STUDY OF A SIMPLE, EXPRESSIVE AND CONSISTENT GRAPH FEATURE REPRESENTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Graphs possess exotic features like variable size and absence of natural ordering of the nodes that make them difficult to analyze and compare. To circumvent this problem and learn on graphs, graph feature representation is required. Main difficulties with feature extraction lie in the trade-off between expressiveness, consistency and efficiency, i.e. the capacity to extract features that represent the structural information of the graph while being deformation-consistent and isomorphism-invariant. While state-of-the-art methods enhance expressiveness with powerful graph neural-networks, we propose to leverage natural spectral properties of graphs to study a simple graph feature: the graph Laplacian spectrum (GLS). We analyze the representational power of this object that satisfies both isomorphism-invariance, expressiveness and deformation-consistency. In particular, we propose a theoretical analysis based on graph perturbation to understand what kind of comparison between graphs we do when comparing GLS. To do so, we derive bounds for the distance between GLS that are related to the *divergence to isomorphism*, a standard computationally expensive graph divergence. Finally, we experiment GLS as graph representation through consistency tests and classification tasks, and show that it is a strong graph feature representation baseline.

1 INTRODUCTION

No matter where and at which scale we look, graphs are present. Social networks, public transport, information networks, molecules, any structural dependency between elements of a global system is a graph. An important task is to extract information from these graphs in order to understand whether they contain certain structural properties that can be represented and used in downstream machine learning tasks. In general, graphs are difficult to use as input of standard algorithms because of their exotic features like variable size and absence of natural orientation. Consequently, graph feature representation with equal dimensionality and dimension-wise alignment is required to learn on graphs.

Any embedding method is traditionally associated to a trade-off between preservation of structural information (expressiveness) and computation time (efficiency) (Cai et al., 2018). In the expressiveness, we particularly consider two key attributes of graph feature representation: consistency under deformation and invariance under isomorphism. The first forces the embedding to discriminate two graphs consistently with their structural dissimilarity. The second enables to have one representation for each graph, which can be a challenge since one graph has many possible orientations. In this paper, we propose to analyze the importance of satisfying the introduced criteria through a known but unused, simple, expressive and efficient candidate graph feature representation: the graph Laplacian spectrum (GLS). The Laplacian matrix of a graph is a well-known object in spectral learning (Belkin & Niyogi, 2002) for several reasons. First, the Laplacian eigenvalues give many structural information like the presence of communities and partitions (Newman, 2013), the regularity, the closed-walks enumeration, the diameter or the connectedness of the graph (Brouwer & Haemers, 2011). It is also interpretable in term of physics or mechanics (Bonald et al., 2018). It is backed by efficient and robust approximate eigen decomposition algorithms enabling to scale on large graphs and huge datasets (Halko et al., 2011). These properties give intuition that GLS can be an appropriate candidate for graph representation. In this paper we go further and analyze additional interesting properties of the Laplacian spectrum through the following contributions: (1) we build a perturbation-based framework to analyze the representation capacity of the GLS, (2) we analyze

the consistency between structural deformation of the graph and its GLS by deriving bounds for the distance between the GLS of two graphs, (3) we validate the consistency and the representational power of the GLS with different experiments on synthetic and real graphs.

The rest of the paper is built as follows. Section 2 gives an overview of interesting properties of the GLS for graph representation. A presentation of the mathematical framework and the theoretical analysis are displayed respectively in Section 3 and 4. Section 5 proposes experiments to illustrate theoretical results and show the representational power of GLS. Finally, Section 6 describes related work for whole-graph representation.

2 THE MEANINGFUL PROPERTIES OF THE LAPLACIAN SPECTRUM

The Laplacian matrix of a graph is a major object in spectral learning (Belkin & Niyogi, 2002). However, most of the attention is usually directed to its eigenvectors and not its spectrum, and spectral learning is generally applied to node clustering or classification, not whole-graph representation. But, Laplacian spectrum holds interesting properties for graph representation. First, Laplacian spectrum is invariant under graph isomorphism. In fact, graph isomorphism is equivalent to simultaneous permutation of lines and columns in the adjacency matrix, and since the determinant of a permutation is 1 or -1, characteristic polynomial of the Laplacian remains unchanged. Second, each eigenvalue can be seen as an entire whole-graph feature representation by itself, containing specific structural information. Hence any subset of Laplacian eigenvalues is a meaningful and valuable embedding. This enables the usage of truncated Laplacian spectrum (t-GLS) instead of GLS as whole-graph feature representation. Using t-GLS reduces the embedding time thanks to eigenvalue algorithms that do not require entire diagonalization to give partial spectrum (Halko et al., 2011). Finally, GLS is related to some interpretable structural information of the graph, as reminded below.

Interpretation of the GLS The smallest non-zero eigenvalue of the Laplacian is the *spectral gap*, corresponding the difference between the two largest eigenvalues of the Laplacian. It contains information about the connectivity of the graph. High spectral gap means high connectivity. For example, given a number of vertices in a connected graph, a minimum spectral gap indicates that the graph is a double kite (Marsden, 2013). The largest eigenvalue gives a lower bound of the maximal node degree of the graph. The spectral gap can also be viewed as the difference in energy between the ground state and first excited state of a dynamical system (Cubitt et al., 2015). More generally each eigenvalue of the Laplacian corresponds to the energy level of a *stable* configuration of the nodes in the embedding space (Bonald et al., 2018). The lower the energy, the stabler the configuration. In (Shuman et al., 2016), the Laplacian eigenvalues correspond to frequencies associated to a Fourier decomposition of any signal living on the vertices of the graph. Thus, the truncation of the Fourier decomposition acts as filter on the signal. Characterizing a graph by the some eigenvalues of its Laplacian is thus comparable to characterizing a melody by some fundamental frequencies.

In summary, Laplacian spectrum contains many graph structural information. Methods to get such information are generally computationally expensive. In the light of these properties, we go further and analyze in the following sections the capacity of GLS to represent graph structure.

3 PERTURBATION APPROACH AND PROBLEM SETUP

We consider two undirected and weighted graphs $G_1 = (V_1, E_1, W_1)$ and $G_2 = (V_2, E_2, W_2)$ with respective adjacency matrix W_1 and W_2, degree matrix D_1 and D_2. These matrices are set with respect to an arbitrary indexing of the nodes. Laplacian matrix L_i of G_i is defined as $L_i = D_i - W_i$. We aim at using the GLS to build fixed-dimensional representation that encodes structural information to compare any graphs G_1 and G_2 that are not aligned nor equally sized. For the rest of the paper, and without loss of generality we postulate that $|V_1| \leq |V_2|$. The rest of this section introduces the definitions, hypothesis and notations needed for our theoretical analysis of the GLS.

