

EASTERN
MICHIGAN UNIVERSITY

Deep Learning for Investigating Theories of Language

Shane Storks, PhD

Assistant Professor of Computer Science @ EMU

EMU Machine Learning Conference, October 31, 2026

Deep Learning 🤝 Linguistics

If you're here, you're probably aware of the powerful AI chatbots we have nowadays…

Large language models (LLMs):

- Input a sequence of **word/token embeddings** into a **transformer** (deep neural network based on attention and multilayer perceptron)
- Train the model to **predict the next word/token**
  - More specifically, its index in a large vocabulary
  - Do this across a very large dataset – virtually all the text on the web
- **Result:** apparently humanlike language processing and generation!



Natural question: *can we use LLMs to understand how human language works, or vice versa?*

My work toward this question

Before joining EMU, I was a postdoc at UMich's Weinberg Institute for Cognitive Science.

There, I collaborated with linguists and psychologists in this area. Some findings:

1. Chang, Huang, Nasir, Storks, Akingbade, & Dai. (2025). **Mind the Gap: How BabyLMs Learn Filler-Gap Dependencies**. *EMNLP 2025.*
2. Gilberti, Storks, & Dai. (2026). **Discovering Properties of Inflectional Morphology in Neural Emergent Communication.** *ACL 2026.*
3. McInnerney*, Storks*, Abney, & Lewis. (2026). **Information Antilocality and the Locality Bias in LLMs.** *Decision pending.*

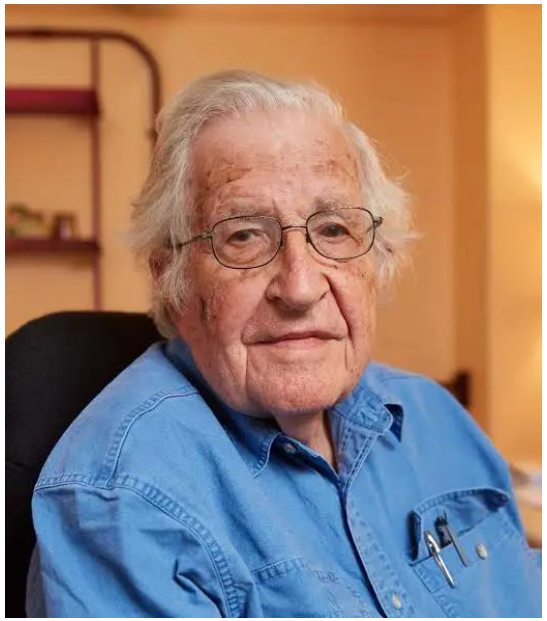
* *indicates equal contribution.*

Disclaimer: I am not a linguist!

(In)famous question for linguists

- ***Is any part of our language learning capability innate?***
  - Or do we only learn language from data?
    (hearing language and associating our surroundings with it)

- **Chomsky (1965):**
  Human language acquisition is constrained by a set of **innate** learning biases! *(universal grammar)*
  - **Locality Principle:** Human brains best process language by linking words that are close together (structurally), so our languages are biased toward information being locally clustered.



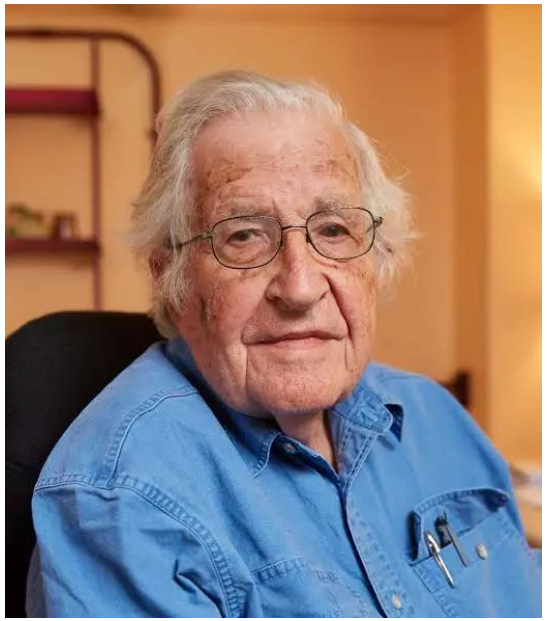
Noam Chomsky

Noam Chomsky. 1965. *Aspects of the Theory of Syntax.* MIT Press, Cambridge, MA.

(In)famous question for linguists

- In contrast, LLMs are almost entirely **data-driven**.
  - Only inductive bias comes from their **transformer** architecture and next-word/token prediction formulation.
  - This threatens Chomsky and co.'s linguistic theories…
  - It shouldn't be possible to learn humanlike language capabilities without humanlike learning biases!

