Chopin's mother, ChatGPT and theoretical computer science

Tomasz Steifer

25.11.2024

Where it started

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

Jakob Uszkoreit* Google Research usz@google.com

Llion Jones* Google Research llion@google.com

Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

[PDF] **Attention is all you need**

A Vaswani - Advances in Neural Information Processing Systems, 2017 - user.phil.hhu.de

Attention is **all you need** **Attention** is **all you need** …

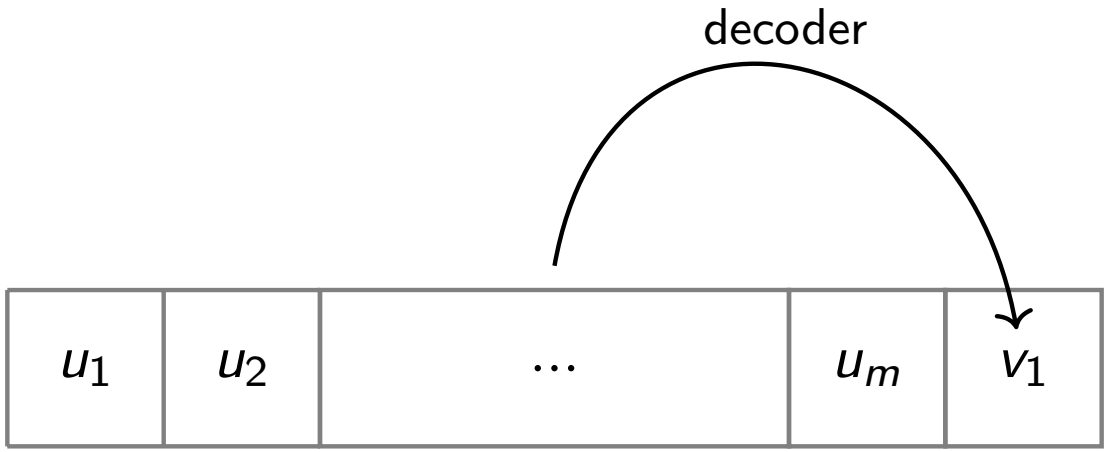
☆ Guardar 99 Citar Citado por 142356 Artículos relacionados »

General picture

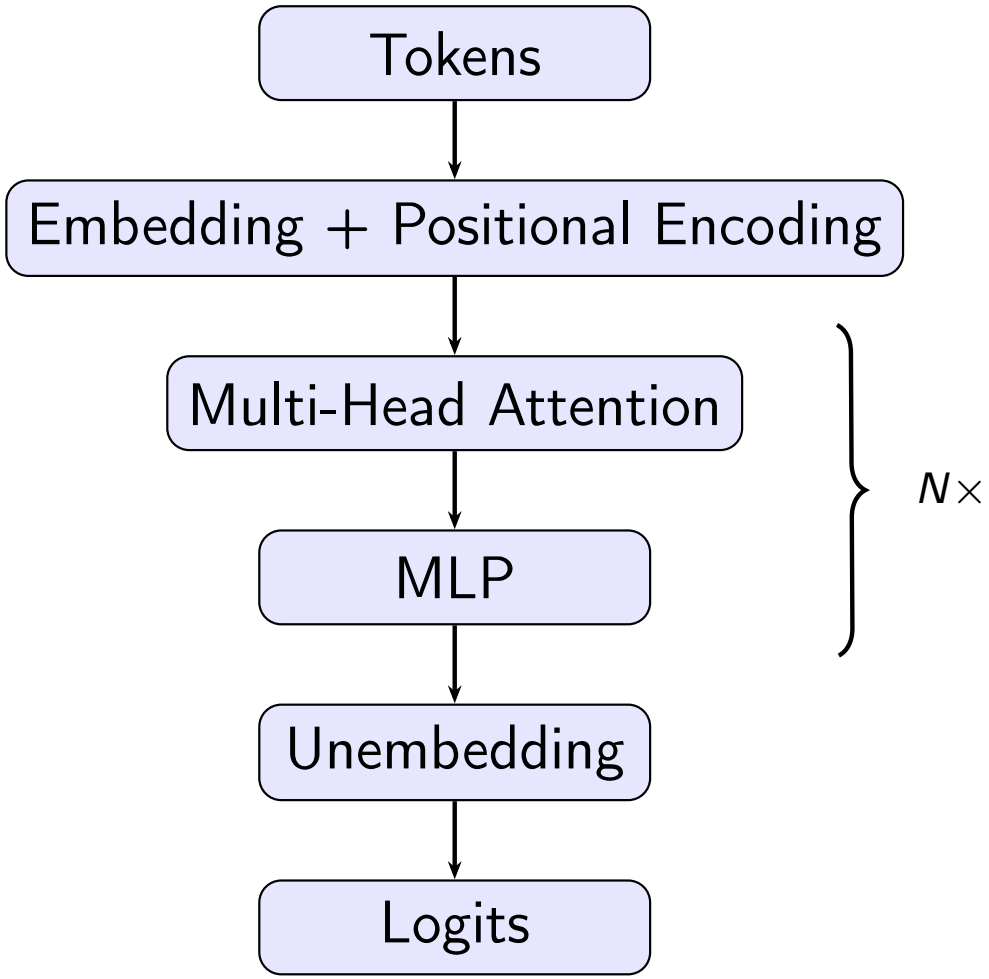
Input: text split into m tokens $u_1, u_2, \ldots, u_m$.