Definition 1. *Let $G = (V, E, W)$ a weighted graph with n nodes, with $W \in \mathcal{M}_{n \times n}$ the $n \times n$ weighted adjacency matrices. We define $P \in \mathcal{M}_{n \times n}$ a symmetric matrix with $P_{ii} = 0$, $P_{ij} \in [-W_{ij}, W_{ij}]$ such that $W_{ij} + P_{ij} \in [0, 1] \, \forall (i, j)$. We define the two following perturbations applied on graph G:*

- *Adding isolated nodes:* $\overline{W} = \begin{bmatrix} W & 0_{n \times m} \\ 0_{m \times n} & 0_{m \times m} \end{bmatrix}$

- *Adding or removing edges:* $W^P = W + P$

We call edge-perturbation the addition or removal of edges, and node-perturbation the addition of nodes. A complete perturbation is done by adding isolated nodes and perturbing the augmented graph with edge-perturbation. We note that the withdrawal of a node is equivalent to removal of all edges around this nodes. Moreover, if graph G is unweighted, i.e. with binary adjacency, then edge perturbations $P_{ij} \in \{-1, 0, 1\}$.

Remark 1. *If $P = -W + \Pi^T W \Pi$ with $\Pi \in \mathcal{P}(n)$ then the perturbation is a permutation of the node indexing. Such a perturbation is not interesting and edge perturbation due to node indexing has to be annihilated by a permutation matrix as in the following Definition 2:*

Definition 2. *We say that G^{P^*} is a perturbed version of G if we have*

$$\begin{cases} W^{P^*} = \Pi^{*T}(\overline{W} + P^*)\Pi^* \\ s.t. \ \Pi^* = \arg\min_{\Pi} \|W^{P^*} - \Pi^T \overline{W} \Pi\|_1 \end{cases}$$

i.e. such that P^* is the sparsest possible i.e. does not include permutations.

Notations We denote P^* *the sparsest perturbation* as defined in Definition 2. We denote $\overline{G}$ the completion of G with isolated nodes. If M is a matrix associated to G, we denote $\overline{M}$ the equivalent matrix for $\overline{G}$. We denote $\lambda(X)$ the eigenvalue of a square matrix X in ascending order, $\lambda_i(X)$ the ith smallest eigenvalue.

Hypothesis 1. *Without loss of generality, we assume that G_2 is a perturbed version of G_1, i.e. $\exists P^*$ the sparsest $|V_2|$-square perturbation matrix and $\Pi^* \in \mathcal{P}(|V_2|)$ a $|V_2|$-square permutation matrix such that $W_2 = \Pi^{*T} \left(\overline{W_1} + P^*\right) \Pi^*$. P^* is a $|V_2|$-square block matrix, with top-left block P_{11}^* being a $|V_1|$-square perturbation matrix for graph G_1. Bottom right block P_{22}^* is the $(|V_2| - |V_1|)$-square adjacency matrix of the additional nodes. P_{12}^* is the $|V_1| \times (|V_2| - |V_1|)$ adjacency matrix representing the links between graph G_1 and the additional nodes $V_2 \setminus V_1$.*

We have defined a notion of continuous deformation of graphs. This deformation has a natural and simple interpretation: any graph G_2 is a perturbed version of graph G_1, and the larger the perturbation the higher the structural dissimilarity between G_1 and G_2.

The next section uses the previously presented mathematical framework to analyze the consistency of the Laplacian spectrum as graph representation and its natural link to graph isomorphism problem.

4   LAPLACIAN SPECTRUM AS GRAPH FEATURE REPRESENTATION

We place ourselves under the Hypothesis 1 saying that the difference between graphs G_1 and G_2 is characterized by the unknown deformation P^*. A good embedding of these graphs should be close when level of deformation is low, and far otherwise. This level of deformation can be quantified by the global and node-wise entries of P^*. These features are by construction present in the Laplacian of P^*, denoted L_{P^*}. We use this idea to propose an analysis of the distance between to GLS.

4.1   CONSISTENCY UNDER DEFORMATION AND RELATION TO GRAPH ISOMORPHISM

Two graphs G_1 and G_2 are isomorphic if and only if $\exists \Pi \in \mathcal{P}(|V_1|)$ such that $L_2 = \Pi^{-1} L_1 \Pi$ (Merris, 1994), hence when they are structurally equivalent irrespective to the vertex ordering. Several papers has proposed to use a notion of *divergence to graph isomophism* (DGI) to compare graphs (Grohe et al., 2018; Rameshkumar et al., 2013). The DGI between graphs G_1 and G_2 is generally the minimal Frobenius norm of the difference between L_1 and $\Pi^{-1} L_2 \Pi$. Considering this definition, the following Lemma links the graph-isomorphism problem and the Laplacian of the hypothetical perturbation P^* and show that this divergence is the norm of L_{P^*}:

Lemma 1. *Using the notations from Hypothesis 1, we have:* $L_2 = \Pi^{*T}\left(\overline{L_1} + L_{P^*}\right)\Pi^*$, *with* $L_{P^*} = diag(P^*\mathbf{1}_{|V_1|}) - P^*$ *the Laplacian of* P^* *and* $\mathbf{1}_n$ *the* n-*dimensional unit vector. In particular,* $\min_{\Pi}\|L_2 - \Pi^T\overline{L_1}\Pi\|_F = \|L_{P^*}\|_F$.

We remind that graph isomorphism is at best solved in quasipolynomial time (Babai, 2016) and can not be used in practice for large graphs and datasets. The following Proposition show how the distance between GLS relaxes the isomorphism-based graph divergence.

Proposition 1. *Using Hypothesis 1 and Lemma 1:* $\|\lambda(L_2) - \lambda(\overline{L_1})\|_2 \leq \|L_{P^*}\|_F$.

The above result tells us that the higher the difference between GLS, the larger the hypothetical perturbation P^*, i.e. the higher the structural dissimilarity.

We now study the implication of GLS closeness. This problem tackles the notion of *non-isomorphic L-cospectrality*, i.e. the idea that two graphs can have equal eigenvalues while having different Laplacian matrix (Brouwer & Haemers, 2011). The following proposition gives a simple insight into the problem of spectral characterization in our perturbation-based framework:

Proposition 2. *We denote* $L_i = Q_i\Lambda_i Q_i^T$ *the SVD of* L_i *such that the diagonal of* Λ_i *is in ascending order. Hence we have* $\|L_{P^*}\|_F \leq \|\lambda(L_2) - \lambda(\overline{L_1})\|_2 + \|\Pi^{*T}\overline{Q_1}\Lambda_2\overline{Q_1}^T\Pi^* - L_2\|_F$.

