Random Spectral Features for Graph Kernel Machines

Valentin de Bassompierre, Jean-Charles Delvenne, Laurent Jacques
UCLouvain, Louvain-la-Neuve, Belgium

Motivation

Graph kernels are a key tool for machine learning on graphs and networks. They

- capture meaningful notions of node **similarity**.
- provide access to an enriched **feature space**.
- serve as an **upstream** input to many ML algorithms, *e.g.*, for signal reconstruction, classification, or link prediction [1].

Challenge: Graph kernels face **scalability** issues, with prohibitive $\mathcal{O}(N^3)$ explicit computation time.

Contributions: Using tools from graph signal processing and randomized linear algebra, we provide

- **low-rank** graph kernel approximations, through
- explicit **random feature** node representations.
- **accuracy** and **scalability** guarantees.

Graph Kernels

Given an undirected weighted graph with N nodes and E edges, graph kernels take the form

$$\mathbf{\Gamma} = h(\boldsymbol{L}) \in \mathbb{R}^{N\times N}, \tag{1}$$

where $\boldsymbol{L}$ is the normalized graph Laplacian, and h a non-negative kernel function [2]. Using the graph Fourier basis (eigenvector basis of $\boldsymbol{L}$),

$$\mathbf{\Gamma} = \boldsymbol{V} h(\mathbf{\Lambda}) \boldsymbol{V}^\top = \textstyle\sum_{k=1}^{N} h(\lambda_k) \boldsymbol{v}_k \boldsymbol{v}_k^\top. \tag{2}$$

Interpretation: h preserves specific eigenvalues, or graph frequencies. Typical kernels emphasize (smooth) low-frequency components and suppress high-frequency ones, such that h is non-increasing.

Problem: Full eigendecomposition (ED) is prohibitive to compute ($\mathcal{O}(N^3)$ time complexity). Need of a method which bypasses explicit computation of the eigenvectors $\boldsymbol{V}$.

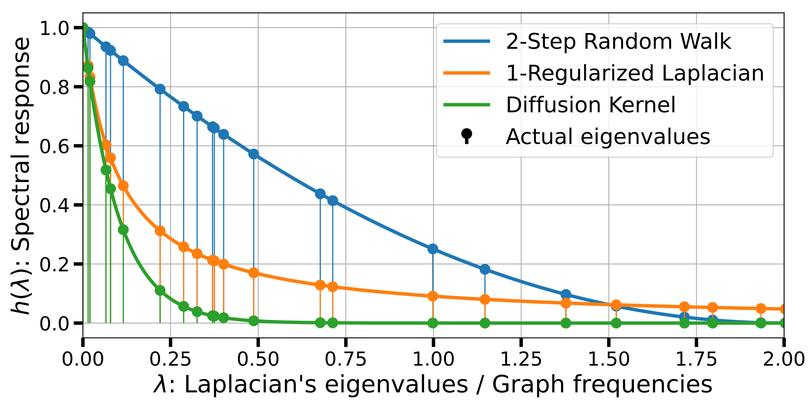


Figure 1: Some examples of graph kernels in the spectral domain (σ controls the bandwidth). Blue: $\mathbf{\Gamma} = (\boldsymbol{I} - \boldsymbol{L}/\sigma)^2$, orange: $\mathbf{\Gamma} = (\boldsymbol{I} + \sigma^2\boldsymbol{L})^{-1}$, green: $\mathbf{\Gamma} = \exp(-\sigma^2\boldsymbol{L})$.

Random Features

Our approach provides a random feature map

$$\phi : i \in \mathrm{V} \mapsto \boldsymbol{\phi}_i = \mathbf{\Phi}\boldsymbol{\delta}_i \in \mathbb{R}^K, \tag{3}$$

with $\mathbf{\Phi}$ a $K \times N$ feature matrix with columns $\{\boldsymbol{\phi}_i\}_{i=1}^N$, such that kernel values can be approximated by simple inner products:

$$\mathbf{\Gamma}_{i,j} \approx \tilde{\mathbf{\Gamma}}_{i,j} = \langle \boldsymbol{\phi}_i, \boldsymbol{\phi}_j \rangle. \tag{4}$$

Benefits:

- **Explicit** node embeddings (feature vectors).
- **Fast** linear kernel evaluation.
- **Reduced** memory requirements.