Output: new token v_1 is produced.

We can feed the extended sequence back to the model to get even more tokens!



Closer look


Tokens
Embedding + Positional Encoding
Multi-Head Attention
MLP
$N\times$
Unembedding
Logits

Self-attention

Definition (Standard attention)

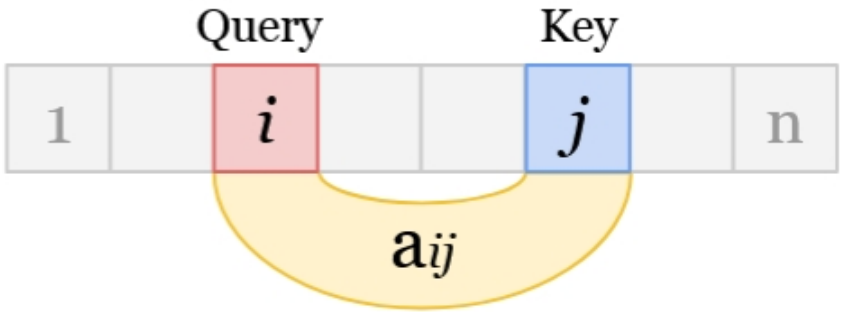
For a sequence of tokens $t = (t_1, ..., t_n)$, 1 head of attention layer produces

$$h(x)_i = \sum_{j=1}^{n} \alpha_{ij} v_j,$$

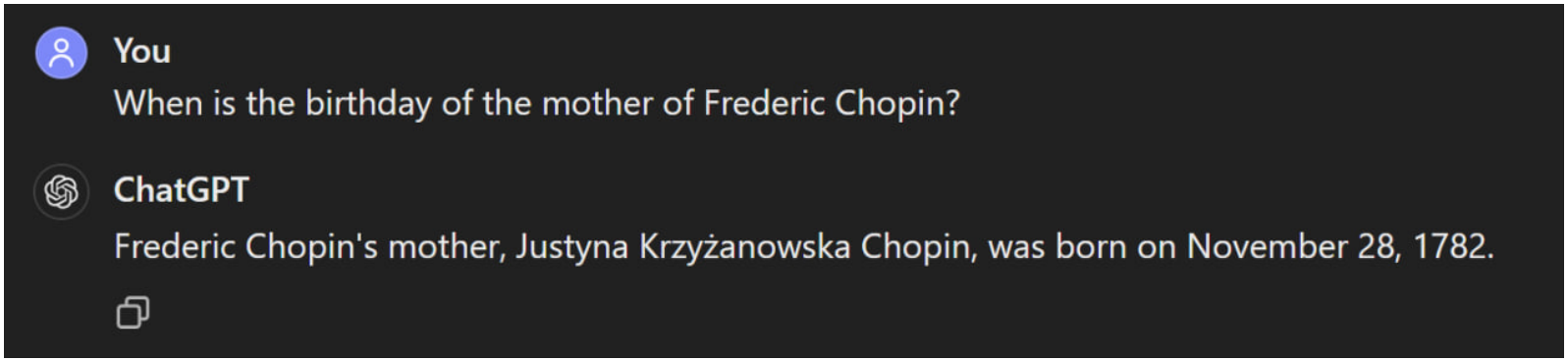
$$\alpha_{ij} = \text{softmax}\{q_i k_j / \sqrt{d} : j = 1, ..., n\},$$

