Spectral Maps for Learning on Subgraphs

Marco Pegoraro PEGORARO@DI.UNIROMA1.IT
Sapienza University of Rome, Italy

Riccardo Marin RICCARDO.MARIN@MNF.UNI-TUEBINGEN.DE
University of Tübingen, Tübingen AI Center, Germany

Arianna Rampini ARIANNARAMPINI@GMAIL.COM
Sapienza University of Rome, Italy

Simone Melzi SIMONE.MELZI@UNIMIB.IT
University of Milano-Bicocca, Italy

Luca Cosmo LUCA.COSMO@UNIVE.IT
Ca' Foscari University of Venice, Italy

Emanuele Rodolà RODOLA@DI.UNIROMA1.IT
Sapienza University of Rome, Italy

Editor: Sophia Sanborn, Christian Shewmake, Simone Azeglio, Nina Miolane.

Abstract

In graph learning, maps between graphs and their subgraphs frequently arise. For instance, when coarsening or rewiring operations are present along the pipeline, one needs to keep track of the corresponding nodes between the original and modified graphs. Classically, these maps are represented as binary node-to-node correspondence matrices, and used as-is to transfer node-wise features between the graphs. In this paper, we argue that simply changing this map representation can bring notable benefits to graph learning tasks. Drawing inspiration from recent progress in geometry processing, we introduce a spectral representation for maps that is easy to integrate into existing graph learning models. This spectral representation is a compact and straightforward plug-in replacement, and is robust to topological changes of the graphs. Remarkably, the representation exhibits structural properties that make it interpretable, drawing an analogy with recent results on smooth manifolds. We demonstrate the benefits of incorporating spectral maps in graph learning pipelines, addressing scenarios where a node-to-node map is not well defined, or in the absence of exact isomorphism. Our approach bears practical benefits in knowledge distillation and hierarchical learning, where we show comparable or improved performance at a fraction of the computational cost.

Keywords: Spectral theory, Learning on graphs, Subgraphs, Maps Representation

1. Introduction

Graph learning offers a powerful set of techniques for understanding complex data, which often call for downsampling or rewiring operations to improve scalability and performance. One common approach is to perform computations and training on a partial or modified version of the graph rather than the entire graph. For example, computationally expensive operations can be performed on a coarsened version of the graph, as demonstrated in works such as Deng et al. (2020). Additionally, graph rewiring, which directly modifies the

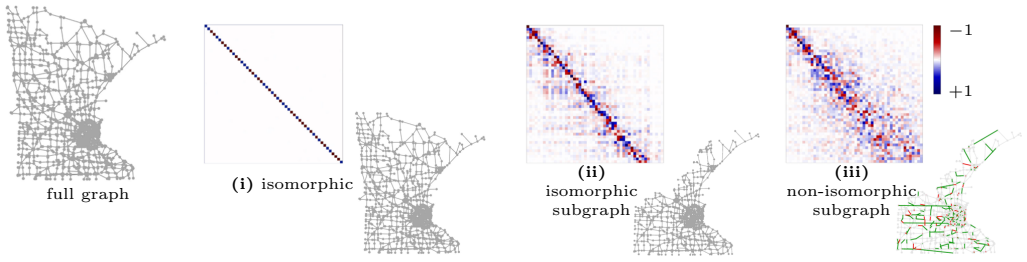


Figure 1: Spectral maps between a full graph (depicted on the left) and three different graphs, respectively: an isomorphic graph (**i**), an isomorphic subgraph(**ii**), and a *non*-isomorphic subgraph obtained by randomly rewiring the former (**iii**) where the green and red edges are added and removed, respectively.

connectivity, creates an even more challenging scenario (Chan and Akoglu, 2016; Brüel-Gabrielsson et al., 2022). In these settings, a crucial aspect that is often taken for granted is the data transfer between graphs and their subgraphs. Recent studies have shown that transferring information such as positional encoding from a graph to its rewired versions can improve GNN performance (Brüel-Gabrielsson et al., 2022), highlighting the importance of effectively transferring information between graphs. However, this task remains challenging in many scenarios, particularly when the involved graphs are not isomorphic. Although correspondences between nodes are often provided, utilizing these correspondences as they are may not always be the optimal solution, leaving room for further improvement.

In this paper, we propose to shift to a *spectral* representation to compactly encode maps between graphs and subgraphs in graph learning pipelines. The new representation is a straightforward replacement into existing models; it is easy to compute, has a regularizing behaviour leading to improved downstream performance, and bears a natural structure that is easy to interpret. From a technical standpoint, the map representation is obtained via a change of basis with respect to the eigenvectors of the graph Laplacian. This idea, introduced a decade ago in the field of geometry processing under then name of *functional maps* (Ovsjanikov et al., 2012), has led to notable advancements in several tasks in graphics and vision. However, the potential application of this concept in graph learning has not been explored so far.

We summarize our main contributions as follows:

- We propose the adoption of spectral representations for maps between graphs and *subgraphs*. For the first time, we show that such maps exhibit a special structure in their coefficients, capturing the similarity between the Laplacian eigenspaces of the two graphs.

- We focus on the problem of feature transfer and include experiments showing practical applications, such as hierarchical embedding and knowledge distillation, on graphs spanning a few dozen to tens of thousands of nodes. We demonstrate key benefits in terms of performance and computational complexity.

- We conduct an empirical examination of the structure of the spectral map across a diverse range of graphs and in various scenarios of partiality, including sub-isomorphic and non-isomorphic graphs. Our findings demonstrate that the map exhibits a distinct structure in these contexts; see Figure 1 for examples.

2. Related work

In this section, we review the literature on using maps in graph learning models, where our method has a direct relevance.

Maps for graph learning. Transferring information between non-isomorphic graphs is a challenging problem in graph learning. This is especially relevant in scenarios such as domain adaptation (Pilanci and Vural, 2020), meta-learning (Yang et al., 2022b), and federated learning (Zhang et al., 2021), where the information collected on a set of graphs needs to be transferred to other graphs. In this paper, we focus on the problem of *representing* maps between graphs, given a (possibly partial) node-to-node correspondence; however, it is worth noting that several methods tackle the complementary problem of determining a correspondence (Singh et al., 2008; Hermanns et al., 2021; Man et al., 2016) when the latter is not provided as input.

Hierarchical graph embedding. Many learning-based graph embedding algorithms, such as DeepWalk (Perozzi et al., 2014) and node2vec (Grover and Leskovec, 2016), do not scale to large graphs and struggle to capture long-distance global relationships (Chen et al., 2018). To overcome these problems, recent works (Chen et al., 2018; Deng et al., 2020; Liang et al., 2021) proposed to compute a hierarchy of coarsened graphs on which to compute the embeddings and then lift the values up to the original graph. In this framework, an important step is the propagation of embeddings through the coarsened graphs, which requires a proper refinement step to ensure the quality of the final embedding. In particular, ensuring smooth propagation between levels has been identified as crucial in enhancing performance. In our experiments, we show how the spectral representation can be easily adapted for this step, with beneficial effects on the graph embedding task; we refer to Section 4.2 for a detailed evaluation.

Knowledge distillation on graphs. The goal of knowledge distillation is to transfer information from a large model to a smaller one (Hinton et al., 2015). Recently, this framework has been extended to graphs (Yang et al., 2020; Chen et al., 2020; Yang et al., 2021). Specifically, Yang et al. (2022a) introduced the concept of geometric knowledge distillation, which involves transferring graph topology information extracted by a GNN model from a graph G (Teacher) to a target GNN model; importantly, the target GNN only has access to a partial view of G (Student). In this paper, we address this task by adopting the spectral representation to enforce the similarity between the intermediate representation learned by the teacher and the student (Section 4.3).

3. Background on spectral representation

Graphs and Laplacian eigenvectors. We consider undirected, unweighted graphs $G = (V, E)$ with nodes V and edges $E \subseteq V \times V$. We denote as $A \in \{0,1\}^{|V|\times|V|}$ the adjacency

matrix of G, which is a binary matrix where $A(i,j) = 1$ if an edge connects node i to node j, and $A(i,j) = 0$ otherwise.

The symmetric normalized Laplacian for G is defined as the square matrix $\mathcal{L} = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$, where D is a diagonal matrix of node degrees, with entries $D(i,i) = \sum_{j=1}^{|V|} A(i,j)$. This linear operator is symmetric and positive semi-definite; it admits an eigendecomposition $\mathcal{L} = \Phi \Lambda \Phi^\top$, where Λ is a diagonal matrix that contains the eigenvalues, and Φ is a matrix having as columns the corresponding eigenvectors concatenated side by side. Throughout this paper, we assume the eigenvalues (and corresponding eigenvectors) to be sorted in non-descending order $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq 2$; this is important for interpreting the spectral maps that we define in the sequel.

