimension of feature vectors, and number of ground truth classes, are reported in Table I.

Table I: **Some statistics of the data sets in our study.**

Data sets	Nodes (N)	Edges	Features (C^0)	Classes (F)
Constructive	$1,000$	$6,541$	500	10
CORA	$2,485$	$5,069$	$1,433$	7
AMiner	$2,072$	$4,299$	500	7
Wikipedia	$20,525$	$215,056$	100	12
Wikipedia I	$2,414$	$8,163$	100	5
Wikipedia II	$1,858$	$8,444$	100	5

1) Constructive example: To illustrate the alignment measure in a controlled setting, we build a constructive example, consisting of $1,000$ nodes assigned to 10 planted communities $C_1, ..., C_{10}$ of equal size. We then generate both a feature matrix and a graph matrix whose structures are aligned with the ground truth assignment matrix. The graph structure is generated using a stochastic block model that reproduces the ground truth structure with some noise: two nodes are connected with a probability $p_{in} = 0.07$ if they belong to the same community C_i and $p_{out} = 0.007$ otherwise. The feature matrix is constructed in a similar way. The feature vectors are 500 dimensional and binary, i.e., a node either possesses a feature or it does not. Each ground truth cluster is associated with 50 features that are present with a probability of $p_{in} = 0.07$. Each node also has a probability $p_{out} = 0.007$ of possessing each feature characterizing other clusters. Using the same stochastic block structure for both features and graph ensures that they are maximally aligned with the ground truth. This constructive example is then randomized in a controlled way to detect the loss of alignment and the impact this loss of alignment has on the classification performance.

2) CORA: The CORA data set is a benchmark for classification algorithms using text and citation data[1]. Each paper is labeled as belonging to one of 7 categories (Case_Based, Genetic_Algorithms, Neural_Networks, Probabilistic_Methods, Reinforcement_Learning, Rule_Learning, and Theory), which gives the ground truth Y. The text of each paper is described by a $0/1$ vector indicating the absence/presence of words in a dictionary of $1,433$ unique words, the dimension of the feature space. The feature matrix X is made from these word vectors. We extracted the largest connected component of this citation graph (undirected) to form the graph adjacency matrix A.

3) AMiner: For additional comparisons, we produced a new data set with similar characteristics to CORA from the academic citation site AMiner. AMiner is a popular scholarly social network service for research purposes only [30], which provides an open database[2] with more than 10 data sets encompassing researchers, conferences, and publication data. Among these, the academic social network[3] is the largest one and includes information on papers, citations, authors, and scientific collaborations. In 2012 the Chinese Computer Federation (CCF) released a catalog including 10 subfields of computer science. Using the AMiner academic social network, Qian *et al.* [31] extracted $102,887$ papers published from 2010 to 2012, and mapped each paper with a unique subfield of computer science according to the publication venue. Here, we use these assigned categories as the ground truth for a classification task. Using all the papers in [31] that have both abstract and references, we created a data set of similar size to CORA. We extracted the largest connected component from the citation network of all papers in 7 subfields (Computer systems/high performance computing, Computer networks, Network/information security, Software engineering/software/programming language, Databases/data mining/information retrieval, Theoretical computer science, and Computer graphics/multimedia) from 2010 to 2011. The resulting AMiner citation network consists of $2,072$ papers with $4,299$ edges. Just as with CORA, we treat the citations as undirected edges, and obtain an adjacency matrix A. We further extracted the most frequent 500 stemmed terms from the corpus of abstracts of papers and constructed the feature matrix X for AMiner using bag-of-words.

4) Wikipedia: As a contrasting example, we produced three data sets from the English Wikipedia. The Wikipedia provides an interlinked corpus of documents (articles) in different fields, which 'cite' each other via hyperlinks. We first constructed a large corpus of articles, consisting of a mixture of popular and random pages so as to obtain a balanced data set. We retrieved the $5,000$ most accessed articles during the week before the construction of the data set (July 2017), and an additional $20,000$ documents at random using the Wikipedia built-in random function[4]. The text and subcategories of each document, together with the names of documents connected to it, were obtained using the Python library *Wikipedia*[5]. A few documents (e.g., those with no subcategories) were filtered out during this process. We constructed the citation network of the documents retrieved and extracted the largest connected component. The resulting citation network contained $20,525$ nodes and $215,056$ edges. The text content of each document was converted into a bag-of-words representation based on the 100 most frequent words. To establish the ground truth, we used 12 categories from the API (People, Geography, Culture, Society, History, Nature, Sports, Technology, Health, Religion, Mathematics, Philosophy) and assigned each document to one of them. As part of our investigation, we split this large Wikipedia data set into two smaller subsets of non-overlapping categories: Wikipedia I, consisting of Health, Mathematics, Nature, Sports, and Technology; and Wikipedia II, with the categories Culture, Geography, History, Society, and People.

