Computationally-efficient Graph Modeling with Refined Graph Random Features

Krzysztof Choromanski* | Avinava Dubey* | Arijit Sehanobish* | Isaac Reid

Google DeepMind | Columbia University | Google Research | University of Cambridge

ICML
International Conference On Machine Learning

Motivation

- **Graph node kernels are fundamental in ML** Essential for anomaly detection, recommender systems, and computational biology.
- **Bottleneck** Exact computation of kernel matrices is prohibitively expensive **$O(N^3)$**.
- **Standard Graph Random Features (GRFs)** approximate via sampling random walks, but face critical limitations:

Sequential & Slow
Long walks are needed for distant nodes.

Suboptimal Accuracy
Fixed Bernoulli termination is rigid.

Hardware Incompatible
Modern accelerators prefer parallelism, not long sequential Markov chains.

Goal: Achieve accurate kernel approximation with short, parallel walks.

From GRF to GRFs++

We introduce two structural modifications to the base Graph Random Features (GRFs) framework to improve computational efficiency and kernel approximation accuracy.

The Walk-Stitching Mechanism

- Replaces explicit, sequential sampling of long walks.
- Composes multiple independent, shorter walks in parallel.
- Approximates the kernel via matrix-matrix multiplications.

Generalized Walk Termination

- Moves beyond the standard Bernoulli trial.
- Employs arbitrary discrete probability distributions (e.g., Poisson) on walk lengths.
- Reduces estimation error while preserving expected walk length.

Overcoming Sequential Limits via Walk-Stitching

Standard GRFs face severe limitations when modeling connections between topologically distant nodes due to the reliance on uninterrupted random walks.

The Base GRF Limitation

The probability of emulating a walk of length $r \geq 2$ scales with the transition probability p_{next}:

$$\mathbb{P}(\text{walk of length } r) \propto p_{next}^{r-2}$$

GRFs++ (Degree): Matrix Walk-Stitching

By independently stitching pairs of shorter walks, we bypass the sequential bottleneck. The kernel approximation becomes:

$$\hat{K}_\alpha(\mathcal{W}) = \prod_{i=1}^{l} K_1^{(i)} (K_2^{(i)})^\top$$

Generalized Walk Termination Strategies

Standard GRFs: Rely on Bernoulli termination (a fixed halting probability at every node)

GRFs++ Solution: Employ any discrete probability distribution on walk lengths.

Conditions for Unbiasedness (Lemma 3.1):

Unbiasedness is theoretically preserved for any valid P provided we can:

1. Efficiently sample random walk lengths
2. Efficiently compute the survival probability

Using a Poisson termination strategy matches the average expected walk length of Bernoulli, but significantly reduces kernel estimation error.

2l-Level De-convolutions

To build valid K_1, K_2 components, the modulation function $f(p)$ must satisfy a $2l$-level discrete convolution. We provide a constructive algorithm for **any** graph kernel:

Step 1: Map graph kernel to an analytical function $g(x)$.

$$g(x) = \sum_{k=0}^{\infty} \alpha_k x^k$$

Step 2: Compute the $(2l)^{th}$-root of the function.

$$h(x) = g(x)^{\frac{1}{2l}}$$

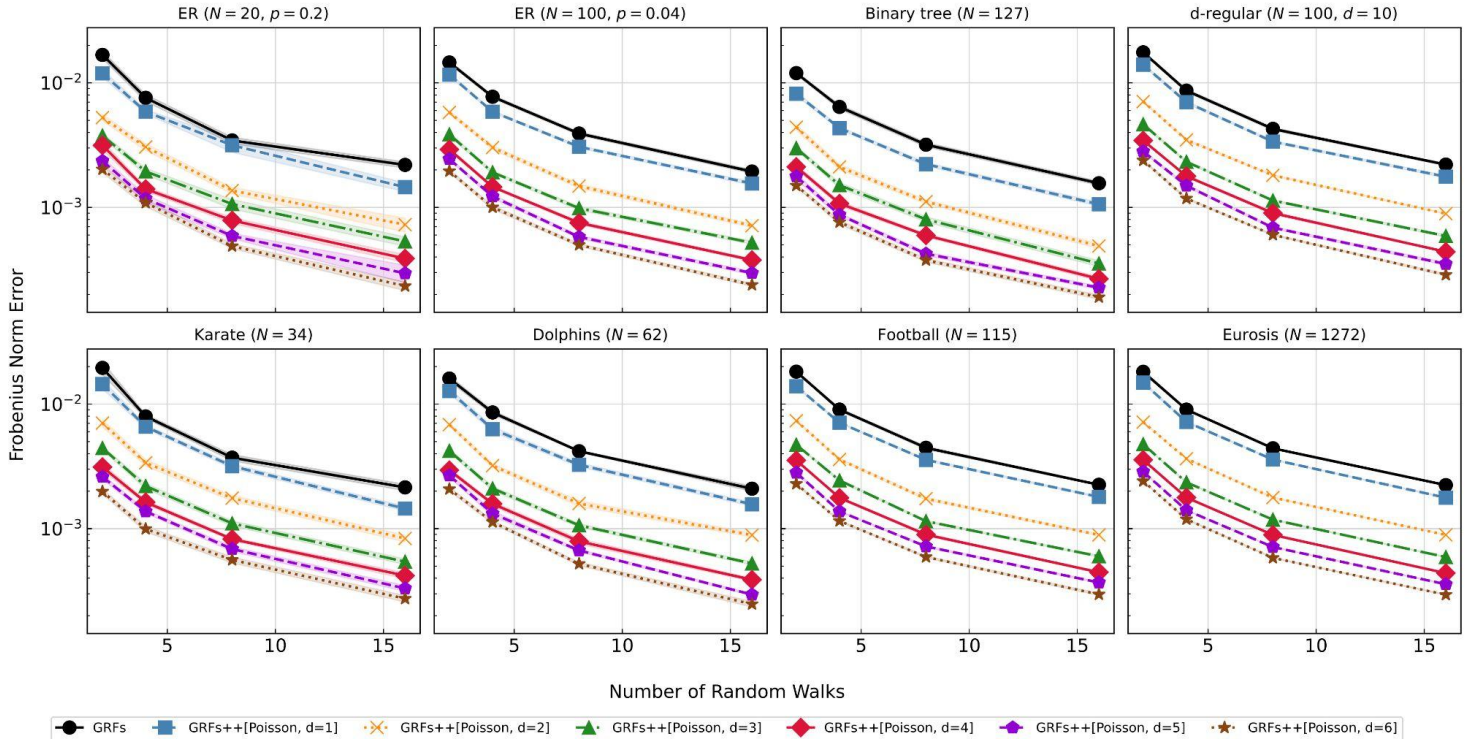
Step 3: Find the Taylor series expansion of $h(x)$ to define $f(p)$.