Each eigenvector ϕ_l for $l = 1, \ldots, |V|$ has length $|V|$, and can be interpreted as a scalar function defined on the nodes of the graph; for this reason, we will refer to them as eigen*functions*. The eigenfunctions form an orthonormal basis for the space of functions defined on the graph nodes (i.e. $\Phi^\top \Phi = Id$). One may consider a subset of eigenfunctions, namely those associated with the k smallest eigenvalues, to compactly approximate a graph signal, employing techniques akin to Fourier analysis.

Spectral maps for graphs. The representation we propose directly derives from the functional maps framework for smooth manifolds Ovsjanikov et al. (2012), extended to the graph setting in Wang et al. (2019); Hermanns et al. (2021).

Consider two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ and a binary matrix $S \in \mathbb{R}^{|V_2| \times |V_1|}$ encoding a node-to-node map $T : G_2 \to G_1$. Applying an orthogonal change of basis w.r.t. bases Φ_1, Φ_2, we get to the representation:

$$C = \Phi_2^\top S \Phi_1 \,, \tag{1}$$

where $\Phi_1 \in \mathbb{R}^{|V_1| \times k}, \Phi_2 \in \mathbb{R}^{|V_2| \times k}$ contain the first k eigenvectors of the symmetrically normalized graph Laplacians of G_1 and G_2 respectively. This matrix C is easy to compute by simple matrix multiplication. The size of C does *not* depend on the number of points in G_1 and G_2, but only on the number k of basis functions. In other words, C represents the linear transformation that maps the coefficients of any given function $f : V_1 \to \mathbb{R}$ expressed as linear combination of Φ_1, to coefficients of a corresponding function $g : V_2 \to \mathbb{R}$ expressed in the eigenbasis Φ_2.

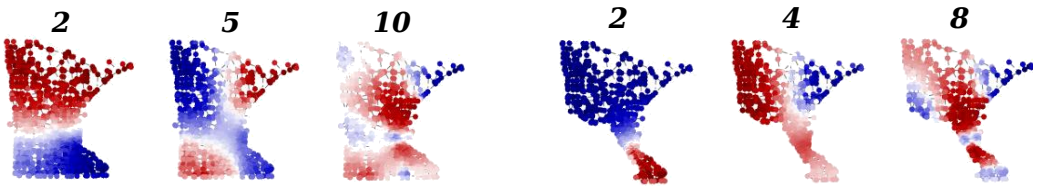
Graph nodes may often come with numerical attributes encoding user identities in social networks, or positional encodings. We can model such data as a collection of functions $f : V_1 \to \mathbb{R}$. From Equation 1 we can transfer a function f from G_1 to G_2 applying the following formula:

$$\hat{g} = \Phi_2 C \Phi_1^\top f \,, \tag{2}$$

where $\Phi_1^\top$ projects f in its coefficients, C apply the spectral transfer, Φ_2 reconstucts the transfered signal $\hat{g}$.

4. Applications on subgraphs

From now on, we consider the setting where we are given a graph G_1 and a possibly noisy subgraph $G_2 = (V_2, E_2)$ of G_1, such that $V_2 \subseteq V_1$. In this paper, we define the node-to-node map $T : G_2 \to G_1$ as the mapping that associates each node $v_i^2 \in G_2$ with the node



(a) G_1 (b) G_2

Figure 2: Subset of eigenfunctions from a complete graph G_1 and its subgraph G_2, with its the ordering index. We observe that the eigenfunctions of G_1 and G_2 still correlate even if not necessarily at the same index.

$v_i^1 \in G_1$, which originally represented v_i^2 before the perturbation was applied to G_1. In this case, Equation 1 still holds. Note that in some cases, one may decide to invert the roles of the graphs G_1 and G_2, as in Section 4.2. This does not affect the spectral representation of the map.

4.1. Motivation

Our motivation starts from the observation that in many practical cases, the eigenspaces of the normalized graph Laplacian are well preserved under *non-isomorphic* transformations of the graph, including strong partiality, topological perturbations, and edge rewiring.

According to Equation (1), each coefficient c_{ij} of C corresponds to a dot product between ϕ_i^2 and $S\phi_j^1$; this measures the correlation *at corresponding nodes* between a Laplacian eigenvector ϕ_i^2 of G_2, and a Laplacian eigenvector ϕ_j^1 of G_1; see Figure 1.

Figure 2 shows how the eigenfunctions of the complete graph G_1, and those of the subgraph G_2 still correlate even if not necessarily at the same index (see pair 5-4) and the correlation may not be exact (see pair 10-8); the extent to which the eigenfunctions correlate is captured precisely by the structure of C. In particular, we can see that the values of the Laplacian eigenfunctions stay approximately the same (up to sign, in the case of simple spectrum) at the nodes that are not directly involved in the perturbation – which is to say that the eigenvectors of the partial graph G_2, encoded in Φ_2, correlate strongly with the those of G_1, encoded in Φ_1.

To the best of our knowledge, this observation is not trivial and has not been reported before. This simple fact leads to the following important observation that is central to our contribution:

The spectral representation allows us to represent the same (or similar) subspace of smooth functions by truncating the functional representation at the first k eigenfunctions

Since eigenfunctions align well, we can exploit the spectral maps and the properties they inherit on the representation and transfer of signals. Classically, maps are represented as binary matrices S whose dimensions scale quadratically with the number of nodes in the graphs. This observation allows us to use the spectral map as a compact and sparse representation that still provides an efficient way of transferring information between graphs. Furthermore, as we will show in the rest of this section, the properties inherited from this

representation provide advantages in applications. In all the following experiments, we inject the spectral representation in learning procedures only where information transfer is needed, leaving the rest of the pipeline unchanged. In Section 5, we show empirical evidence of this behavior and describe its practical consequences.

Table 1: *Hierarchical embedding*: Mean classification accuracy on the task of node classification.

	Node2Vec			Graph Walk			GraphSAGE		
	Cora	Citeseer	Pubmed	Cora	Citeseer	Pubmed	Cora	Citeseer	Pubmed
Graphzoom	0.77	0.64	0.79	0.76	0.65	0.78	0.72	0.55	**0.74**
Ours (5%)	0.78	**0.67**	0.79	0.77	0.67	**0.80**	0.68	0.56	**0.74**
Ours	**0.79**	**0.67**	**0.80**	**0.79**	**0.68**	**0.80**	**0.74**	**0.59**	**0.74**
(% eigs)	(10%)	(2.5%)	(10%)	(15%)	(2.5%)	(0.05)	(20%)	(60%)	(5%)

4.2. Hierarchical Graph Embedding

Hierarchical Graph Embedding aims to learn graph embedding considering a hierarchy of coarsened graphs. First, each level of the hierarchy is constructed from the original graph. Then, an embedding is computed on the last level (i.e. smallest subgraph), and finally, it is lifted to the original graph. In this case, the correspondence between the original graph and its subgraphs is given by construction.

In this section, we show that transferring the embeddings across the hierarchy levels via the spectral map is beneficial in the applications. To this end, we consider the state-of-the-art Hierarchical Graph embedding approach GraphZoom Deng et al. (2020) to compute the coarsened graphs. Then, we transfer the embedding using Equation (2) in the reverse direction, as we transfer the signal from the subgraph to the full graph. Equation (2) still holds, but G_1 is now the subgraph, and G_2 is the full graph. The spectral map is computed from the ground truth correspondences obtained during the coarsening phase.

To evaluate performance, we tackle the task of node classification. The classification is performed by a linear logit regression model that takes as input the embedding lifted up through the hierarchy of graphs. As done in Deng et al. (2020), we consider two levels of coarsening. Table 1 shows the node classification accuracy of our method compared to GraphZoom Deng et al. (2020) and a baseline $n2n$. We consider here three graphs (Cora, Citeseer and Pubmed) and three embedding algorithms (node2vec Grover and Leskovec (2016), DeepWalk Perozzi et al. (2014) and GraphSAGE Hamilton et al. (2017a)), similarly to Deng et al. (2020). The only hyperparameter of our approach is the number of eigenvectors k employed in the spectral map computation. In the last two rows of Table 1, we report the best accuracy obtained using the fixed percentage 5% (*Ours fixed*) and at varying percentages of eigenvectors (*Ours*). The GraphZoom method is replicated using the parameters provided by the official code repository.

Regularizing behavior. Using $k \ll n$ eigenvectors in the construction of C has a regularizing effect on the map, akin to a low-pass filtering of the correspondence. It was already demonstrated in Nt and Maehara (2019) and Li et al. (2019) that smoothing signals can improve performance on graphs. Our results validate this idea once again. We analyze

this behaviour in Appendix A.3 and C, showing qualitative and quantitative results on the smoothing effect.

Table 2: *Knowledge distillation*: Results of node classification accuracy over multiple runs. We compare the spectra representation (Ours) with the methods proposed by Yang et al. (2022a) (GKD-G, GKD-R, GKD-S, PGKD). For Ours we also report the percentages of eigenvectors used in the spectral map.

