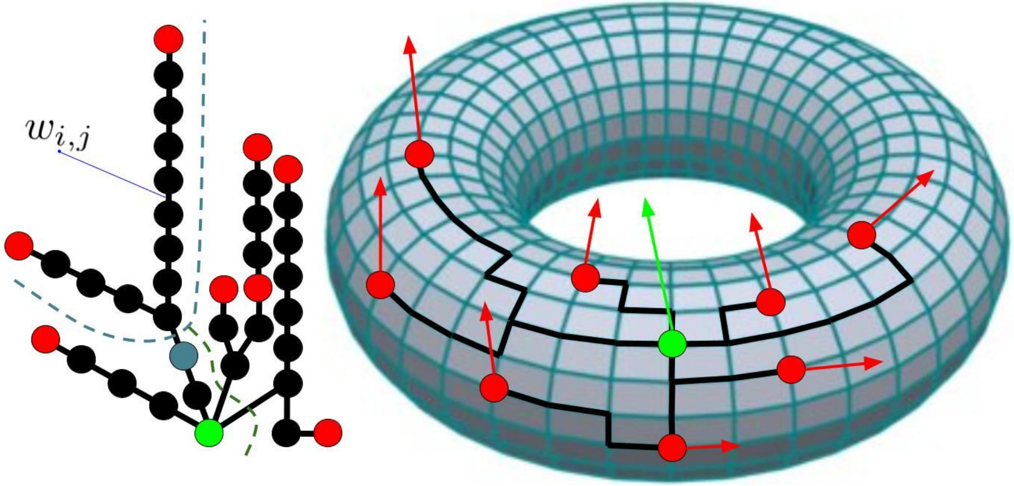
Fast Tree-Field Integrators

From Low-Displacement Rank to Topological Transformers

Krzysztof Choromanski, Arijit Sehanobish*, Somnath Basu Roy Chowdhury*, Han Lin*, Kumar Avinava Dubey*, Tamas Sarlos, Snigdha Chaturvedi*



Google DeepMind

University of North Carolina Chapel Hill

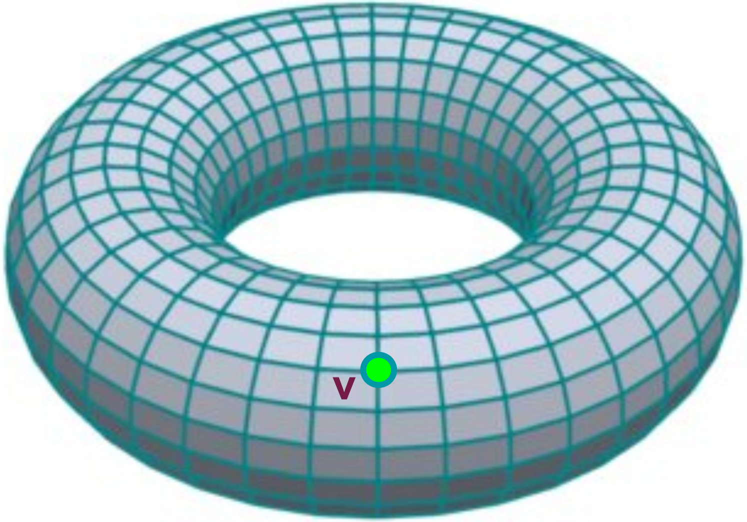
Google Research

* equal contribution

Problem Formulation: Efficient Graph Field Integration

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

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

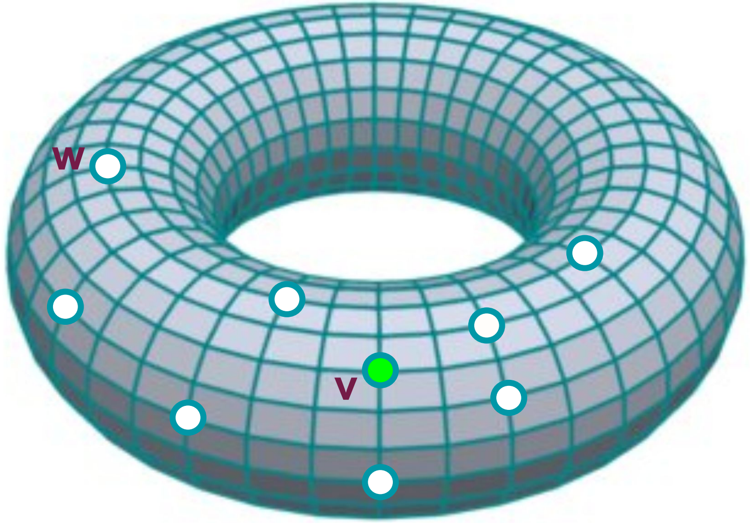


Problem Formulation: Efficient Graph Field Integration

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

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

integration over all the nodes



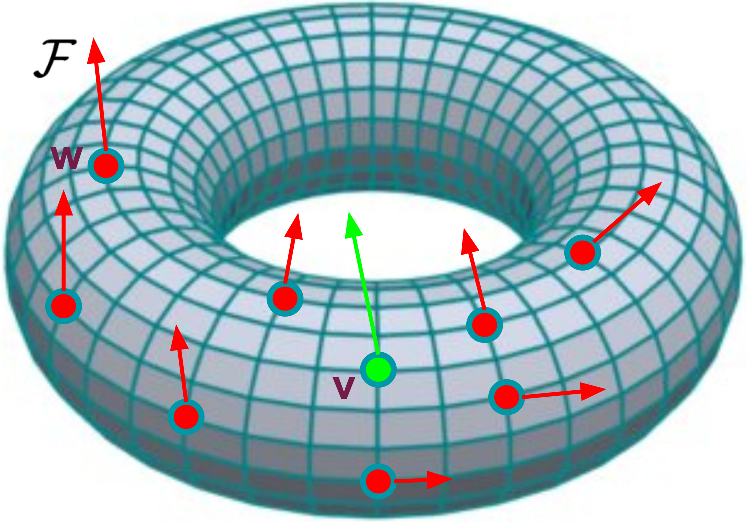
Problem Formulation: Efficient Graph Field Integration

