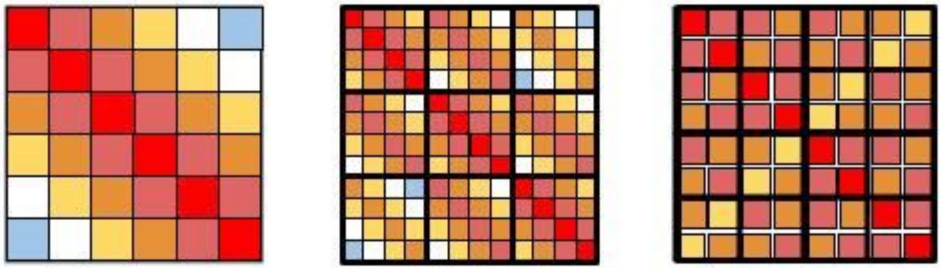
From block-Toeplitz matrices to differential equations on graphs

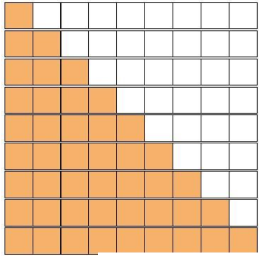
Towards a general theory for scalable masked Transformers…

Krzysztof Choromanski, Han Lin, Haoxian Chen, Tianyi Zhang, Arijit Sehanobish, Valerii Likhosherstov, Jack Parker-Holder, Tamas Sarlos, Adrian Weller, Thomas Weingarten

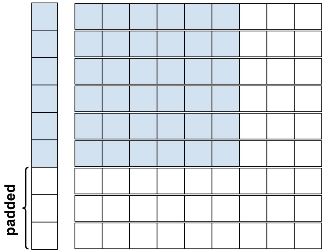


Masking as a powerful inductive bias in Transformers

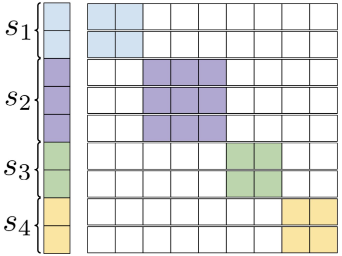
- Causal attention



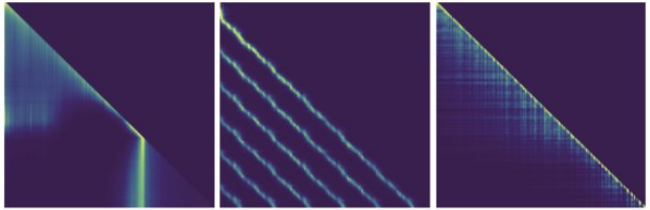
- Padding



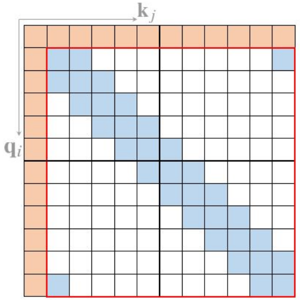
- Packing



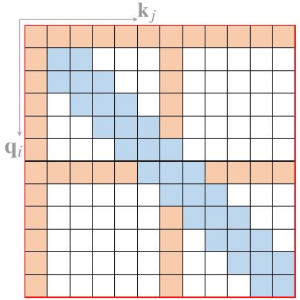
- stochastic-RPE (Liutkus et al 2021)



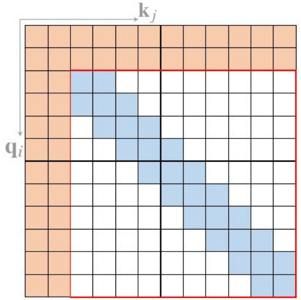
mask M has low-rank decomposition



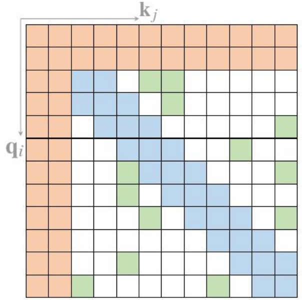
- Star-Transformer



- Longformer

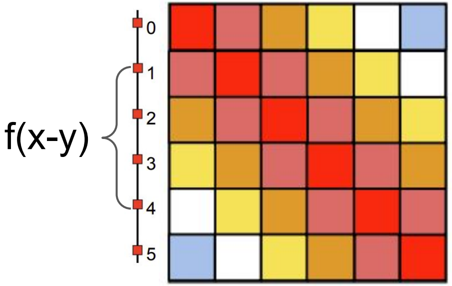


- ETC



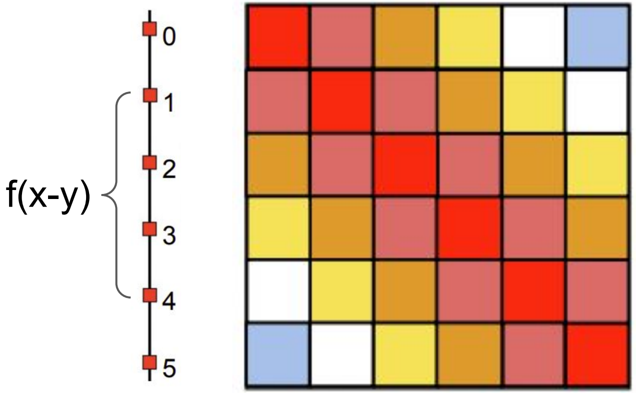
- BigBird

- Relative Position Encoding (RPE)



...