All six data sets used here can be found at https://github.com/haczqyf/gcn-data-alignment/tree/master/alignment/data.

B. GCN architecture, hyperparameters and implementation

We used the GCN architecture [16] and implementation[6] provided by the authors of [16], and followed closely their experimental setup to train and test the GCN on our data sets. We used a two-layer GCN as described in Section II-C with the maximum number of training iterations (epochs) set to 400 [32], a learning rate of 0.01, and early stopping with

[1]https://linqs.soe.ucsc.edu/data
[2]https://aminer.org/data
[3]https://aminer.org/aminernetwork
[4]https://en.wikipedia.org/wiki/Wikipedia:Random
[5]https://github.com/goldsmith/Wikipedia
[6]https://github.com/tkipf/gcn

a window size of 100, i.e., training stops if the validation loss does not decrease for 100 consecutive epochs. Other hyperparameters used were: (i) dropout rate: 0.5; (ii) L2 regularization: 5×10^{-4}; and (iii) number of hidden units: 16. We initialized the weights as described in [33], and accordingly row-normalized the input feature vectors. For the training, validation and test of the GCN, we used the following split: (i) 5% of instances as training set; (ii) 10% as validation set; and (iii) the remaining 85% as test set. We used this split for all data sets with exception of the full Wikipedia data set, where we used: (i) 3.5% of instances as training set; (ii) 11.5% as validation set; and (iii) the remaining 85% as test set. This modification of the split was necessary to ensure the instances in the training set were evenly distributed across categories.

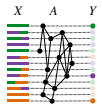
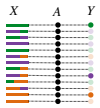
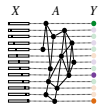
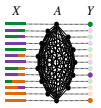
V. Results

The GCN performance is evaluated using the standard *classification accuracy* defined as the proportion of nodes correctly classified in the test set.

A. GCN: original graph vs. limiting cases

For each data set in Table I, we trained and tested a GCN with the original graph and features matrices, and GCN models under the three limiting cases described in Section III-B. We computed the average accuracy of 100 runs with random weight initializations (Table II).

Table II: **Classification accuracy of GCN and limiting cases for our data sets.** The best performance is indicated in bold. Error bars are evaluated over 100 runs. The GCN with original data performs best in most cases, but is outperformed by MLP in the full Wikipedia data set and its subset Wikipedia II.

	GCN (original)	GCN (limiting cases)		
		No graph = MLP *(Only features)* $A = \mathbf{0}\mathbf{0}^T$	No features *(Only graph)* $X = I_N$	Complete graph *(Mean field)* $A = \mathbf{1}\mathbf{1}^T - I_N$
				
Data sets				
Constructive	**0.932** ± **0.006**	0.416 ± 0.010	0.764 ± 0.009	0.100 ± 0.003
CORA	**0.811** ± **0.005**	0.548 ± 0.014	0.691 ± 0.006	0.121 ± 0.066
AMiner	**0.748** ± **0.005**	0.547 ± 0.013	0.591 ± 0.006	0.123 ± 0.045
Wikipedia	0.392 ± 0.010	**0.450** ± **0.007**	0.254 ± 0.037	O.O.M.
Wikipedia I	**0.861** ± **0.006**	0.796 ± 0.005	0.824 ± 0.003	0.163 ± 0.135
Wikipedia II	0.566 ± 0.021	**0.659** ± **0.011**	0.347 ± 0.012	0.155 ± 0.176

The GCN using all the information available in the features and the graph outperforms MLP (the no graph limit) except in the case of the large Wikipedia set. Hence using the additional information contained in the graph does not necessarily increase the performance of GCN. To investigate this issue further, we split the Wikipedia data set into two subsets: Wikipedia I, with articles in topics that tend to be more self-referential (e.g., Mathematics or Technology) and Wikipedia II, containing pages in areas that are less self-contained (e.g., Culture or Society). We observed that GCN outperforms MLP for Wikipedia I but the opposite is still true for Wikipedia II. Finally, we also observe that the performance of 'No features' is always lower than the performance of GCN, and, as expected, the performance of 'Complete graph' (i.e., mean field) is very low and close to pure chance (i.e., $\sim 1/F$).

B. Performance of GCN under randomization

The results above lead us to pose the hypothesis that a degree of synergy between features, graph and ground truth is needed for GCN to perform well. To investigate this hypothesis, we use the randomization schemes described in Section III-A to degrade systematically the information content of the graph and/or the features in our data sets. Fig. 3 presents the performance of the GCN as a function of the percent of randomization of the graph structure, the features, or both. As expected, the accuracy decreases for all data sets as the information contained in the graph, features or both is scrambled, yet with differences in the decay rate of each of the ingredients for the different examples.

