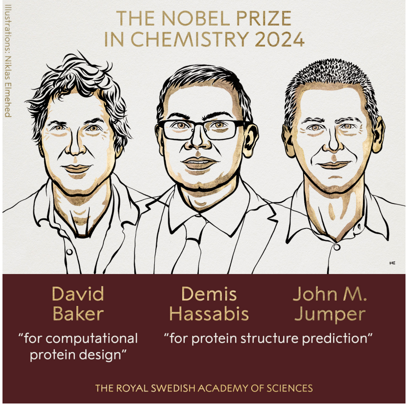
Graph Convolutions Enrich the Self-Attention in Transformers!

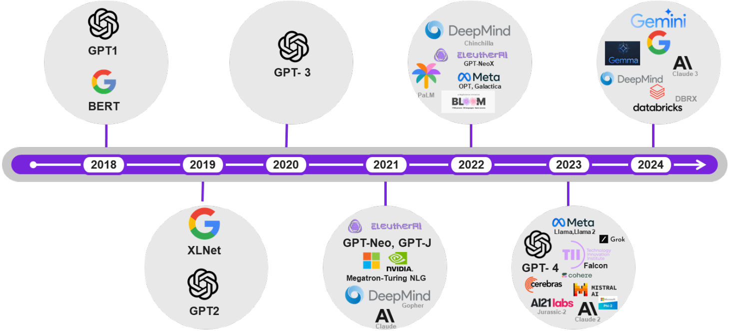
[1]**Jeongwhan Choi** [2]Hyowon Wi [2]Jayoung Kim [2]Yehjin Shin
[3]Kookjin Lee [4]Nathaniel Trask [2]Noseong Park

[1]Yonsei University
[2]Korea Advanced Institute of Science and Technology (KAIST)
[3]Arizona State University
[4]University of Pennsylvania

We Live in the AI Era, and Behind the Success of AI are Transformers!



https://x.com/NobelPrize/status/1843951197960777760

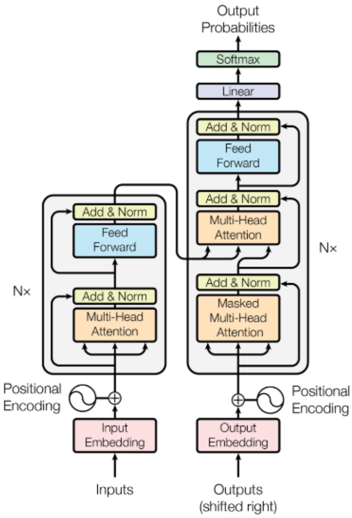


https://www.linkedin.com/pulse/beyond-proprietary-models-how-open-source-llms-ais-role-ramshetti-5wzuc/

Self-Attention is the Heart of Transformers

- The self-attention mechanism, denoted as SA, can be expressed as follows:

$$\mathrm{SA}(\boldsymbol{X}) = \mathrm{softmax}\left(\frac{\boldsymbol{X}\boldsymbol{W}_{\mathrm{key}}(\boldsymbol{X}\boldsymbol{W}_{\mathrm{qry}})^{\intercal}}{\sqrt{d}}\right)\boldsymbol{X}\boldsymbol{W}_{\mathrm{val}}$$
$$= \bar{\boldsymbol{A}}\boldsymbol{X}\boldsymbol{W}_{\mathrm{val}}.$$



Discrete signal processing (DSP)

- Signal $\boldsymbol{x}$ $\rightarrow$ Apply Filter $\boldsymbol{g}$ $\rightarrow$ Output $\boldsymbol{y}$

$$\boldsymbol{y}_i = \sum_{j=1}^{n} \boldsymbol{x}_j \boldsymbol{g}_{i-j}. \tag{1}$$

Graph signal processing (GSP) – generalization of DSP to graph domain

- The graph filter $\boldsymbol{H}$ can be written with a shift operator $\boldsymbol{S}$ (i.e., adjacency matrix $\boldsymbol{A}$):

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} = \sum_{k=0}^{K} w_k \boldsymbol{S}^k \boldsymbol{x}, \tag{2}$$

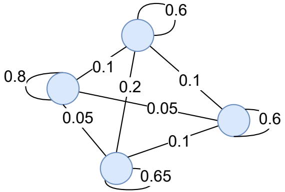
where K is the maximum order of polynomial, and w_k is a coefficient.