How to incorporate general masking into scalable Transformers ?



All You Need is Fast Matrix-Vector Multiplication

masking softmax attention

$$\mathrm{Att}_{\mathrm{SM}}(\mathbf{Q},\mathbf{K},\mathbf{V},\mathbf{N}) = \mathbf{D}^{-1}\mathbf{A}\mathbf{V}$$
$$\mathbf{A} = \exp(\mathbf{N} + \mathbf{Q}\mathbf{K}^{\top}/\sqrt{d_{QK}}), \quad \mathbf{D} = \mathrm{diag}(\mathbf{A}\mathbf{1}_L)$$

$\cap$

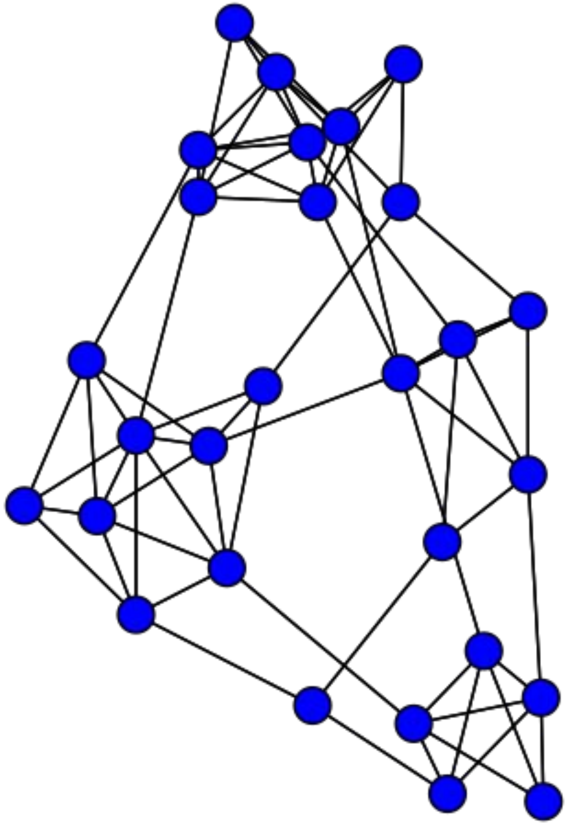
masking kernel attention

$$\mathrm{Att}_{\mathrm{K}}(\mathbf{Q},\mathbf{K},\mathbf{V},\mathbf{M}) = \mathbf{D}^{-1}\mathbf{A}\mathbf{V}$$
$$\mathbf{A} = \mathbf{M} \odot \mathrm{K}(\mathbf{Q},\mathbf{K}), \quad \mathbf{D} = \mathrm{diag}(\mathbf{A}\mathbf{1}_L)$$

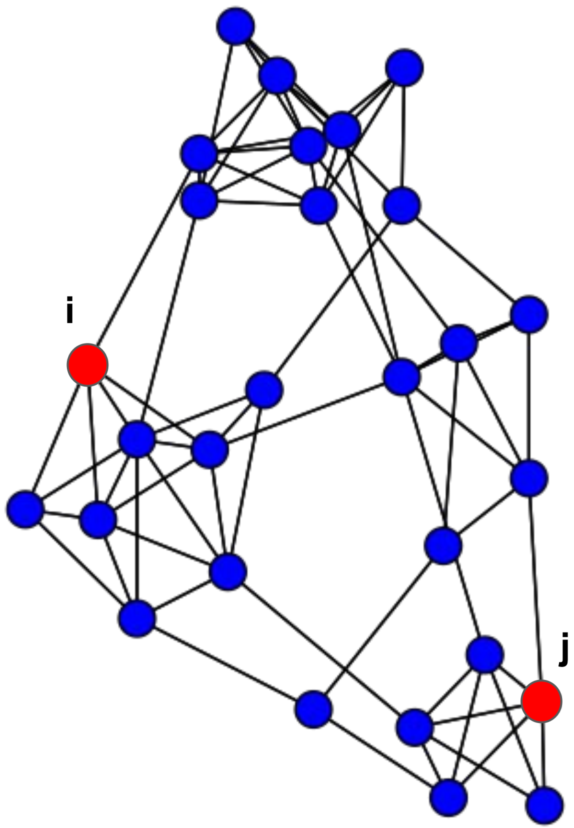
Lemma (Choromanski et al. 2021):

*As long as matrix **M** supports fast (sub-quadratic) matrix-vector multiplication, the corresponding masking mechanism can be incorporated into Performers (low-rank linear attention Transformers) in the sub-quadratic time.*

