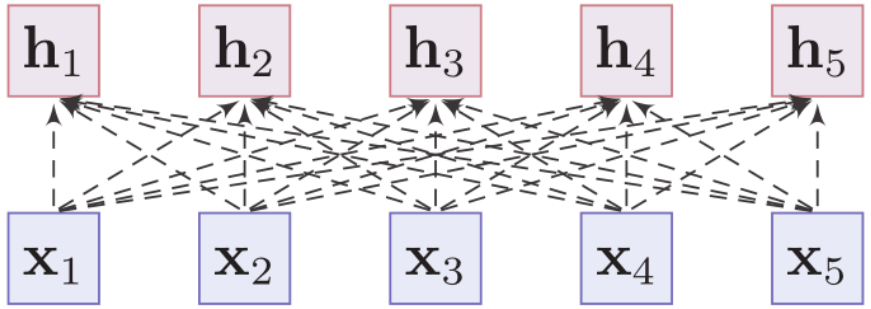
What Dense Graph Do You Need for Self-attention?

Yuxin Wang (Speaker), Chu-Tak Lee ,Qipeng Guo, Zhangyue Yin ,
Yunhua Zhou,
Xuanjing Huang , Xipeng Qiu

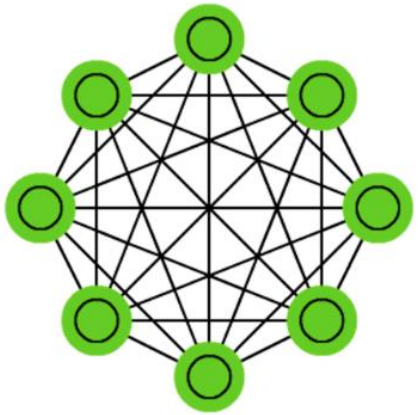
Fudan University

Transformer and GNN

- Transformer is a model built with self-attention module.