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

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

tensor field defined on the graph

integration over all the nodes



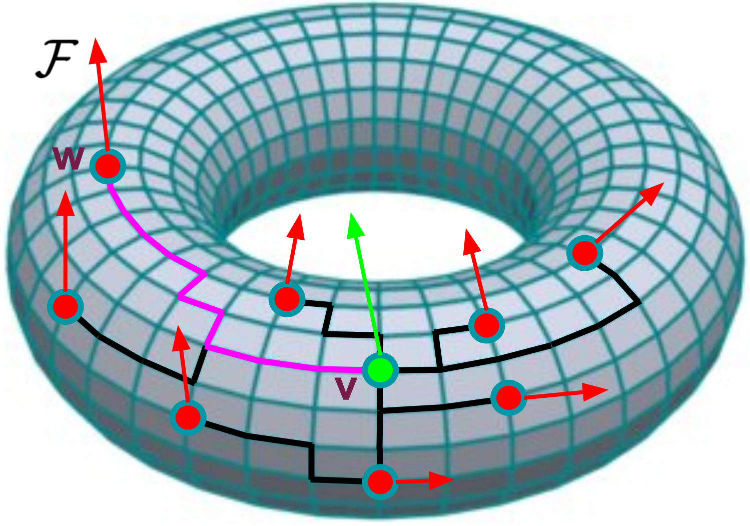
Problem Formulation: Efficient Graph Field Integration

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

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

integration over all the nodes

similarity between two nodes (e.g. a function of the **shortest-path distance** between them)



Problem Formulation: Efficient Graph Field Integration

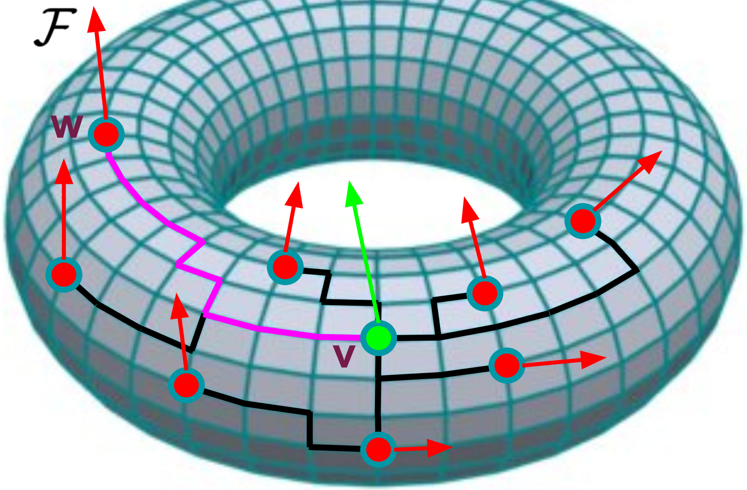
Compute efficiently (in the sub-quadratic time in the number of nodes **N** of the graph) the following expressions **for every node v** of the given graph G

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

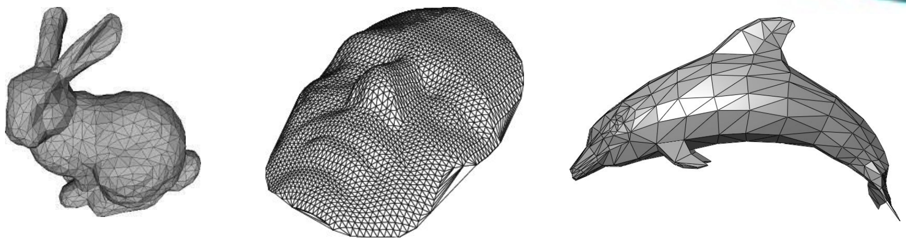
tensor field defined on the graph

integration over all the nodes

similarity between two nodes (e.g. a function of the **shortest-path distance** between them)



Graph as a discretization of the 2-dim manifold:



Problem Formulation: Efficient Graph Field Integration

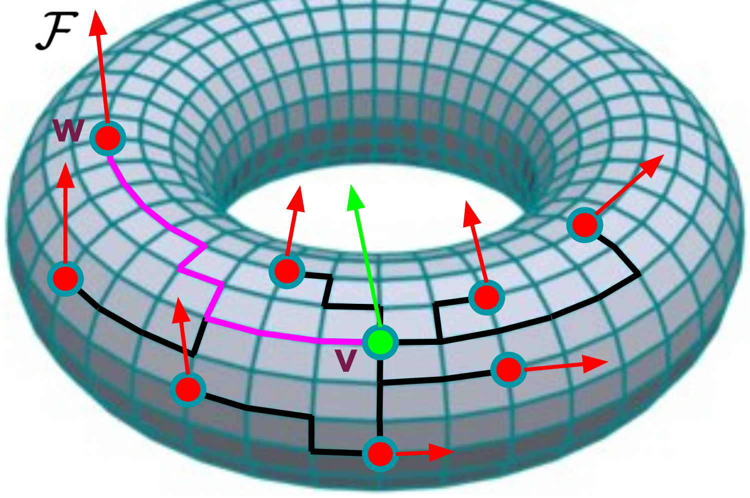
Compute efficiently (in the sub-quadratic time in the number of nodes **N** of the graph) the following expressions **for every node v** of the given graph G

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

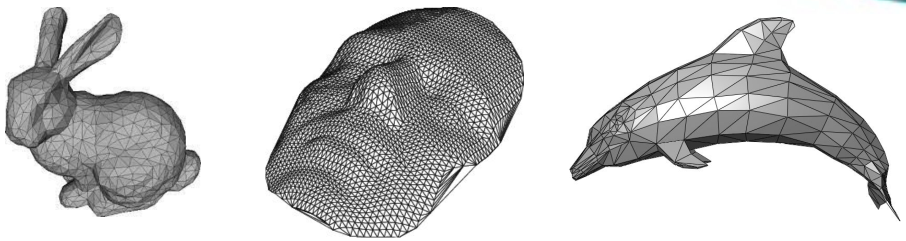
tensor field defined on the graph

integration over all the nodes

similarity between two nodes (e.g. a function of the **shortest-path distance** between them)



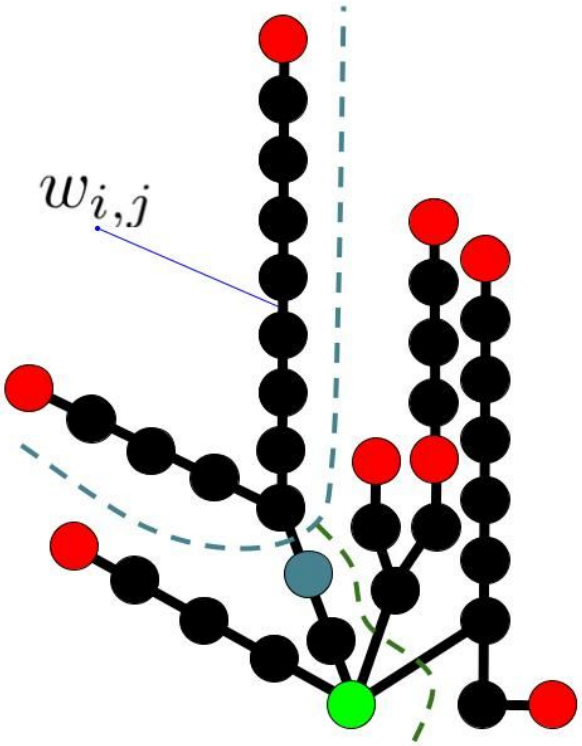
Graph as a discretization of the 2-dim manifold:



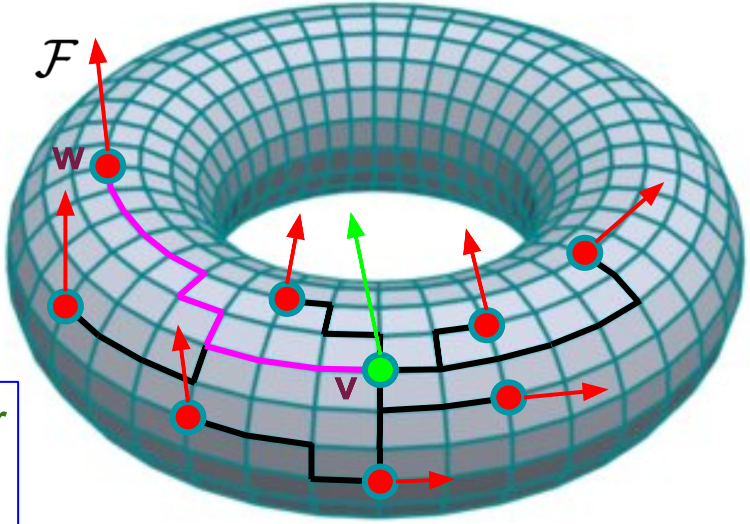
Applications: interpolation on manifolds, topological masking mechanisms for Transformers with structural inputs, physics simulations in curved spaces, Wasserstein barycenter, (Fused) Gromov Wasserstein, …

Integration on the Low-Distortion Trees for $\mathrm{K}(w,v) = f_\theta(d_\mathrm{G}(w,v))$

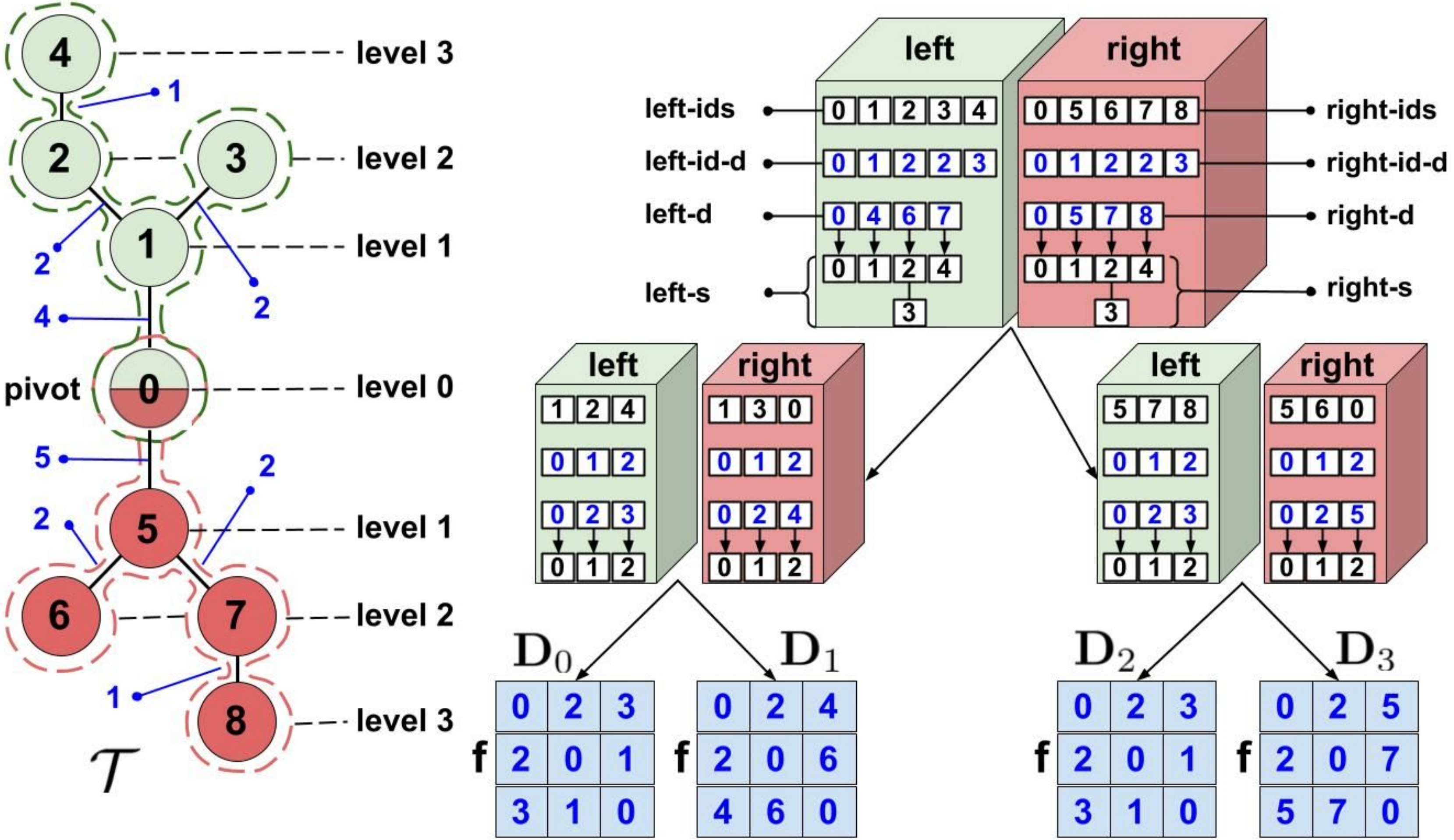
(f-Integration)



- weighted trees approximating original graph metric
- *minimum spanning tree* (MST) in several applications
- integration of **quadratic** time complexity, not an option for large graphs
- we propose **polylog-linear** algorithms working for several classes of f_θ
- based on the divide-and-conquer strategy and FFT



The Algorithm (Fast Tree-Field Integrator: FTFI)