$$h(x) = \sum_{i=0}^{\infty} \beta_i x^i \implies f(p) = \beta_p$$

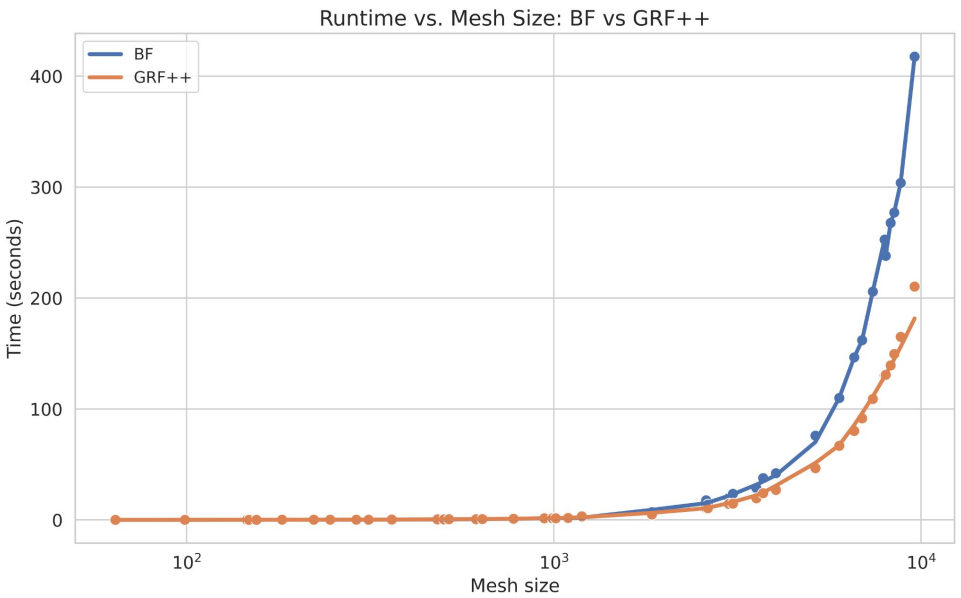
Example (Diffusion Kernel $K(W) = \exp(\lambda W)$):

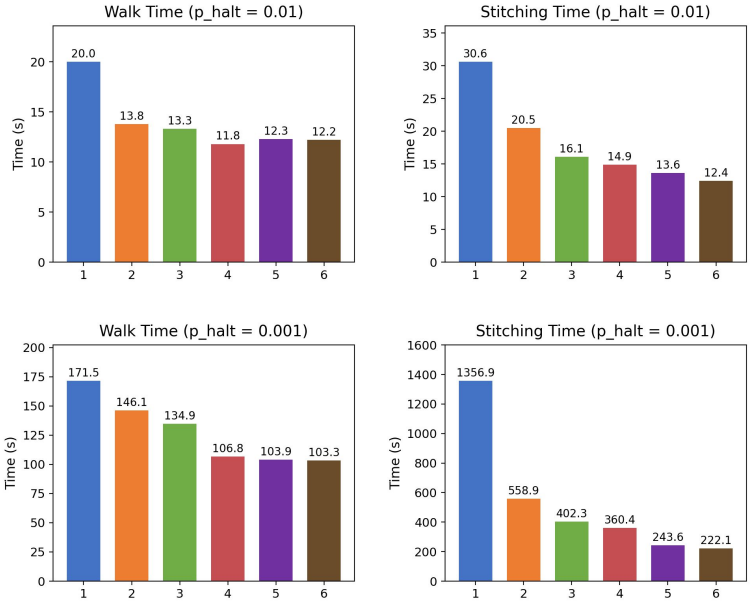
$$g(x) = \exp(x) \implies g^{1/2l}(x) = \exp(x/2l) \implies f(p) = \frac{\lambda^p}{(2l)^p p!}$$