This proposition shows that equal spectrum means equal graphs only when eigenvectors are also equal. Otherwise, L-cospectrality for non-isomorphic graphs tells us that there exists families of graphs that are not fully determined by their spectrum. These families are characterized by some structural properties such that two non-isomorphic graphs with equal Laplacian spectrum share these properties but not their adjacency (Van Dam & Haemers, 2003). In practice, this is not a problem. First, almost all graphs are determined by their spectrum (Brouwer & Haemers, 2011). Second, equal GLS indicates the precious information that graphs share common statistical, mechanical, physical or chemical properties (see Section 2), no matter the adjacency matrix. These physical properties are what we seek to represent when representing graphs for ML tasks. Third, non-isomorphic L-cospectrality concerns equally sized graphs which is not likely with respect to all possible real-life graphs. When the studied dataset contains specifically L-cospectral non-isomorphic graphs and when the task requires unique representation property, GLS is not appropriate and more sophisticated and powerful embedding methods taking for example eigenvectors into account (Verma & Zhang, 2017) should be studied and used. Otherwise, i.e. in almost every situations, according to previously presented results, GLS characterizes the graph and is directly related to the hypothetical perturbation P^*.

Nevertheless, we accordingly propose the Proposition 3 to better understand GLS proximity even when graphs are non-isomorphic cospectral.

Proposition 3. *The closer the GLS, the closer to unitary-similarity the Laplacian matrices.*

We remind that two real n-square matrices A and B are *unitary-similarity* if there exists an orthogonal matrix O such that $B = OAO^T$. Similarity is an *equivalence relation* on the space of square-matrices. Moreover, divergence to unitary-similarity is a relaxed version of the divergence to graph-isomorphism (Grohe et al., 2018), where the permutation matrix space is replaced by a unitary matrix space. Finally from Proposition 1 and 3 we can bound the distance between GLS as follows: $\min_{O \in \mathcal{O}(|V_2|)}\|\overline{L_1} - OL_2O^T\|_F \leq \|\lambda(\overline{L_1}) - \lambda(L_2)\|_2 \leq \|L_{P^*}\|$.

In this section, we have shown that structural similarity (divergence) between graphs can be reasonably approximated by the similarity (divergence) between their GLS.

4.2 LAPLACIAN SPECTRUM AS WHOLE-GRAPH REPRESENTATION IN PRACTICE

Previous section showed the capacity of the distance between Laplacian spectrum to serve as proxy for graph similarity. In practice, a fixed embedding dimension d must be chosen for all graphs in dataset $\mathcal{D}$. According to previous analysis, the most obvious dimension is $d = \max_{G \in \mathcal{D}}|V|$ and all graphs with less than d nodes may be pad with isolated nodes. We note that padding with isolated nodes is equivalent than adding zeros in the GLS. Nevertheless, in some datasets, some graphs can be significantly larger and the padding can become abusive. We therefore propose for these graph to have $d < \max_{G \in \mathcal{D}}|V|$. We simply truncate the GLS such that we keep only the highest d eigenvalues. This method also enables to save computation time.

The problem with this method is that we may lose information for graphs with more than d nodes. In practice, for large graphs, the contribution of the lowest eigenvalues to the distance between GLS as a proxy for graph divergence is negligible. In particular, large graph have many sparse areas, such that many eigenvalues are very low, hence truncating the bottom part of the GLS may not be a problem. We assess the impact of the truncation in the experimental section.

Though, we can also propose several ways to avoid this problem, like embedding the lowest eigenvalues with simple statistics, like moments or histograms. In the experimental section, we do not use this trick.

5 EXPERIMENTS

All experiments can reproduced using the notebook given at: `https://github.com/researchsubmission/ICLR2020/`.

5.1 PRELIMINARY EXPERIMENTS

As a first illustration of deformation-based results presented in Section 4, we propose to use Erdos-Rnyi random graphs (Erdős & Rényi, 1959) with parameter $p = 0.05$. We focus on three simple experiments.

First, the distance between the Laplacian spectrum of a graph and a perturbed version of this graph is related to the number of perturbations. We can find the experimental illustration in Figure 1. We see that the number of perturbations is roughly equally related to the distance between GLS features for edge addition and edge withdrawal, i.e. for graph sparsity decrease or increase. A relation between graph sparsity and Laplacian eigenvalues can be seen for example through the Gershgorin circle theorem (Gershgorin, 1931).
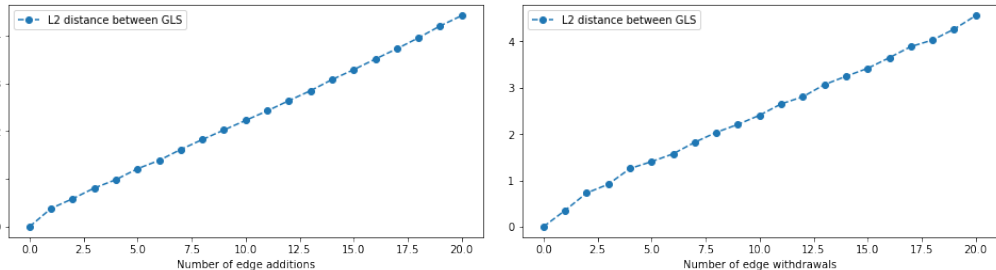

Figure 1: Experimental results to illustrate how GLS behaves under edge addition (left), edge withdrawal (right). In this case, studied adjacency and perturbation matrix are binary.

Second, we mentioned that when a graph is significantly bigger than other graphs of a dataset, w can use a truncated GLS (t-GLS). This method both saves computation time thanks to iterative eigenvalues algorithms and avoid the addition of isolated nodes in all other graphs. In Figure 3, we show results of experiments showing that t-GLS is consistent with node addition. As experimental setup, we take a reference graph with n nodes and compute its GLS. Then we add a randomly connected node and compute the t-GLS of the new graph, by keeping only the n largest eigenvalues. We repeat it 20 times. We compute the L_2-distance to reference GLS, for different levels of connectivity for the additional nodes. We first observe that the t-GLS is consistent with node addition. We also confirm our previous theoretical results by observing that the more connected the additional nodes, the higher the GLS divergence.

5.2 CLASSIFICATION OF MOLECULAR AND SOCIAL NETWORK GRAPHS

We evaluate spectral feature embedding with a classification task on molecular graphs and social network graphs. Experimental setup for classification task is given in Appendix E. We assume here that two structurally close graphs belong to the same class. We challenge this assumption with the following experiments.
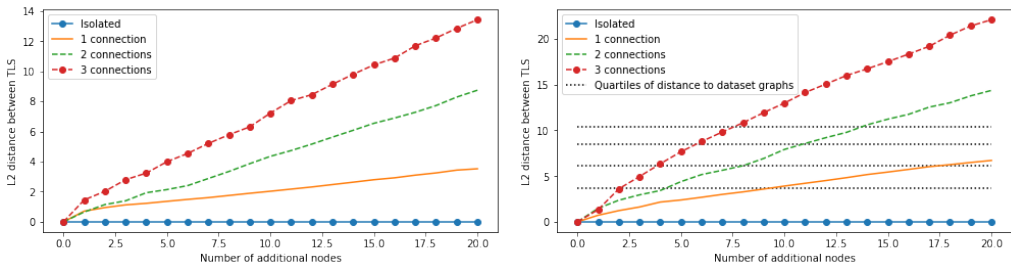
Figure 2: Experimental results illustrate how truncated GLS behaves under iterative addition of 20 new nodes with respectively 0, 1, 2 and 3 random connections with graph, for respectively synthetic a 80-nodes Erdos-Reyni graph (left) and a 28-nodes molecular graph from MUTAG dataset (right). Horizontal dotted lines (right figure) are the quartiles 25, 50, 75 and 100 of the distances between the GLS of the 28-nodes graph and the other 187 graphs of the dataset.

