SIAM J. APPL. MATH.
Vol. 85, No. 2, pp. 413–432

ELLIPSOIDAL EMBEDDINGS OF GRAPHS*

MICHAËL FANUEL†, ANTOINE ASPEEL‡, MICHAEL T. SCHAUB§, AND JEAN-CHARLES DELVENNE¶

Abstract. Due to their flexibility to represent almost any kind of relational data, graph-based models have enjoyed tremendous success over the past decades. While graphs are inherently only combinatorial objects, however, many prominent analysis tools are based on the algebraic representation of graphs via matrices such as the graph Laplacian, or on associated graph embeddings. Such embeddings associate to each node a set of coordinates in a vector space, a representation that can then be employed for learning tasks such as the classification or alignment of the nodes of the graph. As the geometric picture provided by embedding methods enables the use of a multitude of methods developed for vector space data, embeddings have thus gained interest from a theoretical as well as a practical perspective. Inspired by trace optimization problems, often encountered in the analysis of graph-based data, here we present a method to derive ellipsoidal embeddings of the nodes of a graph, in which each node is assigned a set of coordinates in a hyperellipsoid. Our method may be seen as an alternative to popular spectral embedding techniques, with which it shares certain similarities we discuss. To illustrate the utility of the embedding we conduct a case study in which we analyze synthetic and real world networks with modular structure, and compare the results obtained with known methods in the literature.

Key words. graph embedding, community detection, complex networks

MSC codes. 68R10, 05C82

DOI. 10.1137/24M1648363

1. Introduction: Graphs and embeddings.

Graphs enable us to conceptualize many different complex systems in a simple and compact manner. A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of a set of vertices (or nodes) $\mathcal{V}$ and a set of edges (or links) $\mathcal{E}$. The node set $\mathcal{V}$ is used to denote the entities present in the system, and the edges in the set $\mathcal{E}$ designate the interactions between these entities. By specifying a suitable set of nodes and edges, most types of relational data can be abstracted as a graph. Accordingly, graphs have enjoyed enormous success as mathematical modeling tools over the last decades, pervading essentially all areas of science [4, 9, 19, 37, 52], from neurobiology [51] to statistical physics [2].

Arguably, a large part of the success of graphs as modeling tools is due to the minimal, yet versatile, mathematical structure of graphs. We may enrich simple graphs for additional modeling flexibility, e.g., by allowing for weightings or directionality of the edges, or adding some form of multilayer structure. Yet, when discussing graphs

*Received by the editors March 21, 2024; accepted for publication (in revised form) October 3, 2024; published electronically March 3, 2025.
https://doi.org/10.1137/24M1648363
Funding: The work of the first author was supported by ERC grant BLACKJACK (ERC-2019-STG-851866). The work of the third author was supported by the European Union's Horizon 2020 research and innovation program under Marie Sklodowska-Curie grant 702410 and by the Ministry of Culture and Science (MKW) of the German State of North Rhine-Westphalia (NRW Rückkehrprogramm). The work of the fourth author was supported by the Research Project PDR TheCirco of the National Fund for Scientific Research (F.R.S.-FNRS) of Belgium.
†Université de Lille, CNRS, Centrale Lille, UMR 9189-CRIStAL, F-59000 Lille, France (michael.fanuel@univ-lille.fr).
‡Electrical Engineering and Computer Science Department, University of Michigan, Ann Arbor, MI 48109 USA (antoinas@umich.edu).
§RWTH Aachen University, North-Rhine Westphalia, Germany (schaub@cs.rwth-aachen.de).
¶ICTEAM and CORE, Université catholique de Louvain, Louvain, Belgium (jean-charles.delvenne@uclouvain.be).

we often do not think of them in a purely combinatorial fashion. Rather, we tend to reason about graphs in terms of their algebraic representations as matrices, such as an adjacency matrix or a Laplacian, or we consider them in the form of visualizations via diagrams.

Though not inherent to the definition of graphs, in practice, both the algebraic and the visual representation are undeniably important for theory and applications. Algebraic representations of graphs are, for instance, essential for computations and provide links to tools from matrix theory such as spectral analysis that enables a richer understanding of graphs. Similarly, when talking about specific graph structures such as clusters, we often provide geometrical pictures that are supposed to convey the graph structure visually. However, finding a good visualization of a graph in such a Euclidean space is not an easy task, as the (in)famous "hairball" pictures encountered when visualizing many large graphs highlight; see, e.g., Figure 2 (top right).

Before defining formally ellipsoidal embeddings, we provide below a first intuitive picture from the perspective of graph drawing.

1.1. The ellipsoidal embedding. We identify the node set $\mathcal{V}$ of a graph with the natural numbers $\{1,\dots,n\}$. We want to represent the vertex i with a *row* vector $h_i \in \mathbb{S}^{d_0-1} \subset \mathbb{R}^{d_0}$ for all $i = 1,\dots,n$, where $d_0 \geq 2$ is fixed. The integer d_0 is a priori the dimension of the embedding space, as $\mathbb{S}^{d_0-1}$ denotes the d_0-1 dimensional Euclidean unit sphere. The vectors h_i are obtained by minimizing the energy

$$E(H) = -\sum_{i,j=1}^{n} M_{ij}\langle h_i, h_j\rangle, \tag{1.1}$$

which depends on their Euclidean dot product $\langle h_i, h_j\rangle = h_i h_j^\top$ and where M is a descriptor matrix of the graph, such as a modularity matrix or a Laplacian-based matrix as we detail in what follows.

Intuitively, the energy (1.1) is minimized if nodes in the graph for which $M_{ij} > 0$ are positioned close to each other on the sphere $\mathbb{S}^{d_0-1}$, while pairs of nodes for which $M_{ij} < 0$ will repel each other. Unlike the usual force directed visualizations which place the nodes of a graph in the plane with attracting and repelling forces, the positions of the nodes are here constrained to be within a compact set and the force between two nodes i and j depends on the angle between the position vectors h_i and h_j. The nature of the coupling between the nodes is dictated by the choice of the matrix M, which is taken to be a descriptor matrix of the graph, such as the modularity matrix Q or a normalized Laplacian $\mathcal{L}$ that are discussed hereafter. Typically, $M_{ij} > 0$ (resp., < 0) if i and j are strongly (resp., loosely) connected.

To illustrate the embedding, we consider the example graph shown in Figure 1: a toy network arranged in a set of three groups of four nodes (left). By computing a spherical embedding of this graph, we obtain a coordinate vector for each node, interpreted as a "spin" variable valued on $\mathbb{S}^1$; see Figure 1, left, for which $d_0 = 2$. Alternatively, we can consider all those coordinates on a single hypersphere as in Figure 1, right, illustrating the proposed embedding. As should be apparent from Figure 1, neighboring nodes that are more tightly coupled in the graph tend to align their spins. For convenience, the vectors h_i are viewed as the rows of a matrix $H \in \mathbb{R}^{n\times d_0}$, that is, $h_i = H_{i*}$. Then, the energy minimization (1.1) can be rephrased as the following maximization problem:

$$\underset{H\in\mathbb{R}^{n\times d_0}}{\text{maximize}}\ \mathrm{Tr}\left(H^\top M H\right) \text{ subject to } \|H_{i*}\|_2 = 1 \text{ for all } 1 \leq i \leq n. \tag{1.2}$$

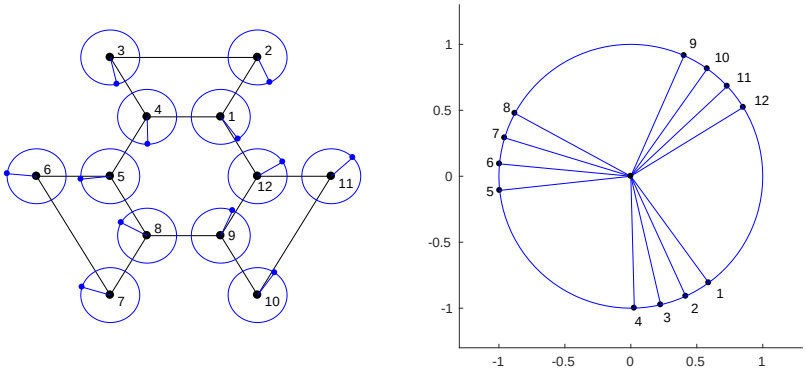

FIG. 1. *Modularity-based spherical embedding. Left: A spherical embedding—see Definition* 1*—on the depicted graph (black lines), which yields a coordinate vector* h_i *for each node* i*, interpreted as a "spin" attached to each node (blue). Right: Each of these spins* h_i *is drawn on the same hypersphere, giving rise to the depicted spherical embedding. The alignment of the spins* h_i *reflects some of the neighborhood structure in the graph. Here we chose* $d_0 = 2$ *and the graph descriptor matrix is the modularity* $M = Q$*; see* (2.3).

Even though the embedding space is a priori of dimension $d_0 - 1$ which can be as large as we want, we observe that an optimal solution $H^\star$ of (1.2) corresponds to an embedding on a subspace of *lower effective dimension* $d_{\text{eff}} \leq d_0$, that is, the embedding is effectively on $\mathbb{S}^{d_{\text{eff}}-1}$.