	Cora	Citeseer	Amazon-photo	Amazon-computer	Coauthor-cs	Pubmed
Node-aware knowledge setting						
ORACLE	87.74 ± 1.41	70.35 ± 1.30	93.00 ± 0.63	90.90 ± 0.56	92.89 ± 0.41	86.33 ± 0.25
TEACHER	85.52 ± 0.51	68.91 ± 1.62	91.93 ± 0.11	89.34 ± 0.42	92.25 ± 0.45	85.16 ± 0.56
STUDENT	82.87 ± 2.09	69.19 ± 2.27	92.30 ± 0.40	85.41 ± 2.30	85.67 ± 2.28	85.09 ± 0.16
GKD-G	**88.08** ± 0.95	**71.07** ± 0.36	92.02 ± 0.41	**90.13** ± 0.19	**92.75** ± 0.61	85.97 ± 0.15
GKD-R	**88.13** ± 1.32	70.83 ± 1.89	**92.44** ± 0.43	88.25 ± 0.79	**92.13** ± 0.86	86.05 ± 0.42
GKD-S	87.64 ± 0.56	**71.15** ± 0.88	**92.44** ± 0.67	89.95 ± 0.50	92.26 ± 0.37	86.01 ± 0.27
PGKD	86.71 ± 0.82	68.95 ± 0.55	92.21 ± 0.67	89.80 ± 0.12	92.02 ± 0.14	**86.36** ± 0.34
Ours	**88.08** ± 1.11	**71.15** ± 1.13	**92.84** ± 0.28	**90.94** ± 0.50	**92.13** ± 0.37	**86.42** ± 0.39
(% eigs)	(12%)	(50%)	(25%)	(50%)	(50%)	(50%)
Edge-aware knowledge setting						
ORACLE	87.74 ± 1.41	70.35 ± 1.30	93.00 ± 0.63	90.90 ± 0.56	92.89 ± 0.41	86.33 ± 0.25
TEACHER	81.88 ± 1.49	67.11 ± 2.05	90.57 ± 1.04	87.47 ± 0.26	91.14 ± 0.54	83.19 ± 0.49
STUDENT	82.82 ± 0.31	70.47 ± 1.42	**92.42** ± 0.54	77.86 ± 3.20	83.38 ± 1.48	84.52 ± 0.31
GKD-G	**87.54** ± 0.23	**71.51** ± 0.82	91.76 ± 0.60	**89.53** ± 0.12	**91.98** ± 0.07	**86.09** ± 0.14
GKD-R	86.76 ± 1.48	71.11 ± 0.70	**92.12** ± 0.36	88.29 ± 0.58	91.73 ± 0.35	86.06 ± 0.64
GKD-S	**87.05** ± 1.20	71.31 ± 2.65	92.00 ± 0.53	88.49 ± 0.64	91.51 ± 0.47	86.07 ± 0.43
PGKD	86.21 ± 0.56	69.59 ± 0.68	**92.42** ± 0.31	89.30 ± 0.61	91.65 ± 0.25	**86.86** ± 0.48
Ours	86.26 ± 0.39	**71.47** ± 0.62	92.14 ± 0.24	**90.16** ± 0.46	**91.80** ± 0.33	85.81 ± 0.10
(% eigs)	(4%)	(4%)	(25%)	(50%)	(50%)	(50%)

4.3. Geometric Knowledge Distillation

The aim of Geometric Knowledge Distillation Yang et al. (2022a) is to transfer topological knowledge from a teacher model to a student model, which has only a partial vision of the graph. In particular, the teacher model is trained on $G_1 = (V_1, E_1)$ and the student on $G_2 = (V_2, E_2)$. In this scenario, we can exploit the spectral map to align the features that the teacher and student models are learning. For this purpose, we define the following loss:

$$||C\Phi_1^T x_t - \Phi_2^T x_s|| \tag{3}$$

where $x_t \in \mathrm{R}^{|V_1|\times d}$ and $x_s \in \mathrm{R}^{|V_2|\times d}$ are the features computed by the teacher and the student, $\Phi_1 \in \mathrm{R}^{|V_1|\times k}$ and $\Phi_2 \in \mathrm{R}^{|V_2|\times k}$ are the eigenvectors on the teacher and student graph respectively and $C \in \mathrm{R}^{k\times k}$ is the spectral map between the two graphs. We remark that both Φ_1 and Φ_2 are precomputed before training time.

In Table 2, we compare the student trained with Equation (3) with the methods proposed in Yang et al. (2022a) using different kernels: gaussian kernel (GKD-G), random kernel (GKD-R), sigmoid kernel (GKD-S) and parametric kernel (PGKD). In the first three rows, we also report the performance of the ORACLE model (trained and tested on G_1), TEACHER (trained on G_1 and tested on G_2) and STUDENT (trained and tested on G_2). We consider two settings: *node-aware* where the subgraph's nodes are a subset of the full

graph's node $V_2 \subset V_1$; *edge-aware* where the subgraph's edges are a subset of the full graph's edges, but the nodes are the same $E_2 \subset E_1$ and $V_2 = V_1$. In both cases, the partiality considered is 50%. We report the node classification accuracy over three runs with random initialization. We train all the models for 500 epochs.

In all the datasets, the spectral representation reaches an accuracy comparable with at least one of the methods proposed by Yang et al. (2022a). In particular, in the node-aware setting, the spectral map always performs as the first or second-best method. The edge-aware setting corresponds to a non-isomorphic transformation of the graph. In this case, as we will show in Section 5.2, the spectral representation still holds a compact representation, but at higher percentages of partiality, the correlation between the eigenspaces of the graphs tends to be weaker. As the results show, this can damage the performance of the spectral representation, even if not drastically. We believe this can lead to further investigation of the non-isomorphic mapping of graphs.

The spectral representation is also a faster method than Yang et al. (2022a). Since the eigenvector can be precomputed at training time, the only additional expense is a simple matrix multiplication. The spectral representation can reach a speedup of 200% compared to Yang et al. (2022a). In Appendix D we show the full table with the computation time per epoch and speed-ups.

5. Empirical results and analysis

So far we have seen how the spectral representation can be easily plugged into existing pipelines showing competitive performances. In this section, we analyse the structure of the spectral map under different kind of partialities to give further insights on its benefits.

5.1. Map structure

In Section 4.1, we highlighted how the preservation of the eigenspaces between a graph and a subgraph is reflected in the structure of the spectral map C. In 3D geometry processing, a similar behaviour was observed for the discrete Laplace-Beltrami operator under partiality transformations (Rodolà et al., 2017; Postolache et al., 2020); however, their theoretical analysis assumes the data to be Riemannian surfaces with a smooth metric – an assumption that does *not* hold in the case of general graphs.

In Appendix A.2, we directly compare the spectral maps computed on smooth surfaces and graphs. Interestingly, in both cases, we observe a slanted-diagonal structure. This structural pattern finds its theoretical explanation on smooth surfaces by applying Weyl's law, as demonstrated in Rodolà et al., 2017, Eq. 9. In contrast, the existence of this diagonal structure in graph-based maps remains enigmatic due to the complete absence of metric information. Despite this theoretical gap, we observe that this diagonal pattern persists even when subjected to various edge removal scenarios, hinting at profound algebraic implications.

5.2. Non-isomorphic subgraphs

In many practical settings, there are cases where the subgraph G_2 is contained in the bigger graph G_1 only up to some topological alterations; for example, in the graph learning liter-

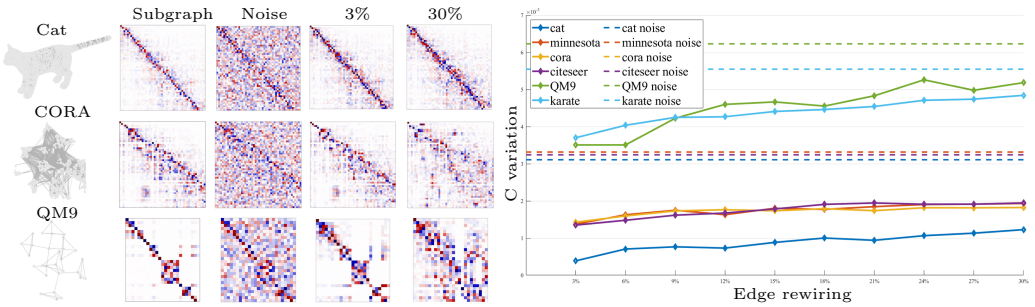


Figure 3: Robustness of the map to the simultaneous action of partiality and rewiring of the subgraph. We compare the addition of Gaussian noise with the impact of increasing rewiring on the spectral map C of size 50×50. On the left, we plot three graphs with their spectral map: no rewiring (Subgraph), the addition of Gaussian noise (Noise), 3% and 30% of edges rewired. On the right, we plot the variation of C at different percentages of rewiring (solid lines) and with the addition of noise (dashed lines) for each graph.

ature, topological perturbations frequently occur due to noise in the data or are explicitly obtained by rewiring operations (Chamberlain et al., 2021) or adversarial attacks (Jin et al., 2021) among others.