- Their response: LLMs could learn any languages, even **impossible languages**, so they don't have humanlike learning bias.
  - **Chomsky et al. (2023):** "incapable of distinguishing the *possible from the impossible*"
  - **Moro et al. (2023):** "the distinction between *possible versus impossible languages* cannot be formulated by definition for LLM"



Noam Chomsky

Noam Chomsky. 1965. *Aspects of the Theory of Syntax.* MIT Press, Cambridge, MA.

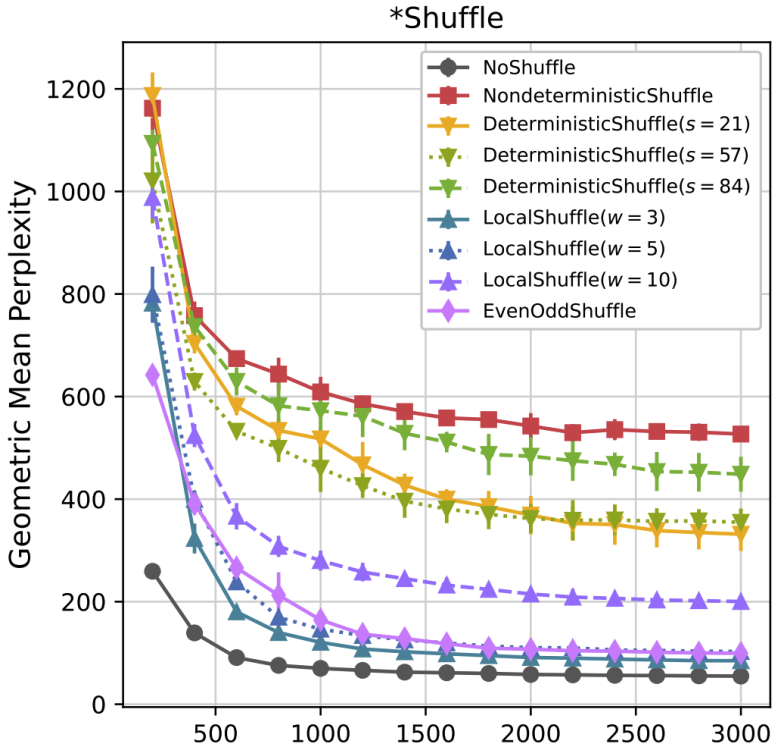
So LLMs have a bias toward locality like humans?

Impossible languages?

Hold on a minute!

- **Kallini et al. (2024):** created impossible languages
  - Take a dataset of natural language sentences
  - Systematically perturb the locality of information by shuffling tokens (and other transformations)
- **Finding:** transformer-based LLMs struggle to learn impossible languages in terms of both **learning success** (converged perplexity) and **learning speed** (time for model to converge)

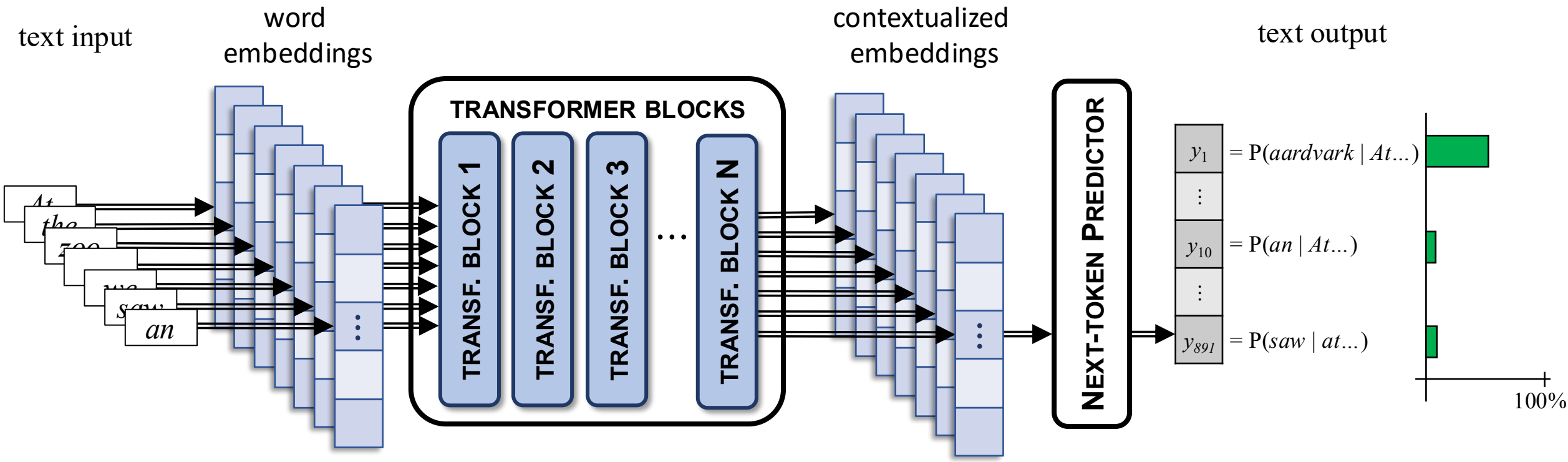
Class	Language	Example 1	Example 2
*Shuffle	NoShuffle	He cleans his very messy books he lf .	They clean his very messy books he lf .
	NondeterministicShuffle	messy books his he very . lf He cleans	his . very he They messy lf books clean
	DeterministicShuffle($s = 21$)	cleans He messy books he lf very . his	clean They messy books he lf very . his
	DeterministicShuffle($s = 57$)	cleans his He messy . he very lf books	clean his They messy . he very lf books
	DeterministicShuffle($s = 84$)	He messy . lf his very books cleans he	They messy . lf his very books clean he
	LocalShuffle($w = 3$)	his He cleans books very messy . he lf	his They clean books very messy . he lf
	LocalShuffle($w = 5$)	his messy very He cleans lf books he .	his messy very They clean lf books he .
	LocalShuffle($w = 10$)	messy books his he very . lf He cleans	messy books his he very . lf They clean
	EvenOddShuffle	He his messy he . cleans very books lf	They his messy he . clean very books lf