As an illustration of the low dimensionality of the embedding in the case of a real networks, the Power Grid of Europe graph (with $n = |\mathcal{V}| = 2712$ and $|\mathcal{E}| = 3580$) is embedded in Figure 2, while a list of other real networks where our method has been applied is given in Table SM1 in the supplementary material. To show that this embedding is indeed quite different to the often considered spectral embedding based on the same descriptor matrix M, the corresponding results are drawn on the right-hand side of Figure 2. This time in order to illustrate that our approach is not restricted to the modularity, we choose the descriptor matrix to be the normalized Laplacian $\mathcal{L}$ as discussed in what follows. As an advantage of the ellipsoidal embedding, we observe on the bottom left of Figure 2 that the effective dimension of the embedding $d_{\text{eff}} = 3$ can be read out from the decay of the spectrum, whereas the spectrum of the normalized Laplacian $\mathcal{L}$ does not exhibit a clear gap. The tail of eigenvalues is found to be numerically small and is therefore neglected. As showed in Figure 1, we may expect the ellipsoidal embedding to highlight structures of graphs as discussed in the next section.

1.2. Contributions. We propose a method for embedding graphs on ellipsoids. The dimension of this embedding is automatically determined as part of the algorithm and is not required as an input; the analyst merely needs to constrain the maximum dimension d_0 of the embedding space for computational purposes. To obtain our embedding, we make use of a *generalized power method with momentum,* a simple deterministic algorithm with a random initialization. In practice, this algorithm yields empirically a small effective embedding dimension, highlighting that many graphs can be represented in terms of a low-dimensional parametrization. For the examples in this paper, the embedding dimension ranges from $d_{\text{eff}} = 2$ for simple graphs to $d_{\text{eff}} = 6$ for graphs with more complicated structures. For graphs with a large embedding dimension, the embedded data can then be further analyzed via multidimensional

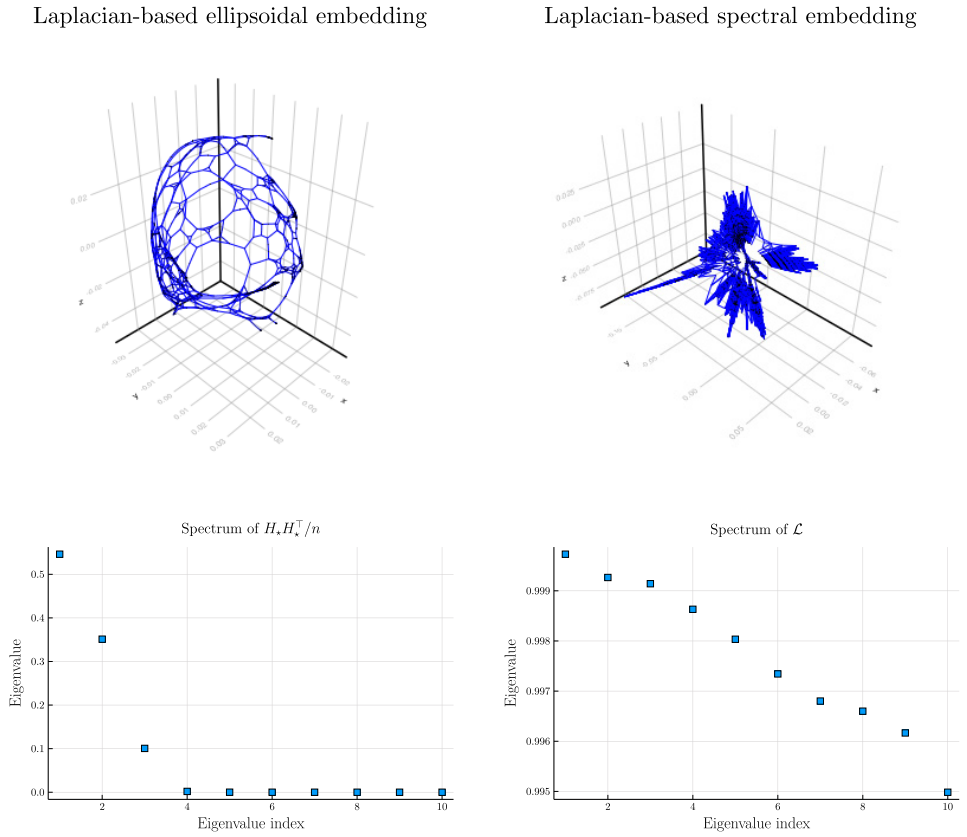

FIG. 2. *Ellipsoidal versus spectral embedding. Laplacian-based ellipsoidal embedding (top left,* $d_0 = 10$, $M = \mathcal{L}$; *see* (3.5)*) and the spectral embedding (top right) of the* POWEREU *graph—see Table* SM1*—thanks to the three leading eigenvectors of the Laplacian* $\mathcal{L}$. *On the bottom left, the spectrum of* $\frac{1}{n} H_\star H_\star^\top$ *associated to the embedding with* $d_{\text{eff}} = 3$. *The spectrum of this normalized Laplacian matrix* $\mathcal{L}$ *is given on the botton right.*

scaling or principal component analysis to visualize the results. In the latter case, the dimensionality reduction relies on a thresholding of components with small variance. At this point, we also mention for completeness that the companion paper [20] presents a kernel-based dimensionality reduction technique for Euclidean data relying on the solution of a discrete problem equivalent to (1.2). In contrast with this work, [20] leverages kernel methods to provide an out-of-sample embedding formula and discusses statistical guarantees for the approximation of a variational problem by this discrete problem.

While the mathematical formulations are different, there are certain analogies with spectral embeddings [7, 42, 43, 53] that we discuss. As a case study to demonstrate the utility of the derived embedding, we show how the embedding can be utilized to perform graph clustering, while spectral embeddings are commonly used as a preprocessing step for clustering. For this case study, we perform an embedding in which the modularity matrix—a well-known tool for graph clustering—is chosen as the descriptor matrix, and then use the resulting embedding to perform graph clustering. Note, however, that any other descriptor matrix could have been chosen in this context, resulting in a different embedding. Given a descriptor matrix, the

corresponding (meta-)algorithm consists in first computing the embedding and then performing a clustering procedure, by using a vector partitioning algorithm, which is of independent interest. We call this strategy *embed-and-partition*. A point of practical interest in this context is that our embedding-based clustering method directly provides an estimate for the relevant number of clusters. Basically, the clustering starts with a given number of "centroid" nodes initially sampled with respect to their degree and the algorithm then optimizes their positions by possibly merging a few of them. This is in contrast to many other methods based on embeddings, which typically rely on specifying a desired number of clusters a priori and then performing a k-means clustering or a similar procedure on the obtained embedding coordinates. Interestingly, for a modularity-based embedding as studied here, its performance is often comparable to the highly successful Louvain method [8] on benchmark graphs in terms of efficiency and for a limited increase of computing time; see Figure 5. As a limitation of our embed-and-partition approach, whereas d_{eff} is often smaller than d_0 as we mentioned above, we observe that the clustering performance is especially good when applied on the d_0-dimensional embedding coordinates rather than on the $(d_{\text{eff}} - 1)$-dimensional sphere or ellipsoid.

We further present empirical evidence that for networks with modular structure, the weaker the community structure is defined, the higher the (automatically inferred) embedding dimension—thus highlighting a connection between the hardness of community detection and the ability to compress the network structure via a low-dimensional ellipsoidal embedding.

A code implementing our embedding and partitioning methods is available at https://github.com/mrfanuel/EllipsoidalGraphEmbedding.jl in the form of a Julia module.

1.3. Outline. The remainder of this paper is organized as follows. To set the scene and provide additional motivation for our embedding, we first provide some concrete examples of trace optimization problems in the context of network analysis in section 2. Subsequently, the mathematical formulation of the embedding problem is explained in section 3, and a novel optimization method is proposed to solve it. Some mathematical properties of the algorithms used for obtaining the ellipsoidal embedding are given in section SM1 of the supplementary material. We discuss relations and differences of the proposed embedding with other problem formulations in network analysis and spectral embeddings. In section 4, we discuss the relationship of ellipsoidal embeddings to community detection in the form of modularity maximization, as a concrete application for our embedding. Specifically, we introduce a greedy algorithm based on the ellipsoidal embedding to obtain the clustering of a graph. Then, we discuss the results of the embedding-based community detection algorithm, by using synthetic and real-world networks with up to one million nodes and several million edges. Our results show that the proposed embedding faithfully captures relevant structural features of a graph. The details of the numerical simulations are given in section SM1. We conclude with a brief discussion in section 5. To improve the readability of the paper, the proofs of the mathematical results are relegated to the supplementary material (M164836_supplement.pdf [local/web 375KB]).

