NetLSD: Hearing the Shape of a Graph

Anton Tsitsulin
Hasso Plattner Institute
anton.tsitsulin@hpi.de

Davide Mottin
Hasso Plattner Institute
davide.mottin@hpi.de

Panagiotis Karras
Aarhus University
panos@cs.au.dk

Alex Bronstein
Technion
bron@cs.technion.ac.il

Emmanuel Müller
Hasso Plattner Institute
emmanuel.mueller@hpi.de

ABSTRACT

Comparison among graphs is ubiquitous in graph analytics. However, it is a hard task in terms of the expressiveness of the employed similarity measure and the efficiency of its computation. Ideally, graph comparison should be invariant to the order of nodes and the sizes of compared graphs, adaptive to the scale of graph patterns, and scalable. Unfortunately, these properties have not been addressed together. Graph comparisons still rely on direct approaches, graph kernels, or representation-based methods, which are all inefficient and impractical for large graph collections.

In this paper, we propose the Network Laplacian Spectral Descriptor (NetLSD): the first, to our knowledge, permutation- and size-invariant, scale-adaptive, and efficiently computable graph representation method that allows for straightforward comparisons of large graphs. NetLSD extracts a compact signature that inherits the formal properties of the Laplacian spectrum, specifically its heat or wave kernel; thus, it *hears the shape of a graph*. Our evaluation on a variety of real-world graphs demonstrates that it outperforms previous works in both expressiveness and efficiency.

1 INTRODUCTION

Graphs are widely used for modeling complex structures such as biological networks, chemical compounds, social interactions, and knowledge bases. Such applications require the means to efficiently and meaningfully compare one graph to others.

Arguably, an ideal means for graph comparison should fulfill the following desiderata: First, it should be indifferent to the order in which nodes are presented; we call this property **permutation-invariance**. Second, it would enable graph comparisons both at a

local level (expressing, e.g., atomic bond differences among chemical compounds) and at the *global* or community level (capturing, e.g., the different topologies of social networks); we call this facility **scale-adaptivity**. Third, it would detect structural similarity regardless of network magnitude (discerning, e.g., the similarity of two criminal networks of different size); we call this aptitude **size-invariance**. Unfortunately, no existing means for graph comparison satisfies all three of these requirements.

Apart from these qualitative requirements, a viable means for graph comparison should be *efficiently computable*. Graph analytics tasks often require pairwise graph comparisons within a large collection of graphs, hence should be ideally done in *constant time*, after preprocessing. Unfortunately, existing methods fare even worse in this respect. A popular distance measure among graphs, *graph edit distance* (GED) [38], defined as the minimum number of edit operations required to turn one graph into another, is **NP**-hard and **APX**-hard to compute [27]; intensive research efforts in GED-based graph comparison [18, 26, 51–53] have not escaped this reality. Similarly, *graph kernel methods* [8, 17, 24, 32, 39, 50] lack an explicit graph representation and they do not scale well either.

In this paper, we develop a permutation- and size-invariant, scale-adaptive, and scalable method for graph comparison. We propose an expressive *graph representation*, NETLSD, grounded on spectral graph theory (Sec. 4), that allows for *constant-time* similarity computations at several scales; NETLSD extracts compact graph signatures based on the heat or wave kernel of the Laplacian, which inherit the formal properties of the Laplacian spectrum.

Figure 1 shows an example: at a small scale, a ring and a wheel are similar, as each node is connected to predecessor(s) and successor(s); at a large scale, they both appear as rings; at intermediate scales their local structures differ.
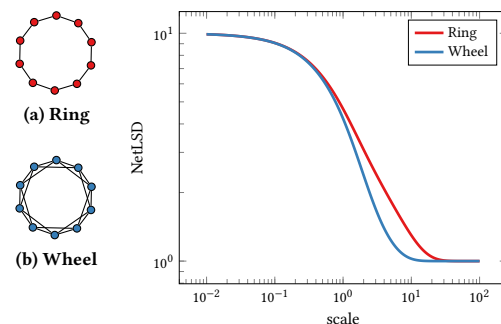

Figure 1: Comparison of similar graphs on multiple scales.

Method	Type	Properties			Complexity	
		Perm.-invariance	Scale-adaptivity	Size-Invariance	Precomputation	Comparison
GED [38]	direct	✔	✘	✘	$O(1)$	NP-complete
DELTACON [25]	direct	✘	✘	✘	$O(1)$	$O(m)$
FTD [4]	direct	✘	✘	✘	$O(1)$	$O(n^2)$
SP [8]	kernel	✔	✘	✘	$O(1)$	$O(n^3)$
MLG [24]	kernel	✔	✔	✘	$O(km + k^2 n)$	$O(k^3)$
NETSIMILE [6]	statistical representation	✔	✘	✔	$O(n \log n)$	$O(1)$
FGSD [45]	statistical representation	✔	✘	✘	$O(n^2)$	$O(1)$
NETLSD	spectral representation	✔	✔	✔	$O(km + k^2 n)$	$O(1)$

Table 1: Related work in terms of fulfilled (✔) and missing (✘) properties, complexity (n nodes, m edges, k eigenvalues).

2 RELATED WORK

We distinguish methods for graph comparison into three categories: *direct*, *kernel methods*, and *statistical representations*. To the best of our knowledge, this is the first work employing a *spectral representation* for graph comparison.

2.1 Direct Methods

Graph edit distance (GED) [38] is the minimal number of edit operations needed to transform one graph into another; unfortunately, its calculation is **NP**-hard [16] and even hard to approximate (**APX**-hard) [27], as it implies determining a *correspondence* among the compared graphs' nodes, a computationally hard task to begin with [17, 24, 32, 39, 50]; some application-specific techniques (e.g., anomaly detection in time evolving graphs) impose assumptions on that correspondence to alleviate the burden [4, 25, 34]; while GED admits heuristic approximation [14, 37] and indexing schemes [26, 52, 54], it is not applicable for generic comparison among large graph collections. Besides, GED treats all edit operations as equal, without discerning the extent to which they may alter the graph topology. Thus, even if the computational obstacle were surmounted, GED would still be unsatisfactory as a measure for multi-scale and multi-size graph comparisons.

In an effort to overcome GED's shortcomings, past works have resorted to more flexible distance definitions, based on vertex and path similarities [34] or propagation models [25]. While such measures are less sensitive to local changes than GED, they usually require node to be aligned in advance, hence are limited to specific applications. An attempt to abandon this requirement relaxes the permutation matrix required to define a full node correspondence and considers a family of tractable distances (FTD) [4]. This relaxed matrix preserves local properties and is easy to compute, yet the ensuing measure is still sensitive to permutations.

2.2 Kernel Methods

Graph kernels [8, 17, 24, 32, 39, 50] are similarity functions among graphs, which typically perform an implicit transformation of graph structure to compare two graphs (e.g. Shortest-path (SP) kernel [39]). While each kernel function has its own valuable properties, to our knowledge, no extent graph kernel achieves both scale-adaptive and size-invariant graph comparison. Besides, kernels require expensive on-demand computations at comparison time, hence are inapplicable for large-scale graph comparisons.

Recently, Kondor and Pan proposed the Multi-scale Laplacian Graph kernel (MLG) [24], which achieves scale-adaptivity by exploiting the propagation of information in the graph and summing the information at each iteration. However, it also raises a computational overhead cubic in Laplacian matrix eigenvalues.