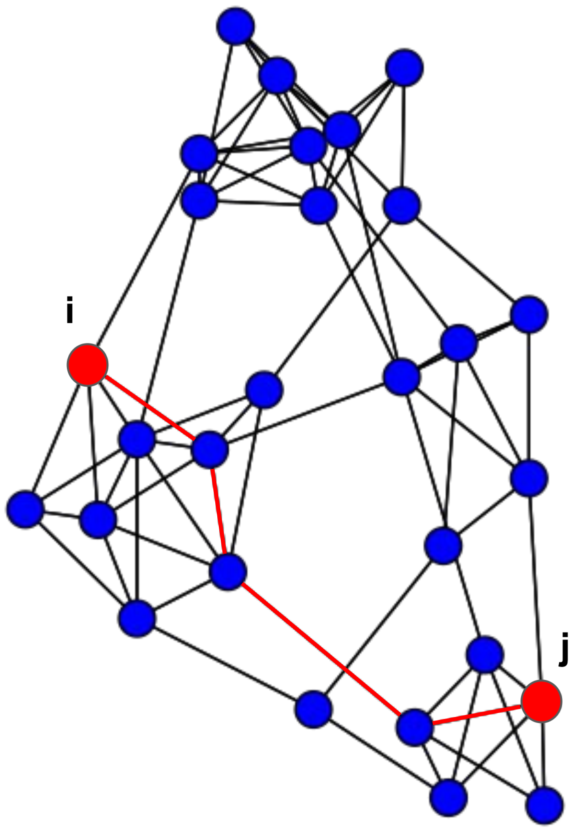
Graph-induced masking



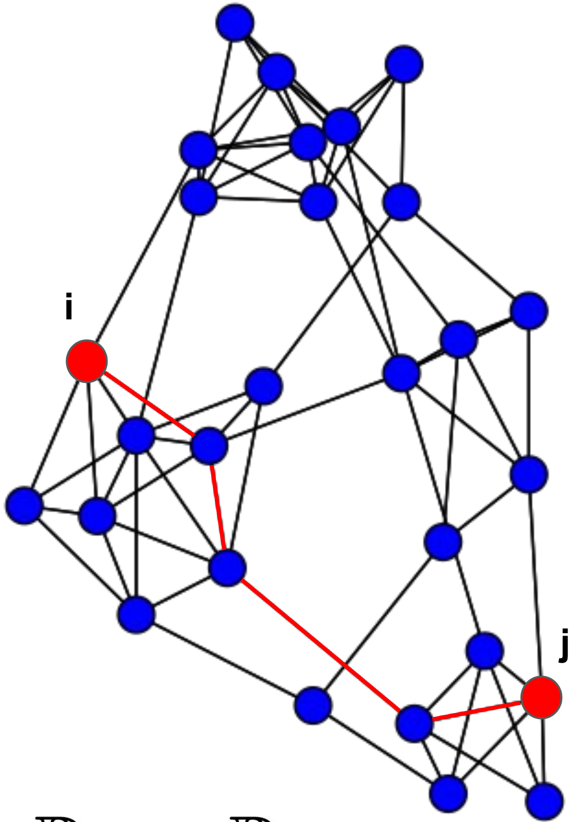
Graph-induced masking



Graph-induced masking

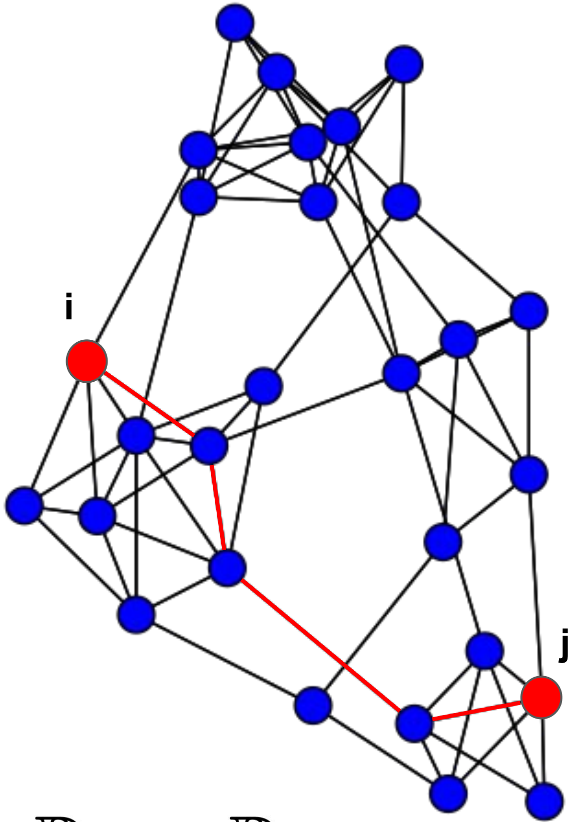


Graph-induced masking



$$f : \mathbb{R} \rightarrow \mathbb{R}$$

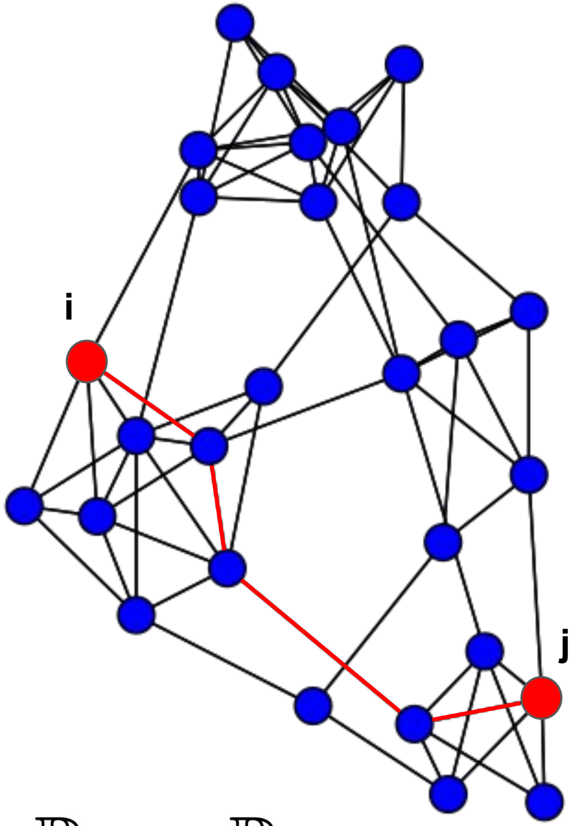
Graph-induced masking



$$\mathbf{M} \overset{\text{def}}{=} [f(\overbrace{\text{dist}_{G_{\text{base}}}(i,j)}^{\text{e.g. shortest-path distance}})]_{i,j=1,\ldots,L}$$

$$f : \mathbb{R} \rightarrow \mathbb{R}$$

Graph-induced masking