1.4. Notation. In terms of notation, we denote by $\mathcal{G}$ a connected graph with vertex set $\mathcal{V}$ and edge set $\mathcal{E}$. The number of nodes in the graph is denoted by $n = |\mathcal{V}|$ and we suppose that the graphs under consideration are undirected. For convenience, we always identify the node set of a graph with the natural numbers $\{1, \ldots, n\}$. The adjacency matrix $A \in \mathbb{R}^{n \times n}$ is then defined such that $A_{ij} = 1$ if and only if i is

connected to j and $A_{ij} = 0$ otherwise. It is customary to introduce the degree vector d with elements $\mathrm{d}_i = \sum_{j=1}^{n} A_{ij}$ and total edge weight $m = \sum_{i=1}^{n} \mathrm{d}_i/2$. Based on the degree matrix $D = \mathrm{diag}(\mathrm{d})$, we further define the combinatorial Laplacian as $L = D - A$, and the normalized Laplacian $L_N = D^{-1/2} L D^{-1/2}$. If M is a square, positive semidefinite (psd) matrix we write $M \succeq 0$ (recall that M is psd if and only if $v^\top M v \geq 0$ for all vectors v). The ℓth column and ith row of a matrix H will be denoted by $H_{*\ell}$ and H_{i*}, respectively. Finally, the nuclear norm of a matrix M is defined as follows: $\|M\|_\star = \mathrm{Tr}(\sqrt{M^\top M})$.

2. Motivation and background.

2.1. Ellipsoidal embedding and graph structures. While graph drawings can help to better understand the topological structure of graphs, we may want to invert this process and try to reason about the graph itself by means of a carefully defined geometrical embedding of the graph. Hence, we may want to design a graph embedding that reflects certain topological properties of the graph geometrically. These are natural ideas, and going back and forth between a geometric embedding and a graph representation of data (sometimes implicitly) underpins a host of successful methods in data science.

- For instance, spectral clustering [39, 49] may be seen as a graph embedding into a Euclidean space on which we perform "classical" clustering afterward (e.g., using k-means). Many different spectral clustering methods exist that rely on choosing the "right" algebraic representation of the graph, such that certain features of the graph are emphasized [44, 53, 54, 56].
- Manifold learning techniques such as diffusion maps [16, 29, 34] provide another interesting example. In this case, we start with a geometric point cloud, from which we construct a graph based on the geometric data. From this graph we then derive a new geometric representation of the data by embedding this graph via a set of diffusion coordinates, thereby providing a data parametrization in a lower dimensional Euclidean space.
- In generative models for graphs we often posit the existence of a "correct" embedding in the construction, e.g., hyperbolic embeddings and hyperbolic latent models have been proposed to model and fit networks [6]. Similarly, random dot-product graphs [7] and other continuous latent graph models [32] posit that an observed graph has been generated with an implicit set of latent geometric coordinates.
- Even for discrete latent random variable models such as the stochastic block model [1], continuous embeddings provided by the spectral properties of the observed graph can provide useful information about the graph, which is harnessed in spectral methods for community detection such as [43].
- Recent approaches for graph clustering (see [45] and [21]) also use specifically a spherical embedding.

What is common among all these approaches is that an embedding of a graph into a metric space provides us with additional means to approximately solve hard problems, such as graph comparisons, clustering, etc., by using the rich toolkit of continuous mathematics within the embedding domain. Indeed, there is a recent surge of interest in graph embeddings because of this reason: some recent works propose to use machine-learning techniques to learn an embedding to reflect certain topological features of the nodes [22, 23, 25].

Typically, a spectral embedding uses the d_0 leading eigenvectors of a symmetric $n \times n$ matrix M for $n \geq d_0$. These eigenvectors are columns of $H_\star \in \mathbb{R}^{n \times d_0}$ which is a solution of the following trace maximization problem:

$$\underset{H\in\mathbb{R}^{n\times d_0}}{\text{maximize}}\ \mathrm{Tr}\left(H^\top MH\right)\ \text{subject to}\ H^\top H=\mathbb{I}_{d_0\times d_0}, \tag{2.1}$$

where the constraint implements the orthonormality of the columns of H. Similar to the spectral embeddings, the ellipsoidal embedding, calculated with (1.2), naturally emerges from trace maximization problems of the form

$$\underset{H}{\text{maximize}}\quad \mathrm{Tr}\left(H^\top MH\right) \tag{2.2a}$$

$$\text{subject to}\quad H\in\mathcal{Z}, \tag{2.2b}$$

where $\mathcal{Z}$ denotes the set of constraints. Here, the matrix M in (2.2a) is an algebraic descriptor of the network. For instance, M could be a Laplacian matrix, or a feature matrix derived from the network such as a matrix M with entry M_{ij} counting all walks up to length k between any two nodes i, j. In order to illustrate the relevance of problems of the type (2.2), we give here a few examples of such problems in the context of the analysis of graphs and networks.

Trace optimization problems in network analysis. Laplacian matrices play a major role in network analysis, as their spectral properties are intimately related to the network structure. Accordingly, they have been analyzed from a variety of angles [15, 33]. One fundamental problem in which the graph Laplacian emerges is the problem of graph partitioning. This problem can be phrased as a *penalized cut* problem [56], which includes other popular notions such as normalized cut [49] and ratio cut [13].

Let $H\in\{0,1\}^{n\times k}$ be a binary indicator matrix associated to a partition of the graph with k clusters, i.e., $H_{ic}=1$ if i belongs to group c and zero otherwise. Based on this definition, the penalized cut problem is to minimize the objective function

$$\mathrm{Tr}\left(H^\top LH(H^\top\Delta H)^{-1}\right)\quad \text{(Penalized Cut)},$$

subject to the constraint that H is a binary indicator matrix of the form described above, and Δ is a positive definite diagonal weighting matrix. Note that this objective may be rewritten in the more compact form $\mathrm{Tr}\left(Z^\top LZ\right)$ with $Z=H(H^\top\Pi H)^{-1/2}$, i.e., can be directly mapped to problem (2.2).

Apart from penalized cut, this class of optimization problems includes many other problems of interest. For instance, the formulation (2.2) includes maximum likelihood estimation of the partitions of certain stochastic block models [3, 24, 41], a type of generative network models that has gained enormous interest in network analysis recently. Furthermore, several synchronization problems [10, 26] can be formulated in this form, such as the U(1)-synchronization problem on graphs [10, 11, 50]:

$$\max_{H\in\mathbb{C}^n}\mathrm{Tr}(\Theta HH^*)\ \text{subject to}\ H\in\mathrm{U}(1)^n\quad \text{(U(1)-synchronization)},$$

where Θ is a Hermitian matrix, with entries such that $\Theta_{ij}=\exp(\mathrm{i}\theta_{ij})$ if ij is an edge of the graph and $\Theta_{ij}=0$ otherwise. Here, H^* denotes the Hermitian conjugate of H.

Another class of important problems of the above form are those associated to modularity optimization, which we will adopt as our running example in the following. We remark, however, that most of our arguments are equally applicable, *mutatis mutandis,* to other problem contexts.

Modularity maximization. For a given network with adjacency matrix A, let Q be the modularity matrix given by

$$Q = \frac{1}{2m}\left(A - \frac{\mathrm{d}\mathrm{d}^\top}{2m}\right). \tag{2.3}$$

The problem of optimizing the modularity can be cast in the form (2.2) as

$$\underset{H}{\text{maximize}} \quad \mathrm{Tr}\left(H^\top Q H\right) \quad \text{(modularity maximization)} \tag{2.4a}$$

$$\text{subject to} \quad H \in \mathcal{Z}, \tag{2.4b}$$

where $\mathcal{Z}$ is the set of partition indicator matrices with any number of groups k, which obey the definition above: each node is in one and only one group and $H_{ic} = 1$ if i belongs to group c and zero otherwise. Let us give a few words of motivation for the expression of the objective in (2.4a). Modularity was introduced in [36] as a way to uncover the "best" community structure as the node partitioning maximizing modularity as in (2.4). Roughly speaking, we can say that the modularity of a partition of a graph is the difference between the number of links within the groups of the graph partition and the expected number of links in these groups for a random graph with the same node set. The distribution of this random graph is described in [36]. Hence, modularity provides a way to quantify community structures by using a comparison between edge density in the graph of interest and a random graph model.

Similar to the other problems discussed above, modularity optimization is an NP-hard problem [12], and accordingly several heuristics have been proposed to solve the above problem, including greedy [8] and spectral algorithms [38].

Before describing in more detail ellipsoidal embeddings in section 3, we pause a moment to discuss a closely related method.

2.2. Extremal graph embeddings by trace maximization. A recent embedding method—called maximal graph realization and put forward in [40]—also relies on solving an optimization problem which can be interpreted as a trace maximization problem. Although it shares some similarities, this optimization problem differs from the problem (1.2) considered here. In the approach of [40], the graph is encoded thanks to its incidence matrix $B = [b_1 \dots b_{|\mathcal{E}|}]^\top \in \mathbb{R}^{|\mathcal{E}|\times n}$ where, for all $e \in \{1, \dots, |\mathcal{E}|\}$,

$$b_{e,i} = \begin{cases} 1 & \text{if } i \text{ is the head of } e, \\ -1 & \text{if } i \text{ is the tail of } e, \\ 0 & \text{otherwise,} \end{cases}$$

with $i \in \{1, \dots, n\}$ and where edges have a fixed arbitrary orientation. Considering a connected graph and a positive integer d, the spirit of this approach is to find a solution to

$$\underset{X \in \mathbb{R}^{n\times d}}{\text{maximize}} \quad \mathrm{Tr}\left(X^\top X\right) \tag{2.5a}$$

$$\text{subject to} \quad 1^\top X = 0,\ \|X^\top b_e\|^2 \le 1 \text{ for all } e \in \{1, \dots, |\mathcal{E}|\}; \tag{2.5b}$$

see equation (2) in [40]. The rows of an optimal solution of (2.5) provide embedding coordinates of the graph nodes. However, the above trace maximization problem is not solved directly, in contrast with the approach used in this paper. In particular, the embedding dimension is obtained as follows. In practice, a convex eigenvalue

maximization problem is solved, aiming to find positive edge weights so that the second least eigenvalue λ_2 of the combinatorial Laplacian is maximal. Then, the embedding dimension d in (2.5) is given by the dimensionality of the eigenspace of λ_2. Thanks to convex duality, [40] shows that a solution to (2.5) is given by a matrix X whose orthonormal rows span the d-dimensional eigenspace of λ_2.