Note that the chance-level performance of the 'Complete graph' (mean field) limiting case is achieved only when *both* graph and features are fully randomized, whereas the accuracy of the two other limiting cases ('No graph - MLP', 'No features') is reached around the half-point ($\sim 50\%$) of randomization of the graph or of the features, respectively. This indicates that using the scrambled information above a certain degree of randomization becomes more detrimental to the classification performance than simply ignoring it.

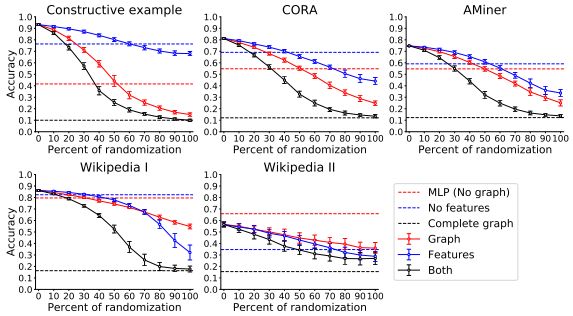


Figure 3: **Degradation of classification performance as a function of randomization.** Each panel shows the degradation of the classification accuracy as a function of the randomization of graph, features and both, for a different data set. Error bars are evaluated over 100 realizations: for zero percent randomization, we report 100 runs with random weight initializations; for the rest, we report 1 run with random weight initializations for 100 random realizations. The horizontal lines correspond to the limiting cases in Table II. The full Wikipedia data set was not analyzed here since the eigendecomposition of $\widehat{A}$ needed to obtain $k^*_X, k^*_{\widehat{A}}$ is computationally intensive.

C. Relating GCN performance and subspace alignment

We tested whether the degradation of GCN performance is linked to the increased misalignment of features, graph and ground truth given by the SAM:

$$\mathcal{S}^*(X, \widehat{A}, Y) = \|D(X, \widehat{A}, Y; k^*_X, k^*_{\widehat{A}}, k^*_Y)\|_{\mathrm{F}} \qquad (12)$$

which corresponds to (10) computed with the dimensions $(k^*_X, k^*_{\widehat{A}}, k^*_Y)$ obtained using (11) (Table III, and see SI for the optimization scheme used). Fig. 4 shows that the GCN accuracy is clearly (anti)correlated with the subspace alignment distance (12) in all our examples (mean correlation $= -0.92$).

As we randomize the graph and/or features, the subspace misalignment increases and the GCN performance decreases. In addition to the Chordal distance, Ref. [26] studies other subspace distances. While all the distances can be expressed in terms of the principal angles θ_j, some rely on all the angles whereas others only use the maximum principal angle. We obtain similar results for distances that use all the principal angles (e.g., Chordal, Grassmann), but we find that extremal distances based on the maximum principal angle (e.g., the Projection distance) do not correlate as well with GCN performance. This highlights the importance of the information captured by all principal angles to quantify the alignment between subspaces. For results based on the Grassmann and Projection distances, see SI Appendix (Section III).

Table III: **Dimensions of the three subspaces obtained according to Eq. 11 for our data sets.**

Data sets	k^*_X	$k^*_{\widehat{A}}$	k^*_Y
Constructive example	287	10	10
CORA	1,291	190	7
AMiner	500	57	7
Wikipedia I	68	1,699	5
Wikipedia II	100	1,125	5

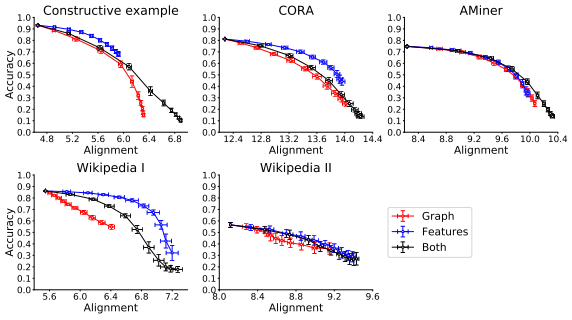


Figure 4: **Classification performance versus the subspace alignment measure (SAM).** Each panel shows the accuracy of GCN versus the SAM (12) for all the runs presented in Fig. 3. Error bars are evaluated over 100 randomizations.