$$q_i = W^Q x_i, k_j = W^K x_j, v_j = W^V x_j$$

in the first layer $x_i = e(t_i)$, e is a d-dimensional embedding layer.



Hallucinations



Tekla Justyna Chopin

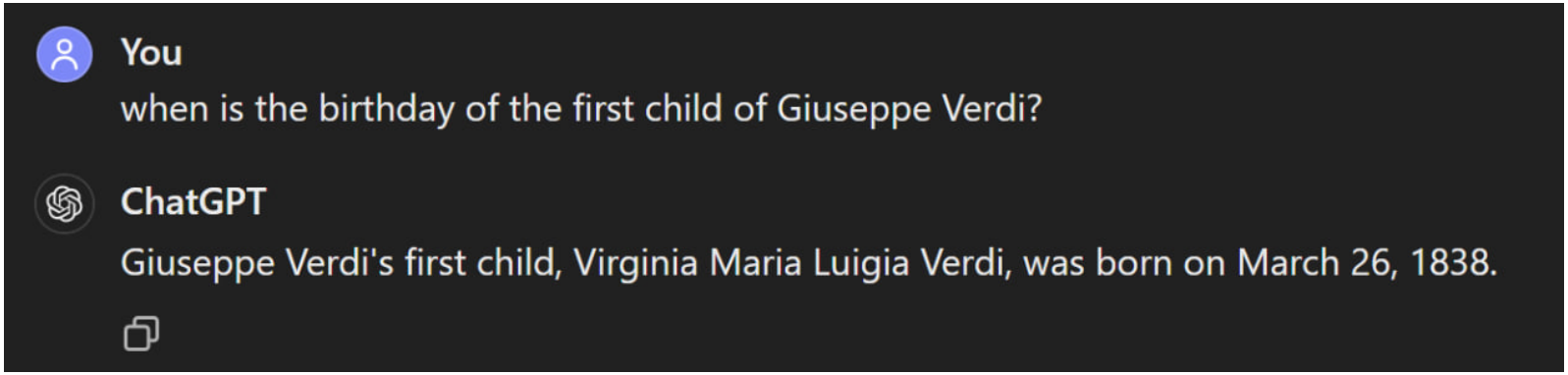
Article Talk

From Wikipedia, the free encyclopedia

Justyna Krzyżanowska ([juˈstɨ.na kʂɨ.ʐaˈnɔf.ska]; c. 14 September 1782 – 1 October 1861) was a Polish nanny, amateur musician, and pianist who became the first music teacher of her son, the pianist and composer Frédéric Chopin. She later ran a boarding house for students who attended the Warsaw Lyceum.

Observed by Guan et al. AAAI'24.

