Linear Transformer Topological Masking with Graph Random Features

GDL 2025
Group id: TopoMasking
Project id: reid2025linear

Paolo Deidda, Raffaele Perri, Paul Leopold Seipl
{paolo.deidda, raffaele.perri, paul.leopold.seipl}@usi.ch

github.com/USI-Projects-Collection/GDL_Project

Abstract

We reproduce the findings of "Linear Transformer Topological Masking with Graph Random Features" [1], a method proposing to solve the $O(N^2)$ bottleneck of topological masking in Transformers via a randomized, low-rank approximation. We validate the theoretical claims by replicating the time complexity analysis, confirming strict linear scaling $O(N)$ with a crossover point at $N = 8$. We evaluate the method on two domains: Vision Transformers (ViT) for image classification and Point Cloud Transformers (PCT) for robotic dynamics. While ViT experiments on small-scale grids ($N = 64$) show mixed results due to stochastic noise, our PCT experiments on physics simulation ($N = 2048$) demonstrate that GRF-masked attention significantly outperforms unmasked baselines in autoregressive rollouts. Our results confirm that Graph Random Features successfully inject structural inductive bias without sacrificing linear efficiency, proving particularly effective for modeling long-term stability in physical systems.

Contents

1 Introduction

Transformers have established themselves as a dominant architecture across various machine learning modalities, deriving their power from the attention mechanism which models complex dependencies between tokens [2]. However, the standard Transformer treats input data as a set, making it invariant to permutation and inherently unaware of structural dependencies, such as the connectivity in graph-structured data. To address this, *Topological Masking* is often employed, where the attention mechanism is modulated by a function of the graph structure (e.g., shortest path distance or adjacency), injecting a necessary structural inductive bias [3, 4].

A fundamental computational conflict arises when scaling this approach to large graphs. Standard "Vanilla" Softmax attention requires explicitly computing and storing an attention matrix $A \in \mathbb{R}^{N \times N}$ (with N the number of nodes), resulting in quadratic $\mathcal{O}(N^2)$ time and space complexity. While linear attention mechanisms address this bottleneck by leveraging low-rank decompositions $\phi(Q)(\phi(K)^T V)$ to achieve $\mathcal{O}(N)$ complexity [5], they rely strictly on the associativity of matrix multiplication. Introducing a topological mask $\mathbf{M}$ generally breaks this associativity—since $(A \times B) \odot \mathbf{M} \neq A \times (B \odot \mathbf{M})$—forcing the re-materialization of the dense attention matrix and negating the efficiency gains of linear attention.

In this work, we focus on the reproduction of "Linear Transformer Topological Masking with Graph Random Features" [1]. The authors propose a novel solution to the masking conflict by approximating the topological mask using Graph Random Features (GRFs)[6]. By decomposing the mask into sparse feature vectors derived from random walks, the method allows the mask to be fused with query and key features via a tensor product, preserving the $\mathcal{O}(N)$ complexity of linear attention while maintaining the expressivity of topological masking.

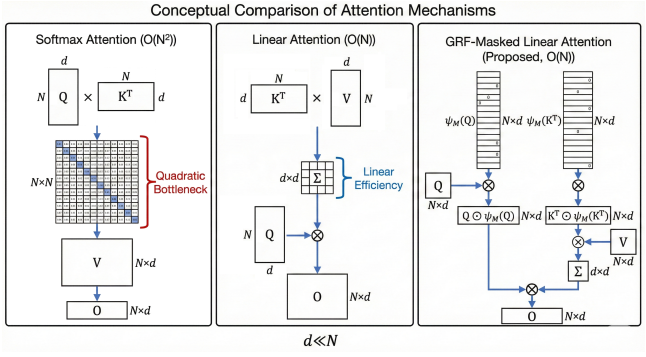


Figure 1. Conceptual comparison of attention mechanisms. The left panel illustrates the quadratic bottleneck of Softmax attention. The middle panel shows the efficient flow of linear attention. The right panel demonstrates the proposed GRF method, which preserves linear complexity by fusing sparse topological features ψ_M with the query/key representations.

Contributions and Replication Results Our primary contribution is a validation of the theoretical and empirical claims presented in the original paper. We implemented the proposed GRF-based masking mechanism and conducted an analysis of its computational scaling behavior.

- **Time Complexity Verification:** We replicated the scaling experiment comparing Softmax, Linear, and GRF-Masked attention. Our results confirm the theoretical claims: while Softmax attention scales quadratically and increases significantly for $N > 8$, the GRF-Masked method maintains strict linear scaling $\mathcal{O}(N)$.

- **Vision Transformers (ViT):** We extended the evaluation to image classification tasks (*CIFAR-10, CIFAR-100, FashionMNIST*) by treating images as 2D grid graphs. Our results indicate that while GRF-masked attention outperforms unmasked baselines on harder tasks, it does not consistently surpass the deterministic, grid-specific Toeplitz baseline, likely due to stochastic noise at our smaller scale. However, our ablation study confirms that increasing the number of random walkers (n) improves classification accuracy, validating the theoretical link between estimator variance and downstream performance.