VI. Discussion

Our first set of experiments (Table II) reflects the varying amount of information that GCN can extract from features, graph and their combination, for the purpose of classification. For a classifier to perform well, it is necessary to find (possibly nonlinear) combinations of features that map differentially and distinctively onto the categories of the ground truth. The larger the difference (or distance on the projected space) between the samples of each category, the easier it is to 'separate' them, and the better the classifier. In the MLP setting, for instance, the weights between layers (W^ℓ) are trained to maximize this separation. As seen by the different accuracies in the 'No graph' column (Table II), the features of each example contain variable amount of information that is mappable on its ground truth. A similar reasoning applies to classification based on graph information alone, but in this case, it is the eigenvectors of $\widehat{A}$ that need to be combined to produce distinguishing features between the categories in the ground truth (e.g., if the graph substructures across scales [34] do not map onto the separation lines of the ground truth categories, then the classification performance based on the graph will deteriorate). The accuracy in the 'No features' column indicates that some of the graphs contain more congruent information with the ground truth than others. Therefore, the 'No graph' and 'No features' limiting cases inform about the relative congruence of each type of information with respect to the ground truth. One can then conjecture that if the performance of the 'No features' case is higher than the 'No graph' case, GCN will yield better results than MLP.

In addition, our numerics show that although combining both sources of information generally leads to improved classification performance ('GCN original' column in Table II), this is *not* always necessarily the case. Indeed, for the Wikipedia and Wikipedia II examples, the classification performance of the MLP ('No graph'), which is agnostic to relationships between samples, is better than when the additional layer of relational information about the samples (i.e., the graph) is incorporated via the GCN architecture. This suggests that, for improved GCN classification, the information contained in features and graph needs to be constructively aligned with the ground truth. This phenomenon can be intuitively understood as follows. In the absence of a graph (i.e., the MLP setting), the training of the layer weights is done independently over the samples, without assuming any relationship between them. In GCN, on the other hand, the role of the graph is to guide the training of the weights by averaging the features of a node with those of its *graph neighbors*. The underlying assumption is that the relationships represented by the graph should be *consistent* with the information of their features, i.e., the features of nodes that are graph neighbors are expected to be more similar than otherwise; hence the training process is biased towards convolving the diffusing information on the graph to extract improved feature descriptions for the classifier. However, if feature similarities and graph neighborhoods (or more generally, graph communities [34]) are not congruent, this graph-based averaging during the training is not beneficial.

To explore this issue in a controlled fashion, our second set of experiments (Fig. 3) studied the degradation of the classification performance induced by the systematic randomization of graph structure and/or features. The erosion of information is not uniform across our examples, reflecting the relative salience of each of the components (features and graph) for classification. Note that the GCN is able to leverage the information present in any of the two components, and is only degraded to chance-level performance when *both* graph and features are fully randomized. Interestingly, this fully randomized (chance-level) performance coincides with that of the 'Complete graph' (or mean field) limiting case, where the classifier is trained on features averaged over all the samples, thus leading to a uniform representation that has zero discriminating power when it comes to category assignment.

These results suggest that a degree of constructive alignment between the matrices of features, graph and ground truth $(X, \widehat{A}, Y)$ is necessary for GCN to operate successfully beyond standard classifiers. To capture this idea, we proposed

a simple SAM (12) that uses the minimal principal angles to capture the consistency of pairwise projections between subspaces. Fig. 4 shows that SAM correlates well with the classification performance and captures the monotonic dependence remarkably, given that SAM is a simple linear measure being applied to the outcome of a highly non-linear, optimized system. The results are consistent for other versions of GCN. In particular, in the SI Appendix (Section II) we show that the alignment measure correlates well with the performance of the recently proposed Simple Graph Convolution (SGC) [35].

The alignment measure can be used to evaluate the relative importance of features and graph for classification without explicitly running the GCN, by comparing the SAM under full randomization of features against the SAM under full randomization of the graph. If $\mathcal{S}^*(X_{100}, \widehat{A}, Y) > \mathcal{S}^*(X, \widehat{A}_{100}, Y)$, the features play a more important role in GCN classification. Conversely, if $\mathcal{S}^*(X_{100}, \widehat{A}, Y) < \mathcal{S}^*(X, \widehat{A}_{100}, Y)$, the graph is more important in GCN classification. While we have focused here on node classification, it would be interesting in future work to extend our measure to other tasks such as graph classification, link prediction, and regression.

VII. Conclusion

Here, we have introduced SAM (12), a measure that quantifies the consistency between the feature and graph ingredients of data sets, and we showed that it correlates well with the classification performance of GCNs. Our experiments show that a degree of alignment is needed for a GCN approach to be beneficial, and that using a GCN can actually be detrimental to the classification performance if the feature and graph subspaces associated with the data are not constructively aligned (e.g., Wikipedia and Wikipedia II). More generally, the SAM has potentially a wider range of applications in the quantification of data alignment including, among others: quantifying the alignment of different graphs associated with, or obtained from, particular data sets; evaluating the quality of classifications found using unsupervised methods; and aiding in choosing the classifier architecture most advantageous computationally for a particular data set.