Self-Attention as a Graph Filter

- Self-attention matrix: $\bar{\boldsymbol{A}} = \boldsymbol{D}^{-1}\boldsymbol{A}$,
  - $\boldsymbol{A}$ is an adjacency matrix
  - $\boldsymbol{D}$ is a degree matrix.
- Self-attention can be considered as a simple graph filter ($\boldsymbol{H} = \bar{\boldsymbol{A}}$)

0.8	0.1	0.05	0.05
0.1	0.6	0.2	0.1
0.05	0.2	0.65	0.1
0.2	0.1	0.1	0.6

Self-attention $\bar{\boldsymbol{A}}$
(Probability score matrix)



Self-Attention as
a weighted graph

- Self-attention is a weighted graph
  - Nodes $\Leftrightarrow$ Tokens
  - Edge weights $\Leftrightarrow$ Attention scores

Oversmoothing Problem in Transformers

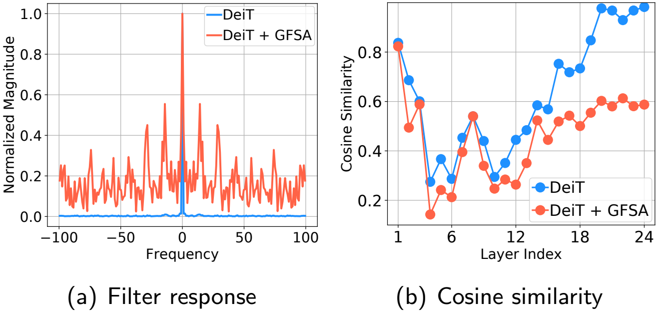


Figure: Filter frequency response and cosine similarity

- Oversmoothing problem:
  - As self-attention is a "low-pass filter", high-frequency information is attenuated.
  - Latent representations tend to become similar to each other.

Graph Filter-based Self-Attention (GFSA)

- We redesign self-attention from a graph signal processing perspective
- Our proposed GFSA is defined with the graph filter $\tilde{\boldsymbol{H}}_{\text{GFSA}}$:

$$\text{GFSA}(\boldsymbol{X}) := \tilde{\boldsymbol{H}}_{\text{GFSA}}\boldsymbol{X}\boldsymbol{W}_{\text{val}}, \tag{3}$$

$$\tilde{\boldsymbol{H}}_{\text{GFSA}} = w_0\boldsymbol{I} + w_1\bar{\boldsymbol{A}} + w_K\underbrace{(\bar{\boldsymbol{A}} + (K-1)(\bar{\boldsymbol{A}}^2 - \bar{\boldsymbol{A}}))}_{\simeq\bar{\boldsymbol{A}}^K}, \tag{4}$$

- We approximate $\bar{\boldsymbol{A}}^K$ with the first-order Taylor approximation:

$$\bar{\boldsymbol{A}}^K \simeq \bar{\boldsymbol{A}} + (K-1)(\bar{\boldsymbol{A}}^2 - \bar{\boldsymbol{A}}). \tag{5}$$

- GFSA learns the appropriate coefficients for downstream tasks, so it can be reduced to a low-pass-only, high-pass-only, or combined filter.

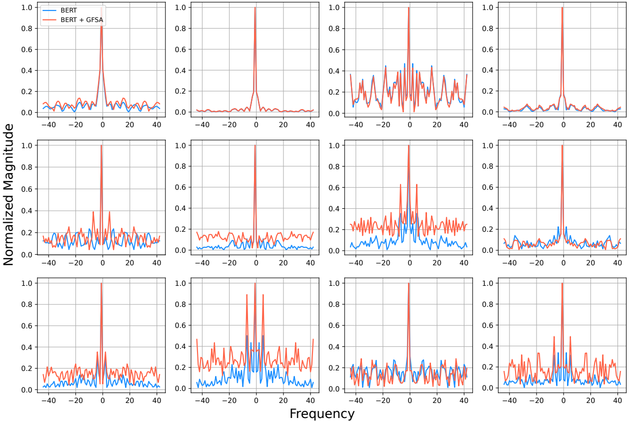


Figure: Visualization of the frequency responses for all 12 layers of BERT trained on STS-B dataset. The top-left figure corresponds to the first layer, and the bottom-right figure corresponds to the last layer.