In Figure 1, we show the spectral map between Minnesota and a subgraph after rewiring (**iii**). We still observe a correspondence between the eigenvectors of the full graph and those of the subgraph. The spectral map has a sparse pattern, but it loosens up as the topological modifications increase. For this to be true, we expect small changes in graph connectivity to lead to small changes in the matrix coefficients. See Appendix B.2 for the formal definition.

In Figure 3, we evaluate the changes of the spectral map at increasing percentages of rewiring of a subgraph. We consider six graphs and compute a subgraph from each one. Then, we apply small incremental changes to the topology of the subgraphs, with increments of 3% of the total number of edges; the changes are performed by removing and adding random edges, obtaining new subgraphs G_i. The plot on the right shows how much the increasing topological changes affect the spectral map representation compared to adding Gaussian noise. In all the cases, the rewiring produces less variation in the spectral map than in adding Gaussian noise. In particular, the functional representation is more robust on larger graphs, such as cat (10000 nodes) or citeseer (2120 nodes), while on smaller graphs, such as QM9 (29 nodes) and Karate (34 nodes), removing or adding an edge has a more significant impact. This observation demonstrates the effectiveness of the spectral representation, especially on larger graphs. We show the complete qualitative analysis in Appendix B.2.

All the remarks directly depend on graph connectivity, and it is hard to find analogies for smooth surfaces. We conjecture that local topological transformations of a graph, while they can undoubtedly induce strong transformations of *some* of its Laplacian eigenspaces (similar to single-point perturbations on planar manifolds, see Filoche and Mayboroda (2012)), are

less likely to distort *all* the eigenspaces at once. This way, the spectral map matrix tends to maintain its global structure intact and exhibits local perturbations.

6. Conclusions

In this paper we highlighted the advantages of the spectral representation for encoding graph and subgraph maps, demonstrating its importance and efficacy in graph learning pipelines.

Further, while in this paper we showed extensive evidence that the spectral map representation bears a special structure depending on the type of partiality, currently we have not taken full advantage of this structure. When the task at hand requires seeking for the subgraph alignment, i.e. whenever the map is unknown, it may be possible to design stronger regularizers to induce sparsity in the matrix representation of the map. This is quite different from the better-known setting of 3D surfaces, where this sparse structure is typically just diagonal or slanted-diagonal.

In light of the increasing interest of the graph learning community toward spectral techniques, adopting a spectral representation for maps between graphs is a natural next step; it is simple to adopt, easy to manipulate, and memory-efficient and has the potential to become a fundamental ingredient in spectral graph learning pipelines in the near future.

Acknowledgments

This work is supported by the ERC grant no.802554 (SPECGEO), PRIN 2020 projects no.2020TA3K9N (LEGO.AI), and PNRR MUR project PE0000013-FAIR. S. Melzi has been partially supported by MUR under the grant "Dipartimenti di Eccellenza 2023-2027" of the Department of Informatics, Systems and Communication of the University of Milano-Bicocca, Italy and through the Academic Hardware Grant Program from the NVIDIA Corporation for the project "Learned representations for implicit binary operations on real-world 2D-3D data". R. Marin has been supported by the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No 101109330. L. Cosmo is supported by the IRIDE project from the Department of Environmental Science, Informatics and Statistics of Ca' Foscari University.

References

Yonathan Aflalo, Haim Brezis, and Ron Kimmel. On the optimality of shape and data representation in the spectral domain. *SIAM Journal on Imaging Sciences*, 8(2):1141–1160, 2015.

Mikhail Belkin and Partha Niyogi. Convergence of laplacian eigenmaps. In *Proc. NIPS*, pages 129–136, 2006.

Rickard Brüel-Gabrielsson, Mikhail Yurochkin, and Justin Solomon. Rewiring with positional encodings for graph neural networks. 2022.

Benjamin Chamberlain, James Rowbottom, Davide Eynard, Francesco Di Giovanni, Xiaowen Dong, and Michael Bronstein. Beltrami flow and neural diffusion on graphs. In

M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 1594–1609. Curran Associates, Inc., 2021.

Hau Chan and Leman Akoglu. Optimizing network robustness by edge rewiring: a general framework. *Data Mining and Knowledge Discovery*, 30(5):1395–1425, 2016.

Haochen Chen, Bryan Perozzi, Yifan Hu, and Steven Skiena. Harp: Hierarchical representation learning for networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Yuzhao Chen, Yatao Bian, Xi Xiao, Yu Rong, Tingyang Xu, and Junzhou Huang. On self-distilling graph neural network. *arXiv preprint arXiv:2011.02255*, 2020.

Luca Cosmo, Emanuele Rodolà, Michael M Bronstein, Andrea Torsello, Daniel Cremers, and Y Sahillioglu. Shrec'16: Partial matching of deformable shapes. *Proc. 3DOR*, 2(9): 12, 2016.

Chenhui Deng, Zhiqiang Zhao, Yongyu Wang, Zhiru Zhang, and Zhuo Feng. Graphzoom: A multi-level spectral approach for accurate and scalable graph embedding. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=r1lGO0EKDH.

Vijay Prakash Dwivedi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Graph neural networks with learnable structural and positional representations. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=wTTjnvGphYj.

Marcel Filoche and Svitlana Mayboroda. Universal mechanism for anderson and weak localization. *Proceedings of the National Academy of Sciences*, 109(37):14761–14766, 2012.

Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016.

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

William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. *CoRR*, abs/1706.02216, 2017b.

Judith Hermanns, Anton Tsitsulin, Marina Munkhoeva, Alex Bronstein, Davide Mottin, and Panagiotis Karras. Grasp: Graph alignment through spectral signatures. In *Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data*, pages 44–52. Springer, 2021.

Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015.

Wei Jin, Yaxing Li, Han Xu, Yiqi Wang, Shuiwang Ji, Charu Aggarwal, and Jiliang Tang. Adversarial attacks and defenses on graphs. *SIGKDD Explor. Newsl.*, 22(2):19–34, jan 2021. ISSN 1931-0145. doi: 10.1145/3447556.3447566.

Qimai Li, Xiao-Ming Wu, Han Liu, Xiaotong Zhang, and Zhichao Guan. Label efficient semi-supervised learning via graph filtering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9582–9591, 2019.

Jiongqian Liang, Saket Gurukar, and Srinivasan Parthasarathy. Mile: A multi-level framework for scalable graph embedding. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 15, pages 361–372, 2021.

Tong Man, Huawei Shen, Shenghua Liu, Xiaolong Jin, and Xueqi Cheng. Predict anchor links across social networks via an embedding approach. In *Ijcai*, volume 16, pages 1823–1829, 2016.

Julian J. McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. Image-based recommendations on styles and substitutes. *CoRR*, abs/1506.04757, 2015.

Andrew Kachites McCallum, Kamal Nigam, Jason Rennie, and Kristie Seymore. Automating the construction of internet portals with machine learning. *Information Retrieval*, 3 (2):127–163, 2000.

Hoang Nt and Takanori Maehara. Revisiting graph neural networks: All we have is low-pass filters. *arXiv preprint arXiv:1905.09550*, 2019.

Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas Guibas. Functional maps: a flexible representation of maps between shapes. *ACM Transactions on Graphics (TOG)*, 31(4):30:1–30:11, 2012.

Maks Ovsjanikov, Etienne Corman, Michael Bronstein, Emanuele Rodolà, Mirela Ben-Chen, Leonidas Guibas, Frederic Chazal, and Alex Bronstein. Computing and processing correspondences with functional maps. In *SIGGRAPH 2017 Courses*. 2017.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. nternational Conference on Knowledge Discovery and Data mining (KDD), 2014.

Mehmet Pilanci and Elif Vural. Domain adaptation on graphs by learning aligned graph bases. *IEEE Transactions on Knowledge and Data Engineering*, 2020.

E. Postolache, M. Fumero, L. Cosmo, and E. Rodolà. A parametric analysis of discrete hamiltonian functional maps. *Computer Graphics Forum*, 39(5):103–118, 2020. doi: https://doi.org/10.1111/cgf.14072.

Emanuele Rodolà, Luca Cosmo, Michael M Bronstein, Andrea Torsello, and Daniel Cremers. Partial functional correspondence. *Computer Graphics Forum*, 36(1):222–236, 2017.

Rohit Singh, Jinbo Xu, and Bonnie Berger. Global alignment of multiple protein interaction networks with application to functional orthology detection. *Proceedings of the National Academy of Sciences*, 105(35):12763–12768, 2008.

Fu-Dong Wang, Nan Xue, Yipeng Zhang, Gui-Song Xia, and Marcello Pelillo. A functional representation for graph matching. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

H. Weyl. Ueber die asymptotische verteilung der eigenwerte. *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse*, 1911: 110–117, 1911. URL http://eudml.org/doc/58792.