Our approach has a number of limitations that could be addressed in future work. First, it contains two parameters (i.e., the dimensions of the subspaces, k_X^* and $k_{\widehat{A}}^*$) which need to be tuned through a computational search. Second, the alignment is not directly comparable across data sets since the subspace dimensions are adjusted for each data set. To facilitate comparisons across data sets, normalized versions of the alignment measure will be the object of future work. Third, the current measure is not suitable for very large data sets as the eigendecomposition of large matrices is computationally demanding. For very large data sets, approximations (e.g., using the Lanczos algorithm to explore only leading eigenvectors) might be necessary to optimize the subspace dimensions.

Acknowledgment

Yifan Qian acknowledges financial support from the China Scholarship Council program (No. 201706020176). Paul Expert and Mauricio Barahona acknowledge EPSRC grant EP/N014529/1 funding the EPSRC Centre for Mathematics of Precision Healthcare. Paul Expert acknowledges support from the NIHR Imperial Biomedical Research Centre (BRC).

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.

[2] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[3] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[4] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: going beyond euclidean data," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 18–42, 2017.

[5] L. Deng and D. Yu, "Deep learning: Methods and applications," *Foundations and Trends in Signal Processing*, vol. 7, no. 3-4, pp. 197–387, 2014.

[6] E. P. Simoncelli and B. A. Olshausen, "Natural image statistics and neural representation," *Annual Review of Neuroscience*, vol. 24, no. 1, pp. 1193–1216, 2001.

[7] D. J. Field, "What the statistics of natural images tell us about visual coding," in *Human Vision, Visual Processing, and Digital Display*, vol. 1077, 1989, pp. 269–277.

[8] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[9] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel, "Handwritten digit recognition with a back-propagation network," in *Advances in Neural Information Processing Systems*, 1990, pp. 396–404.

[10] J. Bruna and S. Mallat, "Invariant scattering convolution networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1872–1886, 2013.

[11] D. Lazer, A. S. Pentland, L. Adamic, S. Aral, A. L. Barabasi, D. Brewer, N. Christakis, N. Contractor, J. Fowler, M. Gutmann *et al.*, "Life in the network: the coming age of computational social science," *Science*, vol. 323, no. 5915, p. 721, 2009.

[12] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" in *International Conference on Learning Representations*, 2019.

[13] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Advances in Neural Information Processing Systems*, 2017, pp. 1024–1034.

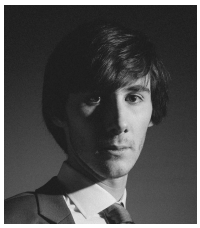
[14] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[15] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Advances in Neural Information Processing Systems*, 2016, pp. 3844–3852.

[16] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations (ICLR)*, 2017.
[17] M. Gori, G. Monfardini, and F. Scarselli, "A new model for learning in graph domains," in *Proceedings of IEEE International Joint Conference on Neural Networks*, vol. 2. IEEE, 2005, pp. 729–734.
[18] Y. Li, D. Tarlow, M. Brockschmidt, and R. S. Zemel, "Gated graph sequence neural networks," in *International Conference on Learning Representations (ICLR)*, 2016.
[19] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, "Convolutional networks on graphs for learning molecular fingerprints," in *Advances in neural information processing systems*, 2015, pp. 2224–2232.
[20] J. Atwood and D. Towsley, "Diffusion-convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2016, pp. 1993–2001.
[21] M. Niepert, M. Ahmed, and K. Kutzkov, "Learning convolutional neural networks for graphs," in *International Conference on Machine Learning*, 2016, pp. 2014–2023.
[22] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," in *International Conference on Learning Representations (ICLR)*, 2014.
[23] P. De Jong, C. Sprenger, and F. Van Veen, "On extreme values of moran's i and geary's c," *Geographical Analysis*, vol. 16, no. 1, pp. 17–24, 1984.
[24] T. Waldhor, "Moran's spatial autocorrelation coefficient," *Encyclopedia of Statistical Sciences, 2nd ed.(S. Kotz, N. Balakrishnana, C. Read, B. Vidakovic and N. Johnson, editors)*, vol. 12, pp. 7875–7878, 2006.
[25] M. E. Newman, "The structure and function of complex networks," *SIAM Review*, vol. 45, no. 2, pp. 167–256, 2003.
[26] K. Ye and L.-H. Lim, "Schubert varieties and distances between subspaces of different dimensions," *SIAM Journal on Matrix Analysis and Applications*, vol. 37, no. 3, pp. 1176–1197, 2016.
[27] A. Björck and G. H. Golub, "Numerical methods for computing angles between linear subspaces," *Mathematics of Computation*, vol. 27, no. 123, pp. 579–594, 1973.
[28] G. H. Golub and C. F. Van Loan, *Matrix Computations*. JHU Press, 2012, vol. 3.
[29] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.
[30] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su, "Arnetminer: extraction and mining of academic social networks," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2008, pp. 990–998.
[31] Y. Qian, W. Rong, N. Jiang, J. Tang, and Z. Xiong, "Citation regression analysis of computer science publications in different ranking categories and subfields," *Scientometrics*, vol. 110, no. 3, pp. 1351–1374, 2017.
[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2015.
[33] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010, pp. 249–256.
[34] R. Lambiotte, J.-C. Delvenne, and M. Barahona, "Random walks, Markov processes and the multiscale modular organization of complex networks," *IEEE Transactions on Network Science and Engineering*, vol. 1, no. 2, pp. 76–90, 2014.
[35] F. Wu, T. Zhang, A. H. d. Souza Jr, C. Fifty, T. Yu, and K. Q. Weinberger, "Simplifying graph convolutional networks," in *International Conference on Machine Learning*, 2019, pp. 6861–6871.