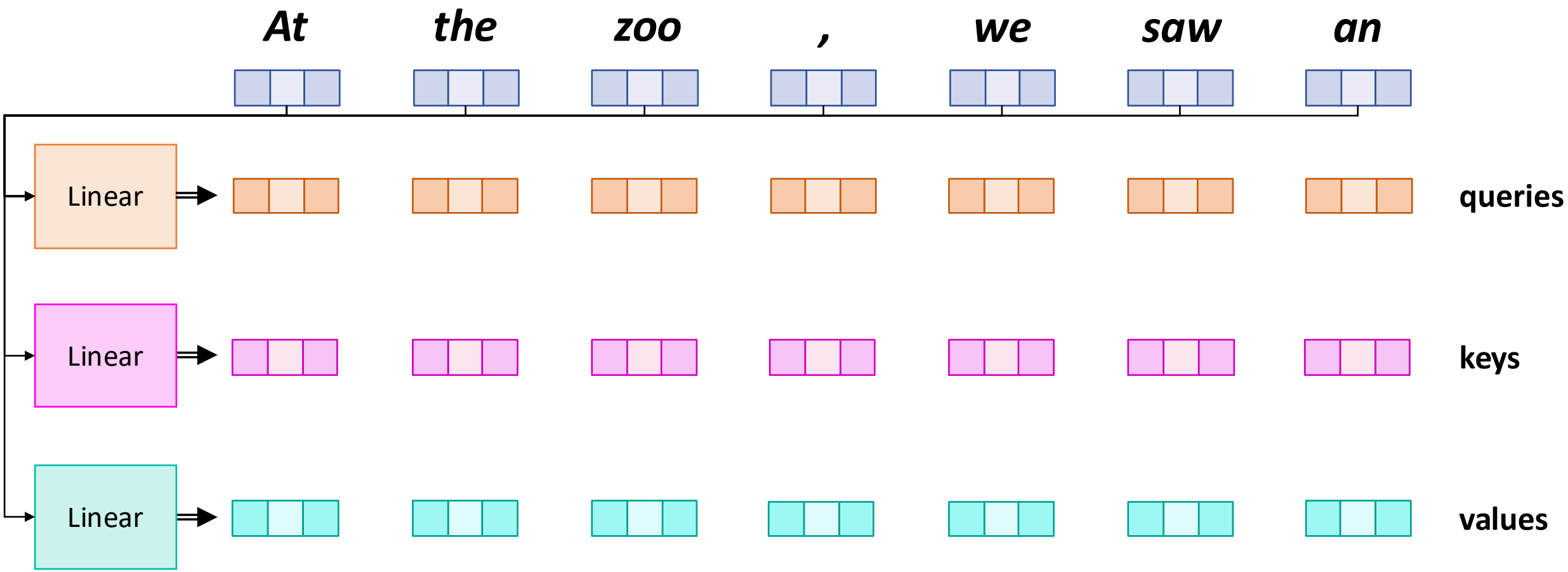
Chomsky, Roberts, and Watumull. 2023. Noam Chomsky: The false promise of ChatGPT. *The New York Times.*
Moro, Greco, and Cappa. 2023. Large languages, impossible languages and human brains. *Cortex*, 167:82-85.
Kallini, Papadimitriou, Futrell, Mahowald, and Potts. 2024. Mission: Impossible Language Models. *ACL 2024.*