Hallucinations



1834–1842: First operas [edit]

Further information: List of compositions by Giuseppe Verdi

In mid-1834, Verdi sought to acquire Provesi's former post in Busseto but without success. But with Barezzi's help, he did obtain the secular post of *maestro di musica*. He taught, gave lessons, and conducted the Philharmonic for several months before returning to Milan in early 1835.[6] By the following July, he obtained his certification from Lavigna.[17] Eventually in 1835 Verdi became director of the Busseto school with a three-year contract. He married Margherita in May 1836, and by March 1837, she had given birth to their first child, Virginia Maria Luigia on 26 March 1837. Icilio Romano followed on 11 July 1838. Both the children died young, Virginia on 12 August 1838, Icilio on 22 October 1839.[1]

$$\mathrm{Birthday}(\mathrm{Mother}(\textit{Frederic Chopin}))?$$

$$\mathrm{Birthday}(\mathrm{FirstChild}(\textit{Guiseppe Verdi}))?$$

A hallucination

Query

```
Advisor of Alice is Charlie.
Advisor of Bob is Dave.
Advisor of Charlie is Bob.
Advisor of Dave is Bob.
Who is the advisor of the
advisor of the advisor of
Alice? Give me just the
name of this person!
```

gemini-1.5-flash

Bob

Figure: 3-fold function composition.

Advisor of Alice is Charlie. Advisor of Bob is Dave. Advisor of Charlie is Bob. Advisor of Dave is Bob. Who is the advisor of the advisor of the advisor of Alice? Give me just the name of this person!

Bob

Why these models give wrong answers?

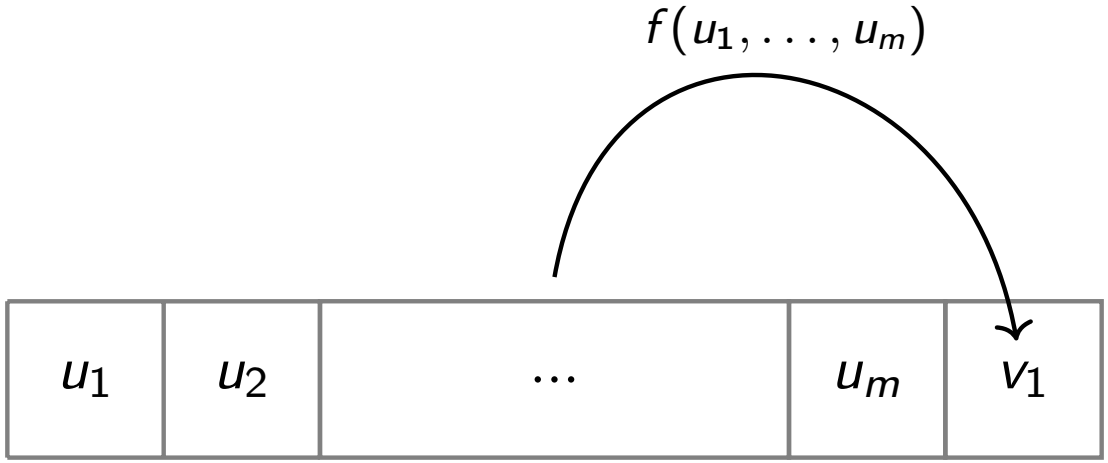
Need more data vs not expressive enough

Expressivity vs learnability

Expressivity: Boolean functions

Input: m bits $u_1, u_2, \ldots, u_m$.

Output: output bit v_1 .