Yifan Qian is a PhD student at Queen Mary University of London (United Kingdom). His research is broadly concerned with computational social science, and combines theories and methods from network science, sociology, machine learning, and data science. He received his MSc in Computer Science from Beihang University (China) in 2017, and his BSc in Information and Computing Science from Beihang University (China) in 2014.

Paul Expert received his PhD in Physics from Imperial College London. He then hold post-doctoral positions in the Department of Neuroimaging at King's College London, the EPSRC Centre for the Mathematics of Precision Healthcare at Imperial College London, and now in the Global Digital Health Unit at Imperial College London. His research is concerned with understanding the interaction between the structure and function of complex systems, with applications ranging from neuroscience to public health and social systems.



Tom Rieu is currently a data analyst at Criteo in Paris, France. He is a graduate from the engineering school CentraleSupélec (Paris) and holds an MSc in Applied Mathematics from Imperial College London. His academic research was focused on machine learning and particularly the application of deep models to networks. Presently, his work in the Criteo Product team is concerned with feature engineering in order to improve the performance of the set of prediction algorithms powering Criteo's online advertising platform and retargeting technology.

Pietro Panzarasa is Professor of Networks and Innovation in the School of Business and Management at Queen Mary University of London. He received his PhD from Bocconi University (Italy), and then became a Research Fellow at the University of Southampton (UK). He also held visiting positions at Columbia University (New York) and Carnegie Mellon University (Pittsburgh). He draws on network science, computational social science and big data analytics to study social capital and dynamics of social interaction in complex large-scale networks.

Mauricio Barahona is Professor in the Department of Mathematics and Director of the EPSRC Centre for Mathematics of Precision Healthcare at Imperial College London. He received his PhD from the Massachusetts Institute of Technology and then held Fellowships at Stanford University and the California Institute of Technology. He is broadly interested in applied mathematics in engineering, physical, social, and biological systems using methods from graph theory, stochastic processes, dynamical systems and machine learning.

Supplementary material "Quantifying the alignment of graph and features in deep learning"

Yifan Qian, Paul Expert, Tom Rieu, Pietro Panzarasa, Mauricio Barahona

This is the supplementary material to the paper entitled "Quantifying the alignment of graph and features in deep learning", submitted to *IEEE Transactions on Neural Networks and Learning Systems*.

I. Finding optimal dimensions

A key element of the subspace alignment measure described in the main paper is to find lower dimensional representations of of the graph, features and ground truth.

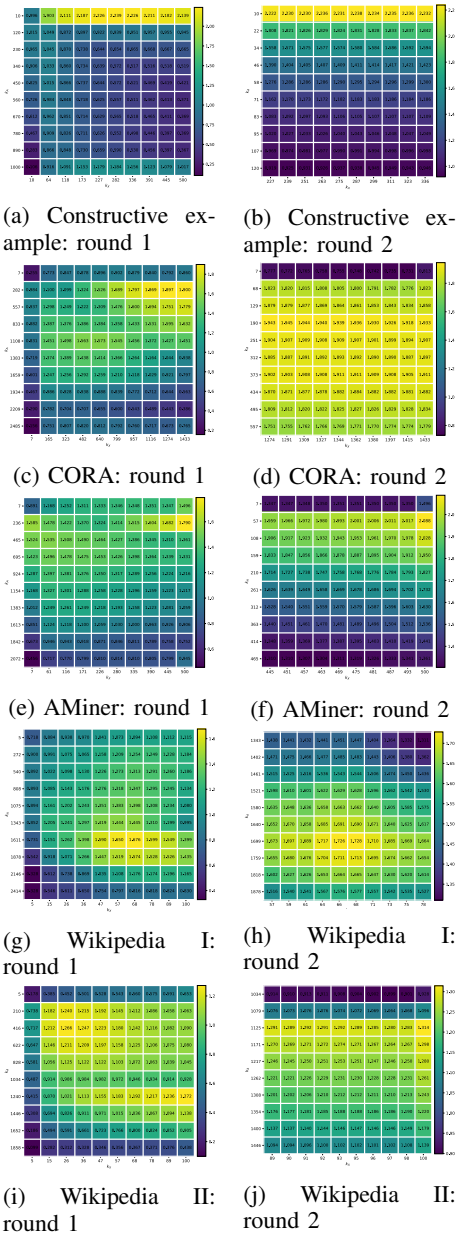
To determine the dimension of the representative subspaces, we propose the following heuristic:

$$(k_X^*, k_{\widehat{A}}^*) = \max_{k_X, k_{\widehat{A}}} \left(\|D(X_{100}, \widehat{A}_{100}, Y)\|_{\mathrm{F}} - \|D(X, \widehat{A}, Y)\|_{\mathrm{F}} \right). \tag{1}$$

We choose k_Y^* to be equal to the number of categories in the ground truth as they are non overlapping. Thus, k_X^* and $k_{\widehat{A}}^*$ range from k_Y^* to their maximum values, C^0, the dimension of the feature vectors, and N, the number of nodes in the graph, respectively.

To find the values for k_X^* and $k_{\widehat{A}}^*$ that maximize $||D||_{\mathrm{F}}$, we scan different possible combinations of k_X and $k_{\widehat{A}}$. We applied two rounds of scanning. In the first scanning round, in the intervals of k_X and $k_{\widehat{A}}$, we picked 10 equally spaced values that contain the minimum and maximum possible values for k_X and $k_{\widehat{A}}$. For example, in CORA, k_Y^* equals 7 because the number of categories in the ground truth is 7. Thus k_X ranges from 7 to $1,433$. At the end of the first round, the optimal values of k_X^* and $k_{\widehat{A}}^*$ are $1,433$ and 282, respectively (see Fig. 1c).

In the second scanning round, we applied a very similar process to the one just described. We set the scanning intervals of k_X and $k_{\widehat{A}}$ as the neighbors of k_X^* and $k_{\widehat{A}}^*$ found in the first round, respectively. For example, in CORA, for the second round, we set the intervals of k_X and $k_{\widehat{A}}$ as $[1415, 1433]$ and $[7, 557]$. Again, we split the new intervals with 10 equally spaced values. We have also shown the scanning results for other data sets in Figure 1.



(a) Constructive example: round 1

(b) Constructive example: round 2

(c) CORA: round 1

(d) CORA: round 2

(e) AMiner: round 1

(f) AMiner: round 2

(g) Wikipedia I: round 1

(h) Wikipedia I: round 2

(i) Wikipedia II: round 1

(j) Wikipedia II: round 2

Figure 1: Summary of results on scanning subspaces.

II. Replicating experiments on a variant of Graph Convolutional Networks

First, we would like to highlight that the alignment metric is independent of the architecture and only relies on the data. Therefore, we expect that the conclusion will be consistent with different variants of GCNs: the convolution operation in GCN (Kipf and Welling) can be seen as a neighborhood aggregation or message passing scheme. Many variants based on the Kipf and Welling version of GCN have been proposed, but they can ultimately be expressed as neighborhood aggregation or message passing schemes. For these different GCNs, we expect that our hypothesis that a certain degree of alignment between ingredients is needed for them to perform well would hold, since the working principles of variants of GCNs and the original version we consider are similar. To substantiate this claim, we have replicated our experiments using a recently proposed variant of GCN: Simple Graph Convolution (SGC) proposed by Wu et al. “Simplifying graph convolutional networks." ICML (2019). SGC is a simplified version of the original GCN proposed by Kipf and Welling that removes nonlinearities and collapses weight matrices between consecutive layers. It has been shown that SGC can achieve competitive performance on node classification tasks and yields up to several orders of magnitude speedup.

We use the implementation provided by Pytorch Geometric[1], which is a popular geometric deep learning extension library for PyTorch. Our results on SGC are shown below in Fig. 2 and Fig. 3. The figure suggests that results based on SGC are consistent with those produced using GCN (Kipf and Welling).

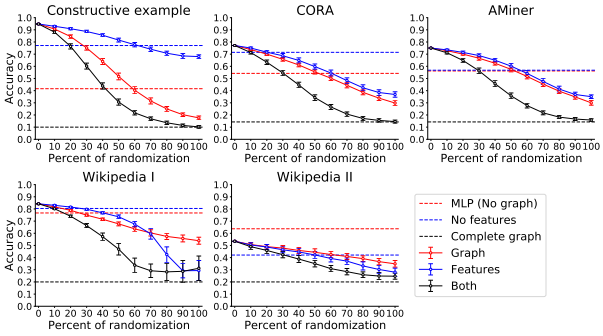


Figure 2: **Degradation of the classification performance as a function of randomization with SGC.** Each panel shows the degradation of the classification accuracy as a function of the randomization of graph, features and both, for a different data set. Error bars are evaluated over 100 realizations: for zero percent randomization, we report 100 runs with random seeds; for the rest, we report 1 run with random seed for 100 random realizations. The horizontal lines correspond to the limiting cases.

