The Power of Hard Attention Transformers on Data Sequences: A Formal Language Theoretic Perspective

38th Annual Conference on Neural Information Processing Systems, Vancouver

Pascal Bergsträßer[1] Chris Köcher[2] Anthony Widjaja Lin[1,2] Georg Zetzsche[2]

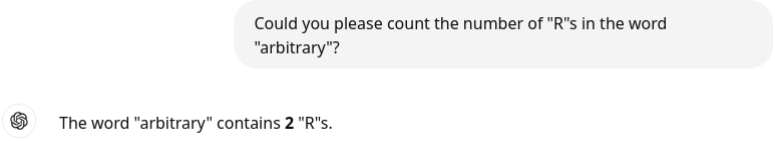
[1] Rheinland-Pfälzische Technische Universität, Kaiserslautern-Landau
[2] Max Planck Institute for Software Systems, Kaiserslautern

Motivation

- Transformers are the basic model in machine learning used in recent LLMs

ChatGPT Gemini Meta AI Apple Intelligence

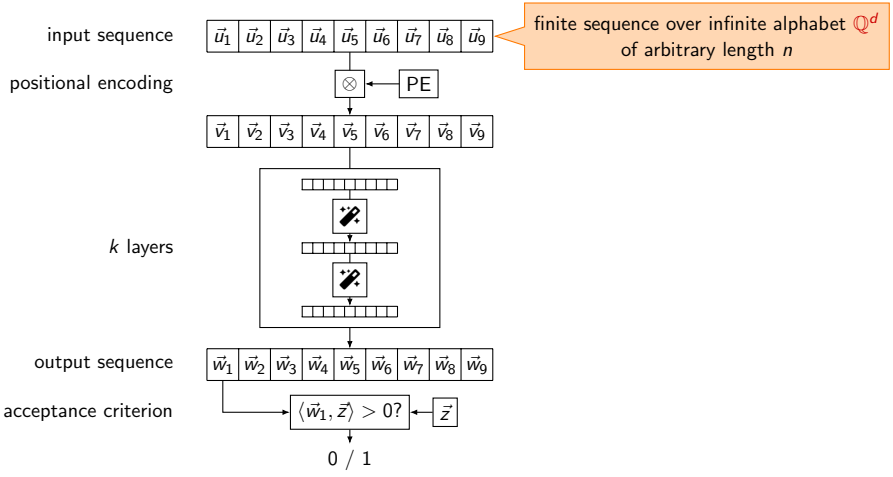
- Sometimes their output is just wrong:



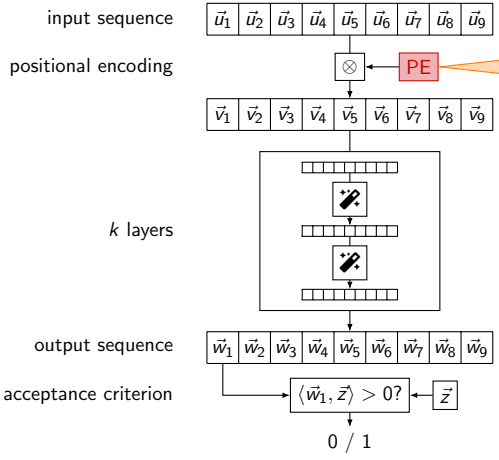
(Spoiler: "arbitrary" contains **three** "R"s)

- Here: we analyze expressibility of transformers that take complex data (like pictures, videos, voice, time series, ...) as input