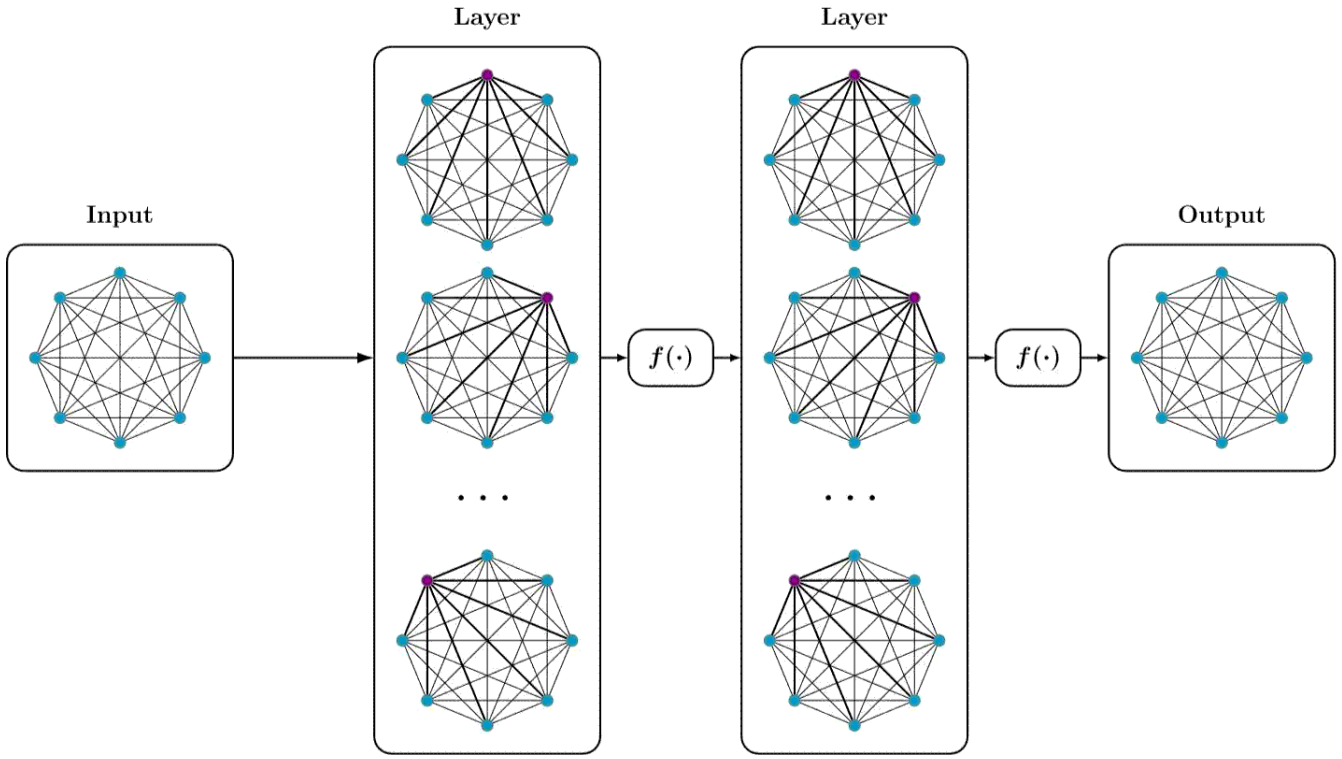
- Fully-connection



Weights α_{ij} are generated dynamically with attention mechanism

Graph Views of Transformers

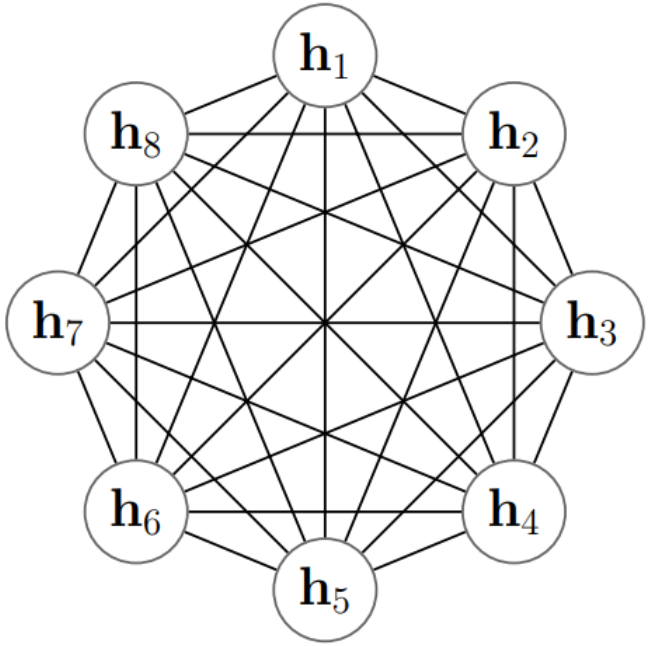
- Transformer is a model built with self-attention module.



Contents

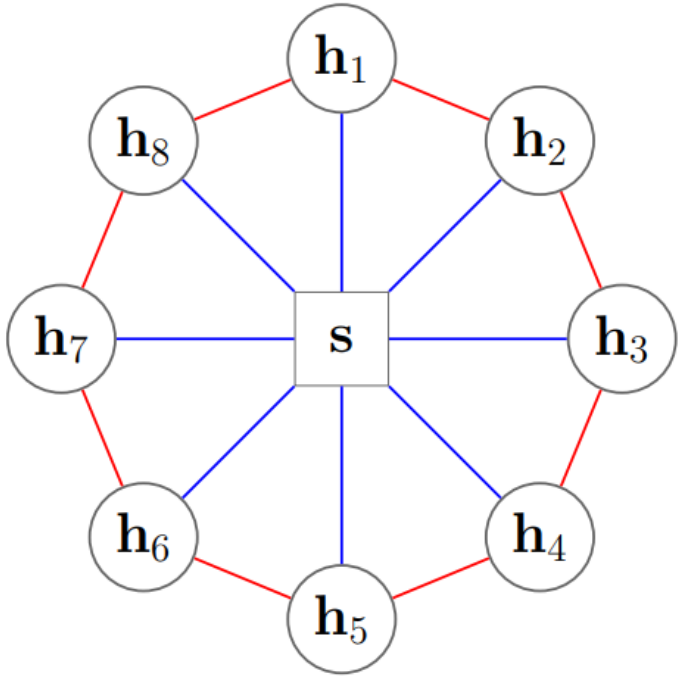
- Sparsification of self-attention
- Normalized Information Payload(NIP)
- Hypercube Transformer
- Experiments

Self-attention as Complete Graph



- Complexity : $\Theta(N^2)$
- Reduce complexity ----> Reduce the number of edges