Our approximation $\tilde{\mathbf{\Gamma}} = \mathbf{\Phi}^\top\mathbf{\Phi}$ is rank-K. The best rank-K approximation to $\mathbf{\Gamma}$ for non-increasing h is

$$\mathbf{\Gamma}^{(K)} = \boldsymbol{V}_{:K} h(\mathbf{\Lambda}_{:K}) \boldsymbol{V}_{:K}^\top = \textstyle\sum_{k=1}^{K} h(\lambda_k) \boldsymbol{v}_k \boldsymbol{v}_k^\top. \tag{5}$$

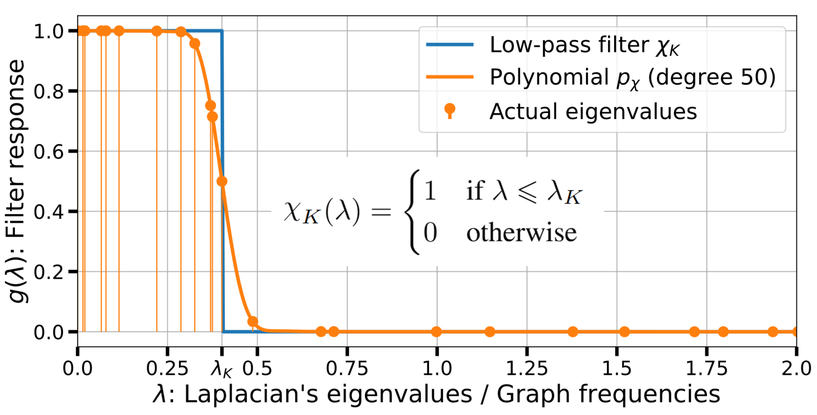


Figure 2: Perfect low-pass filter χ_K and its degree-M polynomial approximation p_χ.

Our Method, Part 1: Range Finding

Task: Approximately identify the subspace spanned by $\boldsymbol{V}_{:K}$, *i.e.*, find an orthogonal matrix $\boldsymbol{Q}$ such that

$$\mathrm{span}(\boldsymbol{Q}) \approx \mathrm{span}(\boldsymbol{V}_{:K}). \tag{6}$$

Method: Filter K Gaussian random signals $\boldsymbol{G} \in \mathbb{R}^{N\times K}$ with a low-pass filter.

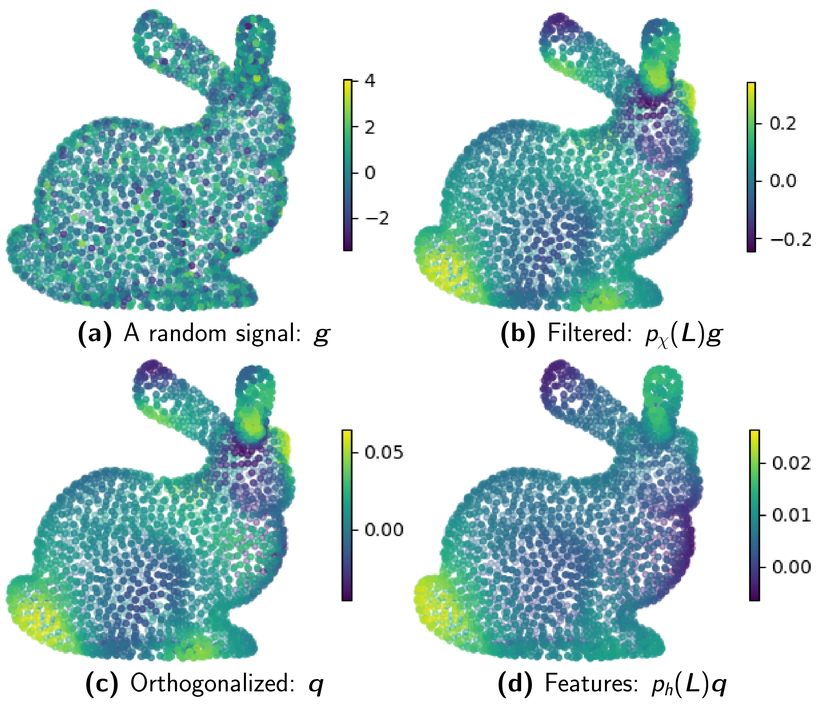
Idea: If $\chi_K := \mathbb{1}_{[0,\lambda_K]}$ is an ideal low-pass filter, then with probability 1,

$$\boldsymbol{B} = \chi_K(\boldsymbol{L})\boldsymbol{G} \implies \mathrm{span}(\boldsymbol{B}) = \mathrm{span}(\boldsymbol{V}_{:K}), \tag{7}$$

and $\boldsymbol{Q} = \mathrm{ortho}(\boldsymbol{B})$ provides an exact solution to (6).