We propose to compare GLS-based classification results to those obtained by feature-based and deep learning methods. Standard graph feature representation methods are: Earth Mover's Distance (Nikolentzos et al., 2017) (EMD), Pyramid Match (Nikolentzos et al., 2017) (PM), Feature-Based (Barnett et al., 2016) (FB) and Dynamic-Based Features (Gómez & Delvenne) (DyF). All of these methods use support vector classifier (SVC) over extracted features. Deep learning methods are: Variational Recurrent Graph Classifier (Pineau & de Lara, 2019) (VRGC), Graph Convolutional Network (Kipf & Welling, 2016) (GCN), Deep Graph CNN (Zhang et al., 2018) (DGCNN), Capsule GNN (Xinyi & Chen, 2018) (CapsGNN), Graph Isomorphism Network (Xu et al., 2019) (GIN) and GraphSAGE (Hamilton et al., 2017). All deep learning methods are end-to-end graph classifers. A description of these models is given in the related work, Section 6.

All values reported in Table 1 and Table 2 are taken from the above-mentioned papers.

On molecular graphs We use five datasets for the experiments: Mutag (MT), Enzymes (EZ), Proteins Full (PF), Dobson and Doig (DD) and National Cancer Institute (NCI1) (Kersting et al., 2016). All graphs are chemical components. Nodes are atoms or molecules and edges represent checmical or electrostatic bindings. We note that molecular graphs contain node attributes, that are used by some models presented in Table 1. We let the question of the relevance of comparing models with slightly different inputs to the discretion of the reader. Description and statistics of molecular datasets are presented in Table 3, Appendix F.

	MT	EZ	PF	DD	NCI1
EMD + SVC	86.1 ± 0.8	36.8 ± 0.8	-	-	72.7 ± 0.2
PM + SVC	85.6 ± 0.6	28.2 ± 0.4	-	75.6 ± 0.6	69.7 ± 0.1
FB + SVC	84.7 ± 2.0	29.0 ± 1.2	70.0 ± 1.3	-	62.9 ± 1.0
DyF + SVC	86.3 ± 1.3	26.6 ± 1.2	73.1 ± 0.4	-	66.6 ± 0.3
VRGC	86.3 ± 8.6	48.4 ± 6.2	74.8 ± 3.0	-	80.7 ± 2.2
GCN*	85.6 ± 5.8	-	76.0 ± 3.2	-	80.2 ± 2.0
DGCNN*	85.8 ± 1.7	51.0 ± 7.3	75.5 ± 0.9	79.4 ± 0.9	74.4 ± 0.5
CapsGNN*	86.7 ± 6.9	54.7 ± 5.7	76.3 ± 3.6	75.4 ± 4.2	78.4 ± 1.6
GIN-0*	89.4 ± 5.6	-	76.2 ± 2.8	-	82.7 ± 1.7
GraphSAGE*	85.1 ± 7.6	-	75.9 ± 3.2	-	77.7 ± 1.5
GLS + SVC	87.9 ± 7.0	40.7 ± 6.3	75.3 ± 3.5	74.3 ± 3.5	73.3 ± 2.1

Table 1: Accuracy (%) of classification with different graph representations, on molecular graphs. SVC stands for support vector classifier. Comparative models are divided into two groups: feature + SVC and end-to-end deep learning. *Models using node attributes.

On social network graphs We use five datasets for the experiments: IMDB-Binary (IMBD-B), IMDB-Multi (IMDB-M), REDDIT-Binary (REDDIT-B), REDDIT-5K-Multi (REDDIT-M) and

COLLAB. All graphs are social networks. The graphs of these datasets do not contain node attributes. Therefore, we can more appropriately compare GLS + SVC to deep learning based classification. Statistics about social networks datasets are presented in Table 4, Appendix F.

	IMDB-B	IMDB-M	REDDIT-B	REDDIT-M	COLLAB
GCN	74.0 ± 3.4	51.9 ± 3.8	-	-	79.0 ± 1.8
DGCNN	70.0 ± 0.9	47.8 ± 0.9	76.0 ± 1.7	-	73.8 ± 0.5
CapsGNN	73.1 ± 4.8	50.3 ± 2.7	-	52.9 ± 1.5	79.6 ± 0.9
GIN-0	75.1 ± 5.1	52.3 ± 2.8	92.4 ± 2.5	57.5 ± 1.5	80.2 ± 1.9
GraphSAGE	72.3 ± 5.3	50.9 ± 2.2	-	-	-
GLS + SVC	73.2 ± 4.2	48.5 ± 2.5	87.4 ± 3.4	52.0 ± 1.8	78.5 ± 1.1

Table 2: Classification accuracy (%) of different deep learning based models plus ours over standard social networks datasets. Graphs of these datasets does not have node features. SVC stands for support vector classifier.

Analysis of the results The classification results above illustrate the capacity of GLS to capture graph structural information, under the assumption that structurally close graphs belong to the same class. The graph neural-networks are globally more expressive since they can leverage specific information for graph classification since is end-to-end. In particular, they obtain strong results when there are node labels (see molecular experiments 5.2). Nevertheless, GLS is a simple way to represent graphs in an unsupervised manner, with theoretical background, simplicity of implementation (eigendecomposition is accessible to anyone interested in any computer) and competitive downstream classification results.

On the reasonability of using truncated GLS We assess the impact of truncating the GLS. Using truncated GLS (t-GLS) enables to (1) reduce the computational cost for large graphs and (2) reduce the dimensionality of the graph representation for all graphs. Results are presented in Figure 3 for molecular datasets.
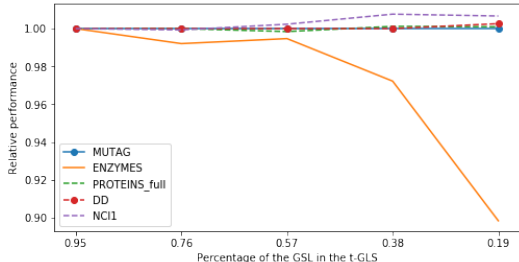

Figure 3: Illustration of the impact of the truncation in term of classification accuracy of the molecular graphs. We represent the impact relatively to the 95-percentile truncation adopted for classification experiments.

We see that truncating GLS is not highly impacting classification results. Only ENZYME multiclass classification, which is a particularly difficult task (see experiments in Section 5.2), suffers from truncation. Additional insight about the t-GLS is given in Appendix G.

6 RELATED WORK

We propose to divide graph feature representation into three categories: graph kernel methods, feature-based methods and deep learning.

