Neurips | Sydney, Australia

RelFlexformer: Efficient Attention 3D Transformers for Integrable Relative Positional Encodings

Seoul National University, Kensho Technologies, Google Research, Google DeepMind, Columbia University

Byeongchan Kim*, Arijit Sehanobish*, Avinava Dubey*, Min-hwan Oh, Krzysztof Choromanski*

https://relflexformer.github.io/

*equal contribution

Transformers in 3D

Attention-based architectures now dominate 3D learning:

- Point Cloud (PC) Transformer *(pc learning)*
- Point Transformers *(point cloud learning)*
- DFormer *(RGB-D images)*

Why Transformers work for 3D:

- Capture long-range dependencies
- Handle irregular, unordered point sets
- Flexible to different input sizes

They suffer from the following drawbacks though.

Why Standard Attention Fails at Scale

$$\text{Attention}(Q, K, V) = \text{softmax}(Q K^T) \cdot V$$

$Q K^T$ is a $N \times N$ matrix.

Memory : $O(N^2)$

Attention Matrix

- N = 1k → 1 million ✓
- N = 16k → 268 million ✗
- N = 100k → 10 billion ✗ ✗
- N = 1 million → 1 trillion ✗ ✗ ✗

⚠ Real 3D scenes: **100K–1M** points

quadratic attention ≠ scalable 3D Transformers

Attention is Geometry-Unaware

Standard attention is permutation-invariant:

♻️ shuffling the token order doesn't change the output.

For 3D point clouds, this does not take geometry into account:

- Nearby points should attend more strongly to each other
- Spatial locality is a powerful geometric prior

We need attention that respects 3D geometry.

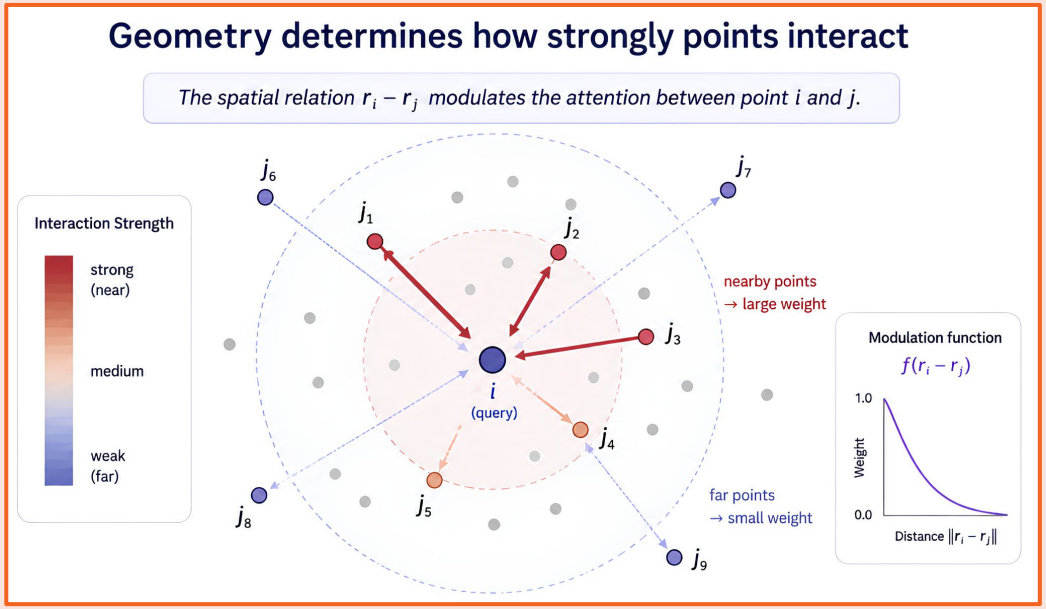
Relative Position Encodings (RPE)

Standard attention : $q_i k_j^T$

Geometry Aware Attention :

$$(q_i k_j^T) \cdot \underbrace{f(r_i - r_j)}$$

geometry-based modulation



Some desirable properties of RPE

- **Exponential decay** ⟶ strong locality
- **Learned function *f*** ⟶ data-driven geometry
- **Translation/rotational invariance** ⟶ 3D-symmetries

Efficiency vs Geometry

① Efficiency

Sub-quadratic attention
(handle large 3D objects)

→ **Performers: O(N)**

② Geometry

Relative position encodings
(spatial inductive bias)

→ **RPE: needs N × N matrix**

These seem incompatible. Can we have BOTH ?