Cheng Yang, Jiawei Liu, and Chuan Shi. Extract the knowledge of graph neural networks and go beyond it: An effective knowledge distillation framework. In *Proceedings of the Web Conference 2021*, pages 1227–1237, 2021.

Chenxiao Yang, Qitian Wu, and Junchi Yan. Geometric knowledge distillation: Topology compression for graph neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022a.

Yi Yang, Yanqiao Zhu, Hejie Cui, Xuan Kan, Lifang He, Ying Guo, and Carl Yang. Data-efficient brain connectome analysis via multi-task meta-learning. *arXiv preprint arXiv:2206.04486*, 2022b.

Yiding Yang, Jiayan Qiu, Mingli Song, Dacheng Tao, and Xinchao Wang. Distilling knowledge from graph convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7074–7083, 2020.

Ke Zhang, Carl Yang, Xiaoxiao Li, Lichao Sun, and Siu Ming Yiu. Subgraph federated learning with missing neighbor generation. *Advances in Neural Information Processing Systems*, 34:6671–6682, 2021.

Appendix A. Interpretation of the spectral map matrix

A.1. Spectral maps on surfaces

Consider two smooth manifolds $\mathcal{M}$ and $\mathcal{N}$, and let $T : \mathcal{N} \to \mathcal{M}$ be a point-to-point map between them. Given a scalar function $f : \mathcal{M} \to \mathbb{R}$, the map T induces a spectral mapping via the composition $g = f \circ T$, which can be seen as a linear map $T_F : f \mapsto g$ from the space of functions on $\mathcal{M}$ to the space of functions on $\mathcal{N}$. As a linear map, the functional T_F admits a matrix representation after choosing a basis for the two function spaces.

To this end, consider a discretization of $\mathcal{M}$ and $\mathcal{N}$, with vertices V_1 and V_2 respectively, and the corresponding discretized version of their Laplace-Beltrami operators (LBOs) (the counterpart of the graph Laplacian on smooth manifolds). The first k eigenfunctions of the two LBOs can be concatenated side by side as columns to form the matrices $\Phi \in \mathbb{R}^{|V_1| \times k}$ and $\Psi \in \mathbb{R}^{|V_2| \times k}$. Further, assume the pointwise map T is available and encoded in a binary matrix S, such that $S(y, x) = 1$ if $y \in V_2$ corresponds to $x \in V_1$, and 0 otherwise. By choosing Φ and Ψ as bases, the spectral map T_F can be encoded in a small $k \times k$ matrix C via the change of basis formula:

$$C = \Psi^\dagger S \Phi \,, \tag{4}$$

where $\dagger$ is the Moore-Penrose pseudoinverse. The size of C does *not* depend on the number of points in $\mathcal{M}$ and $\mathcal{N}$, but only on the number k of basis functions. In other words, C represents the linear transformation that maps the coefficients of any given function $f : \mathcal{M} \to \mathbb{R}$ expressed in the eigenbasis Φ, to coefficients of a corresponding function $g : \mathcal{N} \to \mathbb{R}$ expressed in the eigenbasis Ψ.

When the pointwise similarity S is unknown, one can directly compute the matrix C as the solution of a regularized least-squares problem with k^2 unknowns, given some input features on the two surfaces (e.g., landmark matches or local descriptors). For further details we refer to Ovsjanikov et al. (2012, 2017).

A.2. Comparison with smooth surfaces

In the case of smooth surfaces, it has been shown (Rodolà et al., 2017) that the sparsity pattern of matrix C can be well approximated by a simple formula. Given a surface $\mathcal{M}$ and an isometric part $\mathcal{N}$, the matrix C is approximately diagonal, with diagonal angle α proportional to the ratio of surface areas:

$$\alpha \sim \frac{\text{Area}(\mathcal{N})}{\text{Area}(\mathcal{M})} \,. \tag{5}$$

As a a special case, full-to-full isometric shape matching yields a diagonal matrix C, since $\text{Area}(\mathcal{N}) = \text{Area}(\mathcal{M})$. This result comes directly from an application of Weyl's asymptotic law for Laplacian eigenvalues of smooth manifolds (Weyl, 1911), which relates the eigenvalue growth to the surface area of the manifold via the relation:

$$\lambda_\ell \sim \frac{(2\pi)^2}{\text{Area}(\mathcal{M})^{2/d}} \ell^{2/d} \,, \qquad \ell \to \infty \tag{6}$$

where d is the dimension of the manifold ($d = 2$ for surfaces). We refer to Rodolà et al., 2017, Eq. 9 for additional details pertaining surfaces.

However, Weyl's law (Equation 6) does *not* hold for graphs, since there is not a well-defined notion of "area" of a graph. In fact, when we work with graphs and subgraphs, we observe that matrix C does not necessarily follow a diagonal pattern. More general sparse structures are observed in the coefficients of C, but an explanation rooted in differential geometry is not readily available.

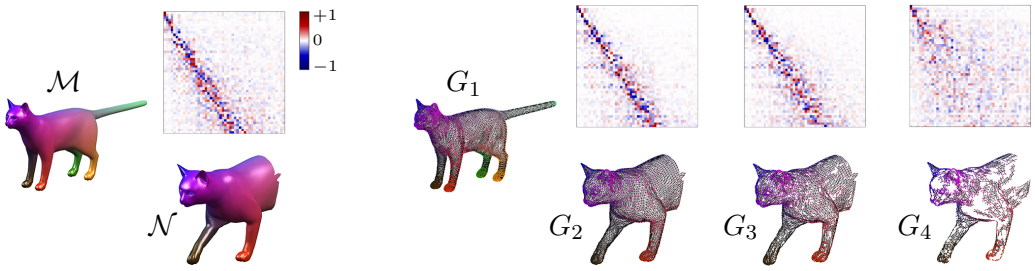


Figure 4: A spectral map between a full and a partial surface (left) compared to the spectral maps between a graph and three different subgraphs (right).

In Figure 4 we show several examples of matrix C for different subgraphs. On the left we show the spectral map matrix between a smooth surface $\mathcal{M}$ and a deformed part $\mathcal{N}$: the slanted-diagonal structure suggests that the eigenspaces of $\mathcal{M}$ are mostly preserved in $\mathcal{N}$. On the right, we show the spectral map matrices between a graph G_1 and different subgraphs: G_2 is obtained by removing 40% of the nodes of G_1, while G_3, G_4 are obtained by removing 55% and 80% of the edges from G_2 respectively. The slanted-diagonal structure can still be observed and gets dispersed only at very high partiality. In the graphs, corresponding nodes have the same color. The slanted-diagonal structure of the map between $\mathcal{M}$ and $\mathcal{N}$ is explained by an application of Weyl's law to 2-dimensional Riemannian manifolds. However, there is no theoretical counterpart to explain the map structure between G_1 and its subgraphs, due to the complete absence of metric information about the underlying surface: the eigenfunctions are computed *purely* from the graph connectivity. Yet, the diagonal structure is preserved even under rather dense removal of edges, suggesting deeper algebraic implications.

One might legitimately ask whether the presence of a structure in the maps of Figure 4 is due to the specific choice of the data, where the subgraphs derive from a 3D mesh (although the normalized graph Laplacian dismisses any edge length information) and where the type of partiality resembles a neat 'cut' (although we also perform random edge removals). However, the same behavior is also observed with abstract graphs. In Figure 6, we report additional examples with large abstract graphs undergoing partiality transformations, showing that clear patterns appear rather consistently across different datasets.

A.3. Number of eigenvectors

Given two graphs G_1 and G_2 with m and n nodes respectively, the node-to-node map S has size $n \times m$, thus scaling quadratically with the number of nodes.

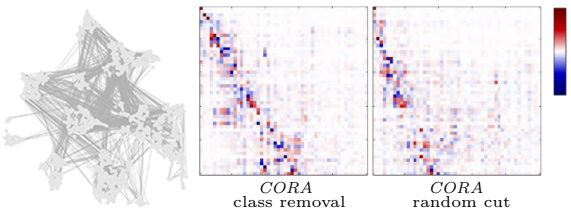


Figure 5: Spectral maps between CORA and two different subgraphs.

By contrast, matrix C as defined in Equation 1 has dimensions that only depend on the number of Laplacian eigenvectors encoded in the matrices Φ_1, Φ_2. If one chooses the first $k_1 \ll m$ Laplacian eigenvectors for G_1 and the first $k_2 \ll n$ Laplacian eigenvectors for G_2, the size of C is $k_2 \times k_1$. Observe that C is rectangular in general, but can be made square by choosing $k_1 = k_2$ if so desired.