As an illustration, ellipsoidal and extremal graph embeddings of small graphs are compared in Figure 3, where both embeddings are found to be qualitatively similar. For these small-scale networks, extremal graph embeddings are quickly computed, whereas we observe that the optimization routine[1] does not scale well to large graphs. For this reason, we were not able to compute extremal embeddings for the 2000 node LFR (Lancichinetti–Fortunato–Radicchi) benchmark of Figure 4.

3. Ellipsoidal embeddings. In view of the embedding interpretation of trace optimization problems such as modularity optimization, we propose here another embedding that consists in finding a generalized label matrix $H \in \mathbb{R}^{n\times d_0}$ whose rows are to be interpreted as coordinate vectors, by solving (1.2) with $M = Q$. Observe that in contrast to the spectral embedding (2.1) formulation, where the *columns* of H were supposed to have unit 2-norm, we here apply a constraint on the *rows* of H. Thus, this constraint cannot be interpreted as a weakening of the constraint appearing in the spectral problem. Specifically, the resulting formulation (1.2) is a relaxation of the partitioning problem (2.4) in that every matrix $H \in \mathcal{Z}$ (membership matrix) fulfills the condition $\|H_{i*}\|_2 = 1$. To better understand the above optimization problem, and how it relates to an ellipsoidal (spherical) embedding, let us comment on a few features of the above formulation. First, notice that the enforced constraints on H imply that the feasible space for H is a Cartesian product of spheres $H \in (\mathbb{S}^{d_0-1})^n$, i.e., every row of H defines a point on a hypersphere of dimension $d_0 - 1$. Hence the matrix H defines embedding coordinates for each node in the graph, that can be interpreted as points on a hypersphere. Note that the resulting set of coordinates is only unique up to unitary transformation. Namely, for any orthogonal matrix U, i.e., satisfying $U^\top U = I = UU^\top$, the matrices H and HU will have exactly the same objective value. Only the matrix $HH^\top$ is invariant under these orthogonal transformations.

Unlike in the spectral case (2.1), d_0 does not correspond directly to the embedding dimension, but rather corresponds to an upper bound of the embedding dimension. Interestingly, in many cases, the optimal embedding can have several "empty" columns in H, which can be dropped without loss of information. In practice, in our simulations, we often chose the integer d_0 to be smaller than 50 and never larger than 250, and we observe that the obtained embedding dimension (see the following sections for a more detailed discussion) is typically much smaller than d_0. This is the empirical reason why the ellipsoidal embedding is often low-dimensional.

3.1. Defining spherical and ellipsoidal embeddings. Let $H_\star$ be an optimal solution of the embedding problem (1.2). For each node i, the ith row of $H_\star$ now defines an embedding of the node in a sphere. However, this solution is only unique up to rotations/reflections. Hence, aiming to reduce embedding invariances, we employ a singular value decomposition (SVD) for H.

DEFINITION 1 (spherical and ellipsoidal embeddings). *Let $H_\star = USV^\top$ be an SVD of a solution of* (1.2) *with $S = \operatorname{diag}(s)$, where $s_1 \geq \cdots \geq s_r > 0$. Further, let U_{i*} and Σ_{i*} be the ith row of U and $\Sigma := US$, respectively. We define the spherical embedding by the map*

[1] https://github.com/braxtonosting/ExtremalGraphRealization.

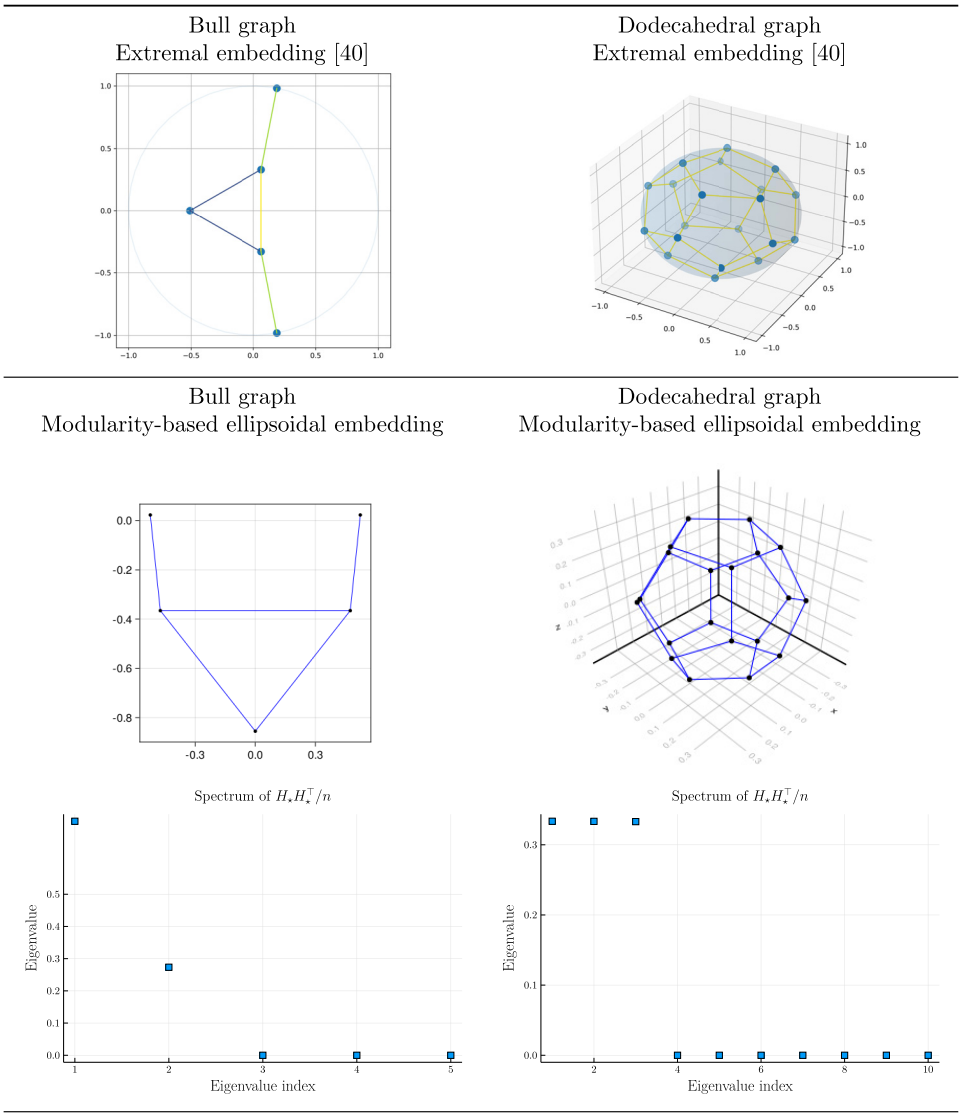

FIG. 3. *Comparison of ellipsoidal (first row) and extremal embeddings (second row) on two simple graphs: the Bull graph (left column) and the dodecahedral graph (right column). On the third row, the spectrum of* $\frac{1}{n}H_\star H_\star^\top$ *associated to the ellipsoidal embedding indicates the effective dimension* ($d_{\text{eff}} = 2$ *on the left and* $d_{\text{eff}} = 3$ *on the right).*

$$i \mapsto \Sigma_{i*} := \begin{bmatrix} s_1 U_{i1} & s_2 U_{i2} & \dots & s_r U_{ir} \end{bmatrix}, \quad 1 \le i \le n,$$

whereas the ellipsoidal embedding is defined as

$$i \mapsto U_{i*} := \begin{bmatrix} U_{i1} & U_{i2} & \dots & U_{ir} \end{bmatrix}, \quad 1 \le i \le n.$$

To see that the above mappings define an embedding on a sphere in the same way as the rows of $H_\star$, consider the matrix $\rho = H_\star H_\star^\top = \Sigma\Sigma^\top \in \mathbb{R}^{n\times n}$, which is a psd matrix with elements given by the inner product $\rho_{ij} = \Sigma_{i*}\Sigma_{j*}^\top$. Since by definition of the embedding problem (1.2), the diagonal elements of ρ have to be equal to 1, we

know that $\|\Sigma_{i*}\|_2 = 1$, and hence the embedding vector Σ_{i*} defines a point on the unit sphere $\mathbb{S}^{r-1}$.

To understand the ellipsoidal embedding, we can define an alternative inner product on $\mathbb{R}^r$ denoted by $\langle \cdot, \cdot \rangle_{S^2}$ based on the diagonal positive definite matrix S^2, where S is the matrix of singular values from the SVD of H. In terms of this inner product the element ρ can be reinterpreted as $\rho_{ij} = \langle U_{i*}, U_{j*} \rangle_{S^2} = U_{i*} S^2 U_{j*}^\top$. Hence, we see that each U_{i*} belongs to an ellipsoid in $\mathbb{R}^r$ determined by the equation $\langle u, u \rangle_{S^2} = 1$.