2.3 Statistical Representations

Representation-based methods generate an one-off graph signature vector, based on statistical properties, and use it in subsequent inter-graph comparisons. Preliminary works in this area [6, 7, 9] handcraft features by aggregating local graph properties such as a node's and its neighbors' degrees. Such representations are easy to compute, yet focus on local characteristics, and are oblivious to global features. A more advanced approach, the Family of Spectral Distances (FSGD), [45] produces a high-dimensional sparse representation as a histogram on the dense biharmonic graph kernel; however, FGSD does not capture graph features at different scales of resolution or graph sizes, and is also inapplicable to reasonably large graphs, due to its quadratic time complexity.

2.4 Spectral Representations

Spectral graph theory is effective in the comparison of 3D objects [9, 15, 33]. Bronstein et al. [10] proposed computing pairwise shape similarities by finding the minimum-distortion embedding of one shape into the other. This procedure is related to the computation of the Gromov-Hausdorff distance [30], which is suitable for shape comparison, yet hard to compute. Most relevant works consider filtering functions corresponding to known diffusion models, including heat [11, 41], wave [2] and commute time [11] distances. While it is known that hitting and commute times degenerate to a function of node degree on large graphs [47], it is unknown whether other filters designed for three-dimensional manifolds maintain their expressiveness on high-dimensional graphs.

While 3D objects have a precise low-dimensional shape, graphs have no rigid form. Yet we can consider a graph as a geometrical object. Günthard and Primas first asked to what extent a graph (in general, a manifold) may be determined by its spectrum [?]; Kac expressed the same question eloquently: "Can one *hear the shape* of a drum?" [20]. Since then, investigations have shown that isospectral graphs tend to be isometric [49], and some graphs are determined by their spectrum [?]. Thus, spectral graph theory forms a solid ground for graph comparison. The Laplacian spectrum is used in graph mining [1, 40], yet, to our knowledge, no previous work has used it as a means for graph comparison.

3 PROBLEM STATEMENT

An undirected graph is a pair $G = (V, E)$, where $V = (v_1, \ldots, v_n)$, $n = |V|$ is the set of vertices and $E \subseteq (V \times V)$ the set of edges. We assume the graph is unweighted even though our method readily applies to the weighted case. A *representation* is a function $\sigma : \mathcal{G} \rightarrow \mathbb{R}^{\mathbb{N}}$ from any graph G in a collection of graphs $\mathcal{G}$ to an infinitely dimensional real-vector; the element j of the representation is denoted as $\sigma_j(G)$. A *representation-based distance* is a function $d^{\sigma} : \mathbb{R}^{\mathbb{N}} \times \mathbb{R}^{\mathbb{N}} \rightarrow \mathbb{R}_0^+$ on the representations of two graphs $G_1, G_2 \in \mathcal{G}$ that returns a positive real number. We aim to devise a constant-time *representation-based distance* among any pair of graphs G_1, G_2.

3.1 Expressive Graph Comparison

Our distance should support data mining tasks, such as clustering, nearest neighbor classification, distance-based anomaly detection. Therefore, we require our distance to be a *pseudometric*; namely, it should fulfill the following properties.

- *Symmetry*, for any $G_1, G_2 \in \mathcal{G}$:
  $d^{\sigma}(\sigma(G_1), \sigma(G_2)) = d^{\sigma}(\sigma(G_2), \sigma(G_1))$
- *Triangle inequality*, for any $G_1, G_2, G_3 \in \mathcal{G}$:
  $d^{\sigma}(\sigma(G_1), \sigma(G_3)) \le d^{\sigma}(\sigma(G_1), \sigma(G_2)) + d^{\sigma}(\sigma(G_2), \sigma(G_3))$

These properties characterize a large family of distances, yet do not reflect their expressiveness. We require *expressive distances* to be permutation-invariant, scale-adaptive, and size-invariant.

Permutation-invariance implies that if two graphs' structure are the same (i.e., if the two graphs are isomorphic) the distance of their representations is zero. A graph $G_1 = (V_1, E_1)$ is isomorphic to another graph $G_2 = (V_2, E_2)$, or $G_1 \simeq G_2$, if there exists a bijective function $\mu : V_1 \rightarrow V_2$ such that $(\mu(u), \mu(v)) \in E_2$ for each $(u, v) \in E_1$.

PROPERTY 1 (PERMUTATION-INVARIANCE). *A distance d^{σ} on representation σ is permutation-invariant iff:*

$$\forall G_1, G_2, \ G_1 \simeq G_2 \Rightarrow d^{\sigma}(\sigma(G_1), \sigma(G_2)) = 0$$

Scale-adaptivity implies that a representation accounts for both local (edge and node) and global (community) graph features. A global feature cannot be captured by *any* combination of features on nodes at distance $r < \mathbf{D}(G) - 1$, where $\mathbf{D}(G)$ is the diameter (longest shortest-path length) of G. Let the set of all subgraphs of G be $\xi(G) = \{g \sqsubset G : \mathbf{D}(g) < \mathbf{D}(G)\}$. We define scale-adaptivity as the property of a representation σ having at least one local feature (i.e., derived only from information encoded in subgraphs $\xi(G)$), and at least one global feature (i.e., derived by strictly more than the information encoded in any $\xi(G)$). Using local features only, a similarity measure would deem two graphs sharing local patterns to have near-zero distance although their global properties (such a page-rank features) may differ, and, in reverse, relying on global features only would miss local structures (such as edge distributions). We aim for a representation adaptive to both local and global structures on demand.

PROPERTY 2 (SCALE-ADAPTIVITY). *A representation σ is scale-adaptive iff it contains both local features σ_i and global features σ_j:*

- *Local Feature:* $\forall G \ \exists f(\cdot) : \ \sigma_i = f(\xi(G))$
- *Global Feature:* $\forall G \ \nexists f(\cdot) : \ \sigma_j = f(\xi(G))$

Size-invariance is the capacity to discern that two graphs represent the same phenomenon at a different magnitudes (e.g., two criminal circles of similar structures but different sizes should have near-zero distance). We can think of a graph as a representation of a metric space (a *manifold* in particular) with a small intrinsic dimension. We would then like to abstract away the particular way of sampling that space. Size-invariance postulates that if two graphs originate from the sampling of the same domain $\mathcal{M}$, they should be deemed similar.

PROPERTY 3 (SIZE-INVARIANCE). *A size-invariant distance d^{σ} on representation σ fulfills:*

$$\forall \mathcal{M} : G_1, G_2 \text{ sampled from } \mathcal{M} \Rightarrow d^{\sigma}(\sigma(G_1), \sigma(G_2)) = 0$$

An expressive means of graph comparison should employ a representation fulfilling the above properties. NETLSD, presented in the sequel, is a graph representation that allows for easy and expressive graph comparison.

4 NETLSD: NETWORK LAPLACIAN SPECTRAL DESCRIPTOR

Defining a representation fulfilling the requirements of permutation-, scale-, size- invariance, and efficiency is tough; in general, structures are hard to compare. Thus, we transfer the problem to the spectral domain. A useful metaphor is that of heating the graph's nodes and observing the heat diffusion as time passes. Another useful metaphor is that of a system of masses corresponding to the graph's nodes and springs corresponding to its edges. The propagation of mechanical waves through the graph is another way to capture its structural invariants. In both cases, the overall process describes the graph in a permutation-invariant manner, and embodies more global information as time elapses. Our representation employs a *trace signature* encoding such a *heat diffusion* or *wave propagation* process over time. We compare two graphs via the L_2 distance among trace signatures sampled at selected time scales.