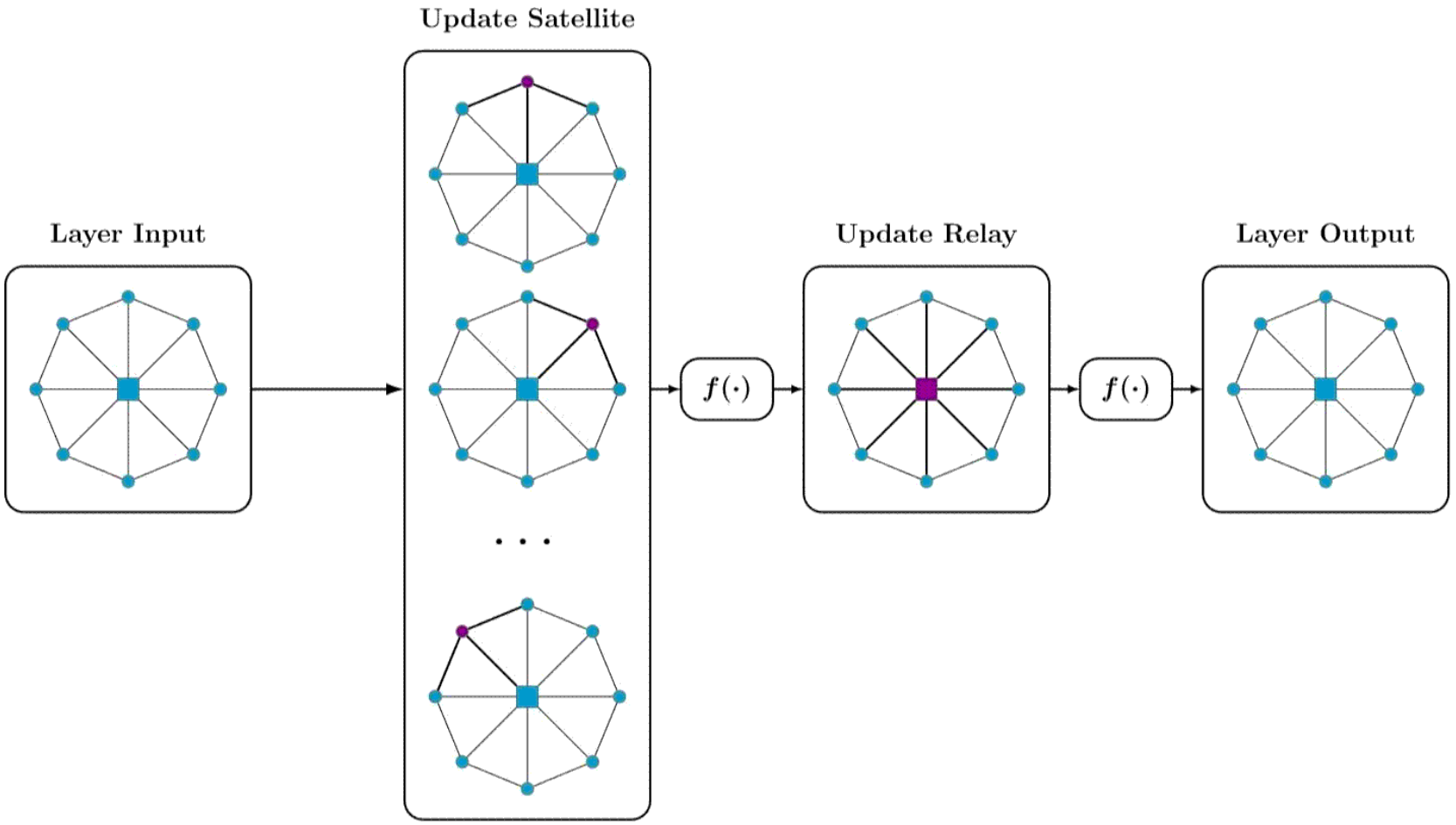
Star-Transformer



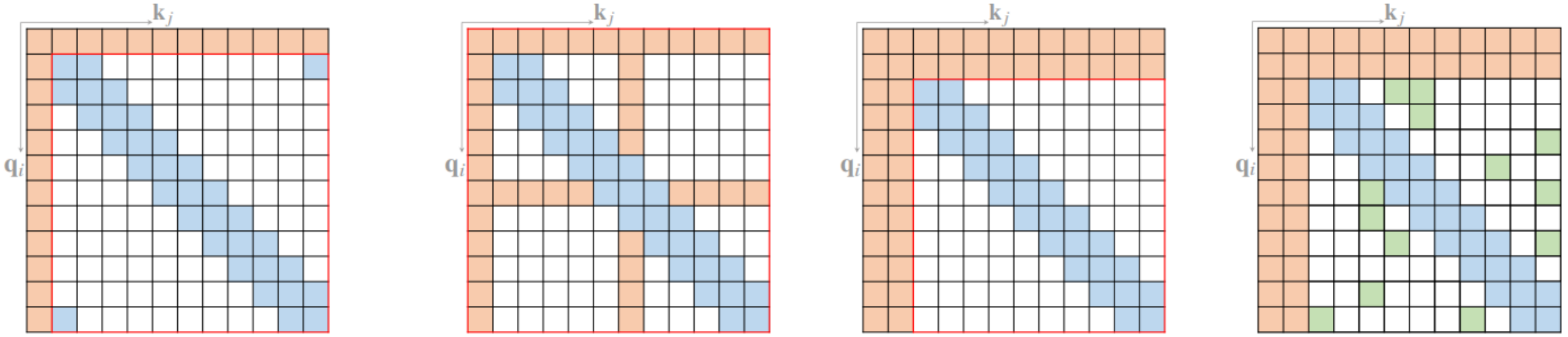
- Reduce Complexity to $\Theta(N)$.
- Preserve the capacity to capture both local composition and long-range dependency.

Qipeng Guo, Xipeng Qiu, Pengfei Liu, Yunfan Shao, Xiangyang Xue, Zheng Zhang. Star-Transformer, NAACL 2019, https://arxiv.org/pdf/1902.09113.pdf

Star-Transformer



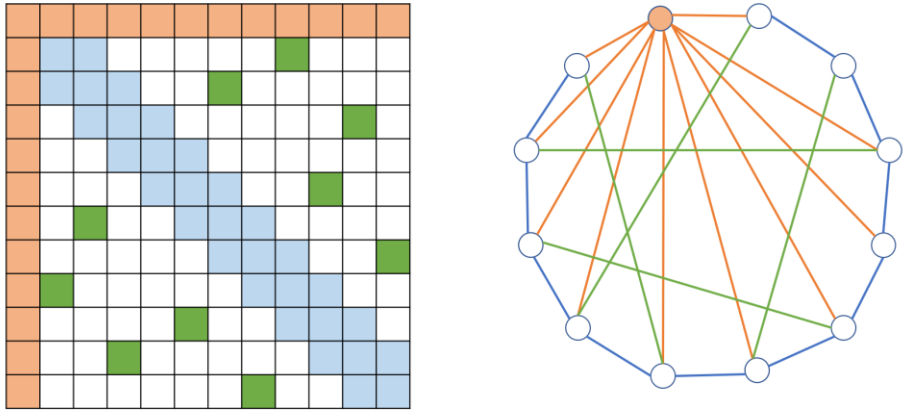
Empirical Sparse Transformers



(a) Star-Transformer (b) Longformer (c) ETC (d) BigBird

Lin, T., Wang, Y., Liu, X., and Qiu, X. A survey of transformers, 2021

Sparse Transformer: A Graph View



- Which property is important for those graphs serving as ground for self-attention?
- How dense do we need the graph to be in order to reduce complexity and at the same time remain performance?