Transformer-based LLMs

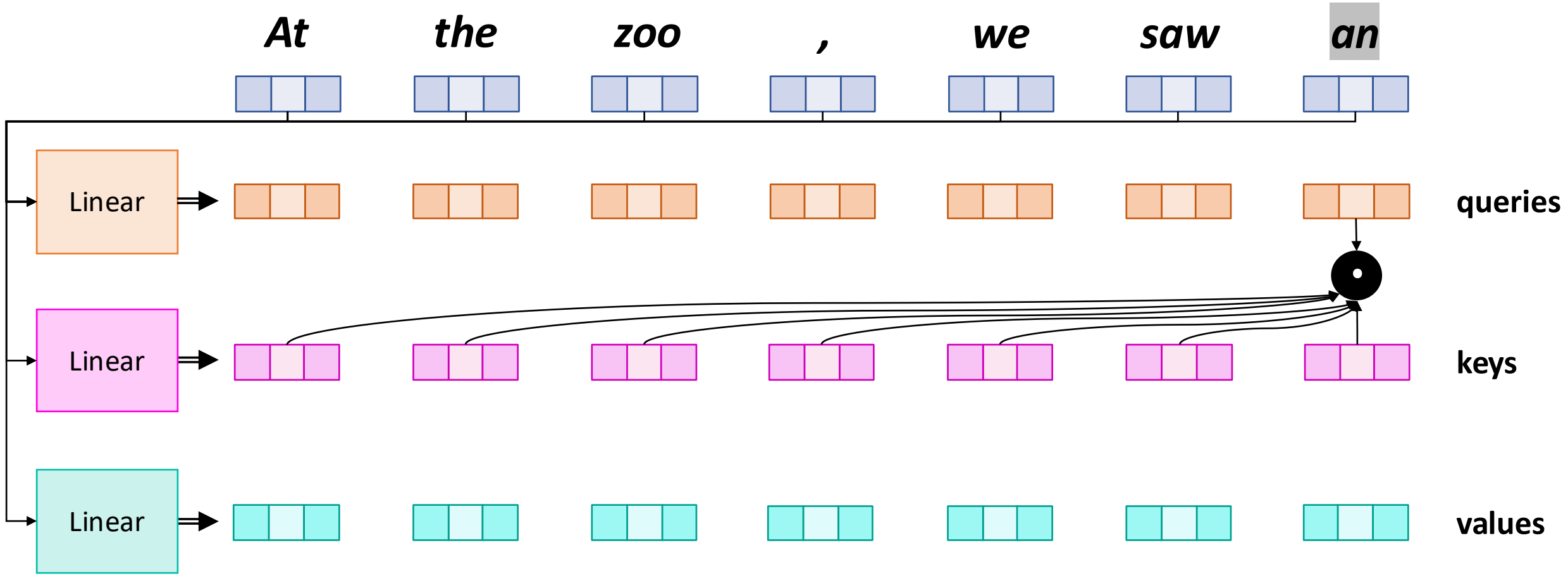
At the zoo, we saw an…



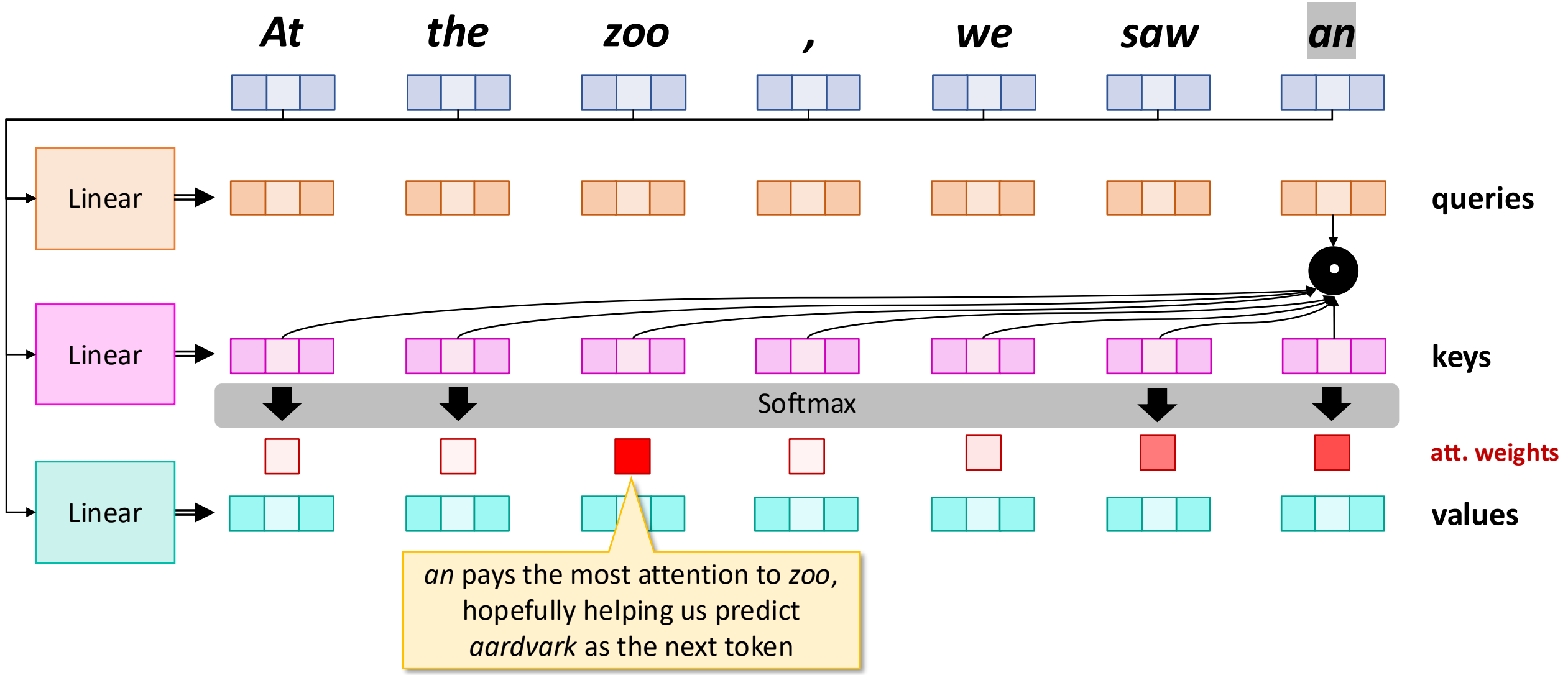