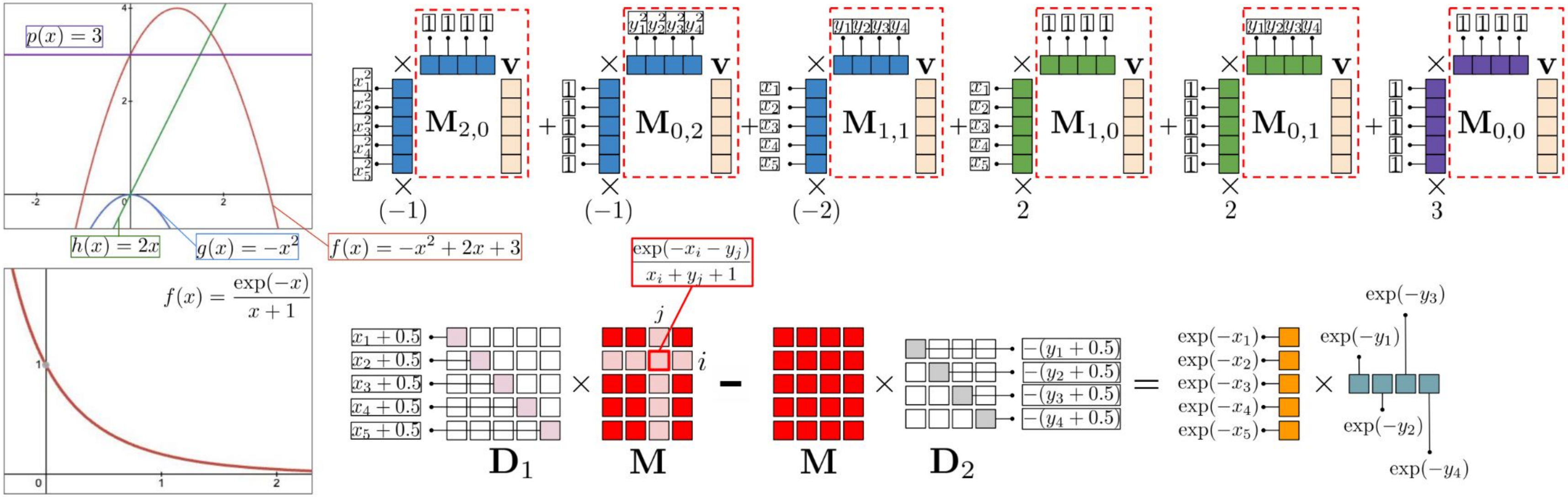


Cordial Functions f_θ

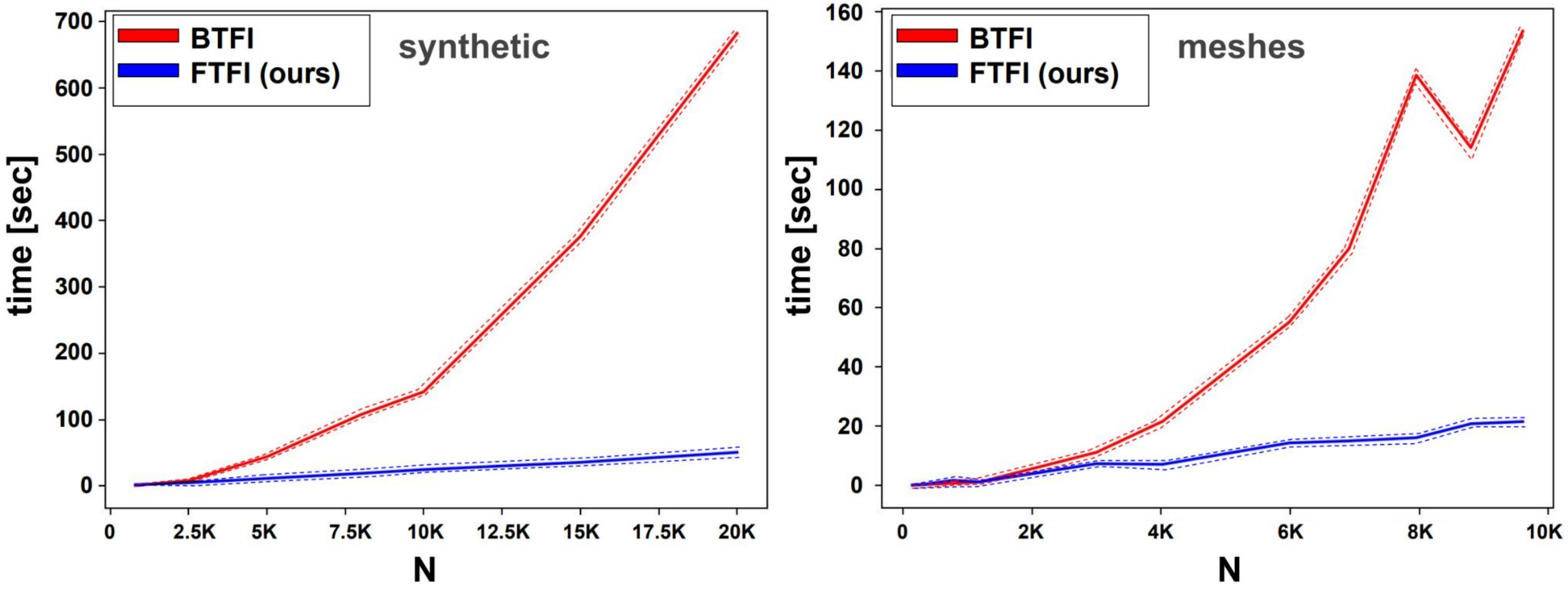
rational, trigonometric, products of exponentials and polynomials,...

Definition[cordial functions]: A function $f : \mathbb{R} \rightarrow \mathbb{R}$ is *d-cordial* (or: *cordial* if d is not specified), if there exists $d \in \mathbb{N}$ such that matrix-vector multiplication with a matrix $\mathbf{M} = [f(x_i + y_j)]_{i=1,\ldots,a}^{j=1,\ldots,b}$ can be conducted in time $O((a+b)\log^d(a+b))$ for every $(x_i)_{i=1}^a, (y_j)_{j=1}^b$.

Lemma [f-integration with cordial functions]: *If f is d-cordial then f-integration for the general weighted tree of N vertices can be conducted in time $O(N \log^{d+1}(N))$.*