Contents

- Sparsification of self-attention
- Normalized Information Payload(NIP)
- Hypercube Transformer
- Experiments

Two views of a graph

- Computational Complexity (CC)
  - Computational Complexity is the computation complexity required to allow the model to grab all interactions among tokens when using graph G for self-attention.
  - For complete graph, it's N.
- Information Payload (IP)
  - Information Payload. measuring how much information a graph can transfer when allowing the model to grab all interactions among tokens.
  - For complete graph, it's $\frac{1}{N-1}$.
- To better compare information transfer on different graphs, we define the Normalized Information Payload (NIP)

$$\mathrm{NIP}(G) := \frac{\mathrm{IP}(G)}{\mathrm{CC}(G)}.$$

Computational Complexity

- κ(G)
  - Given a G-attention layer, to make the whole model grab all interactions among tokens, we need to stack κ(G) G-attention layers.
  - Straightforwardly, κ(G) is the diameter of graph G.
  - For complete graph, it's 1.
- ρ(G)
  - For one G-attention layer, when the input sequence is fixed at length N, the Computational Complexity for one layer is proportional to the mean degree of G, which we denote by ρ(G).
  - For complete graph, it's $\frac{N-1}{2}$.

$$\mathrm{CC}(G) := \rho(G) \times \kappa(G).$$

Information Payload

Definition 2.2. For one path $P_{ab} \in \mathcal{P}_{ab}$, the Information Payload of one path P_{ab}, denoted by $R(P_{ab})$, is defined as

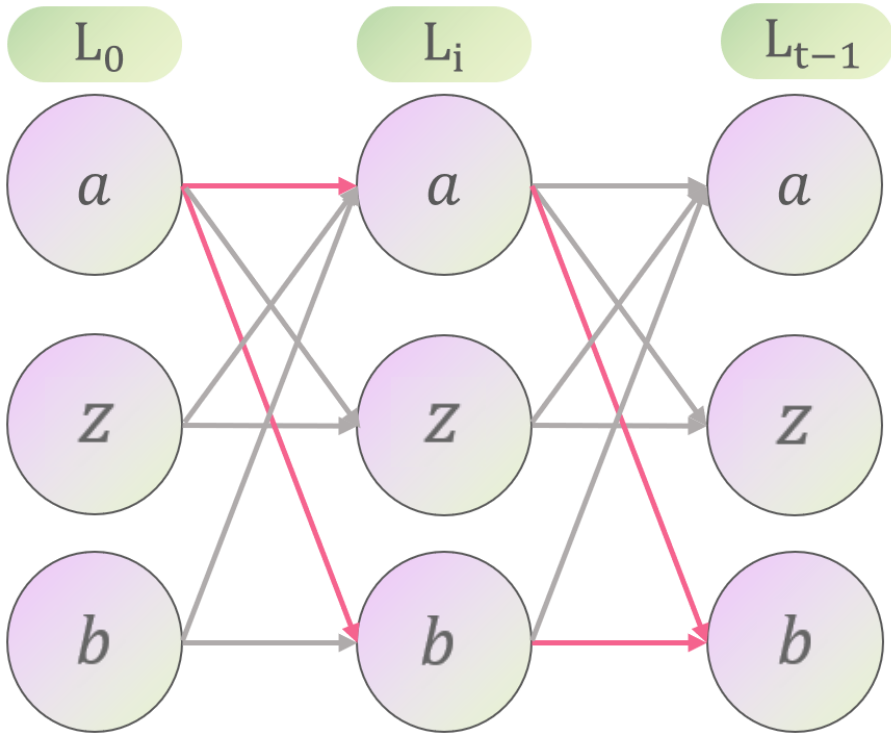
$$R(P_{ab}) := \prod_{v \in P_{ab} \ \& \ v \neq a} \frac{1}{deg(v)}.$$

Definition 2.3. The Information Payload between node pair (a, b), denoted by I_{ab} is sum of Information Payload of all paths that belong to $\mathcal{P}_{ab}$:

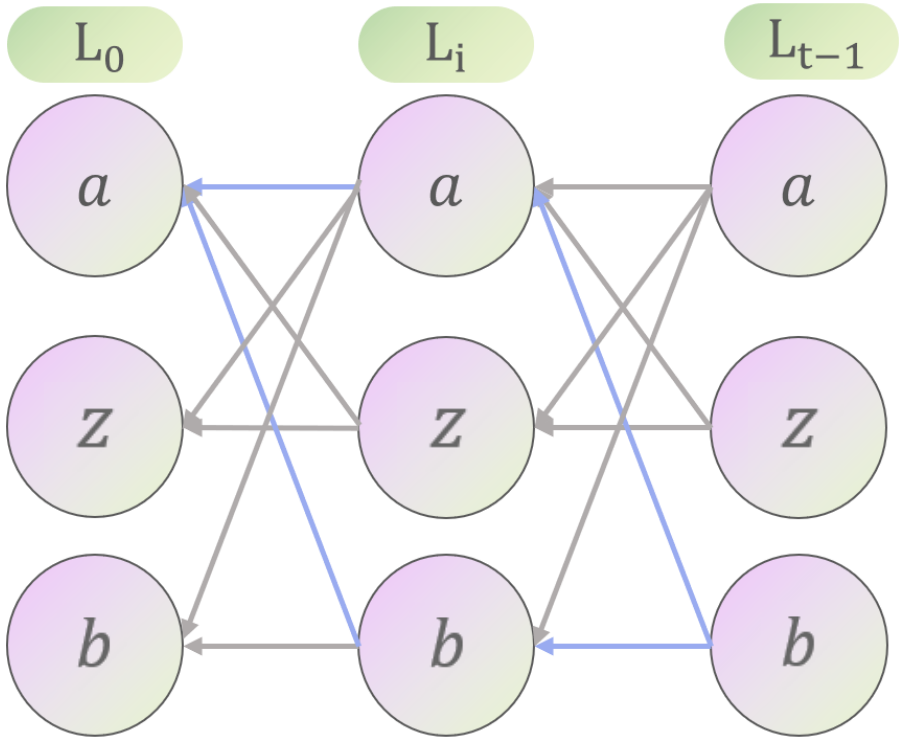
$$I_{ab} = \sum_{P_{ab} \in \mathcal{P}_{ab}} R(P_{ab}).$$

Closely related to Random Walk

Theorem 2.4. *Information Payload between two nodes I_{ab} equals to the probability of a random walk starts from node b that ends in node a at step $len(P_{ab})$.*



Attention forward



Random walk

Information Payload

Definition 2.5. The Information Payload for a graph G $\mathrm{IP}(G)$ is the smallest Information Payload I_{ab} between node pairs (a, b) whose distance is the diameter of the graph. Let Δ be the set of node pairs whose distance is the diameter of the graph, we have

$$\mathrm{IP}(G) := \min_{(a,b)\in\Delta} I_{ab}. \tag{5}$$

Normalized Information Payload (NIP)

Table 1. Normalized Information Payload for commonly used graphs, where w is the number of neighbors in ring lattice. $\star$: $\Theta\left(\frac{1}{N^2}\right)$ after refinement.

Type of graph	$\mathrm{CC}(G)\downarrow$	$\mathrm{IP}(G)\uparrow$	$\mathrm{NIP}(G)\uparrow$
Complete	$\Theta(N)$	$\Theta\left(\frac{1}{N}\right)$	$\Theta\left(\frac{1}{N^2}\right)$
E-R random	$\Theta(\log^2 N)$	$\Theta\left(\frac{(N-2)!}{N^{\log N}(N-\log N)!}\right)$	$\Theta\left(\frac{(N-2)!/(N-\log N)!}{N^{\log N}\log^2(N)}\right)$
Tree	$\Theta(\log N)$	$\Theta(\frac{1}{N^{\log(9)}})$	$\Theta\left(\frac{1}{N^{\log(9)}\log N}\right)$
Star	$\Theta(1)$	$\Theta\left(\frac{1}{N}\right)$	$\Theta\left(\frac{1}{N}\right)^{\star}$
Ring lattice + E-R random	$\Theta(\log N(\log N+w))$	$\Theta\left(\frac{(N-2)!}{(N+\frac{w}{\log N})^{\log N}(N-\log N)!}\right)$	$\Theta\left(\frac{(N-2)!/(N-\log N)!}{(N+\frac{w}{\log N})^{\log N}\log N(\log N+w)}\right)$
Ring lattice + Star (Longformer)	$\Theta(w)$	$\Theta\left(\frac{1}{Nw}\right)$	$\Theta\left(\frac{1}{Nw^2}\right)$
Ring lattice + Star + E-R random (BigBird)	$\Theta(\log N+w)$	$\Theta\left(\frac{1}{N(\log N+w)}\right)$	$\Theta\left(\frac{1}{N(\log N+w)^2}\right)$

Contents

- Sparsification of self-attention
- Normalized Information Payload(NIP)
- Hypercube Transformer
- Experiments

How to design sparse graph with high NIP

- Complete graph : the shortest distance between two neighbors' neighbors (excluding two nodes themselves) equals to zero, meaning that every two neighbor has the same neighbor.

- Unknown graph: the shortest distance between two neighbors' neighbors (excluding two nodes themselves) equals to one. Much sparser while maintaining the connectivity of the graph.

- This Unknown graph is Hypercube.

Normalized Information Payload (NIP)

Table 1. Normalized Information Payload for commonly used graphs, where w is the number of neighbors in ring lattice. $\star$: $\Theta\left(\frac{1}{N^2}\right)$ after refinement.