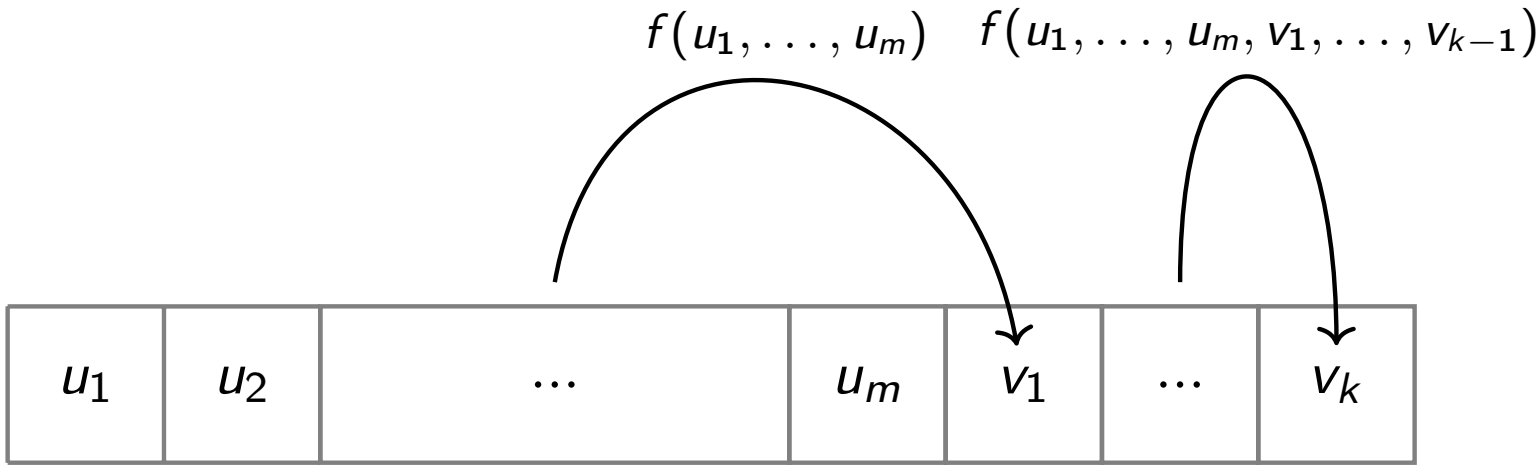
Question

What kind of functions can be computed by a transformer?
Given a language, i.e., set A of binary strings (e.g. PARITY) is there a transformer that decides membership of A?

Expressivity

Several flavors:

- What can be done in 1 iteration and constant model parameters?
- What can be done in many iterations? or: how many iterations we need to compute some function g?



Even more flavors

Standard self-attention uses softmax:

$$Softmax(\vec{x})_i = \frac{e^{x_i}}{\sum_j e^{x_j}}$$

Softmax is difficult to analyze. Some bounds obtained for alternatives:

- unique hard attention = leftmost argmax
- average hard attention = average of values with maximal attention

Hard vs soft attention

Theorem (Hahn 2020)

Transformers with unique hard attention cannot recognize PARITY and DYCK-1

Theorem (Chiang and Cholak 2022)

There is a transformer with soft attention which can recognize PARITY.

Moreover, Hahn's model has no layer normalization!
Bhattamishra et al. 2020 reported that transformer model couldn't generalize PARITY for arbitrary lengths. Similar in my own experiments for small transformer.

Attention is Turing-complete (?)

Theorem (Pérez, Marinković, Barceló. 2019, 2021)

For any Turing machine over alphabet Σ, there is an average hard attention transformer decoder T such that:

- *if $M(w)\downarrow = YES$, then there is k such that f on prompt w outputs YES after k intermediate steps.*
- *if $M(w)\downarrow = NO$, then there is k such that f on prompt w outputs NO after k intermediate steps*
- *if $M(w)\uparrow$, then there is no k such that f on prompt w outputs YES or NO after k intermediate steps.*

Some complexity theory

A language (set of binary strings) $\mathcal{L}$ is recognized by a family of circuits $C_1, C_2, \ldots$ if C_n recognizes the subset of L consisting of strings of length n.

Definition (TC^k)

TC^k is the class of languages recognizes by families of circuits with:

- unbounded fan-in
- $O(poly(n))$ size
- $O(\log(n)^k)$ depth
- MAJORITY gates.

Transformers are in TC^0 (?)

Theorem (Merrill and Sabharwal 2023)

Every language recognized by a transformer (without intermediate steps!) with $O(log(n))$-bit floating point precision is in DLOGTIME-uniform TC^0.

Cannot solve: word problem for permutation groups (if $TC^0 \neq NC^1$, circuits with fan-in 2 and log(n) depth), computing permanent of a matrix.
What about intermediate steps?