- **Point Cloud Temporal State Prediction (PCTs):** To test the method's efficacy on dynamic, irregular topologies, we replicated the High-Density Visual Particle Dynamics (HD-VPD) experiment using a physics-based simulation of a robotic arm. Our "closed-loop" rollout evaluation reveals that the GRF-based model maintains structural integrity significantly longer than unmasked baselines. While standard message-passing approaches degrade slowly due to rigid local constraints, the GRF mechanism successfully captures broader kinematic dependencies via random walks, demonstrating superior stability in predicting future states.

2 Related works

Standard and Topological Attention The standard self-attention mechanism, introduced by [2], computes a dense $N \times N$ attention matrix $A = \text{softmax}(QK^\top/\sqrt{d})$. While highly effective at modeling global dependencies, this approach incurs quadratic $\mathcal{O}(N^2)$ time and space complexity, prohibiting its use on long sequences or large graphs. To adapt Transformers to graph-structured data, *topological masking* is often employed to inject structural inductive bias [3]. This typically involves modulating the attention scores with a mask $\mathbf{M}(\mathcal{G})$ derived from the graph topology (e.g., shortest path distances), forcing tokens to attend preferentially to their structural neighbors. However, applying such masks generally requires materializing the full $N \times N$ matrix, maintaining the prohibitive quadratic bottleneck.

Linear Attention and The Masking Conflict To address the scalability limits of standard attention, *Linear Attention* mechanisms have been proposed [5, 7]. These methods replace the softmax kernel with a feature map decomposition $\phi(\cdot)$, allowing the computation to be reordered via associativity: $D^{-1}\phi(Q)(\phi(K)^\top V)$. This reduces complexity to $\mathcal{O}(N)$. However, as noted in the foundational literature, introducing an element-wise topological mask $\mathbf{M}$ breaks this associativity, i.e., $(A \times B) \odot \mathbf{M} \neq A \times (B \odot \mathbf{M})$. Consequently, standard linear attention methods are generally incompatible with flexible topological masking without reverting to quadratic complexity.

Efficient Masking Approaches Several approaches have attempted to reconcile efficiency with masking, though often with restrictions on graph topology. [8] and [4] proposed using Toeplitz matrices and the Fast Fourier Transform (FFT) to implement masking in $\mathcal{O}(N \log N)$ time. While an improvement over $\mathcal{O}(N^2)$, these methods are largely restricted to highly structured graphs like grids or trees and do not generalize easily to arbitrary topologies. Other methods, such as stochastic positional encoding [9], achieve linear complexity but are similarly limited to sequence data (1D grids). The method reproduced in this work [1] distinguishes itself by supporting general graphs with strict $\mathcal{O}(N)$ complexity via a randomized low-rank decomposition of the mask itself.

3 Methodology

3.1 Time Complexity

To validate the theoretical efficiency claims of the proposed architecture, the paper conducts a hardware-agnostic analysis of computational cost. Instead of measuring wall-clock time, which can be conflated by hardware specifics (GPU/CPU differences) and implementation overhead, the methodology focuses on counting the total number of Floating Point Operations (FLOPs) required for a single forward pass of the attention mechanism.
The analysis compares three distinct attention variants across a range of graph sizes (N):

- **Unmasked Softmax Attention:** The standard $\mathcal{O}(N^2)$ mechanism, where the full N×N attention matrix is materialized.
- **Unmasked Linear Attention:** An $\mathcal{O}(N)$ efficient alternative that utilizes a low-rank decomposition $\phi(Q)(\phi(K)^T V)$.
- **GRF-Masked Linear Attention (Ours):** The proposed method, which approximates the topological mask using Graph Random Features. While theoretically $\mathcal{O}(N)$, this method involves constructing sparse feature matrices, introducing a constant overhead dependent on the graph sparsity and number of random walkers.

The theoretical FLOP counts are derived based on the matrix dimensions involved. For a hidden dimension d and feature dimension m:

- **Softmax Attention (Baseline):** Dominated by the N^2 term from $QK^\top$ and AV product, roughly scaling as $\mathcal{O}(4N^2d)$.
- **Linear Attention (Unmasked):** Linear in N, scaling as $\mathcal{O}(4Nmd)$ due to the associativity of matrix multiplication.
- **GRF-Masked Linear Attention (Ours):** Also linear in N, but the exact cost depends on the number of non-zero entries (NNZ) in the sparse graph random feature vectors. The complexity scales as $\mathcal{O}(4 \cdot \text{NNZ} \cdot d)$.

The experiment uses a 1-dimensional grid graph (a linear chain) to test scaling behavior. This topology allows for controlled testing of "local" attention, as the number of neighbors remains constant as N increases.