Results: Kernel Approximation



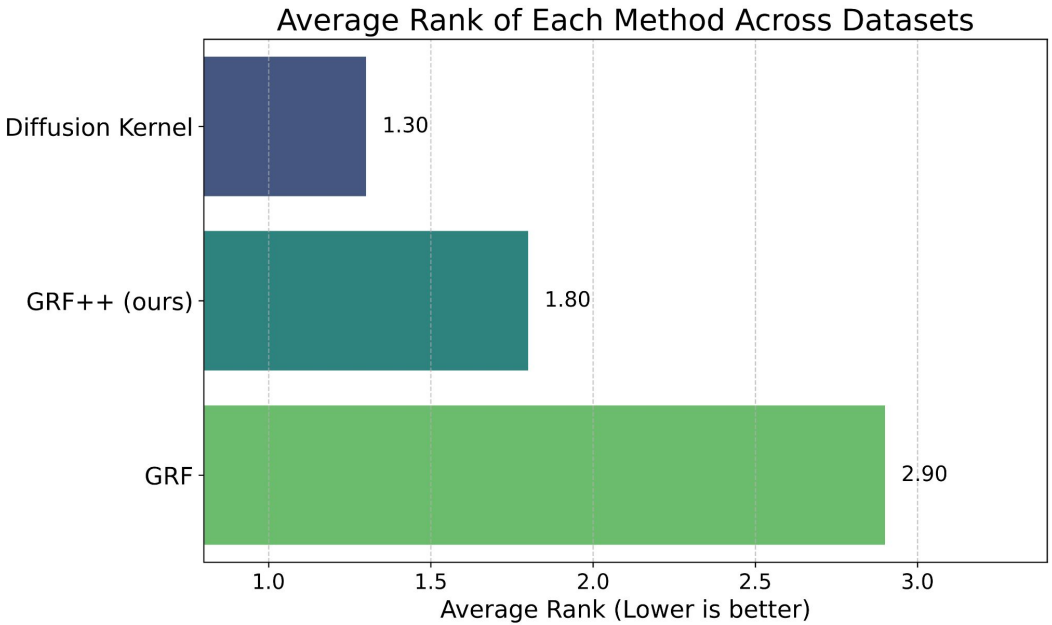
Scaling and Time Results





Left : GRF++ scales favorably compared to naive computation of the kernel matrix. **Right**: GRF++ compares favorably to GRF on both walk time as well as stitching time computations.

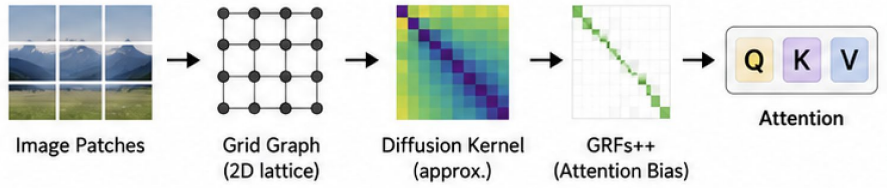
Results : Graph Classification



GRF++ outperforms GRF on 12/12 datasets and approaches the exact diffusion kernel while being more computationally efficient.

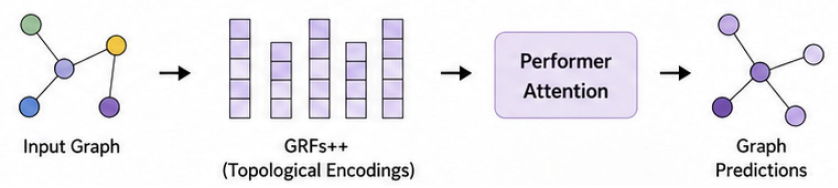
Results : GRF++ in Transformers

GRF++ in Vision Transformers (ViT)



Dataset	Baseline	GRF++	Δ
ImageNet	80.10	**80.31**	+0.21
Places365	55.78	**56.03**	+0.25

GRF++ in Graph GPS



Dataset	Baseline	GRF++	Δ
CIFAR-10	69.96	**70.21**	+0.25
PATTERN	84.91	**85.39**	+0.48
CLUSTER	75.75	**76.28**	+0.53
Peptides-func ↓	27.84	**26.61**	-1.23

Consistent gains on both Image and GNN Benchmarks

Conclusion

- Present GRF++ : unbiased estimators for certain graph kernels.
  - Faster: parallel short walks + matrix multiplication
  - More accurate by stitching more walks
  - General termination strategy

- Broadly applicable: ViTs, graph classification, graph transformers, mesh modeling, and clustering.

Thank you!

Come see us at the poster session or read our paper : https://openreview.net/pdf?id=NvJPE1oiKd