The SVD of $H_\star$—closely related to the spectral decomposition $\rho = H_\star H_\star^\top = \Sigma\Sigma^\top$— further provides us with a simple estimate of the effective dimension of the embedding. We define the effective dimension as

$$d_{\text{eff}}(\epsilon) = \min \left\{ 1 \le r \le d_0 \,\middle|\, \sum_{\ell=1}^{r} s_\ell^2(\rho) > (1-\epsilon) \times \operatorname{Tr}(\rho) \right\}. \tag{3.1}$$

In this paper, we choose $\epsilon = 0.01$. Intuitively, the above definition discounts eigencoordinates which contribute less than 1% to the total variation in the embedding coordinates. The specific value of ϵ may here be interpreted as a "significance" value, which can be chosen by the analyst.

Remark 1 (eigenvalue thresholding). If $H_\star$ is a solution of the embedding problem, then the effective embedding H_{eff} corresponds to a truncation of the SVD of $H_\star$ to its d_{eff} largest singular values. The nuclear norms of the invariants $\rho = H_\star H_\star^\top$ and $\rho_{\text{eff}} = H_{\text{eff}} H_{\text{eff}}^\top$ are related by $\|\rho - \rho_{\text{eff}}\|_\star \le \epsilon \|\rho\|_\star$, where $\|\rho\|_\star = n$. As we choose here $\epsilon = 0.01$, this means that the relative error (as measured by the nuclear norm) between the effective embedding and the optimal embedded is less than 1%.

Remark 2 (orientation ambiguity). The embedding coordinates given in Definition 1 are ordered according to the magnitude of the singular values. Provided that all singular values are distinct, this implies that there is no ambiguity in terms of the ordering of the coordinates. There remains one source of ambiguity, however, namely, a direction change of a coordinate axis.

Note that the ambiguity discussed in Remark 2 is also encountered in spectral embeddings and is essentially unavoidable due to the symmetry of the problem. In the context of spectral embeddings, the above ambiguity corresponds to the fact that any (unit) eigenvector is only defined up to a phase. In practice, these issues of nonuniqueness can be ignored for most applications: typically, we are interested in the relative positions of the nodes, rather than their absolute positions in the embedding space.

3.2. Computing ellipsoidal embeddings. The embedding problem (1.2) can be solved in a number of different ways. In this work we employ a generalized power method, as described in Algorithm 3.1, that is inspired by [27, 10, 14, 5].

To see how the method works, first notice that the diagonal of the descriptor matrix, here illustrated by the modularity matrix Q, does not influence the solution of (1.2) but merely shifts the objective value by a constant. To see this, observe that for any diagonal matrix D, we have $\operatorname{Tr}(H^\top (Q - D) H) = \operatorname{Tr}(H^\top Q H) - \sum_i D_{ii}$. Since in general Q may be indefinite, we thus employ a preprocessing step to make any descriptor matrix positive definite, by simply shifting the spectrum of Q with a diagonal matrix. Specifically we apply the transformation $Q \mapsto K = Q + \operatorname{diag}(v)$.

Algorithm 3.1. Generalized power method [10, 14, 27].

Require: Symmetric positive definite matrix $K \in \mathbb{R}^{n\times n}$; and an initial embedding $x_0 \in \mathbb{R}^{n\times d_0}$ such that $\Pi(x_0) = x_0$; see (3.3). Fix $0 < \text{tol} < 1$, and set $m = 0$.

1: **do**
2: $m = m + 1$
3: $x_m = \Pi(Kx_{m-1})$
4: $o_m = \text{Tr}(x_m^\top K x_m)$
5: **while** ($m \leq 1$ or $|o_m - o_{m-1}|/o_{m-1} \geq \text{tol}$)

Here v is chosen such K is strictly diagonally dominant and therefore $K \succ 0$. Specifically, here we define

$$K_{ij} = \begin{cases} Q_{ij} & \text{if } i \neq j, \\ 1 + \sum_{k\neq i} |Q_{ik}| & \text{if } i = j. \end{cases} \tag{3.2}$$

Using this shifted descriptor matrix, Algorithm 3.1 now solves the embedding problem iteratively, starting from an initially feasible solution. Inspired by [14], here the initialization is obtained by selecting uniformly at random d_0 columns of K and then by projecting the resulting $n \times d_0$ matrix on the product of spheres $(\mathbb{S}^{d_0-1})^n$ by using the projection operator Π, which maps any matrix $H \in \mathbb{R}^{n\times d_0}$ such that $H_{i*} \neq 0$ for all $i = 1, \ldots, n$ onto a spherical embedding matrix, by normalizing the rows of H, namely

$$\Pi(H)_{i*} = H_{i*}/\|H_{i*}\|_2 \text{ for all } 1 \leq i \leq n, \tag{3.3}$$

where we recall that H_{i*} is the ith row of H.

Starting from a feasible initial condition, we alternate between applying our (shifted) descriptor matrix to the current embedding, and then project the result back again onto a hypersphere. Intuitively, the repeated multiplication of K aligns the current iterate with the dominant subspace of K akin to a power method for eigensolvers, thereby increasing the objective value, while the projection step acts as a normalization step and ensures that we maintain feasibility. The algorithm is stopped when the relative variation of consecutive objectives does not exceed a particular tolerance. In this paper we choose the tolerance tol = 1e−08 unless stated otherwise.

Mathematically, Proposition 1 gives a lower bound on the improvement between successive objectives values.

PROPOSITION 1. *Let $K \in \mathbb{R}^{n\times n}$ be symmetric such that $|K_{ii}| > 1 + \sum_{k\neq i} |K_{ik}|$ for all $1 \leq i \leq n$. Let the objective function be $f(x) = \text{Tr}(x^\top K x)$. Then, the sequence of objectives for the iteration $x_{m+1} = \Pi(Kx_m)$ satisfies*

$$f(x_{m+1}) - f(x_m) > \|x_{m+1} - x_m\|_2^2$$

for all $m \geq 0$.

The proof of a more general version of Proposition 1 can be found in section SM2 of the supplementary material.

While Algorithm 3.1 provides us with a practical algorithm to solve our embedding problem, in order to speed up the optimization, we propose Algorithm 3.2, which

Algorithm 3.2. Generalized power method with momentum.

Require: Symmetric positive definite matrix $K \in \mathbb{R}^{n\times n}$; and an initial $x_0 \in \mathbb{R}^{n\times d_0}$ such that $\Pi(x_0) = x_0$. Initialize $y_0 = Kx_0$. Fix $0 < \text{tol} < 1$ and set $n = 0$.

Let r_n be defined as $r_n := (n-1)/(n+2)$ for $n \in \{1, 2, \dots\}$.
do
  $n = n + 1$
  $o_{n-1} = \text{Tr}(y_{n-1}^\top x_{n-1})$,
  $y_n = Kx_{n-1}$,
  $x_n = \Pi(y_n + r_n(y_n - y_{n-1}))$,
while ($n \le 1$ or $|o_{n-1} - o_{n-2}|/o_{n-2} \ge \text{tol}$)

includes a "momentum" term to accelerate the iterations [35]. The advantage of this preprocessing for the convergence of the generalized power method (Algorithm 3.1) as well as a more detailed theoretical analysis of both algorithms are discussed in section SM2 of the supplementary material.

3.3. Effective embedding dimension. As announced above, the embedding dimension is often low. To gain some further insight into this empirical fact, let us introduce a closely related SDP:

$$\text{maximize } \text{Tr}(\rho K) \text{ subject to } \rho \succeq 0 \text{ and } \rho_{ii} = 1 \text{ for all } 1 \le i \le n. \tag{3.4}$$

The similarity with (1.2) is explicit if we take $\rho = HH^\top$ with $H \in \mathbb{R}^{n\times d_0}$, and if we substitute this factorization into (3.4). However, solving a factorization of the form (1.2) does not necessarily give a solution of (3.4) since there is no constraint on the rank of ρ in this convex problem.

Although we will not numerically solve this SDP, it can be shown that a solution of the first order optimality condition of the embedding problem (1.2) also satisfies the complementary slackness condition of (3.4); see section SM2 for more details. Indeed, under certain circumstances the maximum of both problems correspond [11], i.e., the nonconvex embedding problem (1.2) can be effectively solved (up to rotations) by the convex program (3.4). We summarize these results in Proposition 2. To write our results compactly, here we use $\text{ddiag}(M)$ to denote the diagonal matrix obtained by replacing all off-diagonal elements of M by zero.

PROPOSITION 2 (equivalence with a nuclear norm minimization). *Let $K \in \mathbb{R}^{n\times n}$ be a psd matrix with a maximal eigenvalue strictly smaller than $\lambda > 0$ and let $\Sigma \in \mathbb{R}^{n\times n}$ be the invertible matrix with orthogonal rows such that $\lambda\mathbb{I} - K = \Sigma\Sigma^\top$. Then, the optimal solution $X^\star$ of*

$$\underset{X\succeq 0}{\text{minimize}} \ \|X\|_\star, \ \textit{subject to} \ \text{ddiag}\left((\Sigma^{-1})^\top X \Sigma^{-1}\right) = \text{ddiag}(K),$$

has the same rank as the optimal solution of (1.2) $\rho^\star$ and is given by $X^\star = \Sigma^\top \rho^\star \Sigma$.