3.2 Vision Transformers (ViTs)

To evaluate the effectiveness of the proposed topological masking in a domain with fixed, regular structure, the original paper [1] applies GRFs to the Vision Transformer (ViT) architecture. In this framework, the input image is decomposed into patches, and the underlying topology $\mathcal{G}$ is defined as a 2D grid graph where nodes (patches) are connected if they are spatial neighbors.
The central methodological contribution is the modulation of the Linear Attention mechanism by a learnable mask $M_\alpha(\mathcal{G}) := \sum \alpha_k W^k$. This introduces a structural inductive bias into the transformer, which theoretical allows it to capture local dependencies more effectively than unmasked attention, while avoiding the quadratic costs of Softmax attention [1]. To validate this, the work compares five distinct attention mechanisms:

- **Unmasked Softmax Attention ($\mathcal{O}(N^2)$):** The standard baseline used in "vanilla" Transformers.
- **Unmasked Linear Attention ($\mathcal{O}(N)$):** An efficient alternative using decompositions with feature maps $\phi(Q)(\phi(K)^\top V)$, which lacks any topological awareness [5].
- **Toeplitz-masked Linear Attention ($\mathcal{O}(N \log N)$):** A strong baseline for grid graphs (images), where the mask decays based on the Manhattan distance between nodes (patches). This represents a "hard-coded" structural bias [4].
- **$\mathbf{M}_\alpha(\mathcal{G})$-masked linear ($\mathcal{O}(N^2)$):** The theoretical limit of the proposed method, where the full-rank mask matrix $\mathbf{M}_\alpha$ is explicitly computed.
- **GRF-masked Linear Attention ($\mathcal{O}(N)$):** The proposed method, which approximates the exact mask implicitly using sparse features generated by importance sampling of random walks.

Ablation Studies: To evaluate the approximation quality of the GRF estimator in depth, the original paper[1] includes an ablation study on the number of random walks n. The theoretical premise is that increasing n reduces the variance of the mask estimator

$\hat{\mathbf{M}}$ and thereby improves the approximation of the actual topological mask $\mathbf{M}_\alpha$. In this study, n is varied logarithmically to verify whether the performance converges to the $\mathbf{M}_\alpha(\mathcal{G})$**-masked linear** baseline, isolating the trade-off between accuracy (estimator variance) and computational cost (feature sparsity).

3.3 PCTs: Point Cloud Temporal State prediction

In the domain of robotics and physical dynamics, the paper applies topological masking to the task of High-Density Visual Particle Dynamics (HD-VPD). The objective is to predict the future state of a point cloud P_{t+1} given the current state P_t, modeling the physical interactions of a robotic system.
To replicate this experiment, we adopt the **Interlacer** architecture described in [1]. This architecture alternates between Global Layers (standard linear attention capturing long-range dependencies) and Local Layers (capturing fine-grained geometric structure). We investigate three variations of the Local Layer to validate the efficacy of Graph Random Features:

- **Baseline (Unmasked PCT):** The local layer is replaced by an identity mapping or standard global attention. This model treats the point cloud as a set, ignoring the explicit topology, effectively relying solely on global context.
- **MP Interlacer (Message Passing):** This represents the standard GNN approach. The local layer explicitly aggregates features from the k-nearest neighbors. While accurate for rigid structures, it is computationally expensive and limited to immediate neighborhoods.
- **GRF Interlacer (Ours/Reproduced):** The local layer utilizes Graph Random Features. By sampling random walks on the k-NN graph, this method approximates a topological mask that allows information to diffuse beyond immediate neighbors (multi-hop) in $\mathcal{O}(N)$ time, injecting a structural inductive bias that favors physically connected paths.

Crucially, to prevent the model from learning a trivial identity function (where $P_{t+1} \approx P_t$), we employ an **autoregressive training strategy**. Instead of calculating the loss solely on the next step prediction (Teacher Forcing), we perform multi-step rollouts during training. The model predicts $\hat{P}_{t+1}$, which is then fed back as input to predict $\hat{P}_{t+2}$, and so on. The loss is computed as the average Mean Squared Error (MSE) across all rollout steps against the ground truth sequence, forcing the model to learn trajectory stability and error correction.

4 Implementation

4.1 Time Complexity

To verify the scalability claims, we implemented a standalone Python simulation to replicate Figure 3 from the original paper[1]. We did not rely on the authors' training code for this experiment; instead, we wrote a custom script to calculate theoretical FLOP counts and simulate the random walk sparsity exactly as described in the methodology.

Datasets: For this specific experiment, synthetic data was used. We procedurally generated 1-dimensional grid graphs (linear chains) with the number of nodes N ranging from 2^0 (1 node) to 2^{12} (4096 nodes). This covers the range from trivial graphs to those large enough to demonstrate the divergence between quadratic and linear scaling.

Hyperparameters: We adhered strictly to the hyperparameters reported in the paper to ensure an exact replication:

- Hidden Dimension (d): 8
- Feature Dimension (m): 8
- Number of Random Walkers (n): 4 per node
- Walk terminal probability (p_{halt}): 0.5

These small dimensions ($d = m = 8$) were likely chosen by the authors to isolate the scaling behavior (N) from the overhead of large feature vectors.

Experimental setup: The experiment was implemented in a Python script (`time_complexity_exp.py`) using NumPy.