Background: Linear Attention (Performers)

Linear Attention via Kernel Approximation [Choromanski et al. 2021]:

$$\text{softmax}(qk^T) \approx \varphi(q) \cdot \varphi(k)^T$$

linearization of the softmax kernel

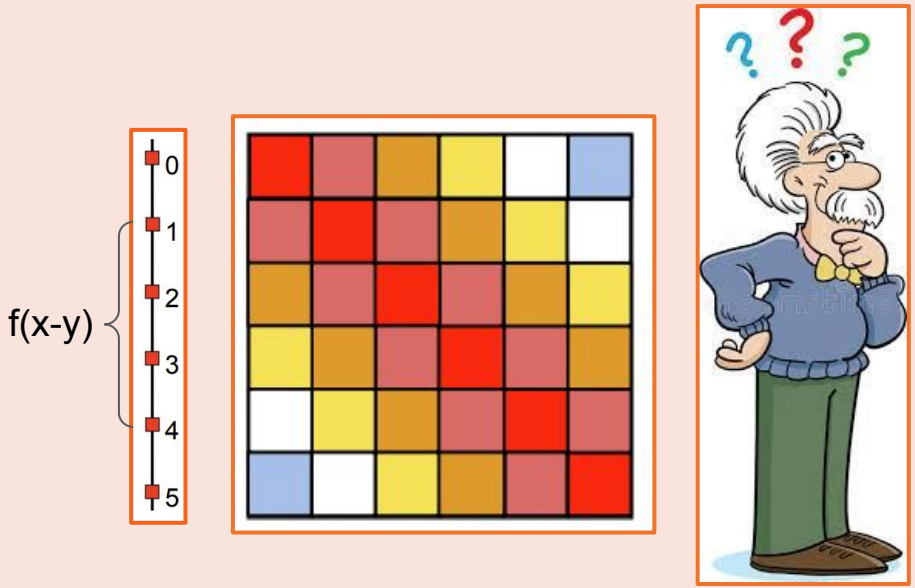
$$\text{Att}(Q, K, V) = \varphi(Q) \cdot \underbrace{(\varphi(K)^T V)}_{\text{Compute first !}}$$

- φ is a nonlinear (randomized) map (ReLU, exp-kernel, etc.)
- Decouples the queries and keys *before* the matrix multiplication.
- Can be constructed to provide **unbiased approximation** of the original softmax kernel.

- Complexity reduced from $O(N^2)$ to $O(N)$!
- Memory : No N × N matrix

*By factoring the matrix this way, the model treats tokens as a bag-of-words—**completely ignoring spatial/geometric structure.***

How to incorporate general masking into scalable Transformers ?

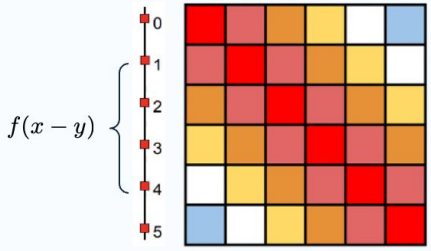


Why Do We Need Fast Matrix-Vector Multiplication?

1 Flexible 3D RPE Mask

RPE defines a geometric mask based on relative positions.

$$M_{ij} = f(r_i - r_j)$$



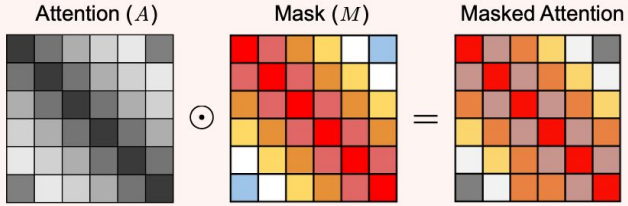
2 Standard RPE Attention

Apply the mask to attention:

$$A_{RPE} = A \odot M$$

Requires explicit $M \in \mathbb{R}^{L \times L}$

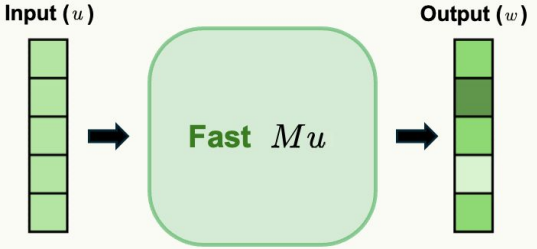
$\Rightarrow O(L^2)$



3 Performer-Compatible RPE

Performer-style attention never materializes the full matrix.
With masking, all we need is

$$w = Mu$$



RelFlexFormer = Performer + FastMult

Standard Performer : O(N)

$$Att = \varphi(Q) \cdot (\varphi(K)^T V)$$

RelFlexFormer: O(N log N)

Geometry-aware via FastMult

Replace each column multiply with FastMult

→ RPE mask applied without forming N×N matrix

Key properties:

- Drop-in replacement for any Performer-based backbone
- Works with PCT, PTv3, DFormer — no architecture changes
- Any L^1-integrable modulation function f
- Small Quadrature size S works in practice
- Scales to **64k+** tokens (Transformer OOMs at **16K**)

Flexible Choice of Spatial Geometry

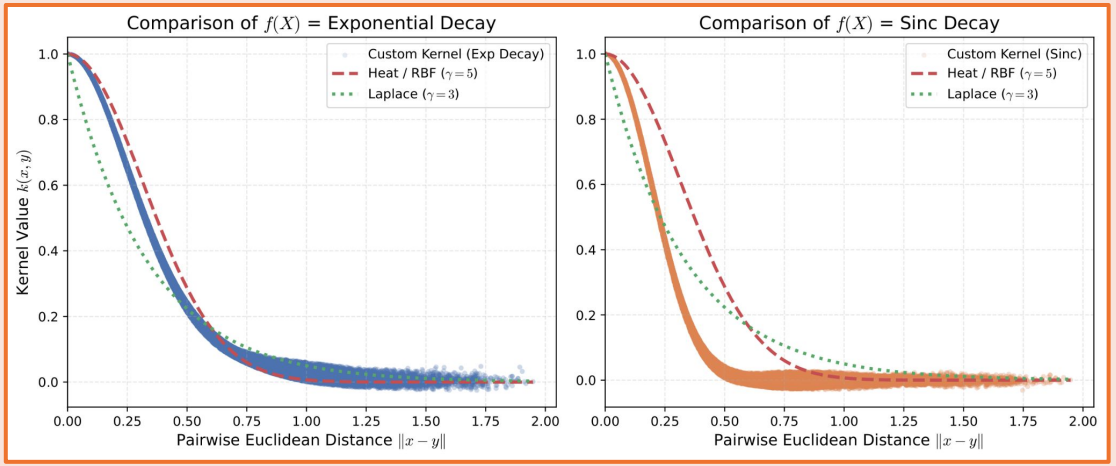
Any L^1-integrable function works as the modulation:

- Exponential decay: $f(x) = \exp(-\lambda\|x\|)$

  ⟶ Locality bias, nearby points upweighted

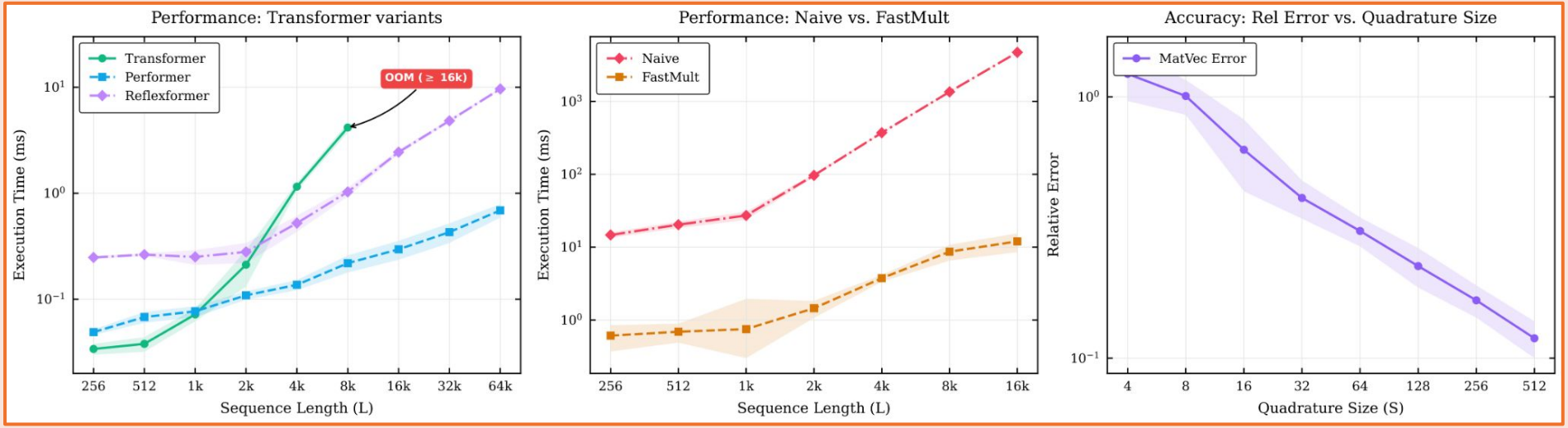
- Sinc function: $f(x) = \mathrm{sinc}(\lambda\|x\|)$

  ⟶ Band-limited attention



λ controls the spatial scale. Can be tuned per dataset or made learnable.

Scalability & Efficiency



- Transformer : OOMs at 16k tokens
- RelFlexFormer handles 64k+ tokens

- FastMult vs naive: orders of magnitude faster at large N

- Quadrature error decays as $O(1/\sqrt{S})$
- S = 8 sufficient in practice

A Unifying Framework

RelFlexFormer subsumes prior efficient RPE methods

RoPE

[Su et al. 2021]

$\xi_{\square} \in \{\pm\theta/2\pi\}$, fixed quadrature basis

Analytically recovers exact RoPE via fixed quadrature.

Block-Toeplitz RPE

[Choromanski et al. 2022]

Grid → FFT

Structured token layouts

When points lie on a grid, the NU-FFT mechanism reduces directly to the standard FFT.

Results: Point Cloud Classification

PCT backbone/ Overall Accuracy (OA) for ModelNet40 and ScanObjectNN

Attention	Complexity	ModelNet40	ScanObjectNN
Transformer (baseline)	$O(N^2)$	93.2	84.0
Performer	O(N)	92.34	83.16
+ PointRoPE	O(N)	92.48	83.88
RelFlexformer (ours)	**O(N logN)**	**92.94**	**84.45**

Beats the quadratic Transformer on ScanObjectNN **(+0.45) while being sub-quadratic!**

Results: 3D Semantic Segmentation

PTv3 backbone/ mIoU

Attention	Complexity	ScanNet	ScanNet200	ScanNet++	nuScenes	S3DIS (6-fold)
Transformer (baseline)	$O(N^2)$	77.6	35.3	48.2	80.4	77.7
Performer	O(N)	74.8	28.2	48.1	72.0	73.2
RelFlexformer (ours)	**O(N logN)**	**76.8**	**34.0**	**48.7**	**80.3**	**76.3**

Consistently closes the gap. Exceeds Transformer on **ScanNet++**.

RelFlexformer + PointRoPE on nuScenes: 81.2 (best overall).

Results: RGB-D Semantic Segmentation

DFormer backbone/ mIOU

Attention	Complexity	NYU v2	SUN RGB-D
Transformer (baseline)	$O(N^2)$	55.6	51.2
Performer	O(N)	54.44	48.49
+ STRING	O(N logN)	54.96	50.87
RelFlexformer (ours)	**O(N logN)**	**55.32**	**51.04**

Summary

RelFlexFormer: geometry-aware efficient attention for 3D

① O(L log L) RPE via Non-Uniform FFT

Any integrable f · any 3D positions · no grid assumption

② Unified across 3D domains

Point clouds • LiDAR • RGB-D • classification • segmentation

③ Closes the linear attention gap

Matches or exceeds softmax accuracy at sub-quadratic cost

Thank You