Proposition 2 is completely analogous to Proposition 3.1 of [20], where a proof is given. Observe that in view of Proposition 2 the problem (3.4), and thus our related embedding problem (1.2), is equivalent to a nuclear norm minimization subject to linear constraints which promotes a low-rank solution and thus a low embedding dimension. Indeed, the minimization of the nuclear norm is a relaxation of the minimization of the rank of a matrix.

3.4. Using other descriptor matrices to derive ellipsoidal embeddings. Our description of the embedding so far has used the modularity matrix as our primary example of a feature matrix. While modularity is certainly one of the most well-known feature matrices related to network analysis, there is nothing about our problem formulation that forces us to stick to modularity. Indeed, it is worth remarking again that we may also use alternative matrices to derive alternative embeddings with different interpretations. The basic requirements on the descriptor matrix M are that it is symmetric and contains both positive and negative entries. For instance, the modularity matrix is designed to detect an assortative group structure in a network, i.e., it is expected to emphasize groups of nodes which are densely connected with each other. However, if we are interested in disassortative structures (e.g., bipartite structure), we may want to consider a descriptor based on the squared adjacency matrix A^2. For instance, we could simply consider a modularity matrix derived from the network with adjacency A^2. Another choice for a descriptor matrix is given by a form of Laplacian matrix. Let $D = \mathrm{diag}(\mathrm{d})$ be the diagonal degree matrix and let $\pi_i = \mathrm{d}_i / \sum_j \mathrm{d}_j$. An embedding related to the normalized Laplacian matrix may then be defined via the following descriptor matrix:[2]

$$\mathcal{L} = D^{-1/2} A D^{-1/2} - \sqrt{\pi}\sqrt{\pi}^{\top}. \tag{3.5}$$

A final choice for a descriptor matrix is given by an autocovariance matrix of a random walk on the graph, as it features in the Markov stability framework [18, 47, 17], which allows us to sweep the graph structures at different scales. More generally, we may consider descriptor matrices derived from more general dynamical (covariance) kernels [46], in order to capture certain dynamical features of the problem at hand. We postpone the study of these alternatives for a further work.

4. Case study: Ellipsoidal embeddings for graph partitioning. One task in network analysis that has enjoyed tremendous interest over the past decades is community detection—the task of partitioning a network into groups of nodes according to some prespecified criterion. In the following, we show how we can use our spherical embedding to perform community detection for networks. While the resulting algorithm may be seen as an independent nonparametric community detection (meta-)heuristics (depending on the chosen descriptor matrix) in its own right, our goal here is primarily to illustrate the utility of the embedding using this task as a case study.

For simplicity, we will use again the modularity matrix as a descriptor matrix of our embedding here. Note that we do not aim to optimize modularity here directly, nor do we advocate modularity optimization as the method of choice for community detection. However, choosing a modularity-based embedding enables us to relate the resulting clustering to the large literature of methods for modularity optimization and thus provides some form of external validation for the utility of the embedding. For comparison we therefore also computed network clusterings according to the Louvain method [8], which is known to perform well for modularity optimization on large graphs.

4.1. Embed-and-partition. In order to find clusters in the embedding, we take inspiration from the well-known k-means algorithm and the vector partitioning

[2]Strictly speaking, the matrix $\mathcal{L}$ and the normalized Laplacian L_N are different matrices. However, note that $\mathcal{L} = I - L_N - \sqrt{\pi}\sqrt{\pi}^{\top}$ simply corresponds to a shifted version of L_N with a rank-1 correction term.

methods proposed in [55, 31]. Let $H \in \{0,1\}^{n\times k}$ be a binary membership matrix associated to a partition of the graph with k clusters, i.e., each node is in one and only one cluster, and $H_{ic} = 1$ if i is in the cluster c and zero otherwise. Also, we denote by $c_i \in \{1,\dots,k\}$ the cluster index of $i \in \{1,\dots,n\}$. Then, we aim to optimize the following objective:

$$\tilde{z} = \mathrm{Tr}\left(H^\top Z H\right) = \sum_{\ell=1}^{k} \left\| \sum_{\{i|c_i=\ell\}} U_{i*} \right\|_2^2, \tag{4.1}$$

where $Z = UU^\top$ in the case of ellipsoidal embedding (see Definition 1) and where H is a binary membership matrix. Like most partitioning problems, the exact maximization of the latter objective function over all binary membership matrices is performed by a greedy approach; see Algorithm 4.1. In the spirit of vector partitioning, each of the sums above is associated to a centroid vector $R_\ell = \sum_{\{i|c_i=\ell\}} U_{i*}$. Following [55], moving node i from community ℓ to community ℓ' yields the following change in the objective: $\Delta\tilde{z} = 2U_{i*}(R_{\ell'} - R_\ell)^\top - 2$. This means that if $U_{i*}R_{\ell'}^\top > U_{i*}R_\ell^\top + 1$, the objective is improved by changing node i from community ℓ to community ℓ'. This remark motivates the iteration given in Algorithm 4.1.

In view of these remarks, the partitioning algorithm proceeds as follows. We first initialize the algorithm according to the following:

1. Draw k centroid vectors $R_1,\dots,R_k$ without replacement from the set of position vectors $\mathcal{U} = \{U_{1*},\dots,U_{n*}\}$ according to the distribution π.
2. For all $1 \le i \le n$, calculate $c_i^{(0)} \in \mathrm{argmax}_{1\le\ell\le k} U_{i*}R_\ell^\top$.
3. For all $1 \le \ell \le k$, compute $R_\ell^{(0)} = \sum_{\{i|c_i^{(0)}=\ell\}} U_{i*}$.

Here the probability distribution π for the initial sampling is chosen to be proportional to the degree of the node $\pi_i = \mathrm{d}_i / \sum_i \mathrm{d}_i$. We then iterate over the cluster-assignments and centroid updates in an alternating fashion as outlined in Algorithm 4.1. In practice, we update the communities as long as the objective $\mathrm{Tr}\left(H^\top Z H\right)$ of the partition associated to H increases; otherwise we stop. We notice empirically that if k clusters are initialized at random, then due to the shape of the embedding several clusters will be associated to empty partitions after a few iterations. In this case our procedure will yield a number of clusters smaller than or equal to the original supplied upper bound k.

As mentioned already above, note that for obtaining partitions with a good modularity value with embed-and-partition in the simulations of this paper, we always define the embedding $U_{i\star}$ of Definition 1 from the untruncated SVD.

4.1.1. Numerical results for embedding-based graph partitioning of benchmark graphs. To perform our synthetic experiments, we created a range of

Algorithm 4.1. Vector partitioning [55].

Require: embedding $\{U_{i*}\}_{i=1,\dots,n}$, initial partition $\{c_i^{(0)}\}_{i=1,\dots,n}$, and centroid vectors $\{R_\ell^{(0)}\}_{\ell=1,\dots,k}$.

1: **do**

2: Find $c_i^{(n)} \in \mathrm{argmax}_{1\le\ell\le k}\, U_{i*}R_\ell^{(n-1)\top}$.

3: Update $R_\ell^{(n)} = \sum_{\{i|c_i^{(n)}=\ell\}} U_{i*}$.

4: **while** modularity of the partition $\{c_i^{(n)}\}_{i=1,\dots,n}$ keeps increasing.

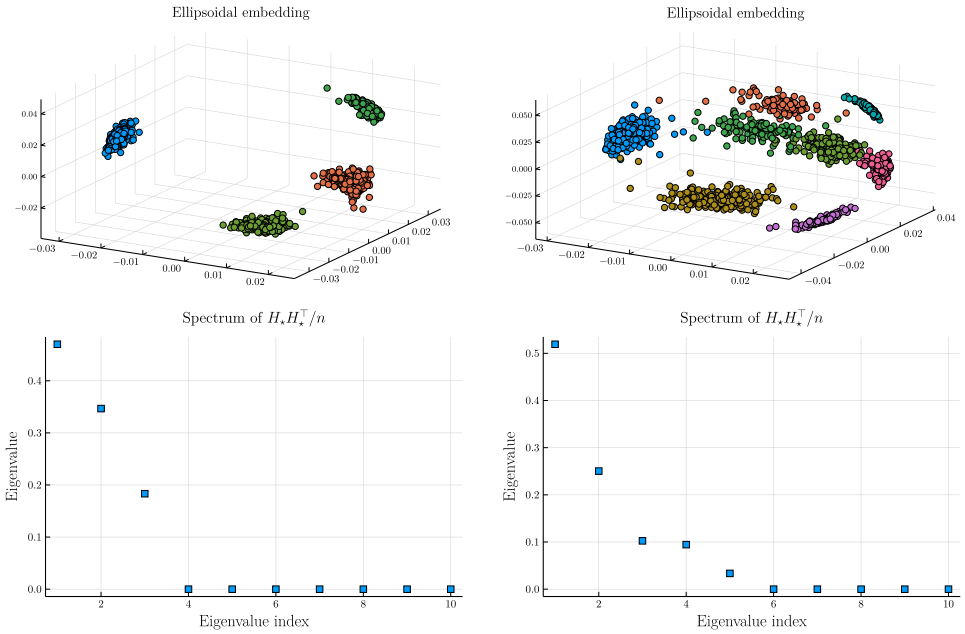

FIG. 4. *Modularity-based ellipsoidal embeddings for graphs with community structure. A visualization of the embedding of two LFR benchmark graphs with* 2000 *nodes with* $d_0 = 10$*;* LFR1 *(left,* 4 *planted communities and* $d_{\text{eff}} = 3$*) and* LFR2 *(right,* 8 *planted communities, a larger mixing parameter, and* $d_{\text{eff}} = 5$*); see* SM3.4 *for details. The colors indicate the true community structure. On the bottom, the eigenvalues of* $\frac{1}{n} H_\star H_\star^\top$*. Our embed-and-partition retrieves the planted communities in both cases.*

different benchmark graphs using the model of Lancichinetti, Fortunato, and Radicchi [30], which simulates graphs with community structures inspired by statistical patterns observed in real-world networks.