The experiments in Figure 10 and 7 show that as the number of eigenvectors increases, the performance also increases. The Mean Average Precision (MAP) is defined as $\frac{1}{n}\sum_{i=1}^{n}\frac{1}{ra_i}$ where ra_i is the rank (position) of positive matching node in the sequence of sorted candidates. In particular, Figure 7 demonstrates that, in most of the cases, a low percentage of eigenvectors (about 5%) suffices to retrieve a good node-to-node correspondence; while at 50% of the eigenvectors on all graphs the error is above 90%. As a general guideline, in this paper we typically use $k = 20 \sim 50$ for a graph with 1000 nodes, leading to an especially compact representation C.

A.4. Choice of Laplace operator

A spectral map can be computed from the eigenbasis of any linear operator. In this paper, we use the symmetrically normalized graph Laplacian $\mathcal{L} = I - D^{\frac{1}{2}}AD^{\frac{1}{2}}$. A valid alternative is the standard Laplacian $L = D - A$, which shows similar behavior to the normalized counterpart. At a practical level, we observed that the Laplacian L suffers from more problems of high multiplicity at lower frequencies, see Figure 8.

In the special case where the graph is constructed on top of a point cloud sampled from a (possibly high-dimensional) manifold $\mathcal{M}$, it has been shown that the eigenvectors of the normalized graph Laplacian converge to the eigenfunctions of the Laplace-Beltrami operator on $\mathcal{M}$ (Belkin and Niyogi, 2006). However, as discussed in Appendix A.2, our case is more general. We consider generic abstract graphs without an explicit underlying manifold, i.e. we do not construct our graphs from input point clouds. Further, in Belkin and Niyogi (2006) it is assumed that $\mathcal{M}$ is a compact infinitely differentiable Riemannian submanifold of $\mathbb{R}^d$ *without* boundary, meaning that partiality transformations, which are the main focus of this paper, are not considered.

Appendix B. Dataset and implementation details

In this section we report additional details about the experimental setup used in the main manuscript.

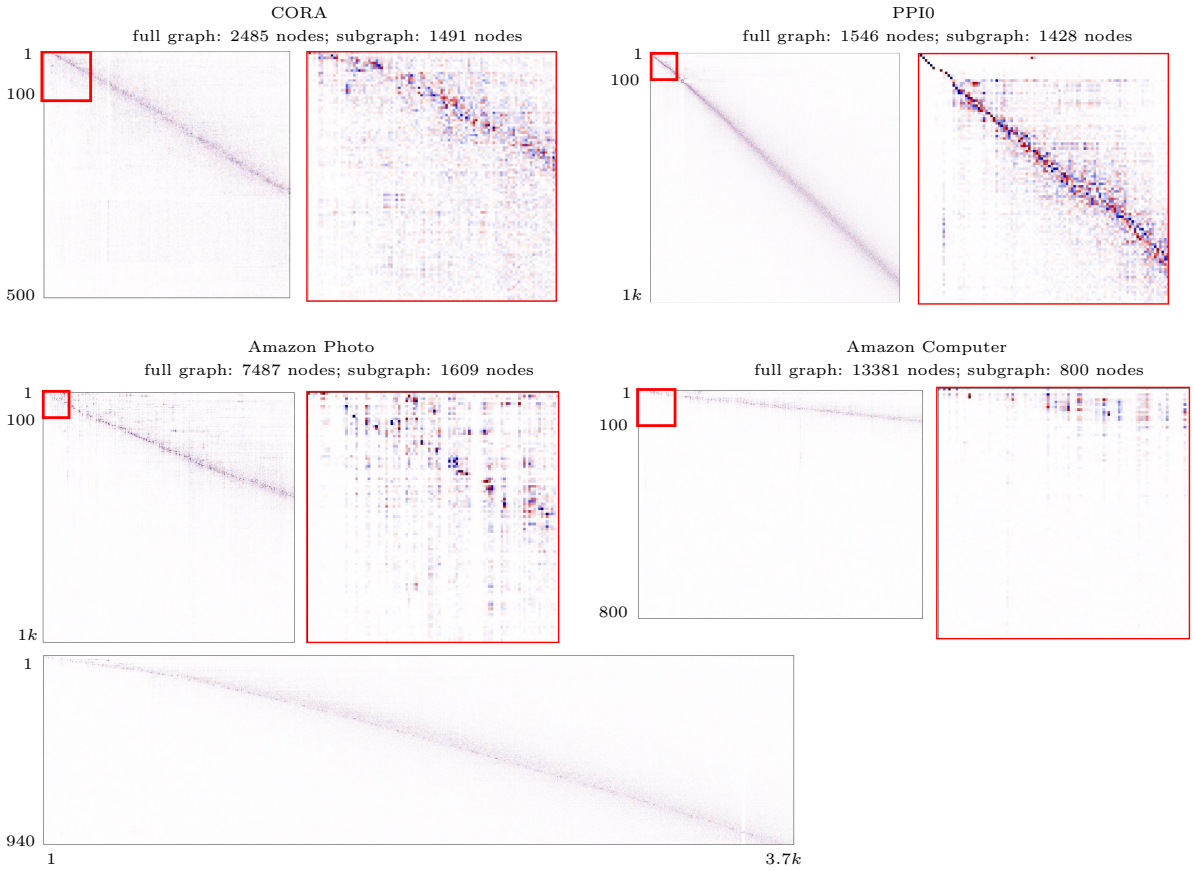


Figure 6: spectral maps computed over abstract graphs from 4 different datasets (CORA (McCallum et al., 2000), PPI0 (Hamilton et al., 2017b), Amazon Photo (McAuley et al., 2015) and Amazon Computer (McAuley et al., 2015)), showing a clear pattern in all cases. For each dataset, we compute the spectral map matrix C between the complete graph and a subgraph; the subgraph is obtained according to a semantic criterion depending on the dataset, e.g., for Amazon Photo, by considering the subgraph of nodes belonging to the same product category. For each spectral map matrix C, we also show a zoom-in (framed in red). All the matrices are sparse, and have a clean structure that in some cases approximates a slanted diagonal. The wide matrix on the bottom is computed on Amazon Photo (using a different subgraph than the one used in the example above it), and shows that the sparse behavior is maintained throughout the entire spectrum.

B.1. Datasets

In Table 3 we sum up the main statistics across all the datasets and benchmarks used in our experiments. In addition to number of nodes, number of edges, graph diameter and

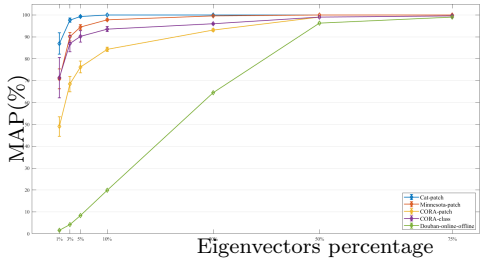


Figure 7: MAP(%) of the correspondence on different datasets at increasing number of eigenvectors (expressed as percentages, growing from 1% to 75%). The correspondences are obtained from ground-truth spectral maps.

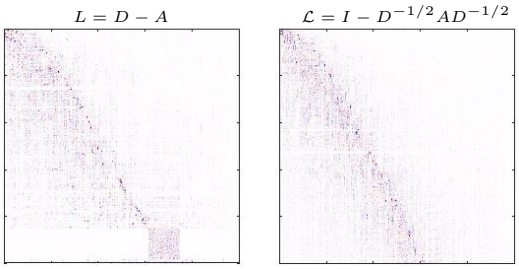
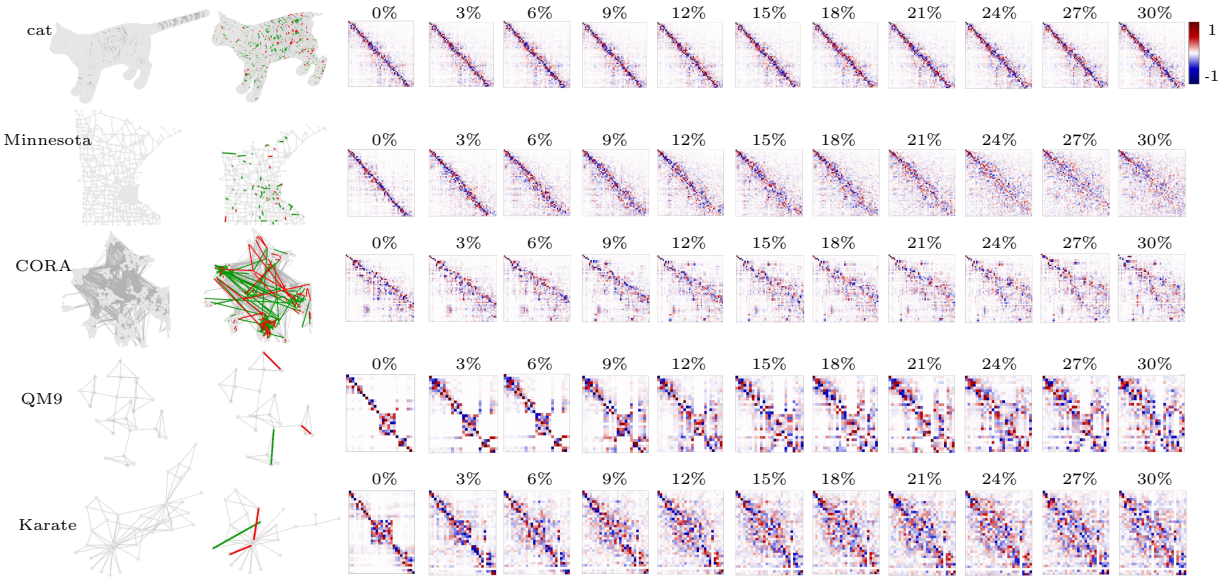


Figure 8: Spectral maps computed with two different Laplacians between the CORA graph and its subgraph.