Unique Hard Attention Transformers (UHAT) on Data Sequences



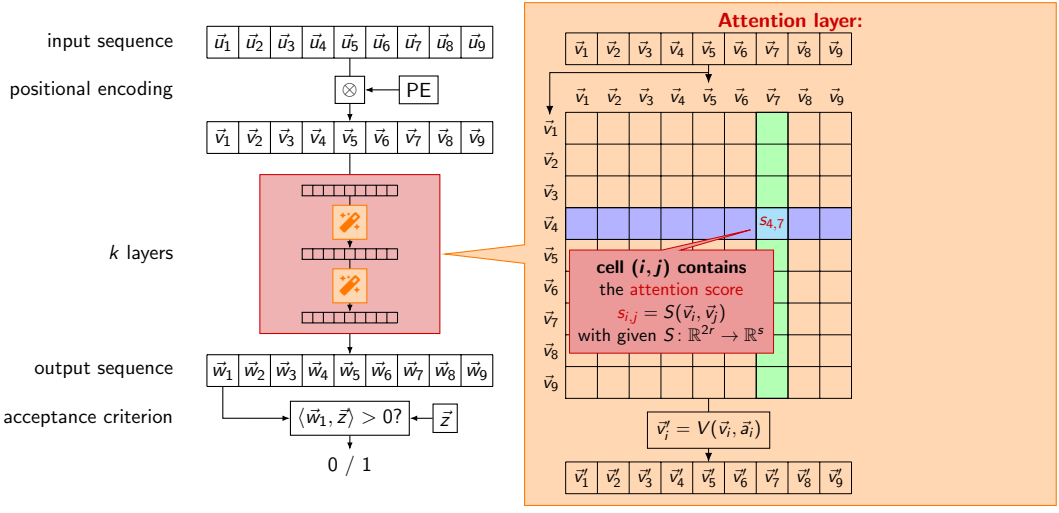
Unique Hard Attention Transformers (UHAT) on Data Sequences



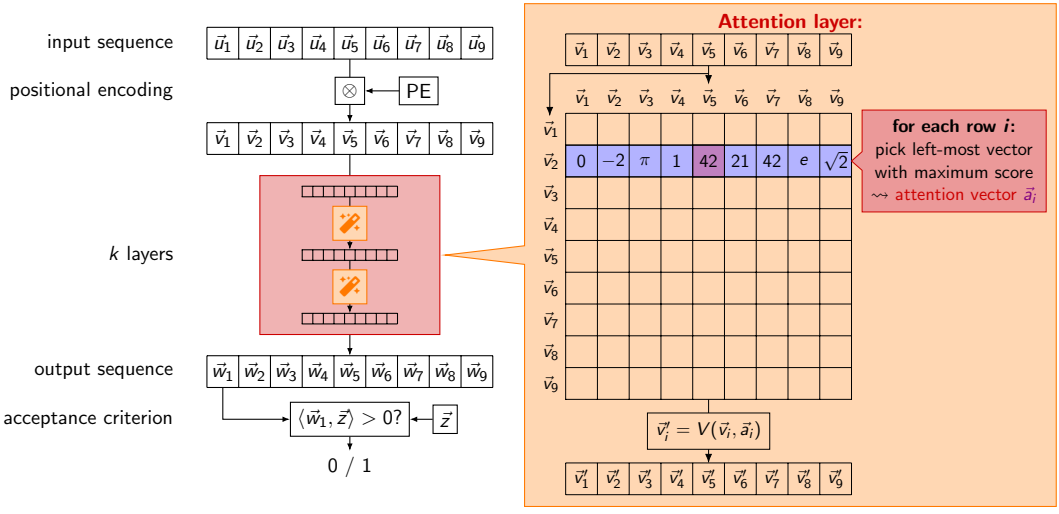
given: arbitrary function $\mathrm{PE}\colon \mathbb{N} \times \mathbb{N} \to \mathbb{R}^e$
e.g., $0, i, \frac{1}{i}, \frac{i}{n}, \sin i, \ldots$

compute for each position $1 \leq i \leq n$:
$\vec{v}_i = \begin{pmatrix} \vec{u}_i \\ \mathrm{PE}(i, n) \end{pmatrix} \in \mathbb{R}^{d+e}$