Self-attention in transformer-based LLMs



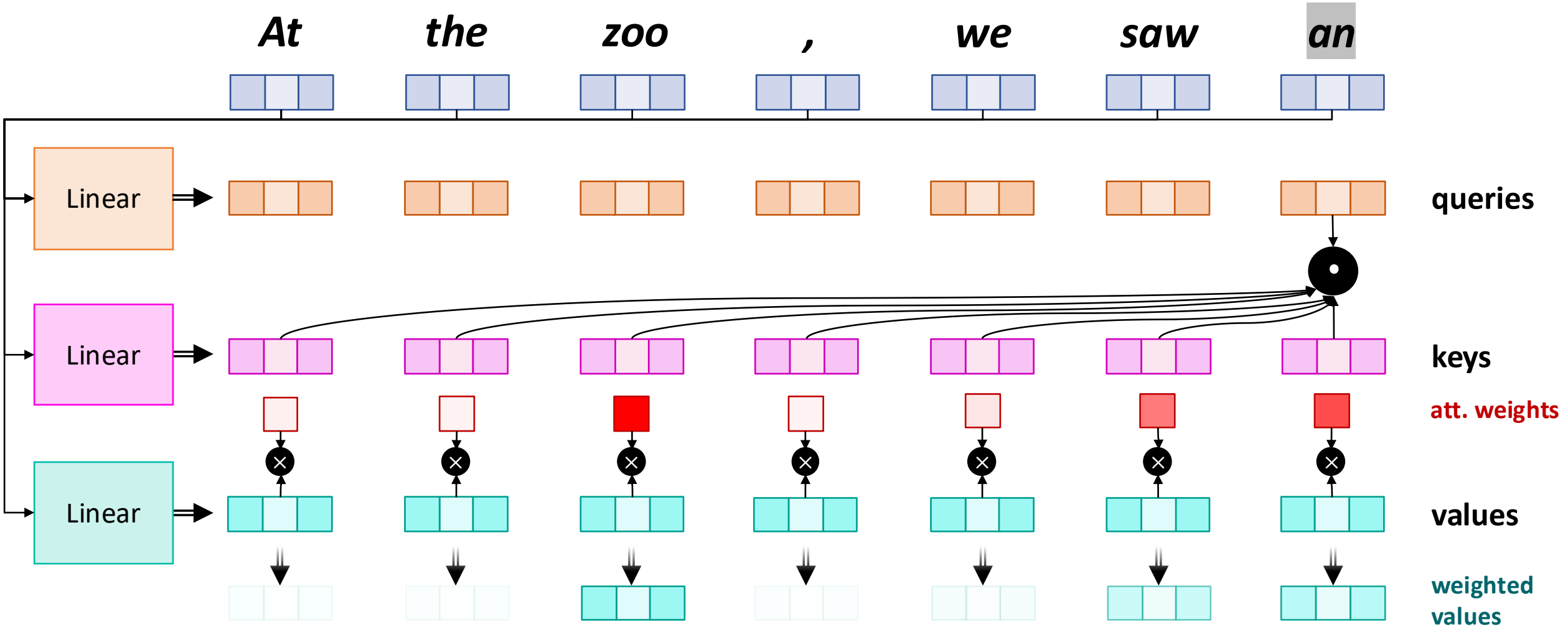
Self-attention in transformer-based LLMs