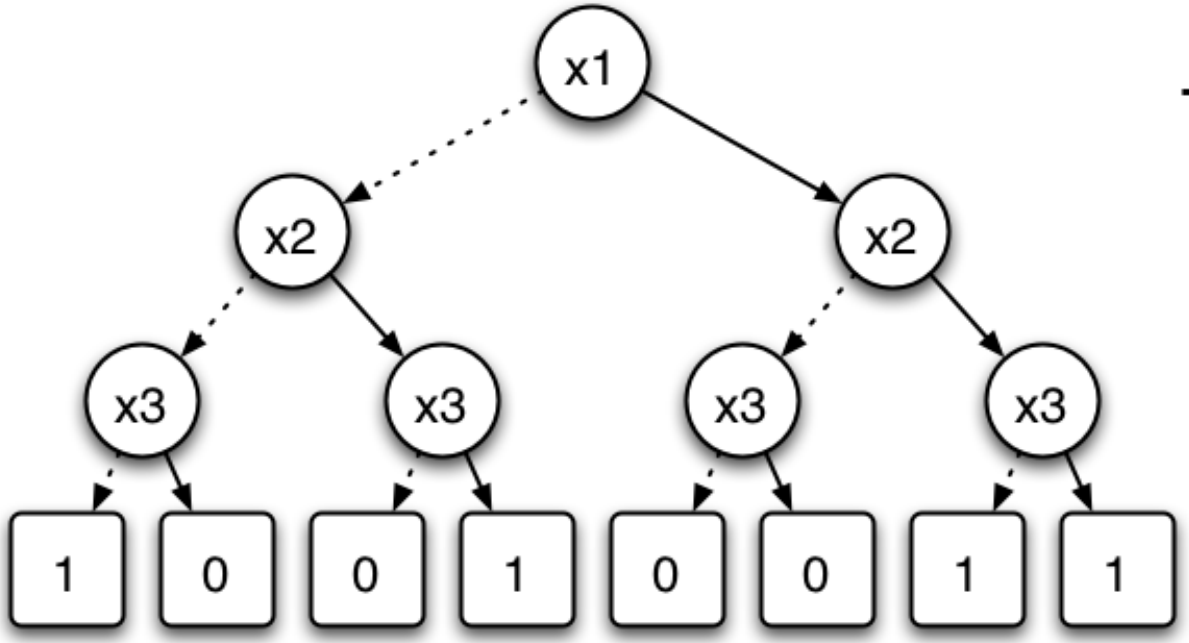
Theorem (Li-Liu-Zhou-Ma 2024)

For any polynomial $T(n)$, transformers, with constant-depth, constant precision, $O(log(n))$ embedding dimension and with $T(n)$ intermediate steps, can recognize languages which are recognized by families of Boolean circuits of size $O(T(n))$.

Corollary

Every regular language can be solved by a transformer with constant-depth, constant precision and with n intermediate steps.

What about 1-layer transformers?



x1	x2	x3	*f*
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Figure: Binary decision tree (via Wikipedia)

Ehrefeucht-Haussler rank

For a rooted binary tree T , the rank of the tree is the rank of the root node, where the rank of each node of the tree is defined recursively as follows: For a leaf node u, $Rank(u) = 0$. For an internal node u with children v, w,

$$Rank(u) = \begin{cases} Rank(v) + 1, & \text{if } Rank(v) = Rank(u). \\ \max(Rank(v), Rank(u)), & \text{otherwise.} \end{cases}$$

Theorem (Barcelo, Kozachinskiy, S. 2024)

A function f can be computed in r iterations of a 1-layer transformer with 1 unique hard attention head if and only if it can be computed by a decision tree of rank r.

We can define a multi-head generalization of the tree rank to obtain:

Theorem (Barcelo, Kozachinskiy, S. 2024)

For any H, r, a function f can be computed in r iteration of a 1-layer decoder with H unique hard attention heads if and only if f can be computed by a rank-r head-H decision tree.

Iterated composition of functions

$k - Comp$ problem:
Input: $f(1), \ldots, f(n)$
Output: $\underbrace{f(f(\ldots,}_{k}(1)\ldots)$

Lemma (Barcelo, Kozachinskiy, S. 2024)

For any H, for all large enough n, we have $rank(H)(k - Comp) = t$.