4.1 Spectra as representations

The *adjacency matrix* of a graph G is a $n \times n$ matrix A having $A_{ij} = 1$ if $(i, j) \in E$ and $A_{ij} = 0$ otherwise. A graph's *normalized Laplacian* is the matrix $\mathcal{L} = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$, where D is the diagonal matrix with the degree of node i as entry D_{ii}, i.e, $D_{ii} = \sum_{j=1}^n A_{ij}$. Since the Laplacian is a symmetric matrix, its eigenvectors $\phi_1, \ldots, \phi_n$, are real and orthogonal to each other. Thus, it is factorized as $\mathcal{L} = \Phi \Lambda \Phi^{\top}$, where Λ is a diagonal matrix on the sorted eigenvalues $\lambda_1 \le \ldots \le \lambda_n$ of which $\phi_1, \ldots, \phi_n$ are the corresponding eigenvectors, and Φ is an orthogonal matrix obtained by stacking the eigenvectors in columns $\Phi = [\phi_1 \phi_2 \ldots \phi_n]$.

The set of eigenvalues $\{\lambda_1, \ldots, \lambda_n\}$ is called the *spectrum* of a graph. The Normalized Laplacian, as opposed to the unnormalized version $L = D - A$, has a bounded spectrum, $0 \le \lambda_i \le 2$.

Belkin and Niyogi [3] showed that eigenvectors of the normalized Laplacian of a point cloud graph converge to the eigenfunction of the Laplace-Beltrami operator [5] on the underlying Riemannian manifold. In general, the normalized Laplacian has more attractive theoretical properties than its unnormalized counterparts [46].

The Laplacian spectrum encodes important graph properties, such as the normalized cut size [40] used in spectral clustering. Likewise, the normalized Laplacian spectrum can determine whether a graph is bipartite, but not the number of its edges [13]. Rather than using the Laplacian spectrum per se, we consider an associated heat diffusion process on the graph, to obtain a more expressive representation, in a manner reminiscent of random walk models [12].

The *heat equation* associated with the Laplacian is

$$\frac{\partial u_t}{\partial t} = -\mathcal{L} u_t, \tag{1}$$

where u_t are scalar values on vertices representing the heat of each vertex at time t. The solution to the heat equation provides the heat at each vertex at time t, when the initial heat u_0 is initialized with a fixed value on one of the vertices. Its closed-form solution is given by the $n \times n$ *heat kernel* matrix:

$$H_t = e^{-t\mathcal{L}} = \sum_{j=1}^{n} e^{-t\lambda_j} \phi_j \phi_j^\top, \tag{2}$$

where $(H_t)_{ij}$ represents the amount of heat transferred from vertex v_i to vertex v_j at time t. We can also compute the heat kernel matrix directly by exponentiating the Laplacian eigenspectrum [13]:

$$H_t = \Phi e^{-t\Lambda} \Phi^\top \tag{3}$$

As the heat kernel involves pairs of nodes, it is not directly usable to compare graphs. We rather consider the *heat trace* at time t:

$$h_t = \text{tr}(H_t) = \sum_j e^{-t\lambda_j} \tag{4}$$

Then our NETLSD representation consists of a **heat trace signature** of graph G, i.e., a collection of heat traces at different *time scales*, $h(G) = \{h_t\}_{t>0}$.

Alternative signatures. The heat kernel can be viewed as a family of *low-pass filters*, $F(\lambda) = e^{-\lambda t}$, parametrized by the scale parameter t, hence the heat trace signature contains low frequency (i.e., large-scale) information at every scale. Other kernels deemphasize the influence of low frequencies. For example, the *wave equation*,

$$\frac{\partial^2 u_t}{\partial t^2} = -\mathcal{L} u_t, \tag{5}$$

which describes the amplitude u_t of a wave propagating in a medium, has, in its turn, a solution given by the *wave kernel*:

$$W_t = e^{-it\mathcal{L}} = \sum_{j=1}^{n} e^{-it\lambda_j} \phi_j \phi_j^\top \tag{6}$$

(note the complex exponential) and a corresponding **wave trace signature** with $t \in [0, 2\pi)$:

$$w_t = \text{tr}(W_t) = \sum_j e^{-it\lambda_j} \tag{7}$$

4.2 Scaling to large graphs

The full eigendecomposition of the Laplacian $\mathcal{L} = \Phi \Lambda \Phi^\top$ takes $O(n^3)$ time and $\Theta(n^2)$ memory. This allows to compute signatures of graphs with over a thousand nodes in less than a second on commodity hardware, yet renders direct computation impossible for larger graphs. Thus, we need to approximate heat trace signatures. We propose two different methods to that end.

Our first proposal is to use a **Taylor expansion**; while this mathematical tool provides a rather dubious approximation of a matrix exponential, as its convergence rate depends on the largest eigenvalue [31], it is useful on *small time scales t*, and allows for an inexpensive computation of its first two terms,

$$h_t = \sum_{k=0}^{\infty} \frac{\text{tr}((-t\mathcal{L})^k)}{k!} \approx n - t\,\text{tr}(\mathcal{L}) + \frac{t^2}{2}\,\text{tr}(\mathcal{L}^2) \tag{8}$$

These first two terms are easily computed, even for very large graphs, as $\text{tr}(\mathcal{L}) = n$ and $\text{tr}(\mathcal{L}^2) = \sum_{ij} \mathcal{L}_{ij}^2$ since $\mathcal{L}$ is self-adjoint. This way, we can compare two graphs locally in $O(m)$.
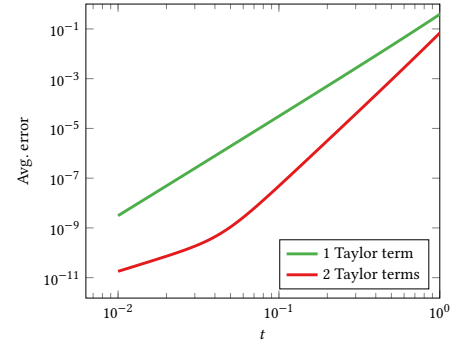

Figure 2: Relative approximation error of normalized h_t for Erdős-Rényi random graphs, varying time scale t.

Figure 2 depicts the error in approximating the normalized heat trace by a Taylor expansion for random graphs of varying sizes; this error is independent of graph size, and stays low until time scale 1. At *larger time scales*, the influence of high frequencies (i.e., the higher part of the spectrum) on the heat trace decreases. Thus, one can approximate the heat trace signature using the lower part of the eigenspectrum, as in shape analysis [11, 28, 41, 44]. Thus, we may apply the low-order Taylor expansion for *small t* and the truncated spectrum approximation for *large t*. However, this approach misses out on the medium scale, as Figure 3 illustrates. Besides, this technique does not lend itself to comparing graphs with different numbers of computed eigenvalues, since the spectrum discretization ratio is not normalized across the networks.
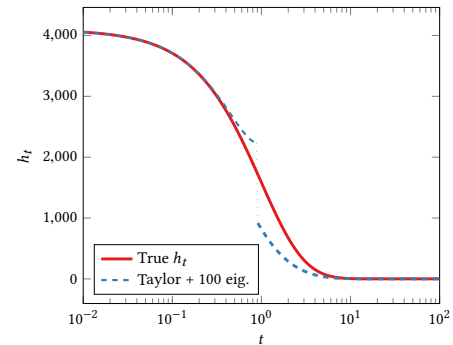

Figure 3: Heat trace approximation by two Taylor terms and 100 eigenvalues on a random blockmodel [21] graph.