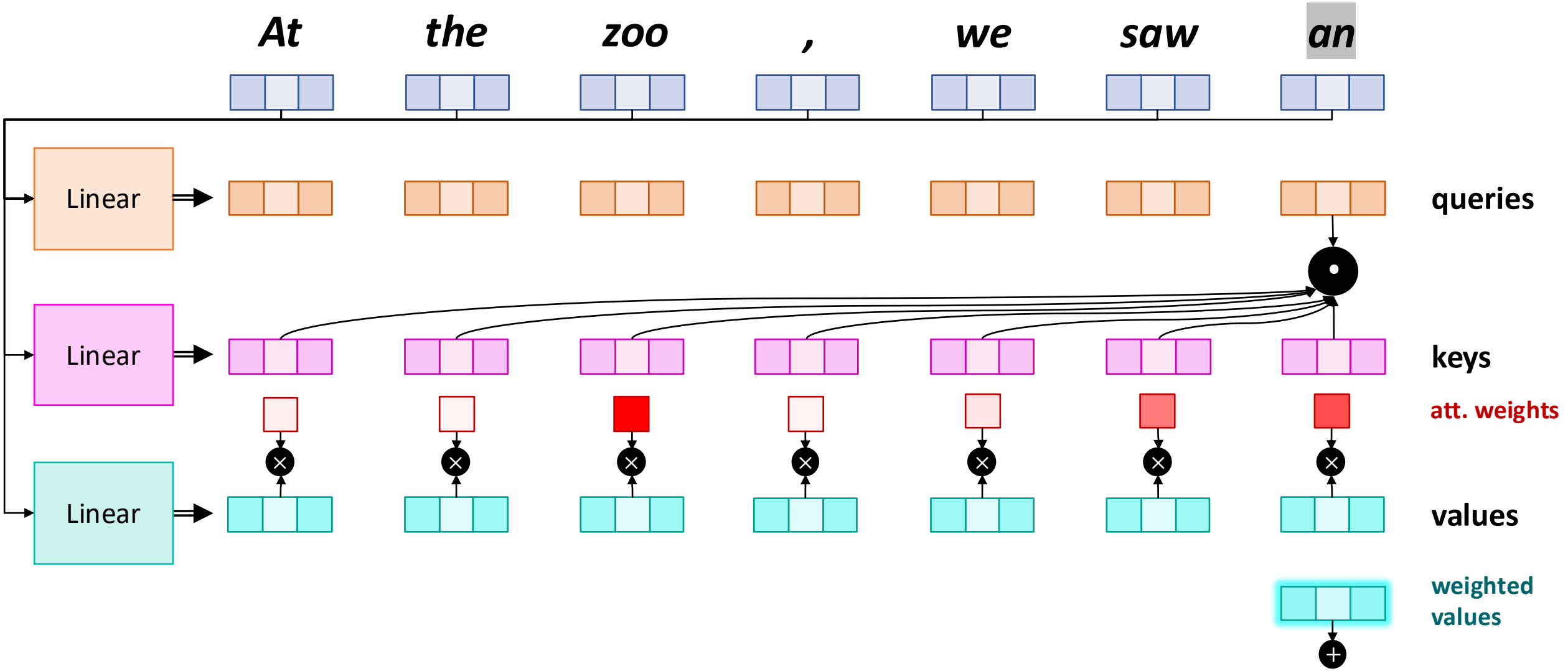
Self-attention in transformer-based LLMs