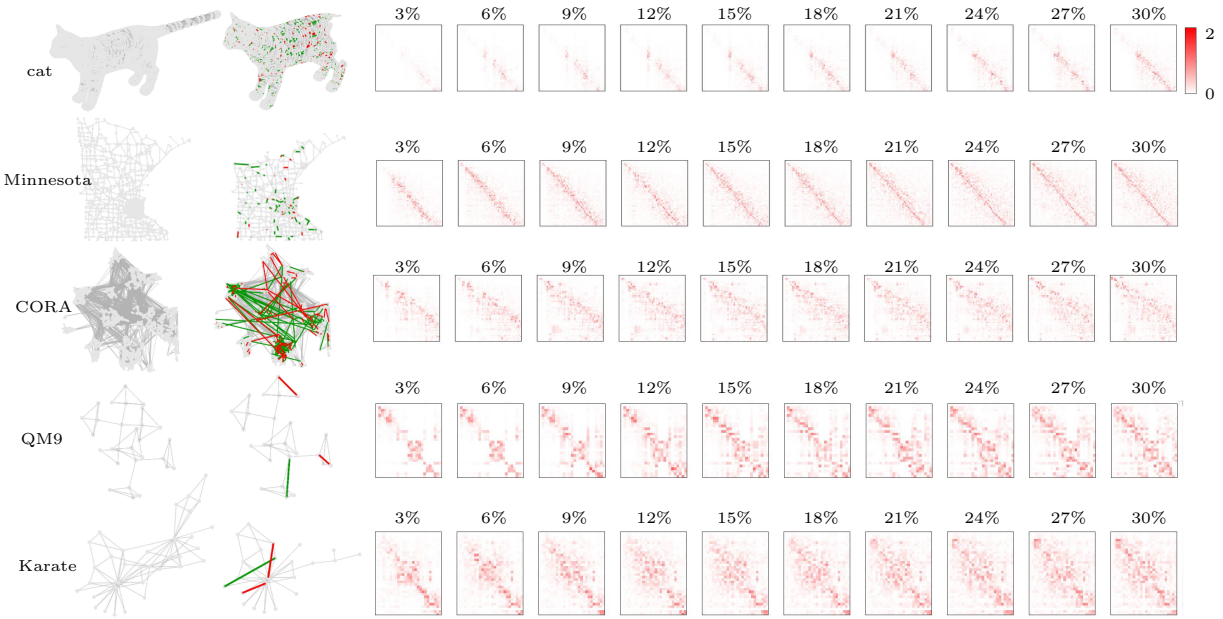
average node degree, in the table we also report the application domain of each dataset, the task where they are used, the type and number of node-wise features (where used). Since PPI and QM9 are collections of graphs, we used only a subset. In particular, from the PPI dataset we used the first and fourteenth graphs (specified with 0 and 13 in the experiments). The Cat graph is derived from the corresponding mesh of the SHREC'16 Partial Deformable Shapes benchmark (Cosmo et al., 2016).

B.2. Robustness to rewiring

In this Section, we formally define the connectivity changes and spectral map robustness used in Section 5.2. Given two graphs $G = (V, E)$ and $G' = (V', E')$, we measure the amount of change from G to G' as the (minimum) number of edits needed to transform E to E', divided by $|E|$: $\frac{(|E-E'|+|E'-E|)}{|E|}$. In our experiments, we consider small changes in the graph connectivity as a perturbation of 3% of the edges. The rewiring operation we applied to the graphs consists of the deletion or addition of the same number of edges.



(*a*) The plotted matrices represent the spectral map between the full and partial graphs from 0% to 30% of rewiring, showing the effect of rewiring on the spectral map structure.



(*b*) The plotted matrices encode the element-wise error of the spectral map after the topological perturbations. Error is encoded as color, growing from white to red.

Figure 9: Robustness of the map to the simultaneous action of partiality and rewiring of the subgraph. The rewiring operations are increasingly **stronger**, with increments of 3% of the total number of edges (starting from 3% and reaching 30%). The second column shows one representative example (per dataset) of such topological modifications, depicting the added edges in green, and the removed edges in red. The plotted matrices represent the spectral map after the topological perturbations, showing the effect of rewiring on the spectral map structure.

Table 3: Summary of statistics about the datasets used in our experiments.

Dataset	Nodes	Edges	Diameter	Average degree	Domain	Task	Features	Number of features
QM9	29	47	6	3.24	Chemistry	Graph regression	-	-
Karate	34	78	5	4.59	Social networks	Node classification	-	-
PPI 0	1546	17699	8	21.90	Chemistry	Graph regression	Gene attributes	50
Citeseer	2120	3731	28	3.50	Citation networks	Node classification	Bag-of-Words	3703
Cora	2485	5069	19	4.08	Citation networks	Node classification	-	-
Minnesota	2635	3298	98	2.5	Roadmap	-	-	1,433
PPI 13	3480	56857	8	31.68	Chemistry	Graph regression	Gene attributes	50
Douban	3906	8164	13	4.18	Social networks	Network alignment	-	-
Amazon Photo	7487	119044	11	31.80	Co-purchase	Node classification	Bag-of-Words	745
Cat	10000	19940	86	5.99	Geometry processing	Shape matching	-	-
FraudAmazon	11944	4417576	4	739.71	Product reviews	Fraud detection	Bag-of-Words	25
Amazon Computer	13381	245778	10	36.74	Co-purchase	Node classification	Bag-of-Words	767
Coauthor-cs	18333	163,788	24	8.93	Citation networks	Node classification	Bag-of-Words	6,805
Pubmed	19717	88,648	18	4.5	Citation networks	Node classification	Bag-of-Words	500

We define the difference between the spectral map C and C' as $\|C - C'\|_F^2$. Note that there is ambiguity in the sign of the eigenfunctions of C'; to factor it out from the error computation, we use the sign that minimizes the error.

In Figure 9 we show the spectral maps generated from the experiment in Figure 3. Figure 9(a) shows the spectral map between the full and partial graphs from 0% to 30% of rewiring; Figure 9(b) shows the variation in the functional representation between the non-rewired case and the different percentages of rewiring.

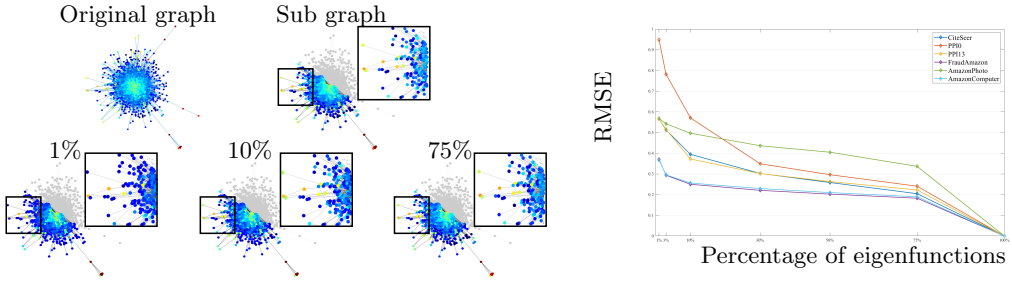


Figure 10: RMSE obtained by transferring positional encodings (PE) using the spectral map with an increasing amount of eigenfunctions. On the left, we show a qualitative example of signal transfer on PPI0. The first row shows the full graph and the partial graph, with the PE plotted on top. The bottom row shows the results of signal transfer with different percentages of eigenfunctions. On the right, we plot the RMSE at increasing percentages of eigenfunctions.