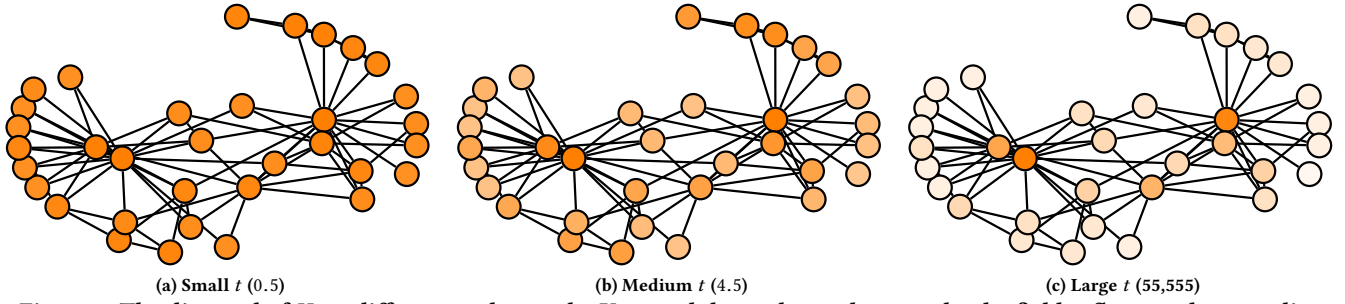(a) Small t (0.5) (b) Medium t (4.5) (c) Large t (55,555)

Figure 4: The diagonal of H_t at different scales on the Karate club graph; at a large scale, the field reflects node centrality.
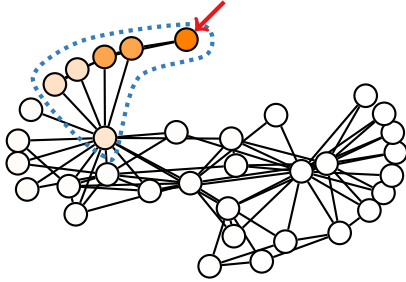

Figure 5: Visualization of the diagonal in the heat kernel matrix H_t for the pointed vertex at scale $t = 0.3$

We conclude that the Taylor expansion is useful on very large graphs, on which eigendecomposition is prohibitive. For manageable graph sizes, we adopt a more accurate strategy [31] based on approximating the *eigenvalue growth rate*, as in [44]: we compute k eigenvalues on *both* ends of the spectrum, and interpolate a *linear growth* of the interloping eigenvalues. This strategy assumes that on the medium scale the manifold defining the graph is two-dimensional, as Weyl's law of asymptotic eigenvalue growth suggests [48]. Since we only need to compute extreme eigenvalues, we use the block Krylov-Schur implementation in SLEPc [19]. Graph Laplacians always have a zero eigenvalue with eigenspace dimension equal to the number of connected components; we deflate the search space for the eigensolver thanks to this property. In our experimental study, we employ this *interpolation technique*; In Section 5.1, we evaluate its approximation quality.

4.3 Properties of the heat trace

Here, we discuss how the heat trace signatures achieves our target representation properties.

Permutation-invariance. The permutation invariance of $h(G)$ follows from the properties of the spectrum: isomorphic graphs are isospectral, hence their respective heat trace signatures are equal.

Scale-adaptivity. The heat kernel can be seen as continuous-time random walk propagation, and its diagonal (sometimes referred to as the autodiffusivity function or the heat kernel signature) can be seen as a continuous-time PageRank [12]. Figure 4 shows the heat kernel signature with small (a), medium (b), and large (c) t; with large t, the heat tends to focus on central nodes.

As t approaches zero, the Taylor expansion yields $H_t \simeq I - \mathcal{L}t$, meaning the heat kernel depicts *local connectivity*. On the other hand, for large t, $H_t \simeq I - e^{-\lambda_2 t}\phi_2\phi_2^\top$, where ϕ_2 is the *Fiedler*

vector used in spectral graph clustering [40], as it encodes *global connectivity*. Thus, the heat kernel localizes around its diagonal, and the degree of localization depends on the scale t; it can thereby be tuned to produce both local and global features.

Figure 5 illustrates heat kernel locality, focusing on a single row of H_t corresponding to the node marked with a red arrow; the kernel is localized in the region marked with a dotted line.

Size-invariance. While the normalized Laplacian alleviates the problem of different edge densities [13], the Taylor expansion in Equation 8 manifests that $h(G)$ contains information about the number of nodes. Fortunately, we can employ *neutral* graphs for normalization, namely the empty and the complete graph with n nodes. Eigenvalues of the normalized Laplacian of an empty graph $\bar{K}_n$ of size n are all zero; for a complete graph K_n, they are given by a vector of one zero and $n-1$ ones. Thus, the heat traces of these graphs are analytically computed as:

$$h_t(\bar{K}_n) = \frac{1}{n}$$
$$h_t(K_n) = 1 + (n-1)e^{-t}.$$

and their wave traces are:

$$w_t(\bar{K}_n) = \frac{1}{n}$$
$$w_t(K_n) = 1 + (n-1)\cos(t).$$

Either option can be used to normalize the heat or wave trace signatures. Such normalization can be interpreted as a modification of the corresponding diffusivity tensor (for the heat trace) or the elasticity tensor (for the wave trace). We provide more details in Section 5.

4.4 Connection to computational geometry

Mémoli [29] suggests a spectral definition of the Gromov-Wasserstein distance between Riemannian manifolds based on matching the heat kernels at all scales. The cost of matching a pair of points (x, x') on manifold $\mathcal{M}$ to a pair of points (y, y') on manifold $\mathcal{N}$ at scale t is given by

$$\Gamma(x, y, x', y', t) = |H_t^{\mathcal{M}}(x, x') - H_t^{\mathcal{N}}(y, y')|.$$

The distance between the manifolds is then defined in terms of the infimal measure coupling

$$d(\mathcal{M}, \mathcal{N}) = \inf_{\mu} \sup_{t>0} e^{-2(t+t^{-1})} \|\Gamma\|_{L^2(\mu \times \mu)},$$
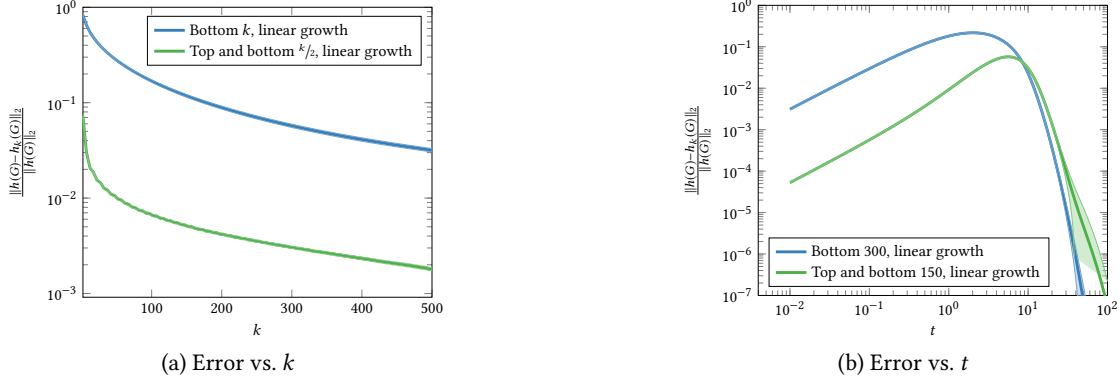
(a) Error vs. k