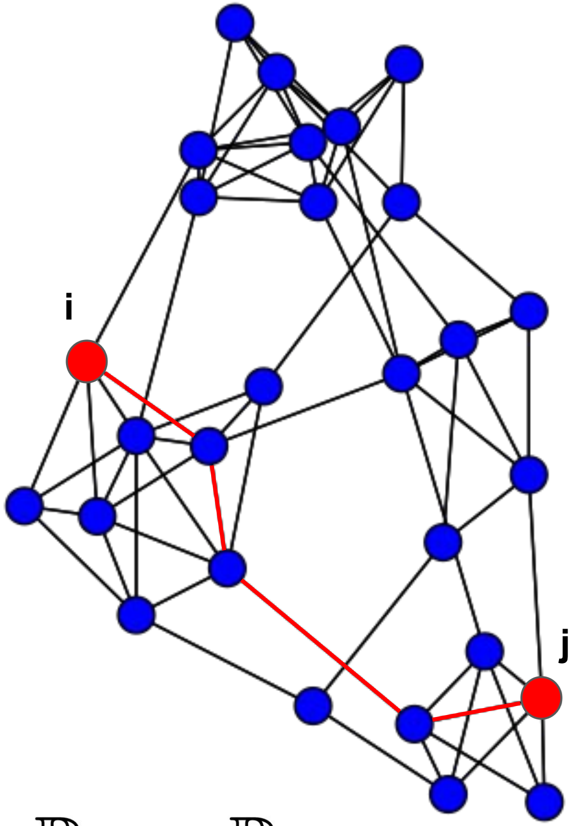


$$\mathbf{M} \stackrel{\text{def}}{=} [f(\underbrace{\text{dist}_{G_{\text{base}}}(i,j)}_{\text{e.g. shortest-path distance}})]_{i,j=1,\ldots,L}$$

(G,f) **tractable** if M supports **sub-quadratic** matrix vector multiplication

$f : \mathbb{R} \rightarrow \mathbb{R}$

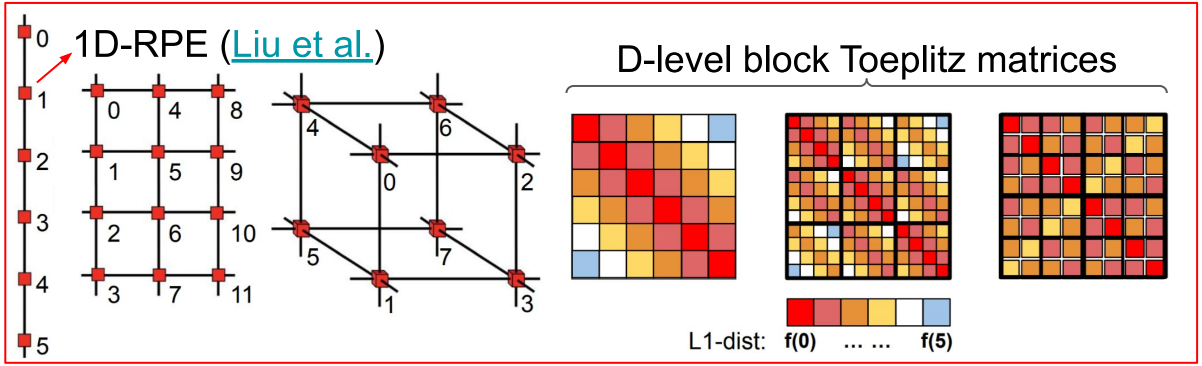
Graph-induced masking



$$\mathbf{M} \overset{\text{def}}{=} [f(\underbrace{\text{dist}_{G_{\text{base}}}(i,j)}_{\text{e.g. shortest-path distance}})]_{i,j=1,\ldots,L}$$