Graph kernel methods Kernel methods create a high-dimensional feature representation of data. The *kernel trick* (Shawe-Taylor et al., 2004) avoids to compute explicitly the coordinates in the feature space, only the inner product between all pairs of data image: it is an *implicit embedding*

methods. These methods are applied to graphs (Nikolentzos et al., 2018; 2017). It consists in performing pairwise comparisons between atomic substructures of the graphs until a good representative dictionary is found. The embedding of a graph is then the number of occurrences of these substructures within it. These substructures can be graphlets (Yanardag & Vishwanathan, 2015), subtree patterns (Shervashidze et al., 2011), random walks (Vishwanathan et al., 2010) or paths (Borgwardt & Kriegel, 2005). The main difficulty lives in the choice of appropriate algorithm and kernel that accept graphs with variable size and capture useful feature for downstream task. Moreover, kernel methods can be computationally expensive but techniques like the Nystrm algorithm (Williams & Seeger, 2001) allow to lower the number of comparison with a low rank approximation of the similarity matrix.

Feature-based methods Feature-based representation methods (Barnett et al., 2016) represent each graph as the concatenation of features. Generally, the feature-based representation can offer a certain degree of interpretability and transparency. The most basic ones are the number of nodes or edges, the histogram of node degrees. These simple graph-level features offers by construction the sought isormorphism-invariance but suffer from low expressiveness. More sophisticated algorithms consider features based on attributes of random walks on the graph (Gómez & Delvenne) while others are graphlet based (Kondor et al., 2009). (Kondor & Borgwardt, 2008) explicitly built permutation-invariant features by mapping the adjacency matrix to a function on the symmetric group. (Verma & Zhang, 2017) proposed a family of graph spectral distances to build graph features. Experimental work in (de Lara & Pineau, 2018) used normalized Laplacian spectrum with random forest for graph classification with promising results. (Wilson & Zhu, 2008) analyzes the cospectrality of different graph matrices and studies experimentally the representational power of their spectra. These two last works are directly related to the current work. Nevertheless, in both cases, the theoretical analysis is absent and comparative experiment with current benchmarks and methods is limited. in this paper we propose a response to these concerns.

Deep learning based methods GNNs learn representation of nodes of a graph by leveraging together their attributes, information on neighboring nodes and the attributes of the connecting edges. When graphs have no vertex features, the node degrees are used instead. To create graph-level representation instead of node representation, node embeddings are pooled by a permutation invariant readout function like summation or more sophisticated information preserving ones (Ying et al., 2018; Zhang et al., 2018). A condition of optimality for readout function is presented in (Xu et al., 2019). Recently, Xinyi & Chen (2018) levraged capsule networks (Sabour et al., 2017), neural units designed to enable to better preserve information at pooling time. Other popular evolution of GNNs formulate convolution-like operations on graphs. Formulation in spectral domain (Bruna et al., 2013; Defferrard et al., 2016) is limited to the processing of different signals on a single graph structure, because they rely on the fixed spectrum of the Laplacian. Conversely, formulation in spatial domain are not limited to one graph structure (Atwood & Towsley, 2016; Duvenaud et al., 2015; Niepert et al., 2016; Hamilton et al., 2017) and can infer information from unseen graph structures. At the same time, alternative to GNN exist and are related to random walk embedding. In Li et al. (2017), neural networks help to sample paths which preserve significant graph properties. Other approaches transforms graphs into sequence of nodes embedding passed into a recurrent neural network (RNN) (You et al., 2018; Pineau & de Lara, 2019) to get useful embedding. These models do not inherently include isomorphism-invariance but greedy learn it by seeing the same graph numerous times with different node ordering and embedding. These methods are powerful and globally obtain a high level of expressiveness (see experimental section 5.2).

7 CONCLUSION

In this paper, we analyzed the graph Laplacian spectrum (GLS) as whole graph representation. In particular, we showed that comparing two GLS is a good proxy for the divergence between two graphs in term of structural information. We coupled these results to the natural invariance to isomorphism, the simplicity of implementation, the computational efficiency offered by modern randomized algorithms and the rare occurrence of detrimental L-cospectral non-isomorphic graphs to propose the GLS as a strong baseline graph feature representation.

REFERENCES

James Atwood and Don Towsley. Diffusion-convolutional neural networks. In *Advances in Neural Information Processing Systems*, pp. 1993–2001, 2016.

László Babai. Graph isomorphism in quasipolynomial time. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pp. 684–697. ACM, 2016.

Ian Barnett, Nishant Malik, Marieke L Kuijjer, Peter J Mucha, and Jukka-Pekka Onnela. Feature-based classification of networks. *arXiv preprint arXiv:1610.05868*, 2016.

Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in neural information processing systems*, pp. 585–591, 2002.

Thomas Bonald, Alexandre Hollocou, and Marc Lelarge. Weighted spectral embedding of graphs. In *2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 494–501. IEEE, 2018.

Karsten M Borgwardt and Hans-Peter Kriegel. Shortest-path kernels on graphs. In *Data Mining, Fifth IEEE International Conference on*, pp. 8–pp. IEEE, 2005.

Andries E Brouwer and Willem H Haemers. *Spectra of graphs*. Springer Science & Business Media, 2011.

Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.

Hongyun Cai, Vincent W Zheng, and Kevin Chang. A comprehensive survey of graph embedding: problems, techniques and applications. *IEEE Transactions on Knowledge and Data Engineering*, 2018.

Gavin C Cawley and Nicola LC Talbot. On over-fitting in model selection and subsequent selection bias in performance evaluation. *Journal of Machine Learning Research*, 11(Jul):2079–2107, 2010.

Toby S Cubitt, David Perez-Garcia, and Michael M Wolf. Undecidability of the spectral gap. *Nature*, 528(7581):207, 2015.

Nathan de Lara and Edouard Pineau. A simple baseline algorithm for graph classification. *Relational Representation Learning, NeurIPS Workshop*, 2018.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems*, pp. 3844–3852, 2016.

David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in neural information processing systems*, pp. 2224–2232, 2015.

Paul Erdős and Alfréd Rényi. On random graphs i. *Publ. Math. Debrecen*, 6:290–297, 1959.

Semyon Aranovich Gershgorin. Uber die abgrenzung der eigenwerte einer matrix. (6):749–754, 1931.

Leonardo Gutiérrez Gómez and Jean-Charles Delvenne. Dynamics based features for graph classification. In *Benelearn 2017: Proceedings of the Twenty-Sixth Benelux Conference on Machine Learning, Technische Universiteit Eindhoven, 9-10 June 2017*, pp. 131.

Martin Grohe, Gaurav Rattan, and Gerhard J Woeginger. Graph similarity and approximate isomorphism. In *43rd International Symposium on Mathematical Foundations of Computer Science (MFCS 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53 (2):217–288, 2011.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pp. 1024–1034, 2017.