(b) Error vs. t

Figure 6: Relative error in spectrum computation of $h(G)$, averaged across 2085 graphs.

where the infimum is sought over all measures on $\mathcal{M} \times \mathcal{N}$ marginalizing to the standard measures on $\mathcal{M}$ and $\mathcal{N}$. Mémoli [29] shows that this distance can be lower bounded by

$$d(\mathcal{M}, \mathcal{N}) \geq \sup_{t>0} e^{-2(t+t^{-1})} |h_t^{\mathcal{M}} - h_t^{\mathcal{N}}|.$$

In other words, the lower bound is the scaled L_∞ distance between the heat trace signatures $h(\mathcal{M})$ and $h(\mathcal{N})$.

We effectively adopt this result *mutatis mutandis* to graphs, substituting the Laplace-Beltrami operator of the manifold with the normalized graph Laplacian; we then sample the heat trace signature at a finite number of scales, rendering it a versatile vector representation of a graph, as other works produce vector representations of graph vertices [36, 43]. This lower bound implies that, if the distance between heat trace signatures is sufficiently large, the compared graphs cannot be similar. Unlike the spectral Gromov-Wasserstein distance, distances between heat trace signatures are easily *indexable*; thus, the derived bound allows for efficiently pruning of dissimilar graphs while working with the index alone, as with any lower-bounding scheme for high-dimensional search [22, 42].

			vertices $|V|$		
dataset	$|\mathcal{G}|$	$|Y|$	min	avg	max
MUTAG	188	2	10	17.93	28
PTC	344	2	2	25.56	109
PROTEINS	1113	2	4	39.06	620
NCI1	4110	2	3	29.87	111
NCI109	4127	2	4	29.68	111
ENZYMES	600	6	2	32.63	126
D&D	1178	2	30	284.3	5748
COLLAB	5000	3	32	74.49	492
IMDB-B	1000	2	12	19.77	136
IMDB-M	1500	3	7	13.00	89
REDDIT-S	9543	2	100	337.8	999
REDDIT-M	2000	2	6	429.6	3782
REDDIT-X	2085	—	1001	1447.2	5242
REDDIT-L	291	2	4445	97491	1632141

Table 2: Dataset properties.

5 EXPERIMENTAL EVALUATION

We evaluate[1] NetLSD against NetSimile [6] and FGSD [45]. We run the experiments on a 20-core Intel Xeon CPU E5-2640v4, 3.20GHz

---

[1]Source code and data are available at https://github.com/xgfs/NetLSD

machine with 256Gb RAM. Each method is assessed on the *best parameters* through cross-validation. We require each method to complete within one day, or else an early termination is issued. We used graph-tool [35] for graph manipulation and synthetic graph generation.

Benchmarks. We compare to the following comparison methods.

NetSimile [6]: a representation using handcrafted features obtained by aggregating statistics on nodes and edges (e.g., average degree, standard deviation of the degree of the neighbors). For each graph, the resulting representation has 35 dimensions. As recommended in [6], we use the Canberra[2] distance for comparison.

FGSD [45]: a method that computes histograms on the biharmonic kernel of the graph. Such histograms typically bear large dimensionality ($\geq$50.000) using the recommended bin-width 0.0001.

Parameter settings. Unless otherwise stated, we repeat each experiment 100 times and report the average across all trials. NetLSD is instantiated to both heat $h(G)$ and wave $w(G)$ trace signature described Section 4.1. In addition, we evaluate our normalized versions: the normalization with empty graph $h(G)/h(\bar{K})$, $w(G)/w(\bar{K})$ and the normalization with complete graph $h(G)/h(K)$, $w(G)/w(K)$. To build NetLSD signature vectors, we need to sample a number of traces, i.e., values of t. After experimentation, we settled for 250 values evenly spaced on the logarithmic scale in the range $[10^{-2}, 10^2]$; we attested this to be a good choice in terms of the quality-size tradeoff, hence use these settings in all experiments, with both heat and wave signatures. With small graphs, we employ the full eigendecomposition to produce the trace for each t value. With the larger REDDIT-L graph, we use 300 eigenvalues, 150 from each side of the eigenspectrum, by default, unless indicated otherwise; we validate these choices in Section 5.1 (cf. Figure 6).

Datasets. We use six graph collections from bioinformatics [50] and three social networks. COLLAB, and IMDB-(B/M) graph collections are obtained sampling neighborhoods of nodes in a collaboration and movie network, respectively. REDDIT-S, REDDIT-M, and REDDIT-L are selected among small (at most 1000 nodes), medium (at most 4000 nodes) and large (> 4000 nodes) subreddits[3], respectively. We report their main characteristics in Table 2: number of graphs $|\mathcal{G}|$, number of labels for classification $|Y|$, minimum, average, and maximum number of vertices.

---

[2]https://en.wikipedia.org/wiki/Canberra_distance
[3]https://dynamics.cs.washington.edu/data.html

dataset	$h(G)$	$h(G)/h(\bar{K})$	$h(G)/h(K)$	$w(G)$	$w(G)/w(\bar{K})$	$w(G)/w(K)$	FGSD	NETSIMILE
MUTAG	76.03	79.12	78.22	78.18	79.72	79.38	77.79	77.11
PTC	56.41	62.53	63.11	58.55	64.28	60.46	54.75	62.12
PROTEINS	91.81	94.90	95.31	93.04	89.00	91.27	60.11	85.73
NCI1	69.74	74.55	69.89	70.54	74.14	70.90	64.08	58.58
NCI109	68.60	73.76	69.48	70.75	73.96	70.67	64.28	58.76
ENZYMES	92.51	95.20	95.70	94.03	90.77	90.10	53.93	87.38
COLLAB	59.82	65.85	69.74	69.01	70.35	71.89	55.18	54.43
IMDB-B	67.18	70.58	69.22	75.26	75.54	74.13	56.23	54.44
IMDB-M	74.45	75.51	75.54	77.99	78.68	76.97	56.31	48.06

Table 3: ROC AUC in detecting whether a graph is real.

dataset	$h(G)$	$h(G)/h(\bar{K})$	$h(G)/h(K)$	$w(G)$	$w(G)/w(\bar{K})$	$w(G)/w(K)$	FGSD	NETSIMILE
MUTAG	86.47	85.32	84.66	83.35	81.72	82.22	84.90	84.09
PTC	55.30	52.76	51.16	54.97	54.53	53.40	60.28	61.26
PROTEINS	64.89	65.73	65.36	66.80	65.58	62.27	65.30	62.45
NCI1	66.49	67.44	64.82	70.78	67.67	62.19	75.77	66.56
NCI109	65.89	66.93	64.78	69.32	67.08	63.53	74.59	65.72
ENZYMES	31.99	33.31	37.19	40.41	35.78	28.75	41.58	33.23
D&D	69.86	68.38	67.09	68.77	65.39	65.12	70.47	64.89
COLLAB	68.00	69.42	69.70	75.77	77.24	67.37	73.96	73.10
IMDB-B	68.04	70.17	69.45	68.63	69.33	61.67	69.54	69.20
IMDB-M	40.51	40.34	40.10	42.66	42.00	39.71	41.14	40.97
REDDIT-M	43.12	40.62	39.08	41.49	38.65	41.24	41.61	41.32
REDDIT-S	83.67	81.77	83.73	84.49	70.47	79.46	88.95	89.65

Table 4: Accuracy in 1-NN Classification.