Type of graph	CC(G) $\downarrow$	IP(G) $\uparrow$	NIP(G) $\uparrow$
Complete	$\Theta(N)$	$\Theta\left(\frac{1}{N}\right)$	$\Theta\left(\frac{1}{N^2}\right)$
E-R random	$\Theta(\log^2 N)$	$\Theta\left(\frac{(N-2)!}{N^{\log N}(N-\log N)!}\right)$	$\Theta\left(\frac{(N-2)!/(N-\log N)!}{N^{\log N}\log^2(N)}\right)$
Tree	$\Theta(\log N)$	$\Theta(\frac{1}{N^{\log(9)}})$	$\Theta\left(\frac{1}{N^{\log(9)}\log N}\right)$
Star	$\Theta(1)$	$\Theta\left(\frac{1}{N}\right)$	$\Theta\left(\frac{1}{N}\right)^{\star}$
Ring lattice + E-R random	$\Theta(\log N(\log N + w))$	$\Theta\left(\frac{(N-2)!}{(N+\frac{w}{\log N})^{\log N}(N-\log N)!}\right)$	$\Theta\left(\frac{(N-2)!/(N-\log N)!}{(N+\frac{w}{\log N})^{\log N}\log N(\log N+w)}\right)$
Ring lattice + Star (Longformer)	$\Theta(w)$	$\Theta\left(\frac{1}{Nw}\right)$	$\Theta\left(\frac{1}{Nw^2}\right)$
Ring lattice + Star + E-R random (BigBird)	$\Theta(\log N + w)$	$\Theta\left(\frac{1}{N(\log N+w)}\right)$	$\Theta\left(\frac{1}{N(\log N+w)^2}\right)$
Hypercube	$\Theta(\log^2 N)$	$\Theta\left(\frac{(\log N)!}{(\log N)^{\log N}}\right)$	$\Theta\left(\frac{(\log N)!}{(\log N)^{\log N+2}}\right)$

Normalized Information Payload (NIP)

Table 1. Normalized Information Payload for commonly used graphs, where w is the number of neighbors in ring lattice. $\star$: $\Theta\left(\frac{1}{N^2}\right)$ after refinement.

Type of graph	NIP(G) ↑
Complete	$\Theta\left(\frac{1}{N^2}\right)$
E-R random	$\Theta\left(\frac{(N-2)!/(N-\log N)!}{N^{\log N}\log^2(N)}\right)$
Tree	$\Theta\left(\frac{1}{N^{\log(9)}\log N}\right)$
Star	$\Theta\left(\frac{1}{N}\right)^{\star}$
Ring lattice + E-R random	$\Theta\left(\frac{(N-2)!/(N-\log N)!}{(N+\frac{w}{\log N})^{\log N}\log N(\log N+w)}\right)$
Ring lattice + Star (Longformer)	$\Theta\left(\frac{1}{Nw^2}\right)$
Ring lattice + Star + E-R random (BigBird)	$\Theta\left(\frac{1}{N(\log N+w)^2}\right)$
Hypercube	$\Theta\left(\frac{(\log N)!}{(\log N)^{\log N+2}}\right)$

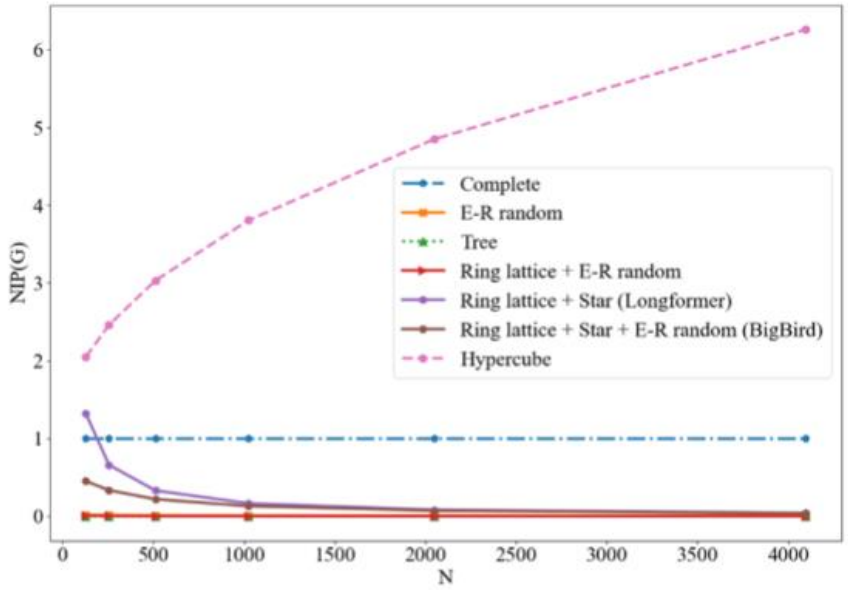


Figure 2. NIP(G) for graphs divided by complete graph in Table 1. We do not include star graph and ring lattice in this Figure because NIP(G) for star graph is too large. The w used for ring lattice is set to $\frac{N}{16}$ according to Longformer at length 4096.