GFSA Improves the Performance of Transformers!

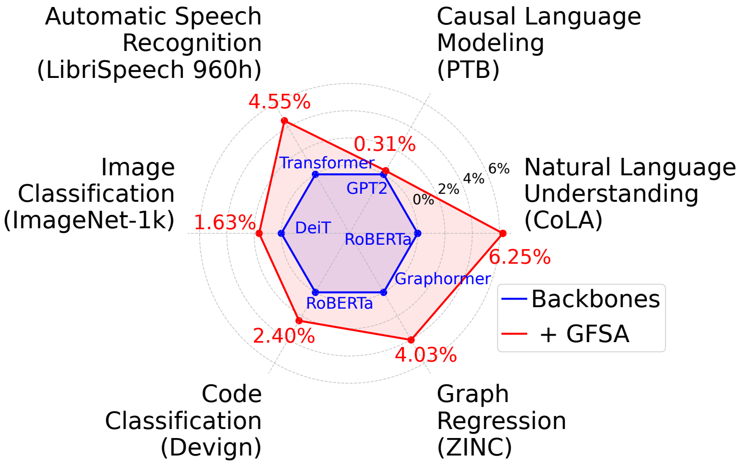


Figure: Performance improvements (%) of our GFSA when integrated with different Transformer backbones in various domains

GFSA with Efficient Design Strategies

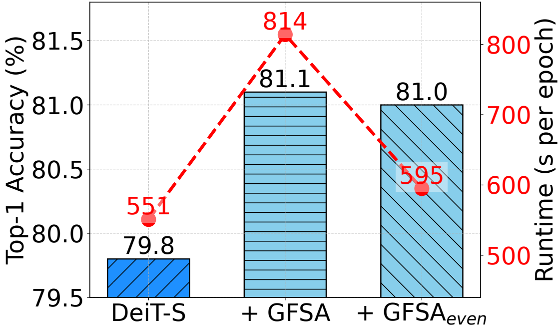


Figure: GFSA in selected layer. Effectiveness of our selective layer strategy on ImageNet-1k

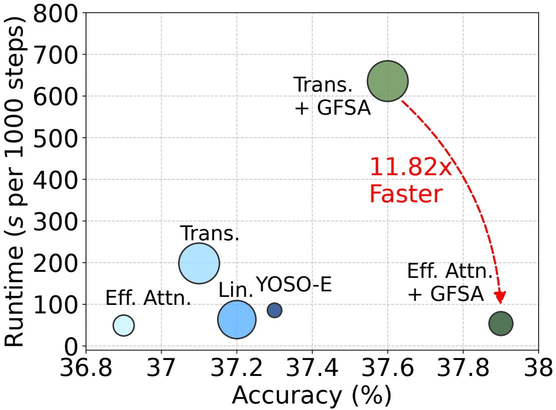
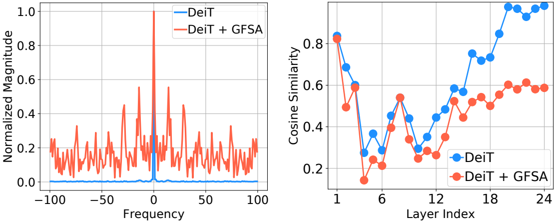
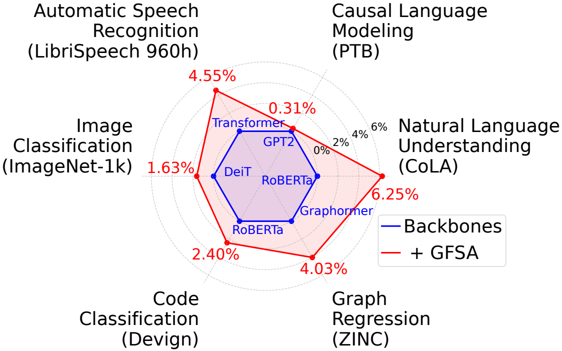


Figure: GFSA in linear Transformers. Performance, runtime, and GPU usage (circle sizes) of models on ListOps (2K) from Long Range Arena benchmark

Conclusion





- Considering the ongoing advancements in large language models, we hope that our approach may offer new insights for enhancing their performance and efficiency.

Thank You!

Email: jeongwhan.choi@yonsei.ac.kr

Paper

GitHub