Appendix C. Hierarchical Graph Embedding: additional results

Figure 11 shows on Cora an example of embedding transferred from the coarsest level to the next in the hierarchy. We use a spectral map computed with the first 5% eigenvectors (last column). The spectral transfer performs an evident smoothing on the embedding, compared to Graphzoom (middle column). As further evidence, we plot the classification

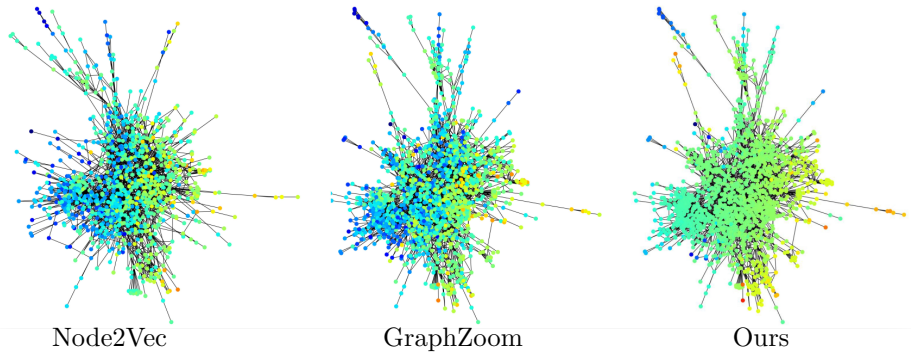
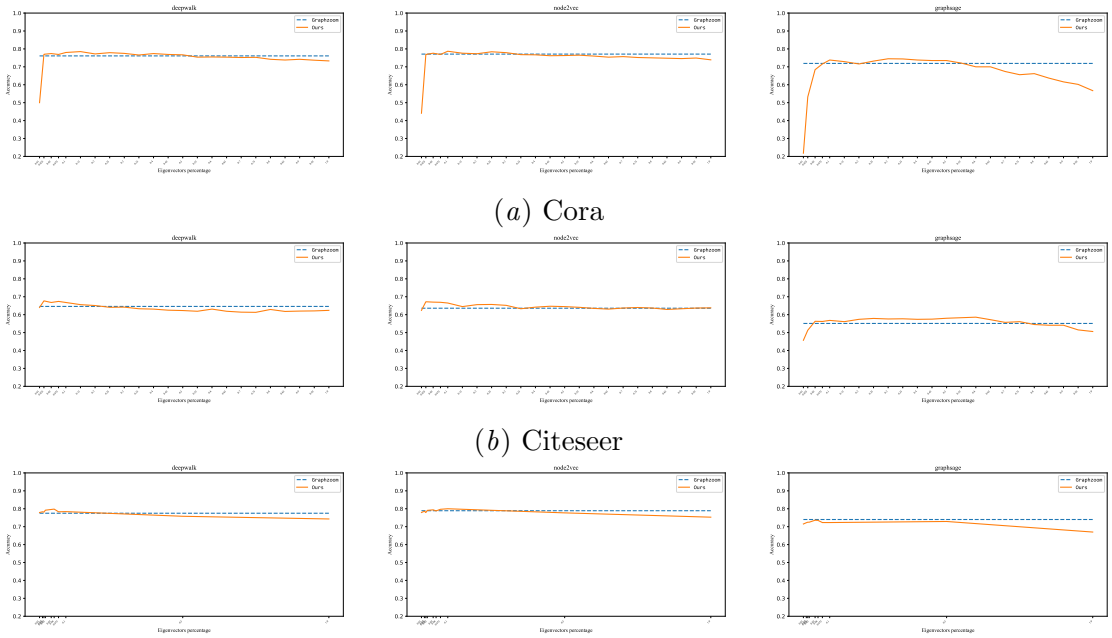


Figure 11: The Node2Vec embedding is, from left to right, applied to the coarsened graph, transferred to the full graph with GraphZoom and with the spectral map. We remark the smoothing effect of the spectral representation.



(a) Cora

(b) Citeseer

(c) Pubmed

Figure 12: Node classification accuracy on the task of Hierarchical Graph Embedding at different percentages of eigenvectors.

accuracy at varying percentages of eigenvectors (Figure 12 in Appendix C). Importantly, the performance peaks with few eigenvectors and then decreases when increasing the number of eigenvectors up to the complete base. In particular, when we use *all* the eigenvectors Φ_1 and Φ_2 to construct C, Equation 1 corresponds to an orthogonal change of basis; therefore, the representations S and C are equivalent and have the same dimensions. Truncating the bases to the first k_1 and k_2 eigenvectors, as described in Appendix A.3, yields a low-rank approximation $C \approx S$.

In Figure 12, we report the accuracy performance for different percentages of eigenvectors in the experiment of Section 4.2. Even if, the regularizing effect is desirable in many cases but is traded off for a loss in accuracy if a precise node-to-node correspondence is desired. On the one hand, if the map C is used to transfer a smooth signal (e.g. node-wise features like spectral positional encodings or carrying semantic information depending on the data), then the loss in accuracy is negligible since Laplacian eigenvectors are optimal for representing smooth signals (Aflalo et al., 2015); on the other hand, transferring non-smooth signals via a small C has the effect of filtering out the high frequencies. If high frequencies are desired, it is often sufficient just to increase the values of k_1, k_2, leading to a bigger matrix C. The performance of the spectral map rapidly increases at low percentages demonstrating the need for a few eigenvectors to obtain a good embedding lifting. When the percentages are higher than 50% the accuracy decreases reaching the values of the node-to-node map at 100%. This phenomenon demonstrates that the spectral map can approximate the node-to-node map at 100% eigenvectors, but it is not the most convenient representation for the Hierarchical Embedding on graphs.

C.1. Signal Transfer

In Section 4.2 and 4.3, we leveraged Equation 2 to transfer information between graphs. To better understand how the spectral representation afflicts the transferred signal, in Figure 10, we analyze the spectral map transfer performance while increasing the number of eigenfunctions used for the map representation. We evaluate the fidelity of the transferred signal with the Root Mean Squared Error between the transferred signal $\hat{g}$ and the ground truth signal g (obtained via the ground truth node-to-node correspondence):

$$RMSE = \sqrt{\frac{1}{n}\sum_{i}^{n} (g(i) - \hat{g}(i))^2}\,, \tag{7}$$

where n is the number of nodes in the subgraph. We consider pairs composed of the original graph and a series of subgraphs extracted according to a semantic criterion, e.g., nodes belonging to the same class or nodes connected by the same edge type. Motivated by the results from Brüel-Gabrielsson et al. (2022), we transfer the Random Walk Positional Encoding (Dwivedi et al., 2022) computed on the full graphs to the subgraphs. We normalize each dimension of the node features of the original graph to exhibit zero mean and unitary standard deviation throughout all the nodes and then transfer this signal through Equation 2. In Figure 10, we can see how the Root Mean Squared Error between the spectral map and the ground truth transfer decreases as the number of eigenfunctions increases. In particular, the error is almost steady between 30% and 75%. This demonstrates the

convenience of using fewer eigenvectors. The qualitative examples on the left of Figure 10 portray the transferred signal on PPI0. The transfer reaches a good approximation at 1% of the eigenfunctions, while at 10% and 75% they are almost identical. This behaviour demonstrates that using a compact representation with few eigenvectors can approximate the signal well.

Appendix D. Geometric Knowledge Distillation: additional results

Table 4: Computation time per epoch in the task of Knowledge Distillation. For each methods we report the mean time per epoch in milliseconds and the speed up with respect to Ours (5%). For the spectral map, we report the performance with both 5% and 10% of eigenvectors.

	Cora	Citeseer	Amazon-photo	Amazon-computer	Coauthor-cs	Pubmed	Coauthor-physics
ORACLE	2.47 (54%)	2.68 (54%)	3.75 (47%)	6.27 (40%)	5.22 (51%)	2.75 (50%)	10.84 (48%)
TEACHER	2.47 (54%)	2.68 (54%)	3.75 (47%)	6.27 (40%)	5.22 (51%)	2.75 (50%)	10.84 (48%)
STUDENT	2.81 (47%)	2.95 (49%)	3.18 (55%)	4.32 (59%)	5.24 (51%)	2.84 (48%)	9.24 (56%)
GKD-G	7.28 (−37%)	7.67 (−32%)	25.79 (−265%)	15.25 (−45%)	18.80 (−76%)	14.76 (−169%)	39.66 (−91%)
GKD-R	9.51 (−79%)	9.86 (−69%)	21.78 (−208%)	16.85 (−60%)	20.27 (−90%)	16.46 (−200%)	41.29 (−99%)
GKD-S	6.27 (−18%)	6.51 (−12%)	17.31 (−145%)	13.90 (−32%)	17.45 (−64%)	13.68 (−149%)	38.35 (−85%)
PGKD	7.90 (−48%)	8.19 (−40%)	17.53 (−148%)	18.44 (−75%)	20.36 (−91%)	15.56 (−183%)	45.02 (−117%)
Ours (5%)	5.32 (0%)	5.83 (0%)	7.07 (0%)	10.53 (0%)	10.66 (0%)	5.50 (0%)	20.77 (0%)
Ours (10%)	5.51 (−4%)	6.18 (−6%)	7.20 (−2%)	10.63 (−1%)	10.97 (−3%)	5.83 (−6%)	21.97 (−6%)

In Table 4 we show the mean epoch time registered during training. For each method and datset we report both the time in millisecond and the speed up compared to *Ours (*5%*)*. The spectral representation is able to reach a speed up of 200% in some cases, demonstrating its convenience in terms of computation efficiency.