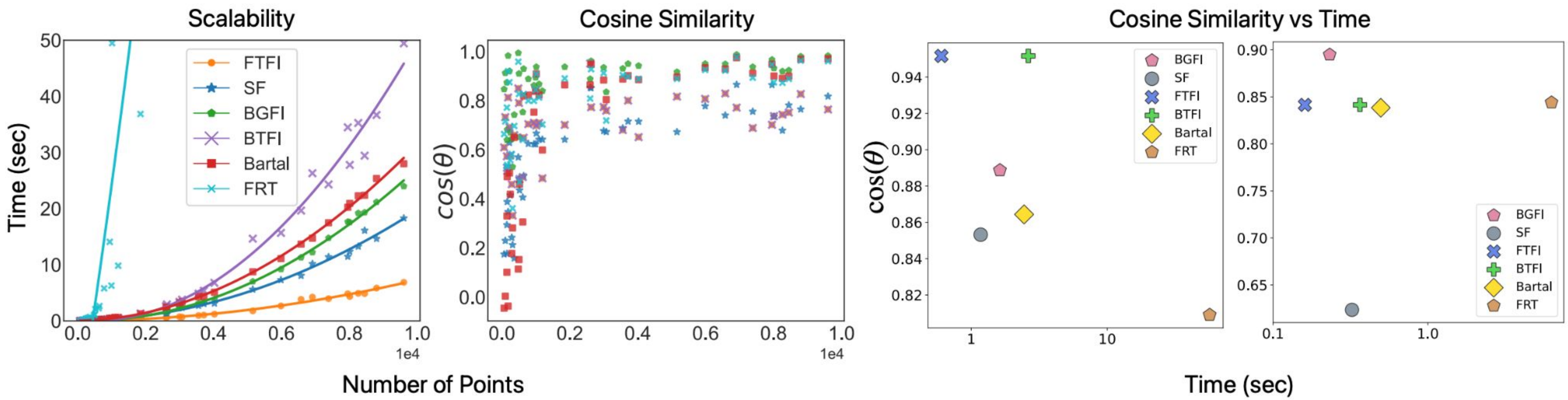


Runtime Efficiency



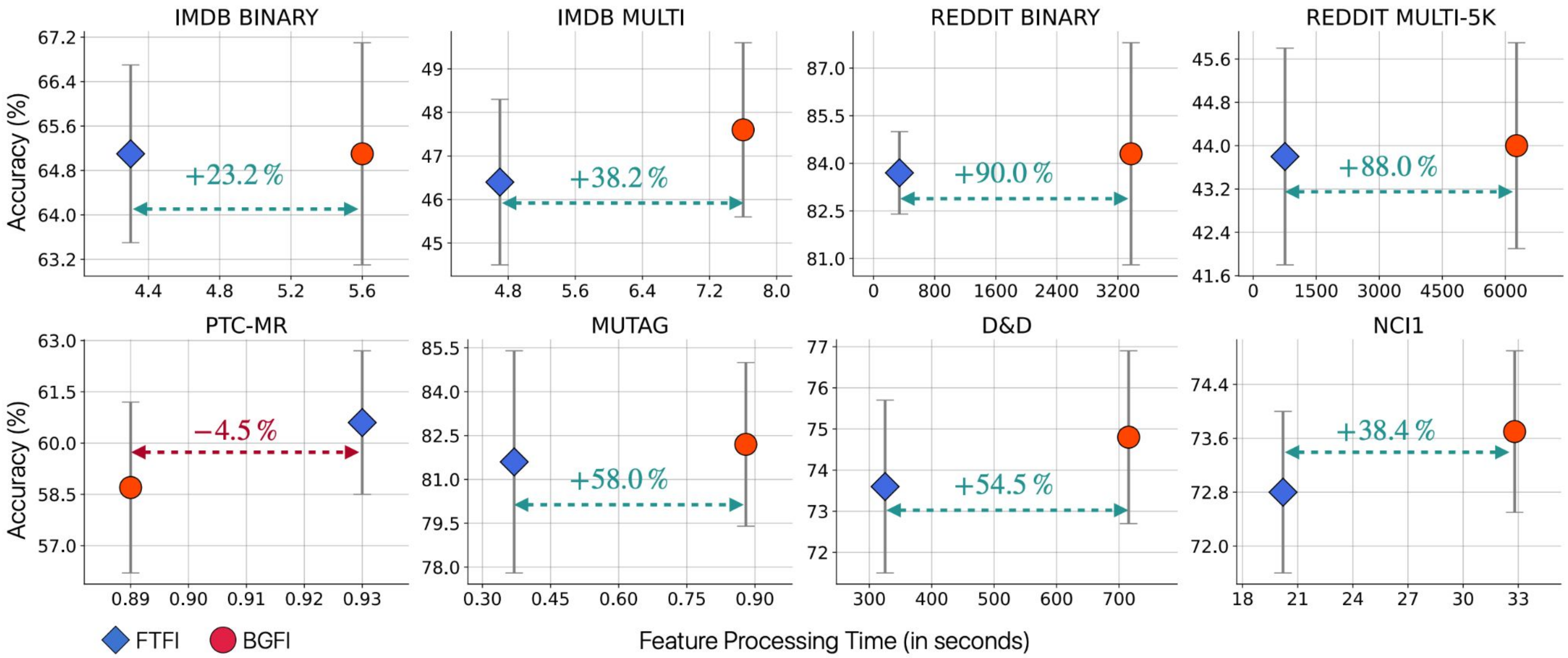
Runtime comparison of FTFI with BTFI as a function of the number of vertices, N. **Left:** Synthetic graphs. **Right:** Mesh-graphs from **Thingi10K**. The speed is not necessarily monotonic in N as it depends on the distribution of lengths of the shortest paths. For each graph, **10** experiments were run (std. shown via dotted lines).

Interpolation on Meshes



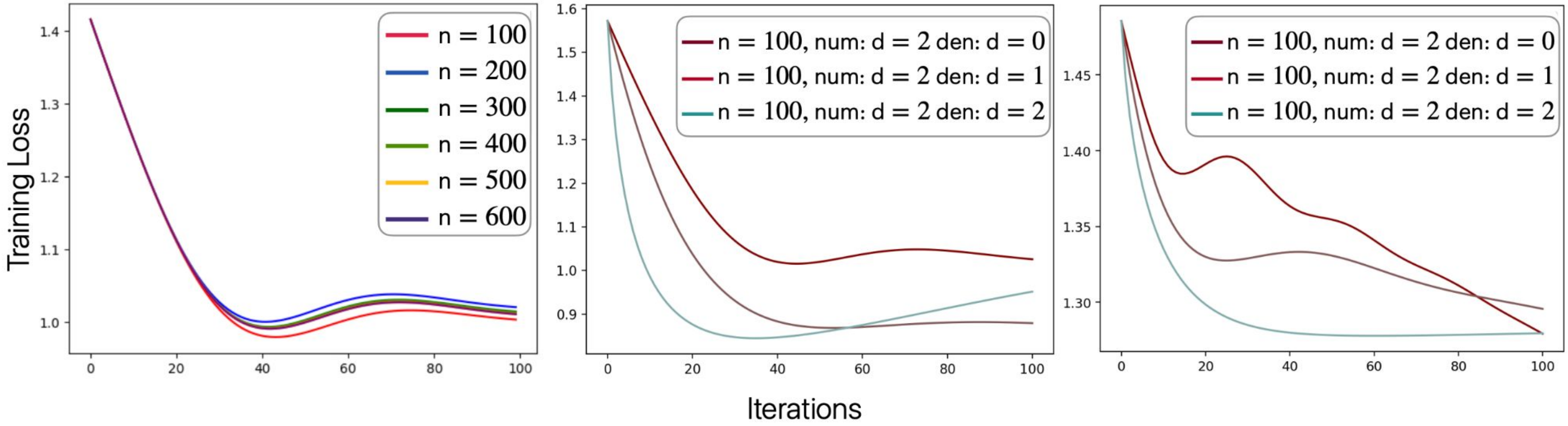
Speed (pre-processing time) and accuracy (cosine similarity) comparison of the FTFI and other baselines for vertex normal prediction on meshes. Cosine similarity of BFFI and FTFI almost overlaps. The last two figures are qualitative examples showcasing the tradeoff between cosine similarity and preprocessing time for meshes of sizes **3K** and **5K** nodes respectively.