5.1 Approximation quality

First, we study the quality of the approximation technique of Section 4.2. Figure 6 reports relative error results in terms of deviation from the exact version, averaged over 2085 REDDIT-X graphs. Specifically, Figure 6a shows the quality of the approximation varying the number of eigenvalues k, using only the k-smallest (blue line) and the $k/2$-smallest and $k/2$-largest eigenvalues (green line). Using eigenvalues from the two ends of the spectrum achieves consistently better performance. Figure 6b shows the impact of t in the approximation. The prediction is easier at large t, as the spectrum converges to the constant value 1; at medium t values, the approximation is harder. Still, the use of the lowest and largest eigenvalues delivers almost an order of magnitude higher accuracy than using only one side of the spectrum, vindicating our choice of approximation.

5.2 Identifying real-world networks

We devise a binary classification task of detecting whether a graph is real or synthetic. Such tasks are critical in anomaly detection and detecting bots and trolls in social networks. To render the task challenging enough, rather than generating purely random graphs (which are easy to detect), we produce synthetic graphs by rewiring real ones while preserving their degree distribution, using 10 iterations of shuffling all their edges via Metropolis-Hastings sampling [35]. We consider a label indicating whether the graph has been rewired. We label 80% of the dataset and test 20%, predicting a graph's label as that of its Nearest Neighbor (1-NN) by

each similarity measure, and report the average across 100 trials. Table 3 shows our results, in terms of ROC AUC, for all datasets and measures. These results confirm the effectiveness of NETLSD.

		k	
Method	100	200	300
$h(G)$	68.91	68.89	68.01
$h(G)/h(\bar{K})$	62.69	61.74	61.88
$h(G)/h(K)$	70.11	69.40	70.88
$w(G)$	71.27	69.93	68.93
$w(G)/w(\bar{K})$	64.79	65.81	65.90
$w(G)/w(K)$	64.51	69.49	72.64

Table 5: Accuracy in 1-NN Classification with REDDIT-L.

5.3 Graph classification

We now assess NETLSD on a traditional task of 1-NN graph classification, using labels as provided in the datasets in question and splitting training and testing as in the previous experiments. Table 4 reports the quality in terms of ROC AUC, averaged over 1000 trials. Again, NETLSD is on a par with other methods.

We additionally report in Table 5 the quality results for the largest dataset at our disposal (REDDIT-L). NETSIMILE and FGSD cannot scale to such large graphs and therefore we do not report on them; on the other hand, NETLSD processes graphs with millions of nodes and attains good overall quality.

5.4 Discerning community structures

Communities are set of vertices sharing common characteristics in terms of connectivity and attributes. Community detection [23, 40] is one of the prototypical, yet only partially solved, tasks in graphs. An expressive graph distance should discriminate graphs with community structure from those without.

To evaluate the expressiveness of NetLSD in terms of communities, we devise a graph classification experiment: We generate some graphs with community structure and some without, and the classifier's task is to predict whether test set graphs have community structure. We employ a simple 1-NN classifier, as our goal is to test the representation's expressiveness rather than the performance of the classifier. We generate 1000 random graphs with Poisson degree distribution $\mathcal{P}(\lambda)$ with mean degree $\lambda = 10$ and fixed size. Then, we sample another 1000 graphs from the stochastic block model (SBM) [21] with 10 communities, following the same degree distribution. The stochastic block model produces graphs with clear community structure as opposed to random ones. We use 80% of the dataset for training, and 20% for testing, repeat the experiment with different training and testing sets, and report the average across trials. Table 6 reports the average quality for discerning SBM graphs by 1-NN classification in terms of ROC-AUC, as we vary the graph size in (64, 128, 256, 512, 1024). NetLSD significantly outperforms the competitors and improves in quality on larger graphs. While discriminating very small communities is intuitively harder than distinguishing large ones, the performance of both FGSD and Net-Simile drops with increasing size, suggesting that these methods only capture local, small-scale variations. This result verifies the capacity of NetLSD to capture global and local characteristics.

	Number of nodes n				
Method	64	128	256	512	1024
$h(G)$	57.40	68.37	77.42	82.83	84.63
$h(G)/h(\bar{K})$	57.42	68.40	77.41	82.84	84.63
$h(G)/h(K)$	56.71	67.96	77.50	83.31	85.12
$w(G)$	57.47	66.97	73.95	78.43	80.26
$w(G)/w(\bar{K})$	57.44	66.98	73.96	78.43	80.25
$w(G)/w(K)$	56.75	66.26	73.06	77.05	78.76
FGSD	58.00	55.73	55.46	53.43	51.57
NetSimile	65.73	61.31	61.51	61.86	61.58

Table 6: Accuracy in detecting graphs with communities.

In the second experiment we evaluate the size-invariance of NetLSD: The task is to discriminate fixed-size communities in graphs of increasing size. Therefore, we sample the number of nodes n from a Poisson distribution $\mathcal{P}(\lambda = n)$ with variance λ, and then generate a network with n nodes. Again, we repeat the process 1000 times with purely random generated graphs and 1000 with random graphs with community structure, and perform a classification experiment as before. Table 7 reports the results, for increasing λ. Once again, NetLSD outperforms the competitors. This result confirms that the normalization proposed in Section 4.3 is effective in detecting community structures in graphs of different size. Table 7 also indicates that, while both normalizations are similarly effective with the wave kernel, the complete-graph normalization produces worse, but still competitive, results with the heat kernel.

	$n \sim \mathcal{P}(\lambda)$				
Method	64	128	256	512	1024
$h(G)$	54.39	59.01	60.82	57.99	53.80
$h(G)/h(\bar{K})$	54.53	62.27	70.83	76.45	78.40
$h(G)/h(K)$	54.37	60.93	66.86	68.24	65.23
$w(G)$	56.23	63.77	69.57	71.66	70.34
$w(G)/w(\bar{K})$	55.51	63.85	72.12	77.59	79.39
$w(G)/w(K)$	56.69	64.92	71.81	75.91	77.50
FGSD	55.44	54.99	53.86	52.74	50.92
NetSimile	59.55	56.57	59.41	66.23	60.58

Table 7: Accuracy in detecting graphs with communities, Poisson distribution of graph size.

Last, we assess the size-invariance of NetLSD in a tough regime: discriminating communities in graphs with a number of nodes *chosen uniformly at random*. We sample the number of nodes n from the uniform distribution $\mathcal{U}(10, \lambda)$; other experimental settings remain the same. Table 8 reports the results with growing λ. NetLSD outperforms the competitors once again, yet the results suggest that this task is more challenging. Complete-graph normalization for the wave kernel and empty-graph normalization for the heat kernel perform best. We conclude that, for fully size-agnostic comparisons, normalization should be chosen carefully.