Hypercube Transformer mapping

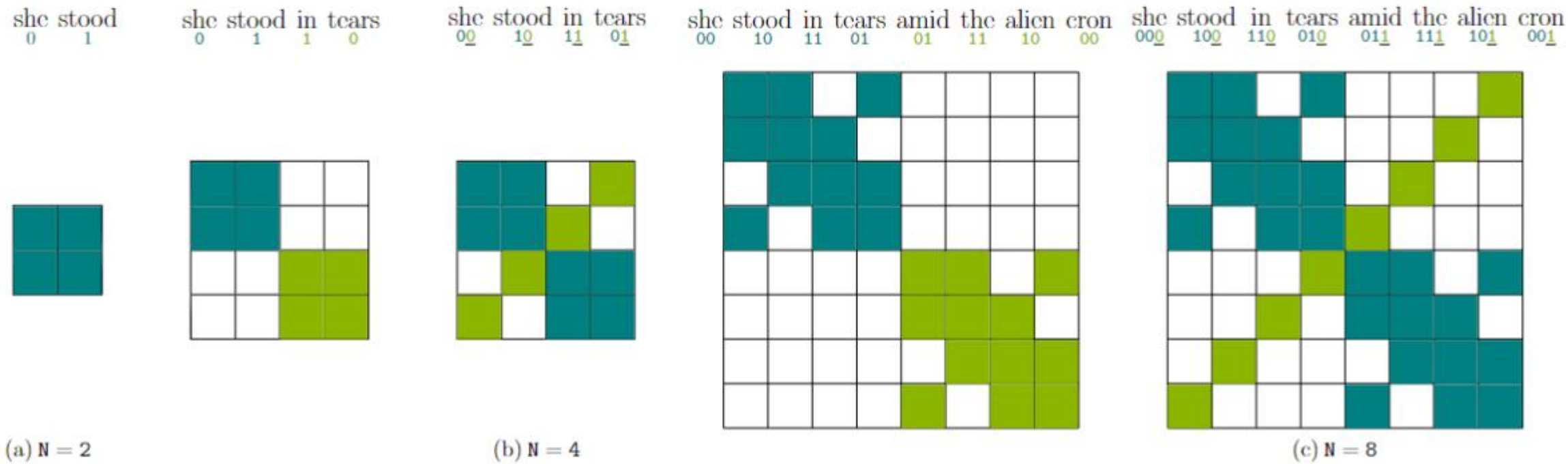


Figure 4. Iteratively mapping a sequence to a hypercube and its attention mask. Figure (a), (b) and (c) is the attention map for input sequences with length $N = 2, 4, 8$ respectively.

Contents

- Sparsification of self-attention
- Normalized Information Payload(NIP)
- Hypercube Transformer
- Experiments

Long-Range-Arena

Table 2. Performances for different graphs on Long-Range Arena. $\star$ means after refinement.

Graph #Length	**ListOps** 2K	**Text** 4K	**Retrieval** 4K	**Image** 1K	**Pathfinder** 1K	**Avg.**	**NIP**(G)	**SpeedUp**
Complete	37.20	63.54	81.00	47.23	**74.39**	60.67	$1\times$	$1\times$
Star	37.58	63.37	79.71	52.19	66.92	59.95	$1\times^{\star}$	-
Ring lattice + E-R random	36.44	**63.81**	80.17	50.88	67.14	59.69	$1.43e^{-10}\times$	-
Ring lattice + Star (Longformer)	37.55	61.12	80.53	52.13	68.66	60.00	$8.25e^{-2}\times$	-
Ring lattice + Star + E-R random (BigBird)	**37.80**	62.34	79.49	52.87	67.44	59.99	$7.21e^{-2}\times$	-
Hypercube	37.48	63.79	**81.16**	**53.79**	74.12	**62.07**	$4.85\times$	$15.8\times$

Long-Range-Arena

Table 2. l

Graph #Length		NIP(G)	SpeedUp
Complete	7	1×	1×
Star	5	1×⋆	-
Ring lattice + E-R rando	9	$1.43e^{-10}$×	-
Ring lattice + Star (Long	0	$8.25e^{-2}$×	-
Ring lattice + Star + E-R	9	$7.21e^{-2}$×	-
Hypercube	**7**	4.85×	15.8×

efinement.

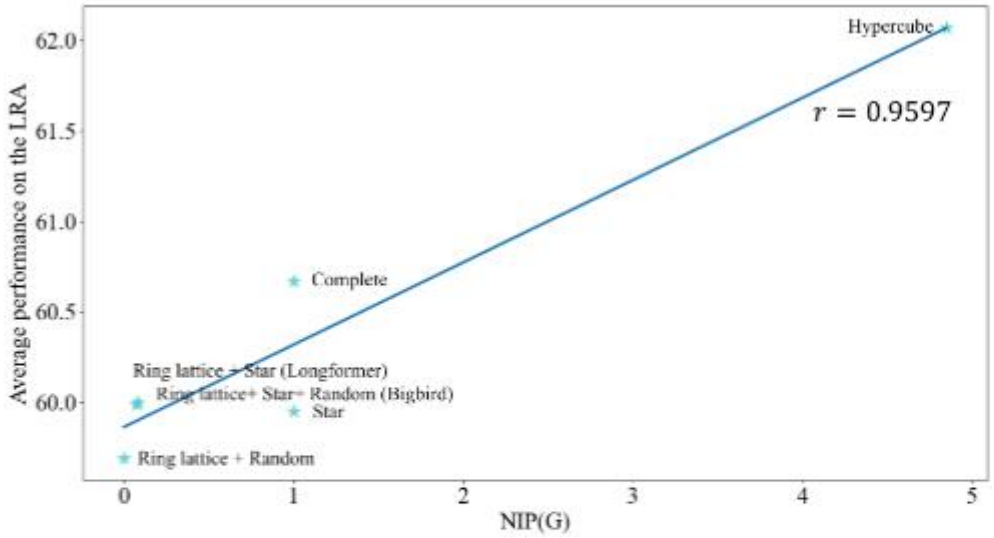


Figure 6. Average performance on the LRA benchmark can have strong proposition with our proposed Normalized Information Payload.