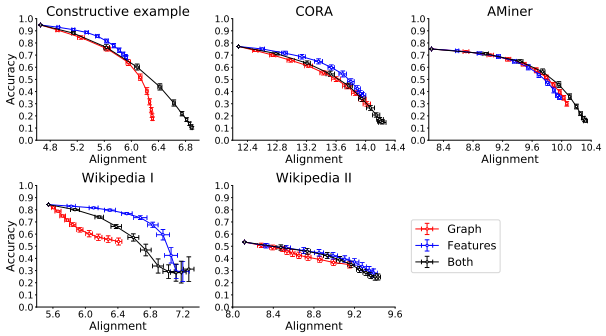


Figure 3: **Classification performance versus the subspace alignment measure (SAM) with SGC.** Each panel shows the accuracy of SGC versus the SAM for all the runs presented in Fig. 2. Error bars are evaluated over 100 randomizations.

III. Choices of distance measures

Within the distances discussed by Ref. (Ye, Ke, and Lim, Lek-Heng. “Schubert varieties and distances between subspaces of different dimensions." SIAM Journal on Matrix Analysis and Applications 37.3 (2016): 1176-1197.), there are two ‘families’:

1) average distances that use *all* the principal angles, e.g., the Chordal distance, $\left(\sum_{j=1}^{\alpha} \sin^2 \theta_j\right)^{1/2}$, and the Grassmann distance, $\left(\sum_{j=1}^{\alpha} \theta_j^2\right)^{1/2}$.
2) extremal distances that use only the maximum principal angle between two subspaces, e.g., the Projection distance, $\sin \theta_\alpha$where θ_α is the maximum angle.

Our numerics show that average distances (the first family) display similar performance, as they leverage information from all the principal angles. Hence these measures produce similar performance to the Chordal distance. To show this, we have replicated our experiments using the Grassmann distance (see Fig. 4 below). The results are consistent with those produced with Chordal distance.

On the other hand, we expect that extremal distances (the second family) will have less expressive power to capture the alignment between subspaces, since they use solely the maximum principal angle and do not consider the information contained in the other principal angles. To demonstrate this point, we replicated our experiments with the Projection distance (see Fig. 5 below). Our results show that the Projection distance is indeed less effective than the Chordal distance in representing the alignment between subspaces.

[1] https://github.com/rusty1s/pytorch_geometric/blob/master/examples/sgc.py

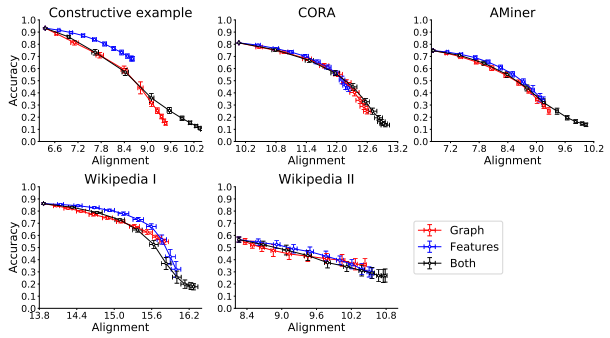


Figure 4: **Classification performance versus the subspace alignment measure (SAM) with Grassmann distance.** Each panel shows the accuracy of GCN versus the SAM. Error bars are evaluated over 100 randomizations.

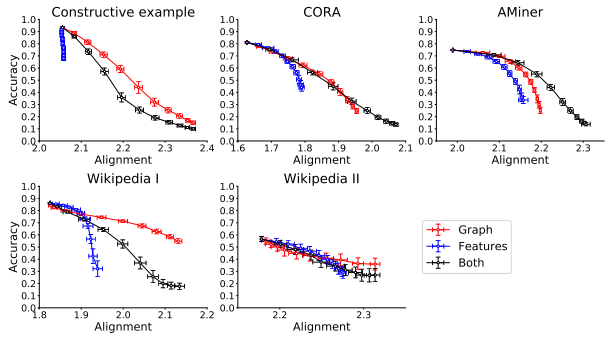


Figure 5: **Classification performance versus the subspace alignment measure (SAM) with Projection distance.** Each panel shows the accuracy of GCN versus the SAM. Error bars are evaluated over 100 randomizations.