Kristian Kersting, Nils M. Kriege, Christopher Morris, Petra Mutzel, and Marion Neumann. Benchmark data sets for graph kernels, 2016. `http://graphkernels.cs.tu-dortmund.de`.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Risi Kondor and Karsten M Borgwardt. The skew spectrum of graphs. In *Proceedings of the 25th international conference on Machine learning*, pp. 496–503. ACM, 2008.

Risi Kondor, Nino Shervashidze, and Karsten M Borgwardt. The graphlet spectrum. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 529–536. ACM, 2009.

Cheng Li, Jiaqi Ma, Xiaoxiao Guo, and Qiaozhu Mei. Deepcas: An end-to-end predictor of information cascades. In *Proceedings of the 26th International Conference on World Wide Web*, pp. 577–586. International World Wide Web Conferences Steering Committee, 2017.

Anne Marsden. Eigenvalues of the laplacian and their relationship to the connectedness of a graph. *University of Chicago, REU*, 2013.

Russell Merris. Laplacian matrices of graphs: a survey. *Linear algebra and its applications*, 197: 143–176, 1994.

Mark EJ Newman. Spectral methods for community detection and graph partitioning. *Physical Review E*, 88(4):042822, 2013.

Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. Learning convolutional neural networks for graphs. In *International conference on machine learning*, pp. 2014–2023, 2016.

Giannis Nikolentzos, Polykarpos Meladianos, and Michalis Vazirgiannis. Matching node embeddings for graph similarity. In *AAAI*, pp. 2429–2435, 2017.

Giannis Nikolentzos, Polykarpos Meladianos, Stratis Limnios, and Michalis Vazirgiannis. A degeneracy framework for graph similarity. In *IJCAI*, pp. 2595–2601, 2018.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

Edouard Pineau and Nathan de Lara. Variational recurrent neural networks for graph classification. In *Representation Learning on Graphs and Manifolds Workshop*, 2019.

A Rameshkumar, R Palanikumar, and S Deepa. Laplacian matrix in algebraic graph theory. *Journal Impact Factor*, pp. 0–489, 2013.

Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *Advances in neural information processing systems*, pp. 3856–3866, 2017.

John Shawe-Taylor, Nello Cristianini, et al. *Kernel methods for pattern analysis*. Cambridge university press, 2004.

Nino Shervashidze, Pascal Schweitzer, Erik Jan van Leeuwen, Kurt Mehlhorn, and Karsten M Borgwardt. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research*, 12(Sep):2539–2561, 2011.

David I Shuman, Benjamin Ricaud, and Pierre Vandergheynst. Vertex-frequency analysis on graphs. *Applied and Computational Harmonic Analysis*, 40(2):260–291, 2016.

Edwin R Van Dam and Willem H Haemers. Which graphs are determined by their spectrum? *Linear Algebra and its applications*, 373:241–272, 2003.

Saurabh Verma and Zhi-Li Zhang. Hunt for the unique, stable, sparse and fast feature learning on graphs. In *Advances in Neural Information Processing Systems*, pp. 88–98, 2017.

S Vichy N Vishwanathan, Nicol N Schraudolph, Risi Kondor, and Karsten M Borgwardt. Graph kernels. *Journal of Machine Learning Research*, 11(Apr):1201–1242, 2010.

Christopher KI Williams and Matthias Seeger. Using the nyström method to speed up kernel machines. In *Advances in neural information processing systems*, pp. 682–688, 2001.

Richard C Wilson and Ping Zhu. A study of graph spectra for comparing graphs and trees. *Pattern Recognition*, 41(9):2833–2841, 2008.

Zhang Xinyi and Lihui Chen. Capsule graph neural network. *International Conference on Learning Representations*, 2018.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *International Conference on Learning Representations*, 2019.

Pinar Yanardag and SVN Vishwanathan. Deep graph kernels. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1365–1374. ACM, 2015.

Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec. Hierarchical graph representation learning with differentiable pooling. In *Advances in Neural Information Processing Systems*, pp. 4800–4810, 2018.

Jiaxuan You, Rex Ying, Xiang Ren, William Hamilton, and Jure Leskovec. Graphrnn: Generating realistic graphs with deep auto-regressive models. In *International Conference on Machine Learning*, pp. 5694–5703, 2018.

Muhan Zhang, Zhicheng Cui, Marion Neumann, and Yixin Chen. An end-to-end deep learning architecture for graph classification. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

A   PROOF OF LEMMA 1

Proof.

$$
\begin{aligned}
L_2 &= D_2 - W_2 \\
&= \operatorname{diag}(W_2 \mathbf{1}) - W_2 \\
&= \operatorname{diag}(\Pi^{*T} \left(\overline{W_1} + P^* \right) \Pi^* \mathbf{1}) - \Pi^* \left(\overline{W_1} + P^* \right) \Pi^* \\
&= \operatorname{diag}(\Pi^{*T} \overline{W_1} \Pi^* \mathbf{1}) + \operatorname{diag}(\Pi^{*T} P^* \Pi^* \mathbf{1}) - \Pi^{*T} \overline{W_1} \Pi^* - \Pi^{*T} P^* \Pi^* \\
&= \Pi^{*T} \overline{D_1} \Pi^* - \Pi^{*T} \overline{W_1} \Pi^* + \Pi^{*T} D_{P^*} \Pi^* - \Pi^{*T} P^* \Pi^* \\
&= \Pi^{*T} \overline{L_1} \Pi^* + \Pi^{*T} L_{P^*} \Pi^*
\end{aligned}
$$

with $L_{P^*} = \operatorname{diag}(P^* \mathbf{1}) - P^* = D_{P^*} - P^*$ and $\mathbf{1}$ the unit vector.

Therefore,

$$
\min_{\Pi} \| L_2 - \Pi^T \overline{L_1} \Pi \|_F = \min_{\Pi} \| \Pi^T L_{P^*} \Pi \|_F = \| L_{P^*} \|_F
$$

$\square$

B   PROOF OF PROPOSITION 1

Proof. From lemma 1 we have $L_2 = \Pi^{*T} \overline{L_1} \Pi^* + \Pi^{*T} L_{P^*} \Pi^*$. Moreover, from Weyl's eigenvalues inequalities and since eigenvalues are isomorphism invariant:

$$
\begin{cases}
\lambda_i(L_2) \leq \lambda_i(\overline{L_1}) + \lambda_{|V_2|}(L_{P^*}) \\
\lambda_i(\overline{L_1}) + \lambda_1(L_{P^*}) \leq \lambda_i(L_2)
\end{cases}
$$

Hence:

$$
\lambda_1(L_{P^*}) \leq \lambda_i(L_2) - \lambda_i(\overline{L_1}) \leq \lambda_{|V|}(L_{P^*})
$$

Now let (λ, x) be any eigen couple of a matrix $M \in \mathcal{M}_{n \times n}$. We can always pick $i \in \{1 \ldots n\}$ and build x such that $|x_i| = 1$ and $|x_{j \neq i}| < 1$. Hence:

$$
\begin{aligned}
(Mx)_i = \lambda x_i &\iff \sum_{j=1}^{n} m_{ij} x_j = \lambda x_i \\
&\iff \lambda^2 \leq \sum_{j=1}^{n} (m_{ij} x_j)^2 \\
&\implies \lambda^2 \leq \sum_{j=1}^{n} (m_{ij})^2 \\
&\implies \lambda^2 \leq \frac{1}{n} \sum_{i,j=1}^{n} m_{ij}^2
\end{aligned}
$$

Using previous results we get:

$$
\sum_{i=1}^{|V_2|} \left(\lambda_i(L_2) - \lambda_i(\overline{L_1}) \right)^2 \leq |V_2| \frac{1}{|V_2|} \sum_{i,j=1}^{|V_2|} L_{P^*}{}_{ij}^2 = \| L_{P^*} \|_F^2,
$$

with $\|X\|_F = \sqrt{\sum_i \sum_j |X_{ij}|^2}$ the Frobenius norm.