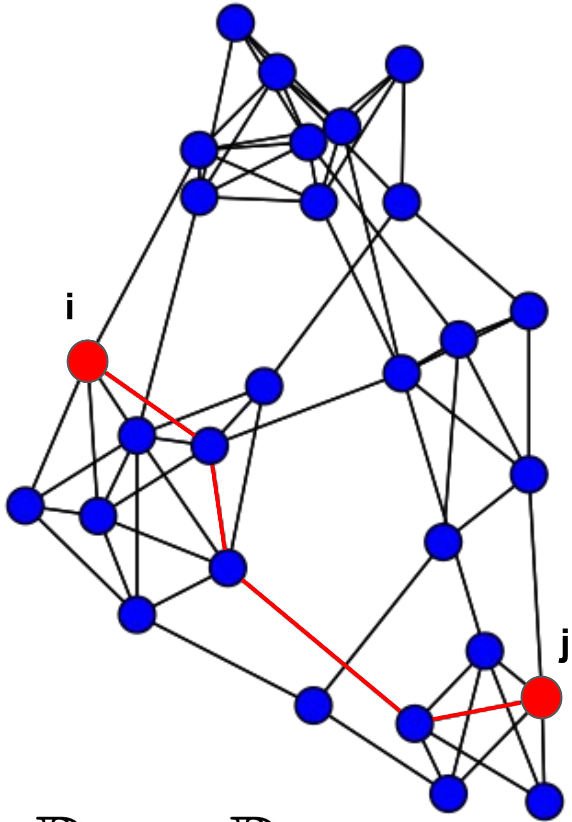
(G,f) **tractable** if M supports **sub-quadratic** matrix vector multiplication

- If G is a d-dimensional unweighted grid then (G, *) is tractable



$$f : \mathbb{R} \rightarrow \mathbb{R}$$

Graph-induced masking



$f : \mathbb{R} \to \mathbb{R}$

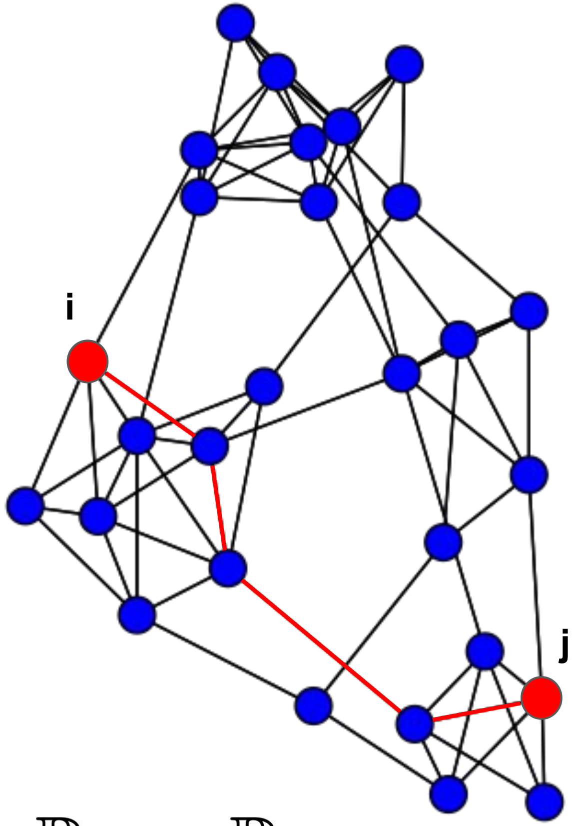
e.g. shortest-path distance

$$\mathbf{M} \stackrel{\text{def}}{=} [f(\underbrace{\text{dist}_{G_{\text{base}}}(i,j)})]_{i,j=1,\ldots,L}$$

(G,f) **tractable** if M supports **sub-quadratic** matrix vector multiplication

- If G is a d-dimensional unweighted grid then (G, *) is tractable
- If G is a forest and: (a) f is exponentiated affine mapping or (b) G is unweighted or (c) G is of sublinear diameter then (G, f) is tractable (*molecular assembly trees*)