Block Sparsity

Theorem 3.1. *For block size* $b \leq \frac{N}{2}$*, larger block size makes star graph and hypercube have less Normalized Information Payload.*

Table 3. Performance of Hypercube Transformer with different block sizes.

Hypercube	Retrieval	Image
Block size 16	81.16	53.79
Block size 32	80.74	51.98
Block size 64	80.75	50.75

Large-scale pretraining

Table 5. Finetuning MLM on Wikitext103.

Model	Loss	Speedup
BERT_{128}	1.18	1×
CubeBERT_{128}	1.05	1.4×

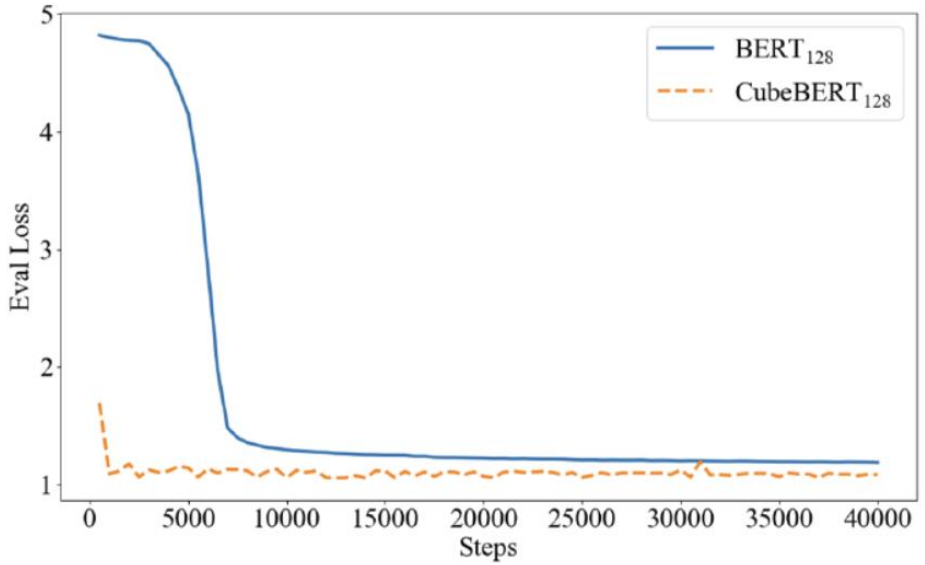


Figure 7. CubeBERT_{128} shows faster dropping rate of eval loss than BERT_{128} when finetuning on Wikitext103.

Large-scale pretraining

Table 5. Finetuning MLM on Wikitext103.

Model	Loss	Speedup

Table 6. Performances on GLUE test sets. For our implementation, results for RTE, STS and MRPC are reported by first finetuning on the MNLI model instead of the baseline pretrained model.

	MNLI-m/mm	QNLI	QQP	RTE	SST-2	MRPC	CoLA	STS-B	Avg.	Speedup
#metric	Acc	Acc	F1	Acc	Acc	F1	Matthew's corr.	Spearman corr.		
#Examples	393k	105k	364k	2.5k	67k	3.7k	8.5k	7k		
BERT	86.0/85.2	92.6	72.0	78.3	94.5	89.9	60.9	87.5	83.0	1×
BERT_{128}	84.9/84.8	91.1	71.0	76.6	93.1	90.4	58.0	88.3	82.0	1×
CubeBERT_{128}	85.9/85.0	90.8	71.3	77.1	**95.3**	86.4	**61.5**	**87.6**	82.3	1.1×

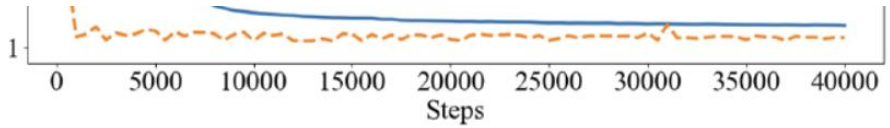


Figure 7. CubeBERT_{128} shows faster dropping rate of eval loss than BERT_{128} when finetuning on Wikitext103.

Open Questions

- How to quantify the Information Interference of one node? (Refinement of Star Graph)
- Expander Graph
  - Cheeger Inequalities to bound edge expansion. The lower bound is achieved for the hypercube (best NIP so far), the upper bound is achieved for a cycle (worst NIP so far).

$$\frac{1}{2}(d-\lambda_2) \leq h(G) \leq \sqrt{2d(d-\lambda_2)}.$$

- Attention weights change with training, making the distribution not uniform. How to model the distribution of attention weights?
  - However, from analysis of BERT, some attention heads, especially in lower layers, have very broad attention, which means the uniform distribution assumption reasonable somehow.

Thank you for listening!