Corollary

For any number of heads H, 1 layer transformer with unique hard attention needs k intermediate steps to solve k-fold composition of functions.

What about softmax? More layers?

Some empirical results

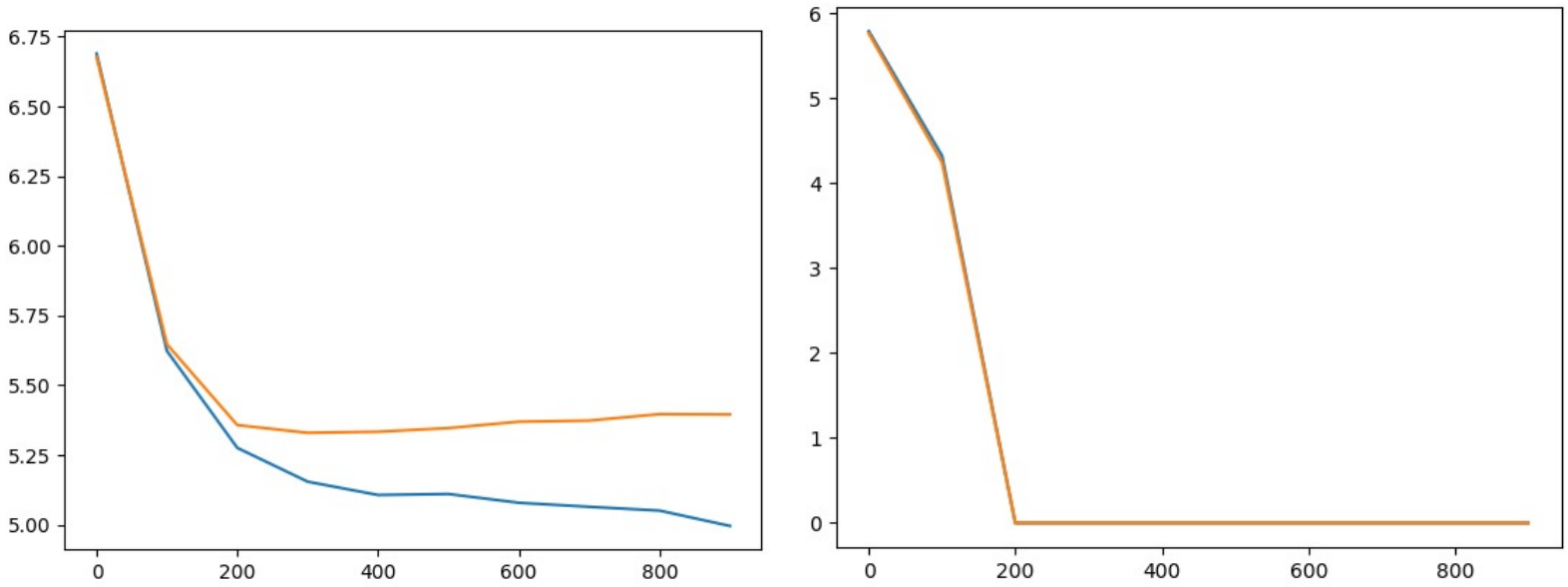


Figure: 2-head 2 iterations vs 1-head 3 iterations for 3-Comp, $n = 20$. Mean L_1-loss vs number of epochs, $d = 20$

Different setting

Transformers (or more broadly attention mechanism) can be applied to non-text data:

- images (Vision transformer),
- time series,
- protein structure (AlphaFold3 - Nobel price in chemistry this year!)
- etc.

Graphs

Composition of relations.
Input: Adjacency matrices of two graphs G_1, G_2
Output: Adjacency matrix of G_3 with $(a, b) \in G_3$ iff there exists $c : (a, c) \in G_1, (c, b) \in G_2$.