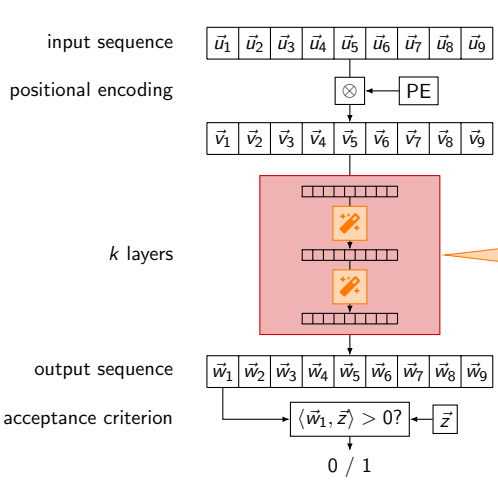
Unique Hard Attention Transformers (UHAT) on Data Sequences

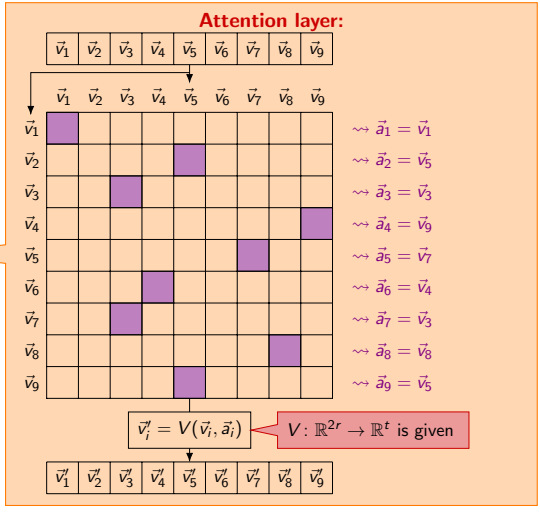


Unique Hard Attention Transformers (UHAT) on Data Sequences

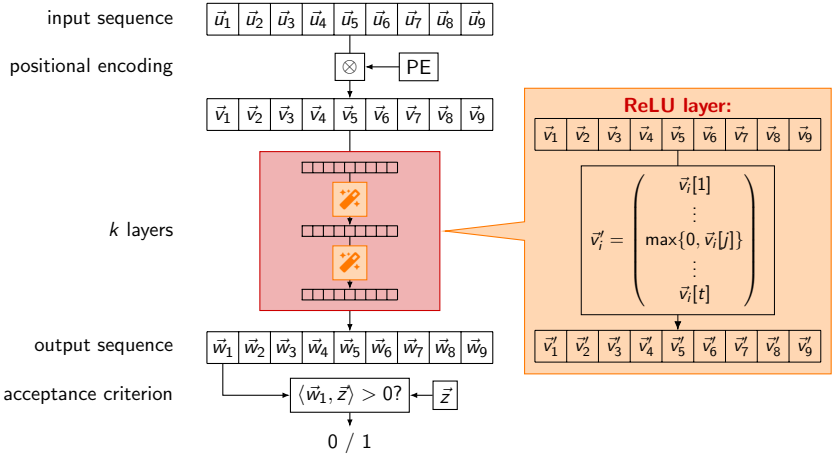


Unique Hard Attention Transformers (UHAT) on Data Sequences

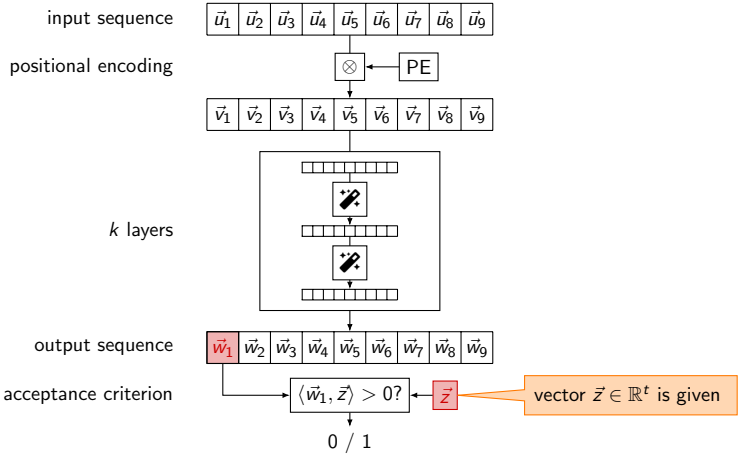




Unique Hard Attention Transformers (UHAT) on Data Sequences



Unique Hard Attention Transformers (UHAT) on Data Sequences



Word vs. Data Languages

	UHATs with finite alphabet	UHATs with infinite alphabet
Circuit complexity	$\subseteq \mathsf{AC}^0$ [Hao et al. 2022] $\neq \mathsf{AC}^0$ [Barceló et al. 2023]	$\subseteq \mathsf{TC}^0$, $\not\subseteq \mathsf{AC}^0$ *
Regularity	$=$ Star-Free * [Angluin et al. 2024]	$\not\subseteq$ Regular *
Logic	$=$ LTL * [Angluin et al. 2024] $\supseteq$ LTL(Mon) [Barceló et al. 2023]	$\supseteq$ LTLTL

Read the paper:

Thank you!

* without positional encoding, but with masking

Word vs. Data Languages

	UHATs with finite alphabet	UHATs with infinite alphabet
Circuit complexity	$\subseteq \mathsf{AC}^0$ [Hao et al. 2022] $\neq \mathsf{AC}^0$ [Barceló et al. 2023]	$\subseteq \mathsf{TC}^0$, $\not\subseteq \mathsf{AC}^0$ *
Regularity	$=$ Star-free * [Angluin et al. 2024]	$\not\subseteq$ Regular * $\supseteq$ LTLTL

Languages accepted by a family of circuits of

- constant depth,
- polynomial size, and
- Boolean gates with unbounded fan-in.

Thank you!

Read the paper:

* without positional encoding, but with masking