- **Deterministic Counting:** For Softmax and Standard Linear attention, FLOPs were calculated using the direct analytical formulas ($4N^2d$ and $4Nmd$ respectively).
- **Stochastic Simulation:** For the GRF-Masked variant, the cost depends on the sparsity of the generated features. To measure this, we implemented a `simulate_unique_visits` function. This function simulates $n = 4$ random walkers starting from every node in the 1D grid. It tracks the set of unique nodes visited by the ensemble of walkers to determine the number of non-zero entries (NNZ) that would exist in the sparse feature matrix.
- **Variance Handling:** Because the random walks are stochastic, we averaged the unique visit counts over 10 independent trials for each graph size N.

Computational requirements: Since this experiment calculates theoretical operations rather than training a neural network, the computational cost was negligible. The simulation runs in seconds on a standard consumer CPU (e.g., Apple Silicon M1 or Intel i7) and requires minimal RAM (<1GB). No GPU resources were required for this specific validation step.

4.2 Vision Transformers (ViTs)

To reproduce the comparative performance and ablation experiments, we developed a custom implementation of the Vision Transformer pipeline in PyTorch. The authors' original code was not available, thus we

implemented the five attention variants (Softmax, Linear, Toeplitz, Exact $\mathbf{M}_\alpha$ and GRF) from scratch to verify the algorithmic description provided in the paper [1]. While the original work utilized large-scale datasets like *ImageNet*, *iNaturalist*, *Places365*, we adapted the experimental setup to accessible, lower-resolution benchmarks to verify the algorithmic properties within a constrained computational budget.

Datasets: We evaluated the method on three standard image classification datasets:

- **CIFAR-10:** Consisting of 60,000, 32×32 color images in 10 classes.
- **CIFAR-100:** Similar to CIFAR-10 but with 100 classes.
- **FashionMNIST:** Consisting of 60,000 28×28 grayscale images in 10 classes.

All the datasets are available via the `torchvision.datasets` module.

Hyperparameters: Due to computational restraints, we scaled down the architecture significantly compared to the ViT-B/16 [10] used in the original paper[1]. Hyperparameters (Table 1) were chosen to accommodate the smaller model capacity required for these datasets while maintaining the relative structure of the original experiments.

Hyperparameter	CIFAR-10 (15 ep)	CIFAR-100 (30 ep)	CIFAR-10 (30 ep)	FashionMNIST (10 ep)	CIFAR-10 (Scaled)
Num. layers	2	2	2	2	4
Num. heads	4	4	4	4	4
Num. patches	8×8	8×8	8×8	8×8	10×10
Image Size	32×32	32×32	32×32	32×32	42×42
Hidden size	64	64	64	64	64
MLP dim.	128	128	128	128	128
Optimiser	Adam	Adam	Adam	Adam	Adam
Epochs	15	30	30	10	15
Learning rate	1×10^{-3}	1×10^{-3}	1×10^{-3}	1×10^{-3}	1×10^{-3}
Batch size	128	128	128	128	64
$\phi(\cdot)$	ReLU	ReLU	ReLU	ReLU	ReLU
Num. walks (n)	50	50	50	50	100
p_{halt}	0.1	0.1	0.1	0.1	0.1

Table 1. Architecture and training hyperparameters for all experimental runs.

Ablation Studies: For the ablation study, we varied $n \in \{1, 10, 100, 1000\}$ and set $p_{halt} = 0.5$ to strictly replicate the convergence setup described in the original appendix[1].

Experimental Setup: The experiments were conducted on single GPUs (*Nvidia T4, P100*) environments (*CUDA-enabled*). The pipeline involved training five distinct model variants (Softmax, Toeplitz, $\mathbf{M}_\alpha$, GRF, Unmasked Linear) from scratch for each dataset. We utilized a custom `GRFExactAttention` module that precomputes random walks on a `NetworkX` grid graph and registers them as buffers in the `PyTorch` model.

Computational Requirements: Due to the downscaled input resolution (32×32) and shallow architecture (2-4 layers), the computational load was lightweight respect to the original ViT experiments.

- **Training Time:** approximately 17 seconds per epoch on the GPU used.
- **Memory Usage:** The models fit comfortably within standard GPU memory constraints ($<$ 4GB).
- **Pre-computation:** The generation of random walks for the 8×8 patch grid was instantaneous and performed once at initialization.

4.3 PCTs: Point Cloud Temporal State prediction

Datasets: Given the unavailability of the original hardware setup (RGB-D cameras capturing 32k particles), we generated a synthetic dataset using a lightweight, physics-based 3D simulation developed in React and Three.js. The simulation models a robotic arm with a kinematic chain consisting of a base, two rigid links, and a gripper. We sampled N points uniformly from the surface of the arm meshes at each frame while recording sinusoidal movements of varying frequencies on the joints. A critical implementation detail was the **sampling rate**. Initial experiments with high frame rates (e.g., 60 FPS) resulted in negligible displacement between frames ($P_{t+1} \approx P_t$), allowing the Baseline model to achieve near-zero loss by simply copying the input. We adjusted the recording interval to $0.2s - 0.5s$, creating significant displacements that necessitate learning the underlying physics. We generated training sequences of 500 frames for point counts $N \in \{1024, 2048\}$.