Graph Classification



Trade-off plot comparing graph classification accuracy and feature processing time for the classifiers using FTFI and BGFI. FTFI achieves similar accuracy as BGFI while significantly reducing fp time across most datasets. We report the reduction in FTFI's processing time (±x%) compared to BGFI using a dotted line.

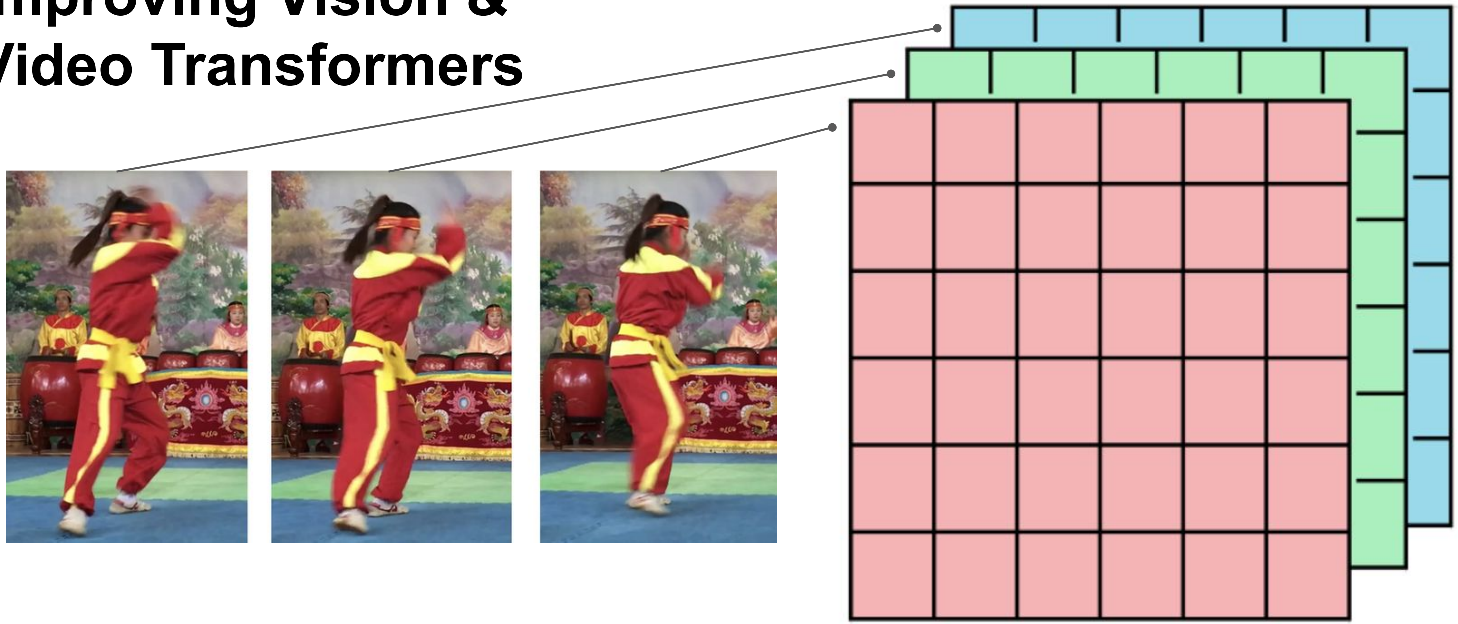
Improving Approximation Quality for Distance Matrices



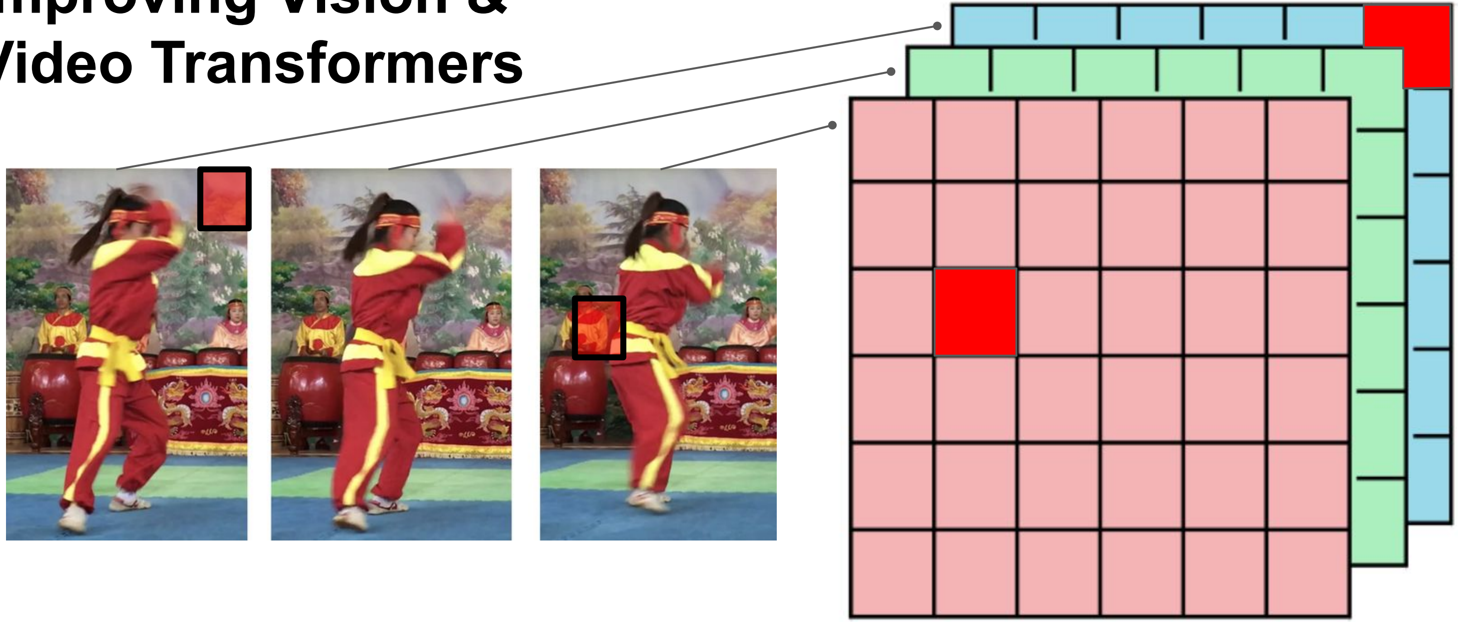
$$f_{b_0,\ldots,b_s}^{a_0,\ldots,a_t}(x) = \frac{a_0 + a_1 x + \ldots + a_t x^t}{b_0 + b_1 x + \ldots + b_s x^s}$$

Left: Relative Frobenius norm error as a function of the number of training iterations for different sizes n and learnable quadratic f. **Middle:** Comparison of the training of different rational functions f with num:d defining the degree of the numerator and den:d, the degree of the denominator for the synthetic graph obtained from a path N = 800 by adding 600 random edges and assigning random weights taken from (0, 1). **Right:** constructed similarly, but for a sampled mesh graphs from **Thingi10k** dataset.