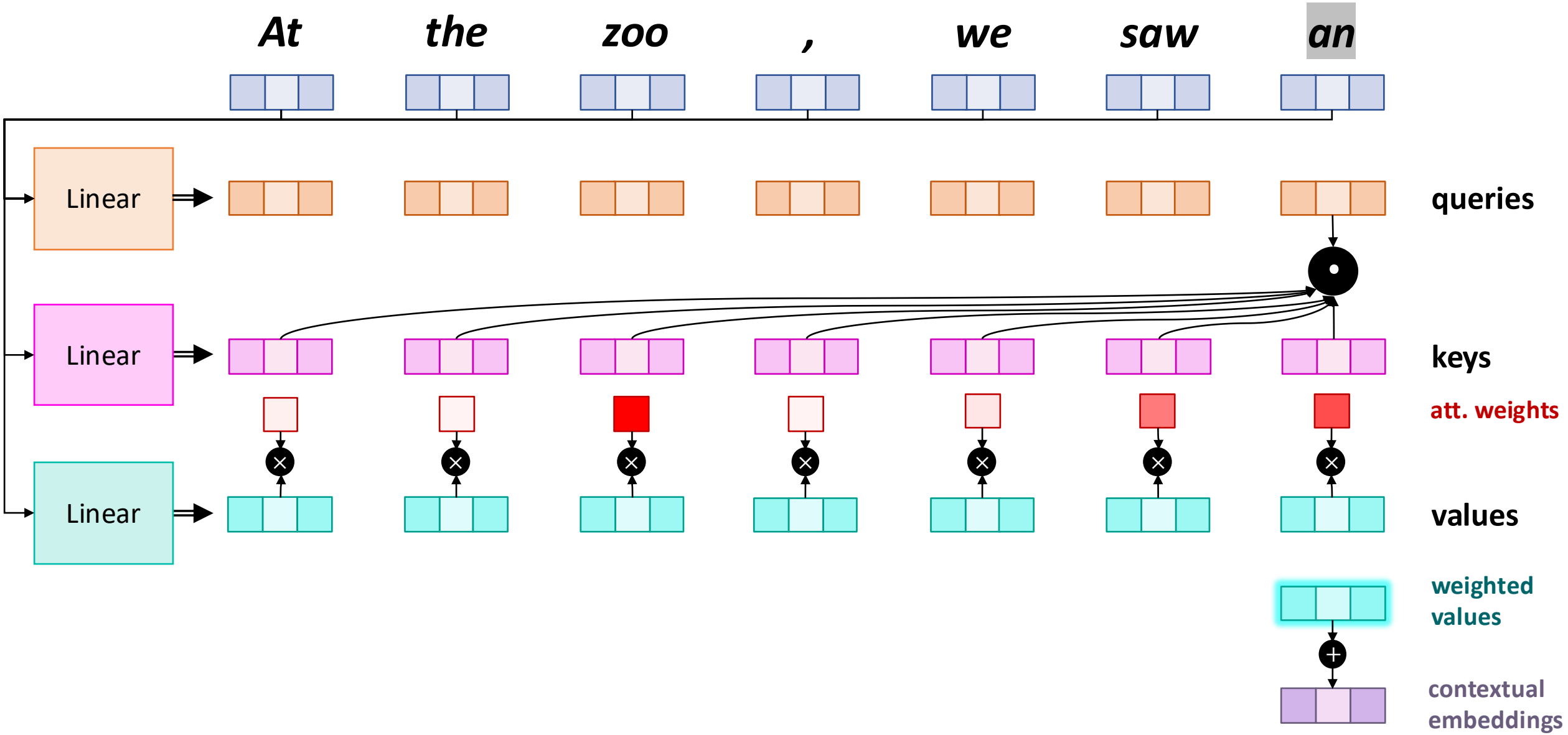
Self-attention in transformer-based LLMs



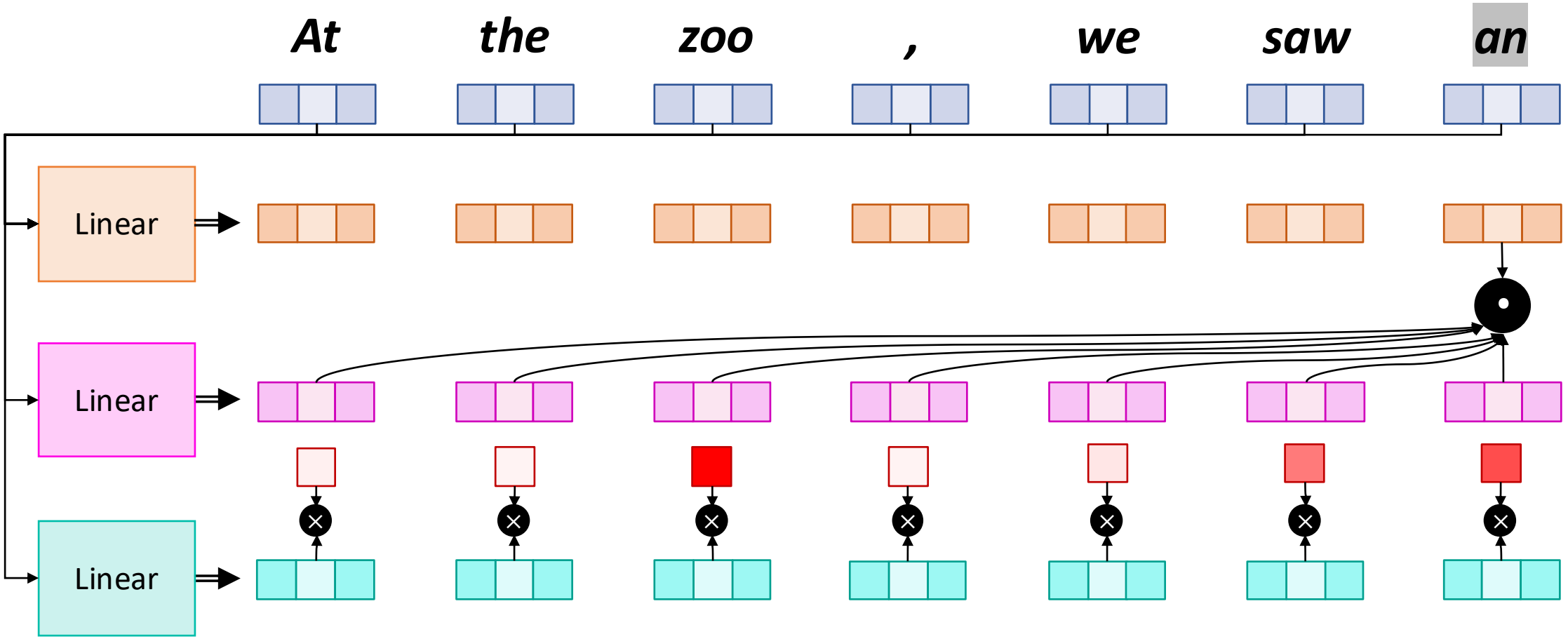
Self-attention in transformer-based LLMs