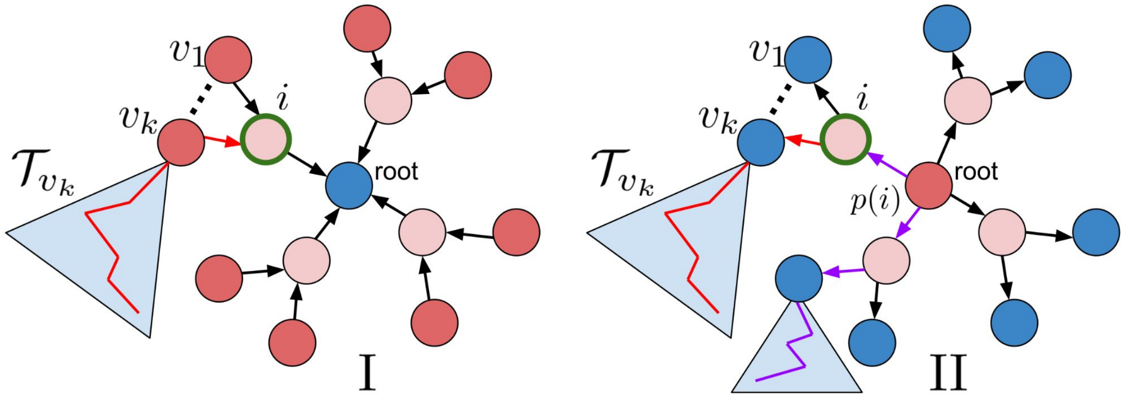
Graph-induced masking



$f : \mathbb{R} \to \mathbb{R}$

$$\mathbf{M} \overset{\text{def}}{=} [f(\overbrace{\text{dist}_{G_{\text{base}}}(i,j)}^{\text{e.g. shortest-path distance}})]_{i,j=1,\dots,L}$$

- If G is a forest and f is exponentiated affine mapping then (G, f) is tractable



Graph Kernel Attention Transformers (GKAT)

- **Main idea:** define M as a graph kernel matrix for a kernel defined on graph nodes.

$$\mathrm{K}: V \times V \rightarrow \mathbb{R}$$

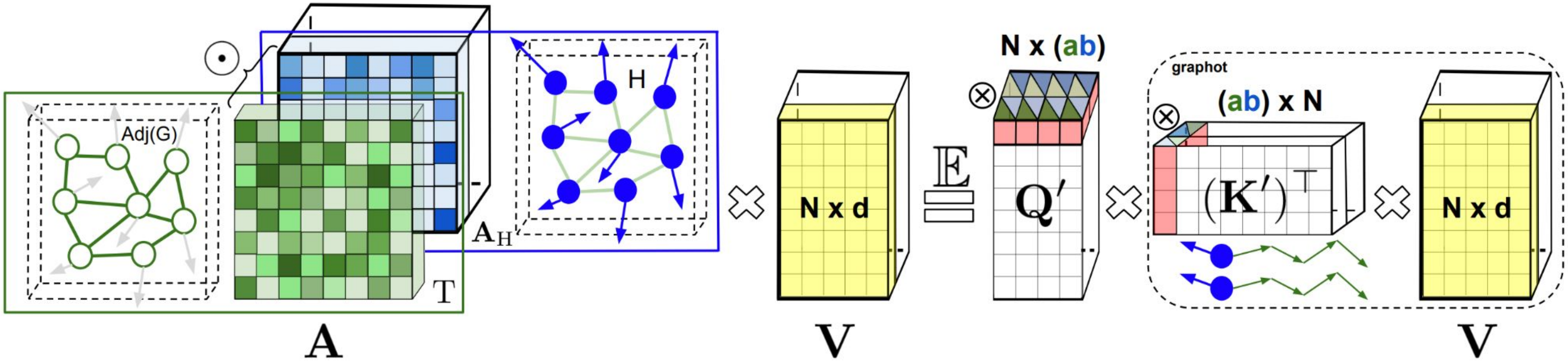
- Examples: **Graph Diffusion Kernels** (GDKs)

$$\mathcal{K}_{\mathrm{K}} = \exp(-\lambda \mathbf{T}) \stackrel{\mathrm{def}}{=} \sum_{i=0}^{\infty} \frac{(-\lambda)^i \mathbf{T}^i}{i!}$$

  - negated adjacency matrix
  - Laplacian matrix
  - normalized Laplacian matrix

- **Execution:** Support fast mask-vector multiplication via: (a) stochastic low-rank decompositions or: (b) spectral graph algorithms coupled with new methods for computing the actions of matrix exponentials.

Low-rank decomposition and Random Walks Graph-Nodes Kernels (RWGNs)