Word vs. Data Languages

	UHATs with finite alphabet	UHATs with infinite alphabet
Circuit complexity	$\subseteq \mathsf{AC}^0$ [Hao et al. 2022] $\neq \mathsf{AC}^0$ [Barceló et al. 2023]	$\subseteq \mathsf{TC}^0$, $\not\subseteq \mathsf{AC}^0$ *
Regularity	$=$ Star-Free * [Angluin et al. 2024]	AC^0 + majority gates
Logic	$=$ LTL * [Angluin et al. 2024] $\supseteq$ LTL(Mon) [Barceló et al. 2023]	$\supseteq$ LTLTL

Read the paper:

Thank you!

* without positional encoding, but with masking

Word vs. Data Languages

	UHATs with finite alphabet	UHATs with infinite alphabet
Circuit complexity	$\subseteq \mathsf{AC}^0$ [Hao et al. 2022] $\neq \mathsf{AC}^0$ [Barceló et al. 2023]	$\subseteq \mathsf{TC}^0$, $\not\subseteq \mathsf{AC}^0$ *
Regular	LTL + monadic predicates Θ et al. 2024]	$\not\subseteq$ Regular *
Logic	$=$ L / L * [Angluin et al. 2024] $\supseteq$ LTL(Mon) [Barceló et al. 2023]	$\supseteq$ LTLTL

Read the paper:

Thank you!

* without positional encoding, but with masking

Word vs. Data Languages

	UHATs with finite alphabet	UHATs with infinite alphabet
Circuit complexity	$\subseteq \mathsf{AC}^0$ [Hao et al. 2022] $\neq \mathsf{AC}^0$ [Barceló et al. 2023]	$\subseteq \mathsf{TC}^0$, $\not\subseteq \mathsf{AC}^0$ *
Regularity	= Sta…	
Logic	$= \mathsf{LTL}$ * [Angluin et al. 2024] $\supseteq \mathsf{LTL(Mon)}$ [Barceló et al. 2023]	$\supseteq$ LTLTL

Locally Testable LTL = LTL(Mon) + arithmetic constraints

Read the paper:

Thank you!

* without positional encoding, but with masking