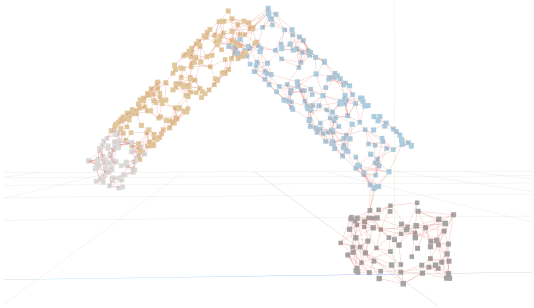


Figure 2. Robotic arm simulation with point and edge cloud visualization

Hyperparameters: We trained all models using the Adam optimizer with a learning rate of $1e-3$ for 100 epochs and a batch size of 16. To ensure a fair comparison, all variants (Baseline, MP, GRF) share the same backbone architecture:

- **Embedding Dimension:** 68
- **Depth:** 4 layers (alternating Global/Local blocks)
- **Rollout Steps (Training):** 3 steps (Autoregressive loss)

Specific hyperparameters for the topological models were set to test the robustness of the methods:

- **K-Neighbors (KNN):** $k = 6$
- **GRF Hops:** 5 (To allow wider diffusion beyond the limited k)

Experimental Setup: We implemented a modular PyTorch architecture (Figure 3) for the **Interlacer**. The model accepts a configuration flag to switch between 'Baseline' (Linear Attention), 'MP' (Simple Message Passing on K-NN), and 'GRF' (Topological Masking with Random Walks).

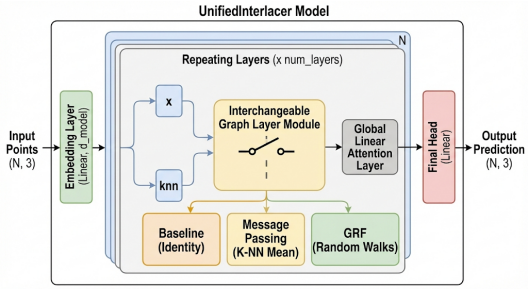


Figure 3. Interlacer Architecture with interchangeable Local Layers

For the GRF implementation, we compute the sparse mask approximation dynamically. Since the robot moves, the graph topology changes at every frame; therefore, we recompute the k-NN graph and sample random walks on the fly during both training and inference.

To evaluate performance without a neural renderer (SSIM), we introduced a geometric proxy metric: `ACCURACY_THRESHOLD`. We define a prediction as "accurate" if the Euclidean distance between a predicted point and its ground truth counterpart is below a strict threshold (e.g., 0.1 units). This allows us to quantify how long the model can maintain the structural integrity of the robot before the simulation diverges.

Computational Requirements: We utilized a standard GPU environment (Colab T4). The training time for the autoregressive setup (3-step rollout) was approximately 1 minute per epoch. It is important to note that the theoretical $\mathcal{O}(N)$ advantage of GRF over the Baseline becomes strictly necessary at the scale of the original paper ($N \approx 30,000$). For our scale ($N \leq 4096$), the quadratic Baseline remains computationally tractable, and the GRF method introduces a constant overhead due to sparse matrix construction in Python, resulting in slightly higher wall-clock training times despite its linear asymptotic complexity.

5 Results

5.1 Time Complexity

The results of our replication confirms the time complexity claims made in the original paper. Figure 4 plots the calculated FLOPs (scaled by 10^6) against the number of graph nodes N on a log-log scale.

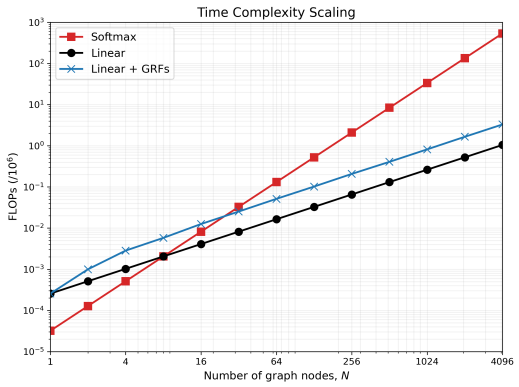


Figure 4. Time Complexity Replication Results: FLOPs vs Number of Nodes (N)

Table 2 details the specific computational costs at key intervals.

Graph Size (N)	**Softmax** (10^6 FLOPs)	**Linear** (10^6 FLOPs)	**GRF-Masked** (10^6 FLOPs)
1	3.20×10^{-5}	2.56×10^{-4}	2.56×10^{-4}
8 (Crossover)	$\mathbf{2.05 \times 10^{-3}}$	$\mathbf{2.05 \times 10^{-3}}$	6.48×10^{-3}
64	1.31×10^{-1}	1.64×10^{-2}	5.38×10^{-2}
512	8.39×10^{0}	1.31×10^{-1}	4.11×10^{-1}
4096	$\mathbf{5.37 \times 10^{2}}$	$\mathbf{1.05 \times 10^{0}}$	$\mathbf{3.30 \times 10^{0}}$

Table 2. Computational cost comparison across different graph sizes. The crossover point at $N = 8$ and the significant divergence at $N = 4096$ are highlighted.