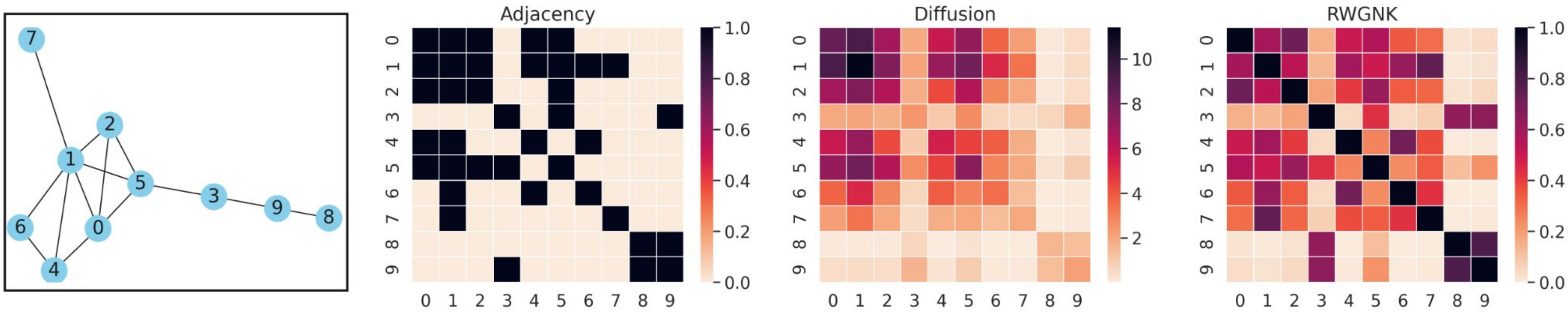
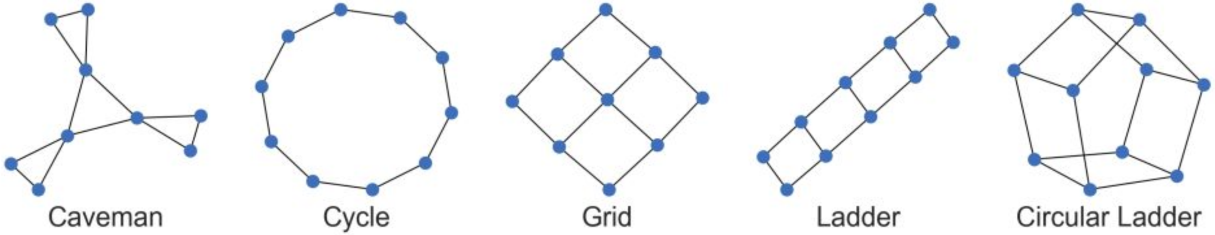


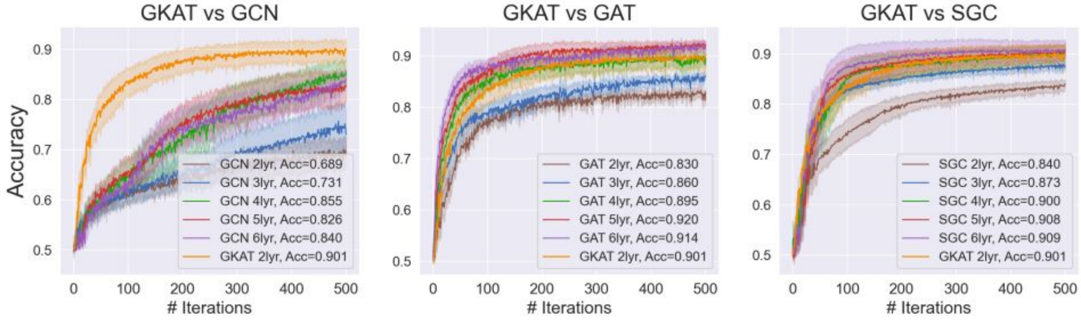
Table 1. Performance of different algorithms on the bioinformatics data. For each dataset, we highlighted/underlined the best/second best method. GKAT is the best on three out of four tasks.

	D&D	NCI1	Proteins	Enzymes
Baseline	78.4 ±4.5%	69.8±2.2%	**75.8±3.7%**	65.2±6.4%
DGCNN	76.6±4.3%	76.4±1.7%	72.9±3.5%	38.9±5.7%
DiffPool	75.0±3.5%	**76.9±1.9%**	73.7±3.5%	59.5±5.6%
ECC	72.6±4.1%	76.2±1.4%	72.3±3.4%	29.5±8.2%
GraphSAGE	72.9±2.0%	76.0±1.8%	73.0±4.5%	58.2±6.0%
RWNN	77.6±4.7%	71.4±1.8%	74.3±3.3%	56.7±5.2%
GKAT	**78.6±3.4%**	75.2±2.4%	**75.8 ±3.8%**	**69.7 ±6.0%**

Table 2. Performance of different algorithms on the social network data. GKAT is among two top methods for four out of five tasks.

	IMDB-B	IMDB-M	REDDIT-B	REDDIT-5K	COLLAB
Baseline	70.8±5.0%	**49.1 ±3.5%**	82.2±3.0%	52.2±1.5%	70.2±1.5%
DGCNN	69.2±5.0%	45.6±3.4%	87.8±2.5%	49.2±1.2%	71.2±1.9%
DiffPool	68.4±3.3%	45.6±3.4%	89.1±1.6%	53.8±1.4%	68.9±2.0%
ECC	67.7±2.8%	43.5±3.1%	OOM	OOM	OOM
GraphSAGE	68.8±4.5%	47.6±3.5%	84.3±1.9%	50.0±1.3%	**73.9±1.7%**
RWNN	70.8±4.8%	47.8±3.8%	**90.4±1.9%**	51.7±1.5%	71.7±2.1%
GKAT	**71.4±2.6%**	47.5±4.5%	89.3±2.3%	**55.3±1.6%**	73.1±2.0%