	$n \sim \mathcal{U}(10, \lambda)$				
Method	64	128	256	512	1024
$h(G)$	51.16	51.98	51.62	51.05	50.08
$h(G)/h(\bar{K})$	51.19	53.36	56.63	59.10	59.71
$h(G)/h(K)$	51.69	53.67	55.43	55.29	53.43
$w(G)$	52.38	55.89	59.81	61.18	59.13
$w(G)/w(\bar{K})$	51.61	54.67	57.83	60.06	61.01
$w(G)/w(K)$	52.63	57.48	62.85	67.95	71.19
FGSD	57.92	55.62	54.94	52.74	52.15
NetSimile	63.63	58.31	55.75	54.34	53.11

Table 8: Accuracy in detecting graphs with communities, Uniform distribution of graph size.

5.5 Case Study

Here, we visualize the discovery potential of using NetLSD as a similarity measure. We run a *furthest pair query* on all graphs in our collection bar those in the REDDIT-L dataset, which are harder to visualize readably; thereby, we discover the two graphs of *lowest similarity*, using the normalized heat kernel. Figure 7 shows those two graphs: a *protein interaction network* from the D&D dataset and an *enzyme's tertiary structure* from the ENZYMES dataset. These two graphs are conspicuously different across multiple scales, from local patterns to global structure. The interaction network in Figure 7a is a small-world graph with large clustering coefficient (0.425) and small diameter ($\mathbf{D} = 9$), whereas the tertiary structure in Figure 7b is a connected collection of paths of big length with negligible clustering coefficient (0.006) and large ($\mathbf{D} = 32$) diameter.

(a) **Protein network.** $|V| = 190$, $|E| = 744$, **D = 9**.

(b) **Enzyme.** $|V| = 125$, $|E| = 141$, **D = 32**.
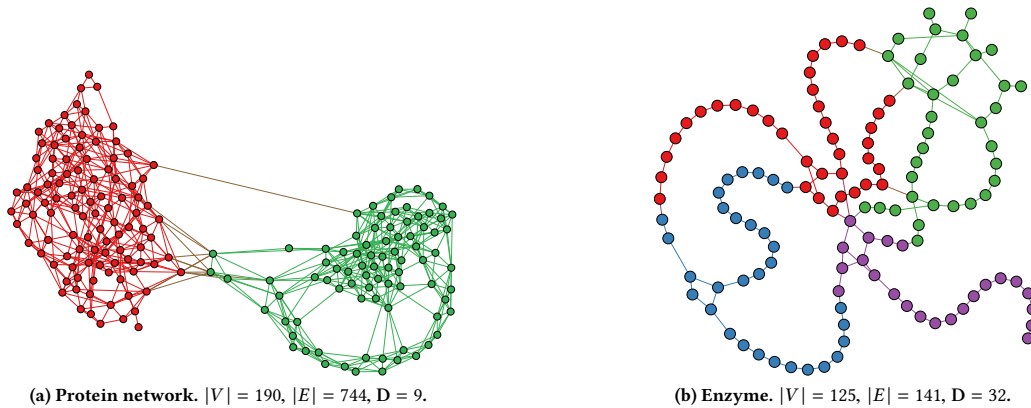
Figure 7: Two most dissimilar graphs by $h(G)/h(\bar{K})$ across all small-size graphs in datasets used. Communities are colored.

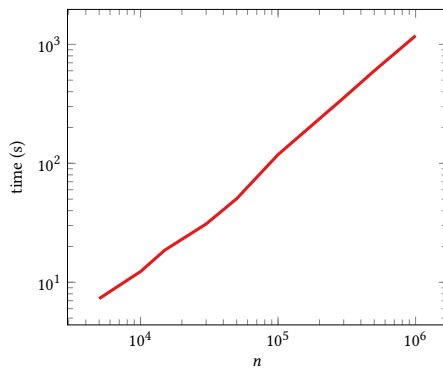**Figure 8: Time to compute** 300 **eigenvalues on both ends of the spectrum and the approximation of** $h(G)$**.**

5.6 Scalability

Last, we corroborate the scalability of NETLSD. Figure 8 shows the time to compute 300 eigenvalues for the approximation of $h(G)$ on graphs of increasing size from the REDDIT dataset. Our method computes the similarity on graphs of one million (10^6) nodes in only 16 minutes, while previous methods could not complete the process within one day. This result illustrates the fitness of NETLSD a scalable comparison method among real-size graphs.

6 CONCLUSIONS

We proposed NETLSD, a representation for graph comparison relying neither on (i) graph alignment operations, nor on (ii) computationally-demanding kernel computations, nor on (iii) supervised representation learning. NETLSD is a multi-scale heat trace signature of a graph Laplacian spectrum, which lower-bounds the Gromov-Wasserstein distance as it incorporates heat traces covering all scales. We derived a novel approximation of heat traces, rendering NETLSD efficiently computable, and a normalization scheme, rendering it size-invariant. To our knowledge, this is the first graph representation that achieves these properties and allows for comparisons at multiple scales. Our experiments show that NETLSD outperforms NETSIMILE and FGSD, two state-of-the-art representation-based methods for graph comparison, on a variety of graph collections on community detection and graph classification.

REFERENCES

[1] Leman Akoglu, Mary McGlohon, and Christos Faloutsos. 2010. Oddball: Spotting anomalies in weighted graphs. In *PAKDD*. 410–421.

[2] Mathieu Aubry, Ulrich Schlickewei, and Daniel Cremers. 2011. The wave kernel signature: A quantum mechanical approach to shape analysis. In *ICCV Workshops*. IEEE, 1626–1633.

[3] Mikhail Belkin and Partha Niyogi. 2007. Convergence of Laplacian eigenmaps. In *NIPS*. 129–136.

[4] Jose Bento and Stratis Ioannidis. 2018. A Family of Tractable Graph Distances. In *SDM*.

[5] Marcel Berger. 2012. *A panoramic view of Riemannian geometry*. Springer Science & Business Media.

[6] Michele Berlingerio, Danai Koutra, Tina Eliassi-Rad, and Christos Faloutsos. 2013. Network similarity via multiple social theories. In *ASONAM*. 1439–1440.

[7] Stephen Bonner, John Brennan, Ibad Kureshi, G Theodoropoulos, and AS Mc-Gough. 2016. Efficient Comparison of Massive Graphs Through The Use Of Graph Fingerprints. In *Twelfth Workshop on Mining and Learning with Graphs (MLG) Workshop at KDD'16*.

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

[9] Alexander M Bronstein, Michael M Bronstein, Leonidas J Guibas, and Maks Ovsjanikov. 2011. Shape google: Geometric words and expressions for invariant shape retrieval. *TOG* 30, 1 (2011), 1.

[10] Alexander M Bronstein, Michael M Bronstein, and Ron Kimmel. 2006. Efficient computation of isometry-invariant distances between surfaces. *SIAM Journal on Scientific Computing* 28, 5 (2006), 1812–1836.

[11] Michael M Bronstein and Alexander M Bronstein. 2011. Shape recognition with spectral distances. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33, 5 (2011), 1065–1071.

[12] Fan Chung. 2007. The heat kernel as the pagerank of a graph. *Proceedings of the National Academy of Sciences* 104, 50 (2007), 19735–19740.