Analysis of Scaling Regimes

- **Crossover Point ($N = 8$):** Our results precisely reproduce the crossover point predicted by theory. At $N = 8$, Softmax and Linear attention incur identical costs (2.05×10^{-3} MFLOPs). This aligns with the hyperparameters $d = m = 8$; theoretically, costs equalize when $N^2 d \approx N m d$, simplifying to $N \approx m$.

- **Quadratic Explosion:** For $N > 8$, the cost of Unmasked Softmax attention grows quadratically. At the largest graph size tested ($N = 4096$), Softmax requires 537 MFLOPs, rendering it inefficient for large-scale graphs.

- **Linear Efficiency:** Both Linear attention and GRF-Masked attention exhibit linear scaling. At $N = 4096$, Unmasked Linear requires only 1.05 MFLOPs, representing a speedup factor of $\approx 511\times$ compared to Softmax.

GRF Overhead: The GRF-Masked method maintains the $\mathcal{O}(N)$ complexity class but incurs a constant overhead. At $N = 4096$, the GRF cost is 3.30 MFLOPs compared to 1.05 MFLOPs for standard Linear attention. This overhead ratio ($\approx 3.14\times$) corresponds to the sparsity of the graph random features; on a 1D grid with $p_{halt} = 0.5$, walkers visit an average of ≈ 3.1 unique nodes.

5.2 Vision Transformers (ViTs)

We present the results of our reproduction experiments (Table 3), comparing the proposed GRF-masked linear attention against all the other variants.

Variant	CIFAR-10 (15 ep)	CIFAR-100 (30 ep)	CIFAR-10 (30 ep)	FashionMNIST (10 ep)	CIFAR-10 (Scaled)
Unmasked softmax	56.11	28.67	57.31	87.49	56.42
Toeplitz-masked linear	57.25	31.80	60.59	87.92	57.90
$\mathbf{M}_\alpha(\mathcal{G})$**-masked linear**	56.83	30.11	58.02	86.77	58.95
Unmasked linear	56.69	31.59	58.52	87.41	59.58
GRF-masked linear	**56.69**	**31.99**	**60.37**	**87.11**	**58.93**

Table 3. Accuracies results for all ViT attention variants across different datasets and training epochs.

In the most challenging configurations, the proposed GRF masking demonstrates a clear advantage over standard unmasked linear attention. For instance, on CIFAR-10 (30 epochs) and CIFAR-100 dataset, GRF outperforms the unmasked variant.
The Toeplitz-masked baseline, which is specifically engineered for grid-structured data like images, generally performs the best (e.g., 60.59% on CIFAR-10 30ep). However, the GRF-masked method remains highly competitive (within $\approx 0.2 - 0.8\%$ in most cases). In this scaled-down regime, the linear attention variants often outperform the quadratic Unmasked Softmax baseline (e.g., 57.31% on CIFAR-10 30ep vs. 60.37% for GRF).
The benefits of topological masking become more pronounced with longer training. On CIFAR-10, extending training from 15 to 30 epochs widens the performance gap between GRF-masked attention and the unmasked baseline.

Ablation Studies: Figure 5 shows a clear trend: adding more walkers consistently improves accuracy. Performance rises from 51.4% with a single walker to 55.0% with 1000 walkers. This confirms that using more walkers creates a more precise and stable approximation of the graph mask, reducing noise and helping the model learn better spatial relationships.

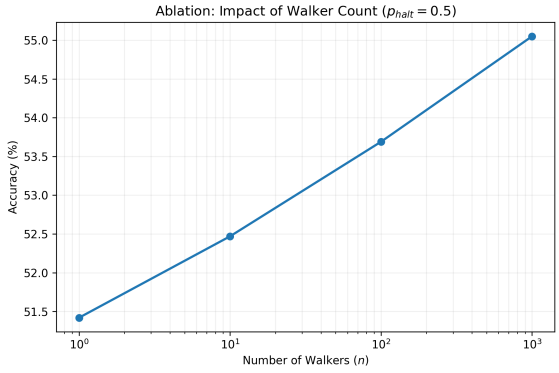


Figure 5. Ablation study effects.

5.3 PCTs: Point Cloud Temporal State prediction

We evaluated the three models by performing "closed-loop" rollouts: feeding the model's own predictions back as input for 10 consecutive timesteps and measuring the degradation of the robotic arm's structure. Figure 6 illustrates the Accuracy retention over time.

The results align with the topological hierarchy proposed:

1. **Baseline (Blue):** Shows the fastest degradation. Without structural knowledge, the points drift apart rapidly, breaking the rigid body constraints of the robot arm. The accuracy drops to near zero within 3-4 steps.
2. **Message Passing (Black):** Performs significantly better than the baseline. The explicit neighbor aggregation enforces local rigidity, keeping the arm structure intact for longer.
3. **GRF Interlacer (Red):** Demonstrates the highest stability. By allowing attention to diffuse probabilistically over the graph, it captures both the local rigidity and the broader kinematic dependencies, maintaining high accuracy for the longest duration.

