Efficient Graph Field Integrators Meet Point Clouds

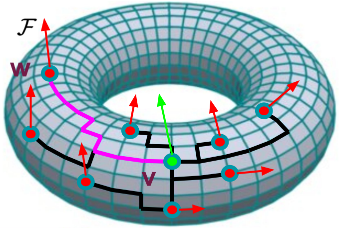
Google DeepMind

Krzysztof Choromanski*, Arijit Sehanobish*, Han Lin*, Yunfan Zhao*, Eli Berger, Tetiana Parshakova, Qingkai Pan, David Watkins, Tianyi Zhang, Valerii Likhosherstov, Somnath Basu Roy Chowdhury, Kumar Avinava Dubey, Deepali Jain, Tamas Sarlos, Snigdha Chaturvedi, Adrian Weller

Columbia University · University of Haifa · Google Research · ICML 40 Years International Conference On Machine Learning

Problem formulation

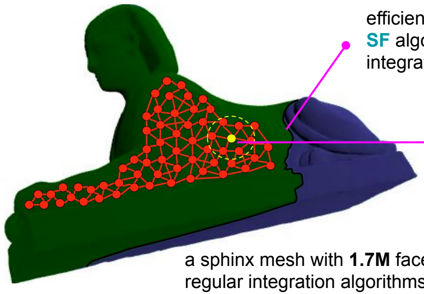
Compute efficiently (in sub-quadratic time in the number of nodes **N** of the graph) the following expressions **for every node v** of the given graph G:



$$i(v) := \sum_{w \in V} \mathrm{K}(w,v)\mathcal{F}(w)$$

- $w \in V$: integration over all the nodes
- $\mathrm{K}(w,v)$: similarity between two nodes (e.g. a function of the **shortest-path distance** between them)
- $\mathcal{F}(w)$: tensor field defined on the graph

Graph as a discretization of the 2-dim manifold:



efficient separator used in our **SF** algorithm is critical for fast integration

our **RFD** algorithm leverages graph structure implicitly via eps-neighborhood defined edges

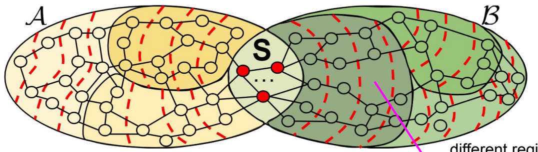
a sphinx mesh with **1.7M** faces; infeasible for regular integration algorithms

Applications: interpolation on manifolds, topological masking mechanisms for Transformers with structural inputs, physics simulations in curved spaces, Wasserstein barycenter, (Fused) Gromov Wasserstein, …

* equal contribution

SeparatorFactorization (SF)

- works with input mesh-graphs
- leverages their low-genus structure ($\rightarrow$ small-size separators)
- applies our new results in structural graph theory on fast graph field integration via separator-based divide-and-conquer methods and Fast Fourier Transform
- $O(N\log^2(N))$ time complexity, for K governed by the exp map of the shortest-path distance: $O(N\log^{1.383}(N))$



different regions encoding deviations of the shortest-path distances to the vertices of **S** from distance to **S**

RFDiffusion (RFD)

- works with point cloud (no mesh needed)
- leverages the implicit graph structure given by the following adjacency matrix: $\mathbf{W}_G(i,j) = f(\mathbf{n}_i - \mathbf{n}_j)$
- linearizes the adjacency matrix via Fourier-Transform based random feature map mechanism
- O(N) time complexity, but for a specific class of graph diffusion kernels, leveraging our novel decomposition of the exponentials of low-rank matrices:

$$\exp(\Lambda \cdot \mathbf{AB}^\top) = \sum_{i=0}^{\infty} \frac{1}{i!}(\Lambda \mathbf{AB}^\top)^i$$

$$= \mathbf{I} + \sum_{i=0}^{\infty} \frac{1}{(i+1)!}\mathbf{A}(\Lambda \mathbf{B}^\top \mathbf{A})^{i+1}\mathbf{A}^{-1}$$

$$= \mathbf{I} + \mathbf{A}[\exp(\Lambda \mathbf{B}^\top \mathbf{A}) - \mathbf{I}](\mathbf{B}^\top \mathbf{A})^{-1}\mathbf{B}^\top$$

$\mathbf{AB}^\top$: low-rank decomposition of **W** via random features

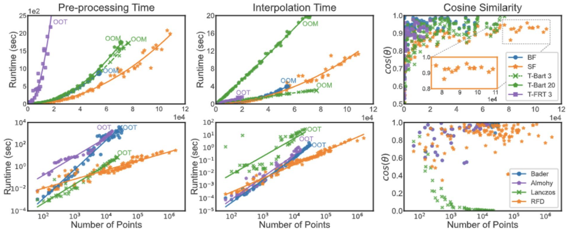
Empirical results

1. Vertex Normal Prediction

- We predict vertex normals from its masked variants.

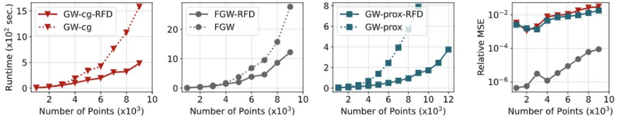
$$\mathbf{F}_i = \sum_{j \in V \backslash V'} \mathrm{K}(i,j)\mathbf{F}_j$$

- Tested on 120 meshes for 3D-printed objects with a wide range of sizes from the Thingi10k dataset.



2. Wasserstein Distances and Barycenters

- We study the OT problem of moving masses on a surface mesh.
- Gromov Wasserstein (GW) discrepancy (resp. Fused Gromov Wasserstein discrepancy (FGW)): extension of Wasserstein distances to graph-structured data.



Mesh	|V|	Total Runtime BF	Total Runtime RFD	MSE
Alien	5212	8.06	**0.39**	0.041
Duck	9862	45.36	**1.10**	0.002
Land	14738	147.64	**2.17**	0.017
Octocat	18944	302.84	**3.36**	0.027

Mesh	|V|	Total Runtime BF	Total Runtime SF	MSE
Dice	4468	6.8	**4.9**	0.063
Duck	9862	39.2	**19.4**	0.002
Land	14738	90.7	**38.9**	0.015
bubblepot2	18633	113.2	**48.3**	0.081