[13] Fan RK Chung. 1997. *Spectral graph theory*. Number 92. American Mathematical Soc.

[14] Andreas Fischer, Ching Y Suen, Volkmar Frinken, Kaspar Riesen, and Horst Bunke. 2015. Approximation of graph edit distance based on Hausdorff matching. *Pattern Recognition* 48, 2 (2015), 331–343.

[15] Ran Gal, Ariel Shamir, and Daniel Cohen-Or. 2007. Pose-oblivious shape signature. *IEEE transactions on visualization and computer graphics* 13, 2 (2007), 261–271.

[16] Michael R Garey and David S Johnson. 2002. *Computers and intractability*. Vol. 29. wh freeman New York.

[17] Thomas Gärtner, Peter Flach, and Stefan Wrobel. 2003. On graph kernels: Hardness results and efficient alternatives. In *Learning Theory and Kernel Machines*. Springer, 129–143.

[18] Karam Gouda and Mosab Hassaan. 2016. CSI_GED: An efficient approach for graph edit similarity computation. In *ICDE*. 265–276.

[19] Vicente Hernandez, Jose E. Roman, and Vicente Vidal. 2005. SLEPc: A scalable and flexible toolkit for the solution of eigenvalue problems. *ACM Trans. Math. Software* 31, 3 (2005), 351–362.

[20] Mark Kac. 1966. Can one hear the shape of a drum? *The american mathematical monthly* 73, 4 (1966), 1–23.

[21] Brian Karrer and Mark EJ Newman. 2011. Stochastic blockmodels and community structure in networks. *Physical review E* 83 (2011).

[22] Shrikant Kashyap and Panagiotis Karras. 2011. Scalable kNN search on vertically stored time series. In *KDD*. 1334–1342.

[23] Kyle Kloster and David F Gleich. 2014. Heat kernel based community detection. In *KDD*. 1386–1395.

[24] Risi Kondor and Horace Pan. 2016. The multiscale laplacian graph kernel. In *NIPS*. 2990–2998.

[25] Danai Koutra, Joshua T Vogelstein, and Christos Faloutsos. 2013. Deltacon: A principled massive-graph similarity function. In *SDM*. 162–170.

[26] Yongjiang Liang and Peixiang Zhao. 2017. Similarity Search in Graph Databases: A Multi-Layered Indexing Approach. In *ICDE*. 783–794.

[27] Chih-Long Lin. 1994. Hardness of Approximating Graph Transformation Problem. In *ISAAC*. 74–82.

[28] Roee Litman and Alexander M. Bronstein. 2014. Learning Spectral Descriptors for Deformable Shape Correspondence. *IEEE Trans. Pattern Anal. Mach. Intell.* 36, 1 (2014), 171–180.

[29] Facundo Mémoli. 2011. A spectral notion of Gromov–Wasserstein distance and related methods. *Applied and Computational Harmonic Analysis* 30, 3 (2011), 363–401.

[30] Facundo Mémoli and Guillermo Sapiro. 2005. A theoretical and computational framework for isometry invariant recognition of point cloud data. *Foundations of Computational Mathematics* 5, 3 (2005), 313–347.

[31] Cleve Moler and Charles Van Loan. 2003. Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM review* 45, 1 (2003), 3–49.

[32] Giannis Nikolentzos, Polykarpos Meladianos, and Michalis Vazirgiannis. 2017. Matching Node Embeddings for Graph Similarity.. In *AAAI*. 2429–2435.

[33] Robert Osada, Thomas Funkhouser, Bernard Chazelle, and David Dobkin. 2002. Shape distributions. *ACM Transactions on Graphics (TOG)* 21, 4 (2002), 807–832.

[34] Panagiotis Papadimitriou, Ali Dasdan, and Hector Garcia-Molina. 2010. Web graph similarity for anomaly detection. *Journal of Internet Services and Applications* (2010), 19–30.

[35] Tiago P. Peixoto. 2014. The graph-tool python library. *figshare* (2014). https://doi.org/10.6084/m9.figshare.1164194

[36] Leonardo Filipe Rodrigues Ribeiro, Pedro H. P. Saverese, and Daniel R. Figueiredo. 2017. *struc2vec*: Learning Node Representations from Structural Identity. In *KDD*. 385–394.

[37] Kaspar Riesen and Horst Bunke. 2009. Approximate graph edit distance computation by means of bipartite graph matching. *Image and Vision computing* 27, 7 (2009), 950–959.

[38] Alberto Sanfeliu and King-Sun Fu. 1983. A distance measure between attributed relational graphs for pattern recognition. *IEEE Trans. Systems, Man, and Cybernetics* 13, 3 (1983), 353–362.

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

[40] Jianbo Shi and Jitendra Malik. 2000. Normalized cuts and image segmentation. *PAMI* 22, 8 (2000), 888–905.

[41] Jian Sun, Maks Ovsjanikov, and Leonidas Guibas. 2009. A concise and provably informative multi-scale signature based on heat diffusion. In *Computer graphics forum*, Vol. 28. Wiley Online Library, 1383–1392.

[42] Yufei Tao, Ke Yi, Cheng Sheng, and Panos Kalnis. 2009. Quality and efficiency in high dimensional nearest neighbor search. In *SIGMOD*. 563–576.

[43] Anton Tsitsulin, Davide Mottin, Panagiotis Karras, and Emmanuel Müller. 2018. VERSE: Versatile Graph Embeddings from Similarity Measures. In *the 2018 Web Conference*. 539–548.

[44] Amir Vaxman, Mirela Ben-Chen, and Craig Gotsman. 2010. A multi-resolution approach to heat kernels on discrete surfaces. In *TOG*, Vol. 29. 121.

[45] Saurabh Verma and Zhi-Li Zhang. 2017. Hunt For The Unique, Stable, Sparse And Fast Feature Learning On Graphs. In *NIPS*. 87–97.

[46] Ulrike Von Luxburg, Mikhail Belkin, and Olivier Bousquet. 2008. Consistency of spectral clustering. *The Annals of Statistics* (2008), 555–586.

[47] Ulrike Von Luxburg, Agnes Radl, and Matthias Hein. 2010. Hitting and commute times in large graphs are often misleading. *arXiv preprint arXiv:1003.1266* (2010).

[48] Hermann Weyl. 1911. Über die asymptotische Verteilung der Eigenwerte. *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse* (1911), 110–117.

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

[50] Pinar Yanardag and SVN Vishwanathan. 2015. Deep graph kernels. In *KDD*. 1365–1374.

[51] Ye Yuan, Guoren Wang, Lei Chen, and Haixun Wang. 2015. Graph similarity search on large uncertain graph databases. *The VLDB Journal* 24, 2 (2015), 271–296.

[52] Zhiping Zeng, Anthony K. H. Tung, Jianyong Wang, Jianhua Feng, and Lizhu Zhou. 2009. Comparing Stars: On Approximating Graph Edit Distance. *PVLDB* 2, 1 (2009), 25–36.

[53] Weiguo Zheng, Lei Zou, Xiang Lian, Dong Wang, and Dongyan Zhao. 2015. Efficient graph similarity search over large graph databases. *TKDE* 27, 4 (2015), 964–978.

[54] Yuanyuan Zhu, Lu Qin, Jeffrey Xu Yu, and Hong Cheng. 2012. Finding top-k similar graphs in graph databases. In *EDBT*. 456–467.