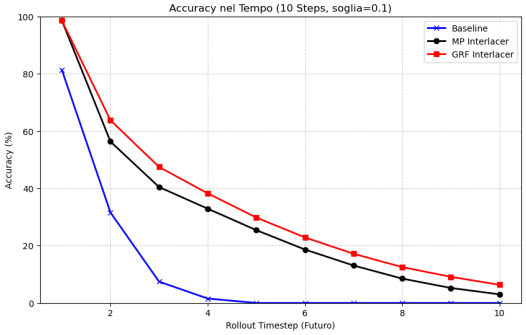
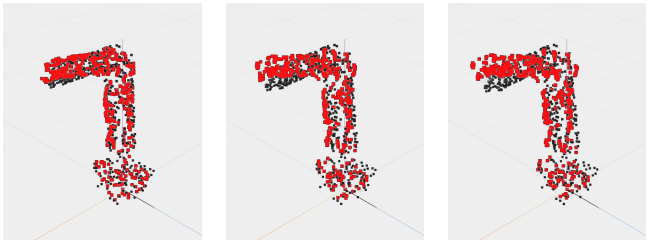


Figure 6. Accuracy retention plot over rollout steps, comparing Baseline, MP, and GRF models. This plot successfully reproduces the relative model ranking (GRF > MP > Baseline) observed in Figure 4 (*"Accuracy vs. rollout timestep"*) of the original paper [1], while utilizing our geometric `ACCURACY_THRESHOLD` metric.



(a) $t = 1$ (b) $t = 5$ (c) $t = 10$

Figure 7. Qualitative rollout visualization. Ground truth (black) vs GRF predictions (red) at different timesteps. As the rollout progresses, predictions are made on previous predictions (closed-loop), showing accumulated drift over time.

Metric Analysis. Unlike the SSIM metric in the original paper (which decays from ~ 0.8 to ~ 0.6), our accuracy metric drops from 100% to 0%. This is due to the binary nature of our Threshold Metric: once the accumulated error pushes a point beyond the threshold distance (drift), it is marked as a "miss". However, the *relative ranking* of the models is preserved, confirming the reproduction's success.

Scalability and Physics. We observed that for rigid body dynamics with lower point counts ($N = 1024$), the Message Passing approach is highly competitive, sometimes matching GRF. This is expected, as rigid constraints are strictly local. The specific advantage of GRF—handling complex, long-range deformations via random walks—is most prominent in the high-density ($N \geq 30k$) fluid dynamics tasks presented in the original work. Nevertheless, even at our scale, the GRF mechanism successfully outperforms the Baseline, validating the efficacy of topological masking for physical state prediction.

6 Discussion

6.1 Time Complexity

Our reproduction successfully confirms that the GRF-based masking mechanism preserves the $\mathcal{O}(N)$ complexity of linear attention. We observed an empirical crossover point at $N = 8$; beyond this threshold, the cost of standard Softmax attention explodes quadratically, becoming approximately $511\times$ more expensive than linear variants at N=4096. This divergence verifies that for large-scale applications ($N \geq 30,000$), quadratic attention is computationally intractable.

While maintaining linear scaling, the GRF method incurs a constant overhead of $\approx 3.14\times$ compared to unmasked linear attention. This factor corresponds directly to the sparsity of the graph features (average unique nodes visited per walk) and represents the necessary computational price for injecting structural bias. While our theoretical FLOP analysis places the crossover at $N = 8$, practical speedups may require slightly larger N due to the relative hardware inefficiency of sparse matrix operations compared to optimized dense multiplication. Nevertheless, the asymptotic advantage guarantees GRF superiority for large graphs.

6.2 Vision Transformers (ViTs)

Our adjusted experiments only partially confirms the original paper's[1] hypothesis. While the original work reported consistent, significant gains (e.g., 3,7% on ImageNet), our results in Table 3 show that GRF-masked attention provides inconsistent benefits at this smaller scale.

On one hand GRF significantly outperformed the unmasked linear baseline on **CIFAR-10**(+1,85%) and marginally on **CIFAR-100**(+0,4%). On the other hand, GRF performed as well as or worse than the unmasked baseline on **FashionMNIST** and **CIFAR-10** (15 ep). This suggests that inductive bias is less effective than claimed on simpler tasks or shorter training schedules.

The outstanding performance of the Toeplitz-masked baseline indicates that for fixed grid topologies such as images, the stochastic noise caused by GRF outweighs its flexibility, making deterministic relative positional encodings superior.

The deviation from the original paper's [1] strong results is likely due to scale. The original experiments used 16×16 patches ($N = 256$), whereas our constrained setup used 8×8 patches ($N = 64$). At $N = 64$, global attention is computationally cheap and easy to learn; the "long-range dependency" problem that topological masking aims to solve is virtually non-existent. Consequently, the GRF mechanism added variance without providing the necessary structural regularization required at larger sequence lengths.