To gain some further intuition of how our method operates in this task, a visualization of the embedding of the LFR benchmark graph is shown in Figure 4. In both cases, the effective dimension of the embedding is indeed small, as may be seen from the spectra at the bottom of Figure 4, while the planted communities are recovered by our partitioning method. In a more extensive study, our clustering results are compared in Figure 5 on LFR benchmarks of various mixing parameters with the Louvain method [8]. A conclusion that can be drawn from those comparisons is that our method yields competitive partitions in terms of quality.

4.1.2. Numerical results for real-world graphs. Several real networks given in Table SM1 were also used to compare partitions obtained with modularity-based ellipsoidal embeddings with the following baselines: the Louvain method, node2vec + k-means, and a spectral method based on the modularity matrix followed by vector partitioning. The results can be found in Table SM2. We observe that in the case of those real-world networks, our partitioning method often obtains relatively good modularity values.

5. Conclusions. Taking inspiration from spectral relaxations of trace optimization problems, we have proposed a general ellipsoidal embedding algorithm for networks. We have discussed several connections of this approach to spectral clustering and other methods proposed in the literature and provided a simple, efficient algorithm

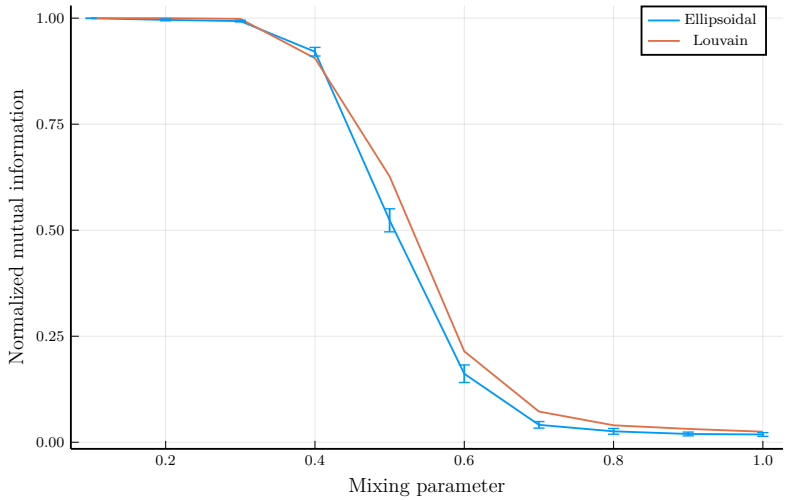

FIG. 5. *Normalized mutual information versus mixing parameter* `mu` *of LFR benchmark graphs with* $n = 1000$ *nodes. These graphs were generated with different mixing parameters ranging from* 0.1 *to* 1*; see* SM3.5 *for the numerical setting. An ellipsoidal embedding was computed with* $d_0 = 30$ *and communities were retrieved thanks to Algorithm* 4.1 *with* $k = 100$ *initialized centroids. The normalized mutual information between the planted and the retrieved community structure is here displayed as a function of the mixing parameter. The whole procedure was repeated independently* 3 *times and averages as well as standard deviations are reported. We refer to Figure* SM2 *for a study of the sensitivity to the choice of* k *and* d_0*.*

to compute such an embedding. We have further shown that our embedding can be utilized for community detection by applying a vector partitioning algorithm in the embedding space derived from the modularity matrix, which may be of independent interest. Interestingly, the computed embedding dimension, which can be selected in an automatic fashion, appears to be indicative of the "structural complexity" of the studies network and can be used as a lower bound for the amount of clusters present in the network.

There are a number of interesting directions based on this work worth pursuing in future research. For instance, it would be interesting to characterize the relationship between the network structure and the optimal embedding dimension in more detail. In particular, while numerically we have observed that the embedding dimension can serve as a robust proxy for the complexity of the network, it would be interesting to see whether this observation can be formalized. One possible way forward here would be to study a generative model which could be related to a spherical embedding such as the $\mathbb{S}_1$ model [48], or random dot-product graphs [7]. In this context it would also be insightful to understand the relationship to associated spectral embeddings better, which follow a related yet distinct paradigm.

From the algorithmic perspective, a new preprocessing method for the generalized power method was proposed in this work as well as a new algorithm: the generalized power method with momentum. Although we have no proof yet of the convergence for this new algorithm, it was showed empirically to converge markedly faster. Especially since the proof techniques of the accelerated gradient methods [35] do not seem to be applicable in our context, we think it is of theoretical interest to study its convergence properties in more detail.

Finally, there are some interesting interpretations of the proposed method, as discussed in section 1.1, which will be worth exploring further. In particular, the

connection of the ellipsoidal embeddings to quantum dynamics (density matrices) suggests further investigation. There has been significant interest recently in quantum dynamics such as random walks on networks [28] and it would be of interest, e.g., to explore "quantum descriptor matrices" of graphs and their resulting embeddings.

REFERENCES

[1] E. Abbe and C. Sandon, *Community detection in general stochastic block models: Fundamental limits and efficient algorithms for recovery*, in Proceedings of the 56th Annual Symposium on Foundations of Computer Science, IEEE, 2015, pp. 670–688.

[2] R. Albert and A.-L. Barabási, *Statistical mechanics of complex networks*, Rev. Modern Phys., 74 (2002), pp. 47–97, https://doi.org/10.1103/RevModPhys.74.47.

[3] A. A. Amini and E. Levina, *On semidefinite relaxations for the block model*, Ann. Statist., 46 (2018), pp. 149–179, https://doi.org/10.1214/17-AOS1545.

[4] A. Arenas, A. Díaz-Guilera, J. Kurths, Y. Moreno, and C. Zhou, *Synchronization in complex networks*, Phys. Rep., 469 (2008), pp. 93–153, https://doi.org/10.1016/j.physrep.2008.09.002.

[5] A. Aspeel, *Community Detection in Large-Scale Time-Varying Networks, A Modularity Based Approach*, Master's thesis, Université catholique de Louvain, 2017.

[6] D. M. Asta and C. R. Shalizi, *Geometric network comparisons*, in Proceedings of the 31st Conference on Uncertainty in Artificial Intelligence, AUAI Press, Arlington, VA, 2015, pp. 102–110.

[7] A. Athreya, D. E. Fishkind, M. Tang, C. E. Priebe, Y. Park, J. T. Vogelstein, K. Levin, V. Lyzinski, Y. Qin, and D. L. Sussman, *Statistical inference on random dot product graphs: A survey*, J. Mach. Learn. Res., 18 (2018), pp. 1–92, http://jmlr.org/papers/v18/17-448.html.

[8] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, *Fast unfolding of communities in large networks*, J. Stat. Mech. Theory Exp., 2008 (2008), P10008, https://doi.org/10.1088/1742-5468/2008/10/P10008.

[9] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang, *Complex networks: Structure and dynamics*, Phys. Rep., 424 (2006), pp. 175–308, https://doi.org/10.1016/j.physrep.2005.10.009.

[10] N. Boumal, *Nonconvex phase synchronization*, SIAM J. Optim., 26 (2016), pp. 2355–2377, https://doi.org/10.1137/16M105808X.

[11] N. Boumal, V. Voroninski, and A. S. Bandeira, *The non-convex Burer–Monteiro approach works on smooth semidefinite programs*, in Proceedings of the 30th International Conference on Neural Information Processing Systems, 2016, pp. 2765–2773.

[12] U. Brandes, D. Delling, M. Gaertler, R. Goerke, M. Hoefer, Z. Nikoloski, and D. Wagner, *Maximizing Modularity is Hard*, https://arxiv.org/abs/physics/0608255, 2006.

[13] P. K. Chan, M. D. F. Schlag, and J. Y. Zien, *Spectral k-way ratio-cut partitioning and clustering*, IEEE Trans. Comput. Aid. Design Integ. Circ. Syst., 13 (1994), pp. 1088–1096, https://doi.org/10.1109/43.310898.

[14] Y. Chen and E. Candes, *The Projected Power Method: An Efficient Algorithm for Joint Alignment from Pairwise Differences*, arXiv:1609.05820, 2016.

[15] F. R. K. Chung, *Spectral Graph Theory*, AMS, Providence, RI, 1997.

[16] R. R. Coifman, S. Lafon, A. B. Lee, M. Maggioni, B. Nadler, F. Warner, and S. W. Zucker, *Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps*, Proc. Natl. Acad. Sci. USA, 102 (2005), pp. 7426–7431, https://doi.org/10.1073/pnas.0500334102.