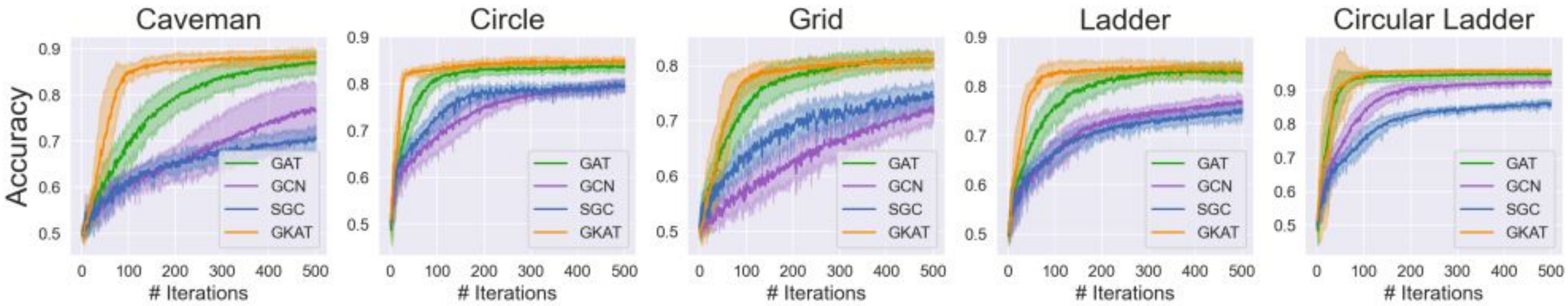
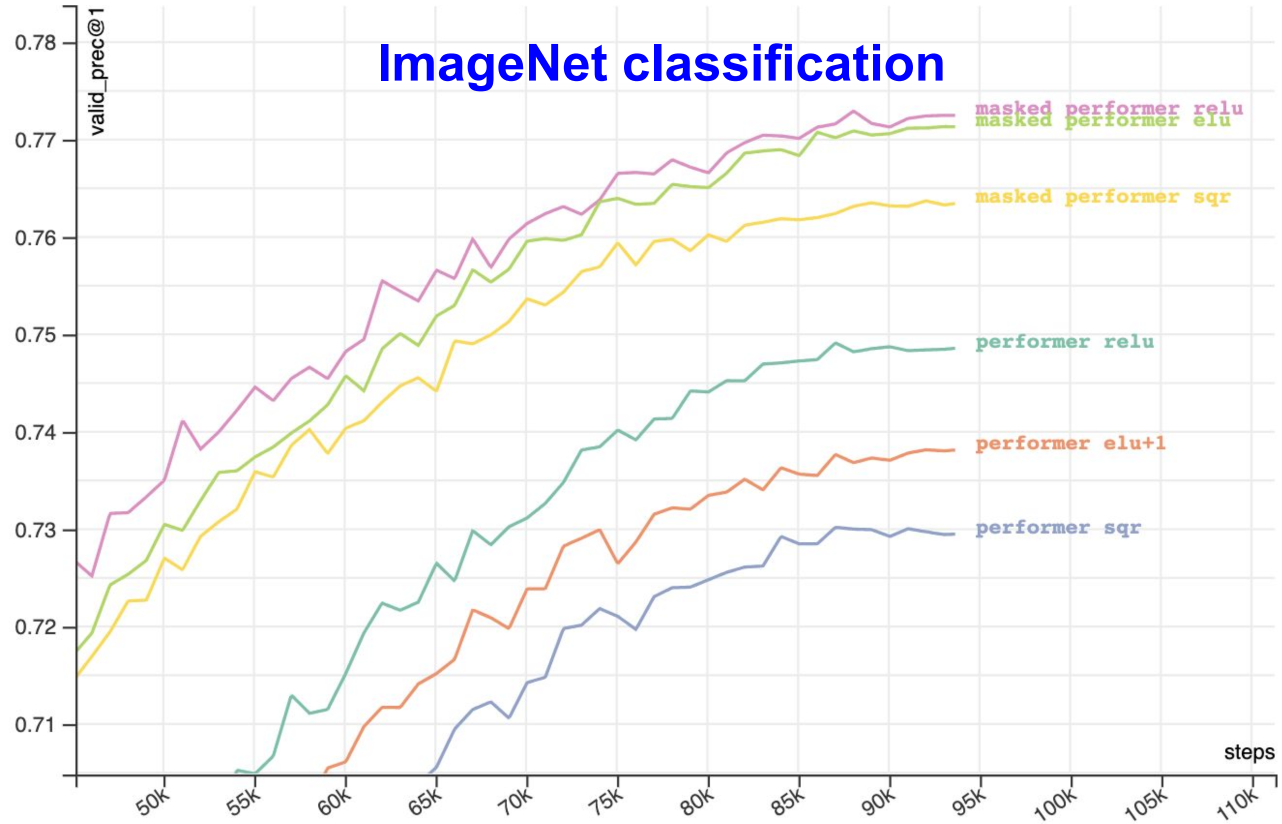


Figure: Model accuracy comparison of all four methods: GKAT, GAT, GCN and SGC on the motif-detection task. All architectures are 2-layer. GKAT outperforms other algorithms on all the tasks. See also Appendix:Sec. 7.4 for the tabular version with $100K$-size graphs.

2-level block Toeplitz masking for images



Code: https://github.com/google-research/google-research/tree/master/topological_transformer

Fig: Masked Transformer in action.

Thank you for your Attention !