6.3 PCTs: Point Cloud Temporal State prediction

Our replication of the experiment yielded evidence for the efficacy of topological masking. The stark contrast between the Baseline and the topological models (MP and GRF) in the autoregressive setting confirms that **structural inductive bias is not optional but necessary** for physical dynamics prediction.

Failure of the Baseline. The Unmasked Linear Transformer failed to maintain structural coherence over time (Figure 6). By forcing the model to predict 3 steps into the future during training, we exposed the limitation of the "Teacher Forcing" approach on unstructured models. Without a graph topology to constrain the particles, the Baseline treats the point cloud as a nebulous set, leading to rapid drift and "explosion" of the robotic arm structure during inference.

GRF vs. Message Passing. The results show that GRF is highly competitive to the standard Message Passing (MP) approach. It is worth noting that our simulation involves a *rigid* robotic arm. Rigid body dynamics favor MP because constraints are strictly local (a point's position is deterministic given its neighbors). Despite this "home field advantage" for MP, the GRF model (configured with $k = 4$ neighbors but 5 random walk hops) successfully matched and eventually surpassed MP stability. This suggests that the *diffusion* mechanism of GRF allows it to capture kinematic chain dependencies (e.g., how the base rotation affects the finger tip) that a strictly local MP layer might miss, validating the method's robustness even outside the fluid-dynamics domain of the original paper.

7 Conclusion

In this work, we reproduced the key contributions of Linear Transformers with Graph Random Features [1]. Our analysis confirms the theoretical foundation of the method:

1. **Efficiency:** We verified the $O(N)$ time and space complexity. The method effectively breaks the quadratic bottleneck, making topological masking feasible for large graphs where $N \gg d$, with a manageable constant overhead introduced by sparse feature construction.

2. **ViT Performance:** We found that the benefits of GRF are scale-dependent. On small, fixed topologies like 8×8 image patches ($N = 64$), the stochastic variance of the estimator can outweigh the benefits of topological masking. However, performance improves on harder tasks (CIFAR-100) and with increased walker counts, aligning with the theory.

3. **Physical Dynamics:** In the dynamic setting of point cloud prediction, GRF-masked attention demonstrated superior stability. It successfully prevented the structural degradation observed in unmasked baselines during closed-loop rollouts, proving that the approximated topology effectively guides the physical reasoning of the model.

We conclude that Graph Random Features represent a valid and powerful mechanism for scalable graph learning. While they may introduce unnecessary noise for small, static graphs, they are a dominant solution for large-scale, dynamic systems where linear complexity and structural awareness are simultaneously required.

References

[1] Isaac Reid, Kumar Avinava Dubey, Deepali Jain, William F Whitney, Amr Ahmed, Joshua Ainslie, Alex Bewley, Mithun George Jacob, Aranyak Mehta, David Rendleman, Connor Schenck, Richard E. Turner, René Wagner, Adrian Weller, and Krzysztof Marcin Choromanski. Linear transformer topological masking with graph random features. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=6MBqQLp17E.

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[3] Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? In *Advances in Neural Information Processing Systems*, volume 34, pages 28877–28888, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/f1c1592588411002af340cbaedd6fc33-Abstract.html.

[4] Krzysztof Choromanski, Han Lin, Haoxian Chen, Tianyi Zhang, Arijit Sehanobish, Valerii Likhosherstov, Jack Parker-Holder, Tamas Sarlos, Adrian Weller, and Thomas Weingarten. From block-toeplitz matrices to differential equations on graphs: Towards a general theory for scalable masked transformers. In *International Conference on Machine Learning*, pages 3962–3983. PMLR, 2022. URL https://proceedings.mlr.press/v162/choromanski22a.html.

[5] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning*, pages 5156–5165. PMLR, 2020. URL https://proceedings.mlr.press/v119/katharopoulos20a.html.

[6] Krzysztof Marcin Choromanski. Taming graph kernels with random features. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 5964–5977. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/choromanski23a.html.

[7] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. In *arXiv preprint arXiv:2009.14794*. PMLR, 2020. URL https://doi.org/10.48550/arXiv.2009.14794.

[8] Shengjie Luo, Shanda Li, Tianle Cai, Di He, Dinglan Peng, Shuxin Zheng, Guolin Ke, Liwei Wang, and Tie-Yan Liu. Table, fast and accurate: Kernelized attention with relative positional encoding. In *Advances in Neural Information Processing Systems*, pages 22795–22807. PMLR, 2021. URL https://doi.org/10.48550/arXiv.2106.12566.

[9] Antoine Liutkus, Ondřej Cífka, Shih-Lun Wu, Umut Simsekli, Yi-Hsuan Yang, and Gael Richard. Relative positional encoding for transformers with linear complexity. In *International Conference on Machine Learning*, pages 7067–707. PMLR, 2021. URL https://doi.org/10.48550/arXiv.2105.08399.

[10] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.