$\square$

C   PROOF OF PROPOSITION 2

Proof. We remind that the Forbenius norm is unitarily invariant thanks to the cyclic property of the trace. For any $\tilde{P} \in \mathcal{O}(|V_2|)$ we have:

$$\begin{aligned}
\|L_{P^*}\|_F &= \|\Pi^{*T}\overline{L_1}\Pi^* - L_2\|_F \\
&= \|\Pi^{*T}\overline{L_1}\Pi^* - \tilde{P}^T L_2 \tilde{P} + \tilde{P}^T L_2 \tilde{P} - L_2\|_F \\
&\leq \|\Pi^{*T}\overline{L_1}\Pi^* - \tilde{P}^T L_2 \tilde{P}\|_F + \|\tilde{P}^T L_2 \tilde{P} - L_2\|_F.
\end{aligned}$$

In particular if $\tilde{P} = Q_2 \overline{Q_1}^T \Pi^*$:

$$\begin{aligned}
\|\tilde{P}^T L_2 \tilde{P} - L_2\|_F &= \|(Q_2\overline{Q_1}^T\Pi^*)^T L_2 Q_2\overline{Q_1}^T\Pi^* - L_2\|_F \\
&= \|(Q_2\overline{Q_1}^T\Pi^*)^T Q_2\Lambda_2 Q_2^T Q_2\overline{Q_1}^T\Pi^* - L_2\|_F \\
&= \|\Pi^{*T}\overline{Q_1}\Lambda_2\overline{Q_1}^T\Pi^* - L_2\|_F,
\end{aligned}$$

We also have that

$$\begin{aligned}
\|\Pi^{*T}\overline{L_1}\Pi^* - \tilde{P}^T L_2 \tilde{P}\|_F &= \|Q_1\Lambda_1 Q_1^T - \Pi^*(Q_2\overline{Q_1}^T\Pi^*)^T Q_2\Lambda_2 Q_2^T Q_2\overline{Q_1}^T\Pi^*\Pi^{*T}\|_F \\
&= \|\overline{\Lambda_1} - \Lambda_2\|_F.
\end{aligned}$$

Hence:

$$\|L_{P^*}\|_F \leq \|\overline{\Lambda_1} - \Lambda_2\|_F + \|\Pi^{*T}\overline{Q_1}\Lambda_2\overline{Q_1}^T\Pi^* - L_2\|_F.$$

$\square$

D   PROOF OF PROPOSITION 3

Proof. Denoting $\mathcal{O}(n)$ the n-orthogonal matrices group (orthogonal since real), we want to show that:

$$\min_{O \in \mathcal{O}(|V_2|)} \|\overline{L_1} - O^T L_2 O\|_F \leq \|\lambda(\overline{L_1}) - \lambda(L_2)\|_2$$

We denote $L_i = Q_i \Lambda_i Q_i^T$ the eigendecomposition of the Laplacian L_i with $\Lambda_i = \text{diag}(\lambda(L_i))$. Since $\overline{Q_1}$ is unitary and using property of Frobenius norm, we have, $\forall O \in \mathcal{O}(|V_2|)$:

$$\|\overline{L_1} - O^T L_2 O\|_F = \|\overline{\Lambda_1} - \overline{Q_1}^T O^T Q_2\Lambda_2 Q_2^T O\overline{Q_1}\|_F,$$

We know that $\overline{Q_1}$ and Q_2 are orthogonal since they are respectively eigenvector matrices of symmetric matrix $\overline{L_1}$ and L_2. We therefore have:

13

$$(Q_2^T O \overline{Q_1})^T (Q_2^T O \overline{Q_1}) = \overline{Q_1}^T O^T Q_2 Q_2^T O \overline{Q_1} = I_{|V_1|}$$

Moreover $\forall \tilde{\Pi} \in \mathcal{P}(|V_2|) \subset \mathcal{O}(|V_2|)$, if $O = Q_2 \tilde{\Pi} \overline{Q_1}^T$ then $O \in \mathcal{O}(|V_2|)$.

Hence,

$$
\begin{aligned}
\min_{O \in \mathcal{O}(|V_2|)} \|\overline{L_1} - O^T L_2 O\|_F &= \min_{O \in \mathcal{O}(|V_2|)} \|\overline{\Lambda_1} - \overline{Q_1}^T O^T Q_2 \Lambda_2 Q_2^T O \overline{Q_1}\|_F \\
&\leq \min_{\tilde{\Pi} \in \mathcal{P}(|V_2|)} \|\overline{\Lambda_1} - \overline{Q_1}^T (Q_2 \tilde{\Pi} \overline{Q_1}^T)^T Q_2 \Lambda_2 Q_2^T (Q_2 \tilde{\Pi} \overline{Q_1}^T) \overline{Q_1}\|_F \\
&= \min_{\tilde{\Pi} \in \mathcal{P}(|V_2|)} \|\overline{\Lambda_1} - \tilde{\Pi}^T \Lambda_2 \tilde{\Pi}\|_F \\
&= \min_{\tilde{\Pi} \in \mathcal{P}(|V_2|)} \|\overline{\Lambda_1} - \tilde{\Pi} \Lambda_2 \tilde{\Pi}^T\|_F \\
&= \min_{\tilde{\Pi} \in \mathcal{P}(|V_2|)} \|\overline{\Lambda_1} - \tilde{\Pi} \Lambda_2 \tilde{\Pi}^T\|_F \\
&= \min_{\sigma \in \mathcal{S}(|V_2|)} \|\lambda(\overline{L_1}) - \lambda(L_2)_{\sigma(1:|V_2|)}\|_2 \\
&\leq \|\lambda(\overline{L_1}) - \lambda(L_2)\|_2
\end{aligned}
$$

with $\mathcal{S}(n)$ the permutation group of $\{1 \ldots n\}$.