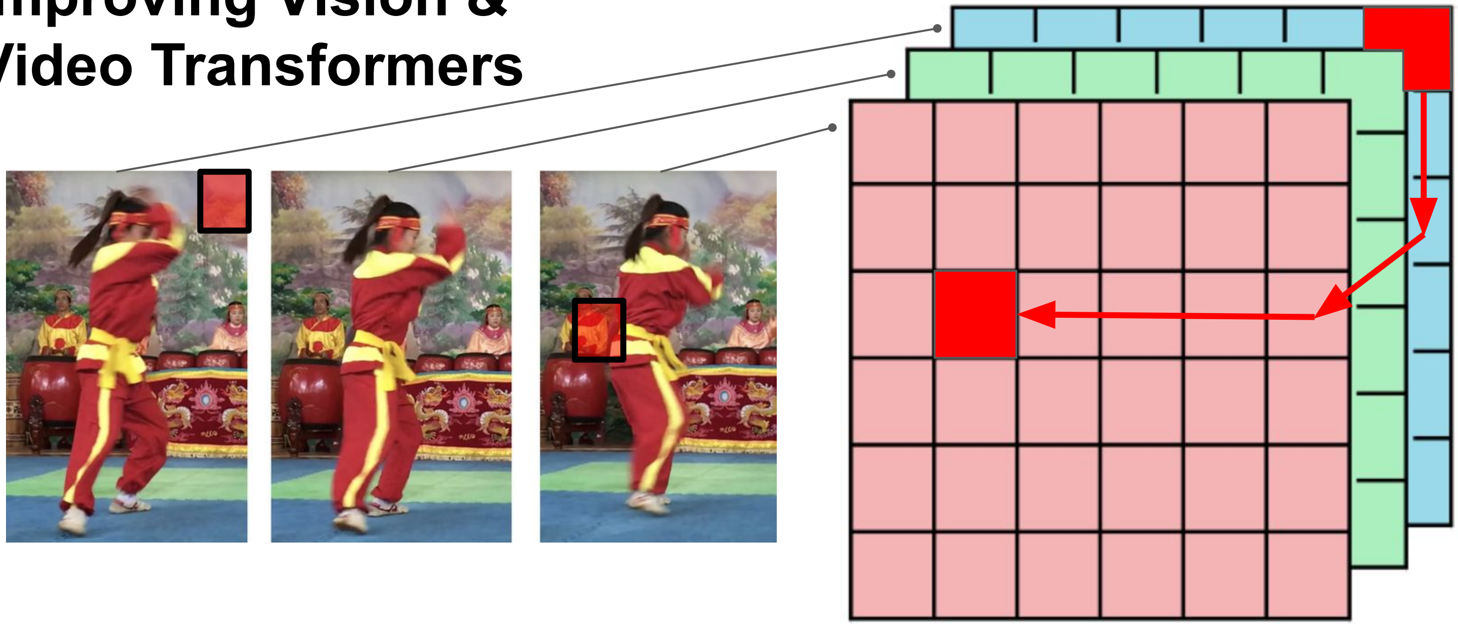
Improving Vision & Video Transformers



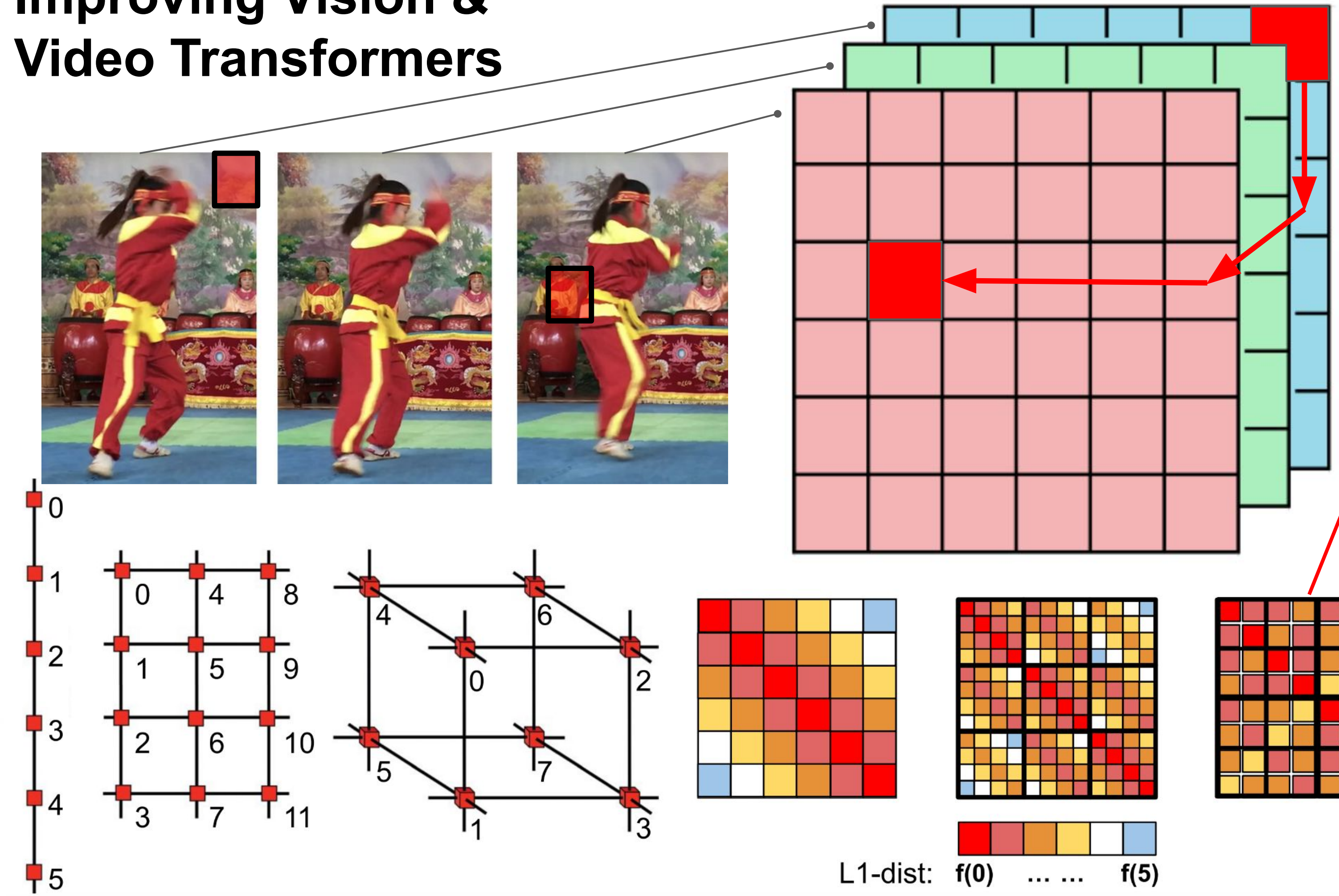
Improving Vision & Video Transformers



Improving Vision & Video Transformers



Improving Vision & Video Transformers



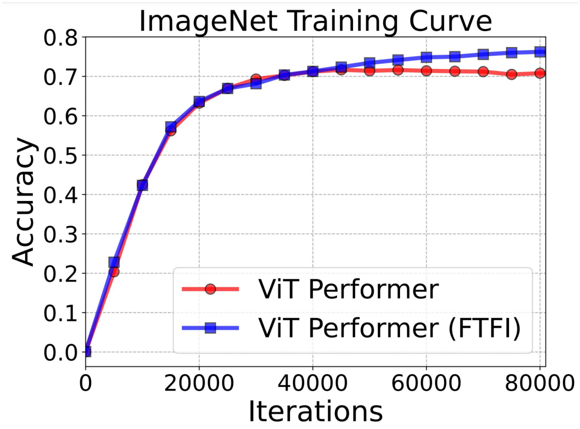
topological masking can be thought of as modulating regular attention with a particular graph kernel matrix

From block-Toeplitz matrices to differential equations on graphs: towards a general theory for scalable masked Transformers; Choromanski et al., ICML 2022

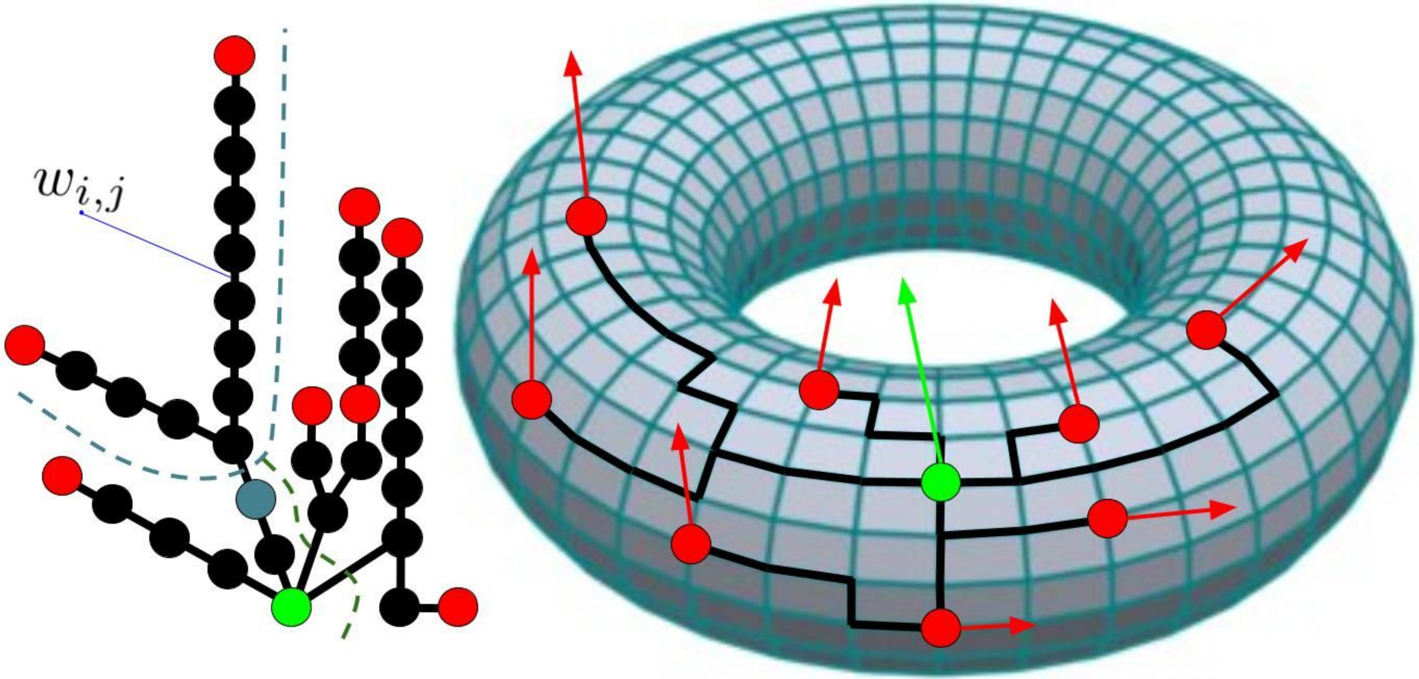
Improving Vision & Video Transformers

ImageNet																Place365			
$\phi :=$RELU				$\phi := x \to x^2$				$\phi := x \to x^4$				$\phi :=$ exp				$\phi :=$ ReLU			
synced	**g**	**t**	**Acc. (%)**	**synced**	**g**	**t**	**Acc. (%)**	**synced**	**g**	**t**	**Acc. (%)**	**synced**	**g**	**t**	**Acc. (%)**	**synced**	**g**	**t**	**Acc. (%)**
NA	NA	NA	76.23	NA	NA	NA	75.03	NA	NA	NA	76.37	NA	NA	NA	76.76	NA	NA	NA	54.80
✓	exp	1	77.28	✓	exp	1	76.66	✓	exp	1	77.84	✗	exp	1	**78.79**	✗	exp	1	56.69
✓	exp	2	76.60	✓	exp	2	75.91	✓	exp	2	77.23	✗	exp	2	78.51	✗	$z \to z^{-1}$	1	56.44
✗	exp	1	**77.79**	✗	exp	1	**76.76**	✗	exp	1	77.94	✗	$z \to z^{-1}$	1	77.39	✗	$z \to z^{-1}$	5	56.32
✗	exp	2	77.43	✗	exp	2	76.27	✗	exp	2	**78.15**	✗	$z \to z^{-1}$	2	77.69	✗	$z \to z^{-1}$	10	56.51

Performance of Topological Vision Transformers with tree-based masking. For each attention kernel, we present the results of the best variant in **bold** and Performer baselines in blue.



Experiments with the RPE mechanism for ViT-L and on ImageNet. We observe that FTFI provides **7%** accuracy gain compared to the Performer variant.



Thank You for Your Attention !