Theorem (Buc et al (S.∈al)

Fix precision p, d embedding dimension and H number of heads. If a 1 *layer transformer with soft attention solves composition of graphs with n nodes, then:*

$$(d + 3)pH \geq (n - 2)$$

Generalized attention

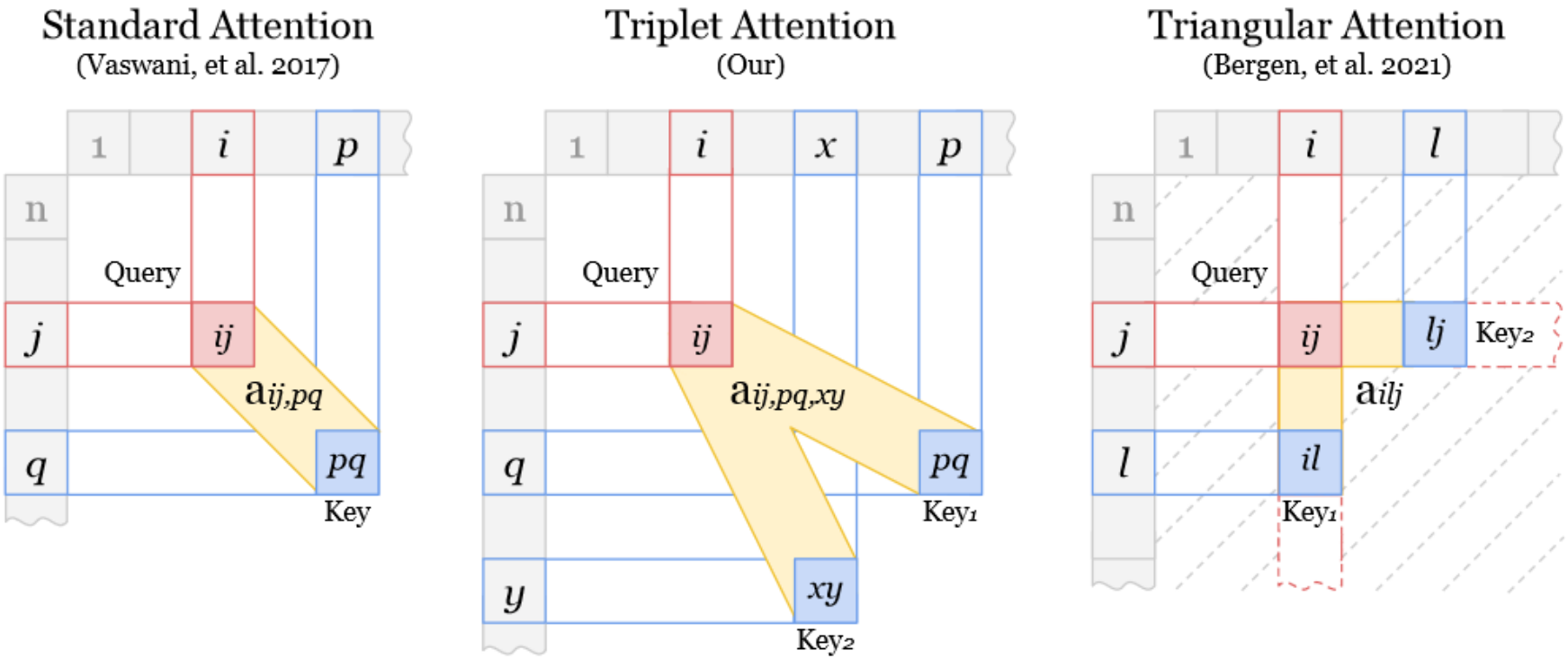


Figure: Generalized attention

Theorem

Both triplet and triangular attention can solve composition of relations with one layer and constant parameters.

Empirical verification: in progress (ICML 2025?)

Future directions

Some open problems:

- Expressivity vs empirical learnability
- Can softmax simulate unique hard attention?
- What attention patterns are learned in real models?
- Decision tree rank for multiple layers?
- New applications for generalized attention mechanisms (FENG First-Team proposal submitted!)

Google, Stanford, Microsoft Research, Columbia, New York University etc.

Can IPPT play a role in this?

Thank you for your attention!