$\square$

E   EXPERIMENTAL SETUP FOR CLASSIFICATION OF GRAPHS

For classification, we use standard 10-folds cross validation setup. Each dataset is divided into 10 folds such that the class proportions are preserved in each fold for all datasets. These folds are then used for cross-validation i.e, one fold serves as the testing set while the other ones compose the training set. Results are averaged over all testing sets. All figures gathered in the tables of results are build using this setup. For the dimension $d \in [\![1, \max_{G \in \mathcal{D}} |V|]\!]$, representing the number of eigenvalues we keep to build the truncated GLS, we chose the percentile 95 of the distribution of graph sizes in each dataset, i.e. we truncate the 5% smallest eigenvalues. Considering weak truncation impact (see Section 5.2, when we have large datasets containing large graphs, like the two REDDIT datasets, we can truncate more severely to make the problem computationally more efficient. In particular considering that GLS approached as a simple baseline more than a final graph representation for large scale usage.

We use the support vector classifier (SVC) from scikit-learn (Pedregosa et al., 2011). We impose Radial Basis Function as kernel, i.e. $\mathcal{K}(\lambda(\overline{L_1}), \lambda(L_2)) = \exp(-\gamma \|\lambda(\overline{L_1}) - \lambda(L_2)\|_2^2)$. It is a similarity measure related to L_2-norm between GLS. Hence, our theoretical results remain consistent with our experiments. Hyper parameters C and γ are tuned among respectively $\{0.5, 1, 5\}$ and $\{0.0001, 0.001, 0.01, 0.1, 0.5, 1, 5\}$ for the molecular datasets, and $\{0.5, 1, 5, 25, 50\}$ and $\{0.0001, 0.001, 0.01, 0.1\}$ for the social network datasets. In practice, using a global pool for all the datasets gives equivalent results, but hyperparameter inference becomes expensive with a too large grid, in particular in a 10-fold cross validation setup. We use a nested hyperparameter search cross-validation for each of the 10 folds: in each 90% training fold we perform a 5-fold random search cross-validation before training. We therefore avoid the problem of overfitting related to model selection that appear when using non-nested cross-validation (Cawley & Talbot, 2010).

We additionally put our results with k-nearest neighbor algorithm with L_2 norm, in order to illustrate the notion of proximity we introduced: neighboring GLS is related to structurally close graphs (i.e. eventually graphs from the same class).

F CHARACTERISTICS OF THE REAL DATASETS

We use five molecular datasets and five social network datasets for the experiments (Kersting et al., 2016). Tables 3 and 4 gives statistics of the differents datasets. All used datasets can be found at `https://ls11-www.cs.tu-dortmund.de/staff/morris/graphkerneldatasets` (Kersting et al., 2016).

Molecular graphs datasets are Mutag (MT), Enzymes (EZ), Proteins Full (PF), Dobson and Doig (DD) and National Cancer Institute (NCI1). In MT, the graphs are either mutagenic and not mutagenic. EZ graphs are tertiary structures of proteins from the 6 Enzyme Commission top level classes. In DD, compounds are secondary structures of proteins that are enzyme or not. PF is a subset of DD without the largest graphs. In NCI1, graphs are anti-cancer or not. The graphs of these datasets have node labels that can be leverages by graph neural networks.

	MT	EZ	PF	DD	NCI1
# graphs	188	600	1113	1178	4110
# classes	2	6	2	2	2
bias (%)	66.5	16.7	59.6	58.7	50.0
min./max. $|V|$	10/28	2/125	4/620	30/5736	3/106
avg. $|V|$	18	33	39	284	30
avg. $|E|$	39	124	146	1431	65
Node attributes	✓	✓	✓	✓	✓

Table 3: Molecular datasets statistics. Bias indicates the proportion of the dominant class.

Social networks datasets are IMDB-Binary (IMBD-B), IMDB-Multi (IMDB-M), REDDIT-Binary (REDDIT-B), REDDIT-5K-Multi (REDDIT-M) and COLLAB. REDDIT-B and REDDIT-M contain graphs representing discussion thread, with edges between users (nodes) when one responded to the other's comment. Classes are the subreddit topics from which thread have originated. IMDB-B and IMDB-M contain networks of actors that appeared together within the same movie. IMDB-B contains two classes for *action* or *romance* genres and IMDB-M three classes for *comedy*, *romance* and *sci-fi*. COLLAB graphs represent scientific collaborations, with edge between two researchers meaning that they co-authored a paper. Labels of the graphs correspond to subfields of Physics. The graphs of these datasets have no node attributes and therefore enable fair comparison with deep learning methods.

	IMDB-B	IMDB-M	REDDIT-B	REDDIT-M	COLLAB
# graphs	1000	1500	2000	4999	5000
# classes	2	3	2	5	3
bias (%)	50.0	33.3	50.0	20.0	52.0
min./max. $|V|$	12/136	7/89	3/3760	22/3606	32/492
avg. $|V|$	20	13	426	501	75
avg. $|E|$	97	66	496	590	2458
Node attributes	✗	✗	✗	✗	✗

Table 4: Social network datasets statistics. Bias indicates the proportion of the dominant class.

G ADDITIONAL INSIGHT ON THE ACCEPTABILITY OF USING TRUNCATED GLS

Figure 4 illustrates the reasonability of using only the highest eigenvalues of the Laplacian spectrum as whole-graph feature representation. We take the original and final graphs of the deformation-consistency test presented in Figure 3. We compute the L_2 distance between t-GLS with dimension d and divide it by d, for d varying from 1 to 15 . The objective is to confirm that first eigenvalues
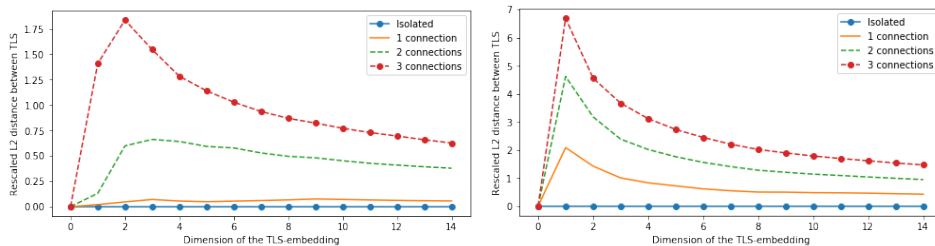
15

Figure 4: Illustration of the relative importance of the dimensionality of GLS-embedding, after the iterative addition of 20 new nodes with respectively 0, 1, 2 and 3 random connections with graph, for respectively synthetic a 80-nodes Erdos-Reyni graph (left) and a 28-nodes molecular graph from MUTAG dataset (right). We see that the first largest eigenvalues of the Laplacian are the most important to discriminate a graph and its perturbed version.

are relatively more important to discriminate to structurally different graphs, which is the case. We note that for the Erdos-Reyni case with few connected additional nodes, first eigenvalues are not as relatively important as for the other example. In fact, adding nodes with stochastic connections is the construction process of Erdos-Reyni graphs. Hence, discriminating augmented graph from the original one is difficult based only on the structural information.