In practice: Constructing the $\chi_K(\boldsymbol{L})$ is infeasible without access to $\boldsymbol{V}_{:K}$. We replace χ_K with a degree-M polynomial approximation p_χ (see Figure 2), and obtain

$$\boldsymbol{Q} = \mathrm{ortho}(p_\chi(\boldsymbol{L})\boldsymbol{G}). \tag{8}$$



(a) A random signal: $\boldsymbol{g}$ **(b)** Filtered: $p_\chi(\boldsymbol{L})\boldsymbol{g}$

(c) Orthogonalized: $\boldsymbol{q}$ **(d)** Features: $p_h(\boldsymbol{L})\boldsymbol{q}$

Figure 3: Example of random feature extraction.

Our Method, Part 2: Feature Construction

Task: Construct $\mathbf{\Phi} \in \mathbb{R}^{K\times N}$ such that

$$\mathbf{\Phi}^\top\mathbf{\Phi} = \tilde{\mathbf{\Gamma}} \approx \mathbf{\Gamma} \tag{9}$$

Idea: The ideal feature matrix for kernel approximation is

$$\mathbf{\Phi}^* = (h^{\frac{1}{2}}(\boldsymbol{L})\boldsymbol{V}_{:K})^\top,$$
$$\implies (\mathbf{\Phi}^*)^\top\mathbf{\Phi}^* = \textstyle\sum_{k=1}^{K} h^{\frac{1}{2}}(\lambda_k)\boldsymbol{v}_k\boldsymbol{v}_k^\top h^{\frac{1}{2}}(\lambda_k) = \mathbf{\Gamma}^{(K)}.$$

In practice: Replace $\boldsymbol{V}_{:K}$ with $\boldsymbol{Q}$ from (8), and approximate $h^{\frac{1}{2}}$ with a degree-M polynomial p_h, enabling the fast estimation of

$$\mathbf{\Phi} = (p_h(\boldsymbol{L})\boldsymbol{Q})^\top \quad \to \text{filter } \boldsymbol{Q} \text{ with } p_h. \tag{10}$$

The full feature extraction process is illustrated on Figure 3.

Error Analysis

We can decompose the overall error $\mathcal{E} := \|\mathbf{\Gamma} - \tilde{\mathbf{\Gamma}}\|$ as

$$\mathcal{E} \leqslant \underbrace{\mathcal{E}_R^2}_{\text{Range Finding Error}} + \underbrace{\mathcal{E}_P}_{\text{Polynomial Approx. Error}}. \tag{11}$$

Proposition

Let $\epsilon_{h,M} := \max_{k\in[N]} |h^{\frac{1}{2}}(\lambda_k) - p_h(\lambda_k)|$. Then,

$$\mathcal{E}_P \leqslant 2\epsilon_{h,M} + \epsilon_{h,M}^2,$$

and

$$\mathbb{E}\mathcal{E}_R \leqslant h^{\frac{1}{2}}(\lambda_{K+1}) + 2\sqrt{\tfrac{K}{r-1}}\, p_\chi(\lambda_{K+1}) + 2e\tfrac{\sqrt{K+r}}{r}\big(\textstyle\sum_{j>K} p_\chi(\lambda_j)^2\big)^{1/2}.$$

Here $r \ll K$ is an oversampling parameter such that $K+r$ is the effective embedding dimension. From the Weierstrass theorem, $\epsilon_{h,M}$ vanishes for large M, as does $p_\chi(\lambda_j)$ for $j > K$.
Notice that $h^{\frac{1}{2}}(\lambda_{K+1}) = \|\mathbf{\Gamma} - \mathbf{\Gamma}^{(K)}\|$ is the error committed by the best rank-K approximation.

Link to Paper.

Link to Code.

Results: Effect of Bandwidth

We compare our method to g-GRFs [3] on a Swiss-Roll graph ($N = 5000$), for the diffusion kernel $\mathbf{\Gamma} = \exp(-\sigma^2\boldsymbol{L})$ with varying bandwidth σ. Our approach

- achieves low approximation error for **spectrally localized** kernels.
- captures **global** geometric structure.
- is **complementary** to g-GRFs, which better captures localized behavior.

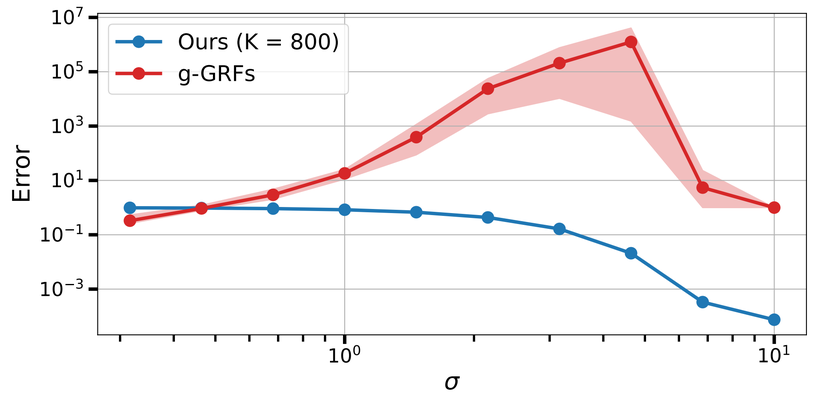


Figure 4: Average relative spectral norm of the error as a function of the bandwidth parameter σ.