[17] J.-C. Delvenne, M. T. Schaub, S. N. Yaliraki, and M. Barahona, *The stability of a graph partition: A dynamics-based framework for community detection*, in Dynamics On and Of Complex Networks, Vol. 2, A. Mukherjee, M. Choudhury, F. Peruani, N. Ganguly, and B. Mitra, eds., Model. Simul. Sci. Eng. Technol., Springer, New York, 2013, pp. 221–242.

[18] J.-C. Delvenne, S. N. Yaliraki, and M. Barahona, *Stability of graph communities across time scales*, Proc. Natl. Acad. Sci. USA, 107 (2010), pp. 12755–12760, https://doi.org/10.1073/pnas.0903215107.

[19] S. N. Dorogovtsev, A. V. Goltsev, and J. F. F. Mendes, *Critical phenomena in complex networks*, Rev. Modern Phys., 80 (2008), pp. 1275–1335, https://doi.org/10.1103/RevModPhys.80.1275.

[20] M. Fanuel, A. Aspeel, J.-C. Delvenne, and J. A. K. Suykens, *Positive semi-definite embedding for dimensionality reduction and out-of-sample extensions*, SIAM J. Math. Data Sci., 4 (2022), pp. 153–178, https://doi.org/10.1137/20M1370653.

[21] M. Gösgens, R. van der Hofstad, and N. Litvak, *The hyperspherical geometry of community detection: Modularity as a distance*, J. Mach. Learn. Res., 24 (2023), pp. 1–36, http://jmlr.org/papers/v24/22-0744.html.
[22] A. Grover and J. Leskovec, *node2vec: Scalable feature learning for networks*, in Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016.
[23] L. Gutiérrez Gómez, B. Chiêm, and J.-C. Delvenne, *Dynamics Based Features for Graphs Classification*, arXiv:1705.10817, 2017.
[24] B. Hajek, Y. Wu, and J. Xu, *Achieving exact cluster recovery threshold via semidefinite programming*, IEEE Trans. Inform. Theory, 62 (2016), pp. 2788–2797, https://doi.org/10.1109/TIT.2016.2546280.
[25] W. L. Hamilton, R. Ying, and J. Leskovec, *Representation Learning on Graphs: Methods and Applications*, preprint, arXiv:1709.05584, 2017.
[26] A. Javanmard, A. Montanari, and F. Ricci-Tersenghi, *Phase transitions in semidefinite relaxations*, Proc. Natl. Acad. Sci. USA, 113 (2016), pp. E2218–E2223, https://doi.org/10.1073/pnas.1523097113.
[27] M. Journée, Y. Nesterov, P. Richtárik, and R. Sepulchre, *Generalized power method for sparse principal component analysis*, J. Mach. Learn. Res., 11 (2010), pp. 517–553, https://www.jmlr.org/papers/volume11/journee10a/journee10a.pdf.
[28] J. Kempe, *Quantum random walks: An introductory overview*, Contemp. Phys., 44 (2003), pp. 307–327, https://doi.org/10.1080/0010715031000110776.
[29] S. Lafon and A. Lee, *Diffusion maps and coarse-graining: A unified framework for dimensionality reduction, graph partitioning, and data set parameterization*, IEEE Trans. Pattern Anal. Mach. Intell., 28 (2006), pp. 1393–1403, https://doi.org/10.1109/TPAMI.2006.184.
[30] A. Lancichinetti, S. Fortunato, and F. Radicchi, *Benchmark graphs for testing community detection algorithms*, Phys. Rev. E, 78 (2008), 046110, https://doi.org/10.1103/PhysRevE.78.046110.
[31] Z. Liu and M. Barahona, *Geometric multiscale community detection: Markov stability and vector partitioning*, J. Complex Networks, 6 (2018), pp. 157–172.
[32] L. Lovász, *Large Networks and Graph Limits*, Amer. Math. Soc. Colloq. Publ. 60, AMS, Providence, RI, 2012.
[33] B. Mohar, *The Laplacian spectrum of graphs*, in Graph Theory, Combinatorics, and Applications, Wiley, New York, 1991, pp. 871–898.
[34] B. Nadler, S. Lafon, R. Coifman, and I. G. Kevrekidis, *Diffusion maps, spectral clustering and reaction coordinates of dynamical systems*, Appl. Comput. Harmon. Anal., 21 (2006), pp. 113–127.
[35] Y. E. Nesterov, *A method for solving the convex programming problem with convergence rate* $\mathcal{O}(1/k^2)$, in Dokl. Akad. Nauk SSSR, 269 (1983), pp. 543–547.
[36] M. Newman, *Modularity and community structure in networks*, Proc. Natl. Acad. Sci. USA, 103 (2006), pp. 8577–8582.
[37] M. E. J. Newman, *The structure and function of complex networks*, SIAM Rev., 45 (2003), pp. 167–256, https://doi.org/10.1137/S003614450342480.
[38] M. E. J. Newman, *Finding community structure in networks using the eigenvectors of matrices*, Phys. Rev. E, 74 (2006), 036104, https://doi.org/10.1103/PhysRevE.74.036104.
[39] M. E. J. Newman, *Spectral methods for community detection and graph partitioning*, Phys. Rev. E, 88 (2013), 042822, https://doi.org/10.1103/PhysRevE.88.042822.
[40] B. Osting, *Extremal graph realizations and graph Laplacian eigenvalues*, SIAM J. Discrete Math., 37 (2023), pp. 1630–1644, https://doi.org/10.1137/22M1504421.
[41] T. P. Peixoto, *Nonparametric bayesian inference of the microcanonical stochastic block model*, Phys. Rev. E, 95 (2017), 012317, https://doi.org/10.1103/PhysRevE.95.012317.
[42] T. Qin and K. Rohe, *Regularized spectral clustering under the degree-corrected stochastic blockmodel*, in Proceedings of the 26th International Conference on Neural Information Processing Systems, Vol. 2, 2013, pp. 3120–3128.
[43] K. Rohe, S. Chatterjee, and B. Yu, *Spectral clustering and the high-dimensional stochastic blockmodel*, Ann. Statist., 39 (2011), pp. 1878–1915, https://doi.org/10.1214/11-AOS887.
[44] M. Saerens, F. Fouss, L. Yen, and P. Dupont, *The principal components analysis of a graph, and its relationships to spectral clustering*, in Machine Learning: ECML 2004, Lecture Notes in Comput. Sci. 3201, Springer, Berlin, 2004, pp. 371–383.
[45] F. Sanna Passino, N. A. Heard, and P. Rubin-Delanchy, *Spectral clustering on spherical coordinates under the degree-corrected stochastic blockmodel*, Technometrics, 64 (2022), pp. 1–12, https://doi.org/10.1080/00401706.2021.2008503.

[46] M. T. Schaub, J.-C. Delvenne, R. Lambiotte, and M. Barahona, *Multiscale dynamical embeddings of complex networks*, Phys. Rev. E, 99 (2019), 062308, https://doi.org/10.1103/PhysRevE.99.062308.

[47] M. T. Schaub, J.-C. Delvenne, S. N. Yaliraki, and M. Barahona, *Markov dynamics as a zooming lens for multiscale community detection: Non clique-like communities and the field-of-view limit*, PLoS ONE, 7 (2012), e32210, https://doi.org/10.1371/journal.pone.0032210.

[48] M. A. Serrano, D. Krioukov, and M. Boguná, *Self-similarity of complex networks and hidden metric spaces*, Phys. Rev. Lett., 100 (2008), 078701, https://doi.org/10.1103/PhysRevLett.100.078701.

[49] J. Shi and J. Malik, *Normalized cuts and image segmentation*, IEEE Trans. Pattern Anal. Mach. Intell., 22 (2000), pp. 888–905, https://doi.org/10.1109/34.868688.

[50] A. Singer, *Angular synchronization by eigenvectors and semidefinite programming*, Appl. Comput. Harmon. Anal., 30 (2011), pp. 20–36, https://doi.org/10.1016/j.acha.2010.02.001.

[51] O. Sporns and E. Bullmore, *Complex brain networks: Graph theoretical analysis of structural and functional systems*, Nat. Rev. Neurosci., 10 (2009), pp. 186–98, https://doi.org/10.1038/nrn2575.

[52] S. H. Strogatz, *Exploring complex networks*, Nature, 410 (2001), pp. 268–276, https://doi.org/10.1038/35065725.

[53] U. von Luxburg, *A tutorial on spectral clustering*, Stat. Comput., 17 (2007), pp. 395–416, https://doi.org/10.1007/s11222-007-9033-z.

[54] U. von Luxburg, M. Belkin, and O. Bousquet, *Consistency of spectral clustering*, Ann. Statist., 36 (2008), pp. 555–586, https://doi.org/10.1214/009053607000000640.

[55] X. Zhang and M. E. J. Newman, *Multiway spectral community detection in networks*, Phys. Rev. E, 92 (2015), 052808, https://doi.org/10.1103/PhysRevE.92.052808.

[56] Z. Zhang and M. I. Jordan, et al., *Multiway spectral clustering: A margin-based perspective*, Stat. Sci., 23 (2008), pp. 383–403, https://doi.org/10.1214/08-STS266.