Results: Effect of Target Rank

We examine the influence of the parameter K. We verify that

- the error **decreases** with K.
- the error **matches** that of $\mathbf{\Gamma}^{(K)}$.
- our algorithm effectively recovers the **dominant kernel subspace**.

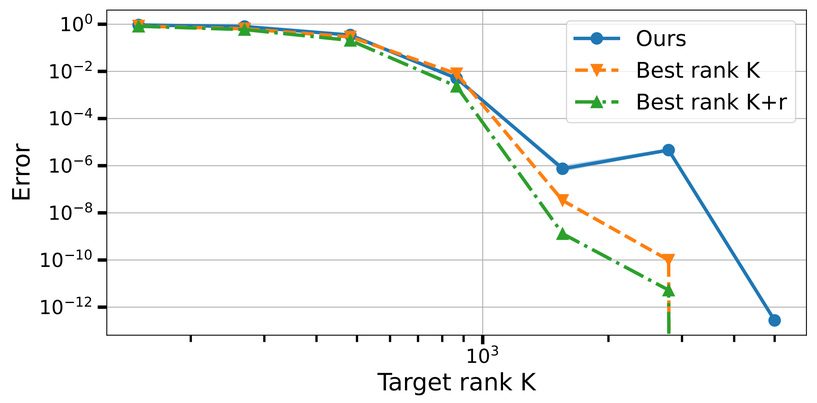


Figure 5: Average relative spectral norm of the error as a function of the target rank K, on a Swiss-Roll graph ($N = 5000$) and for the diffusion kernel ($\sigma = 5$).

Complexity Analysis

Our algorithm achieves **linear time** complexity for sparse graphs and **linear memory** complexity.

	Time	Memory
Full ED	$\mathcal{O}(N^3)$	$\mathcal{O}(N^2)$
Ours	$\mathcal{O}(EMK)$	$\mathcal{O}(NK)$

Table 1: Complexity comparison between explicit eigendecomposition and our approximation method. N: number of nodes, E: number of edges, M: order of polynomial approximations.

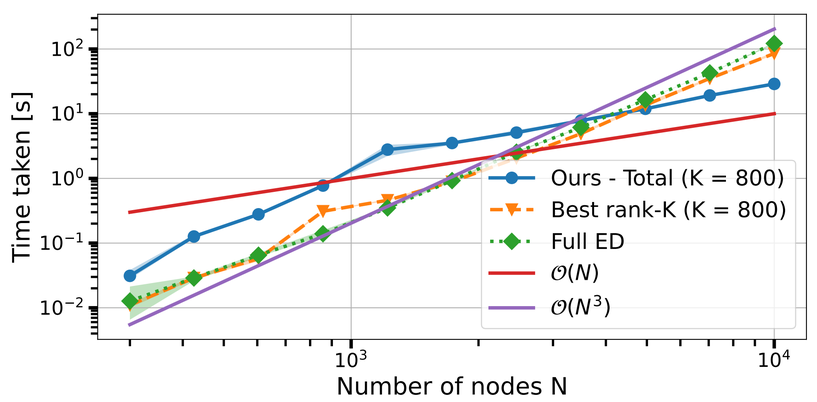


Figure 6: Average computation time as a function of the graph size N, on Swiss-Roll graphs.

References

[1] A. Ortega, P. Frossard, J. Kovačević, J. M. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," in *Proc. IEEE*, vol. 106, no. 5, pp.808-828, 2018.

[2] A. J. Smola and R. Kondor, "Kernels and regularization on graphs," in *16th annu. conf. learn. theory and 7th kernel wksp.*, 2003, pp. 144–158.

[3] I. Reid, K. Choromanski, E. Berger, and A. Weller, "General graph random features," *Int. conf. learn. represent.*, vol. 2024, pp. 28526-28543, 2024.

This work was supported by the Fonds de la Recherche Scientifique - FNRS under Grants n° 40038614 and FNRS T.0160.24. During this research, LJ benefited from an INRIA Chair at the Institute for Advanced Study, Collegium of Lyon, working in the OCKHAM team, at ENS Lyon.