Self-attention in transformer-based LLMs



Self-attention in transformer-based LLMs



Takeaway: When predicting the next word/token, a transformer-based LLM can use information from any other words/tokens across the whole prompt. It shouldn't matter how far away they are.

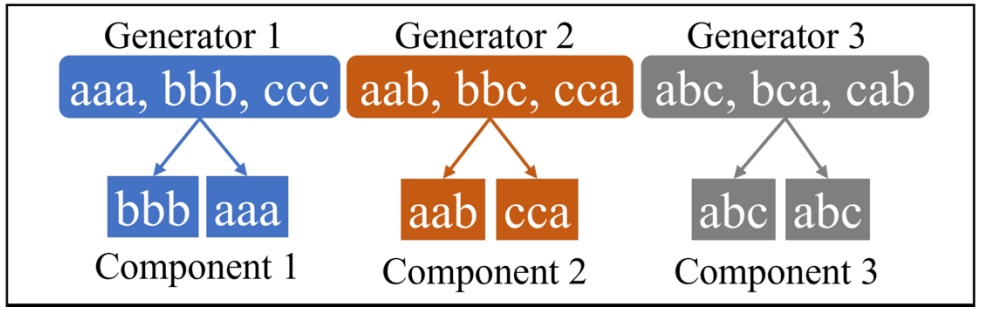
So how could they be biased toward local information???





k-back languages

- **Goal:** investigate apparent "locality bias" in transformer-based LLMs more systematically
- We synthetize several **antilocal** "*k*-back" languages, such that the next token (character) can not be predicted from the previous *k* tokens
  - Only the token *k* + 1 back provides useful information
- Formulate *k* + 1 distinct **generators** which each have their own unique trigrams (sequences of 3 tokens), then **interleave** their trigrams
  - In order to predict which token comes next, we need to find the previous token from that same trigram, which must be *k* + 1 tokens back
  - **Control:** don't interleave, just **concatenate** them back-to-back



2-back: b a a b a b b b c a c a a c b a a c

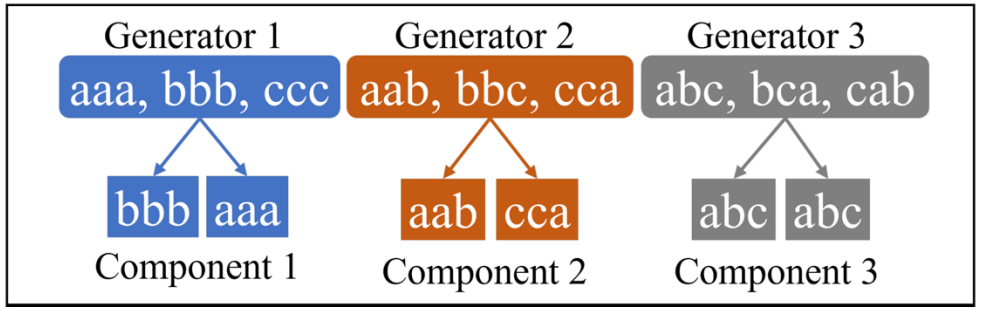
control: b b b a a a a a b c c a a b c a b c

k-back languages

Advantages of this framework:

1. We can systematically measure and vary how *antilocal* our languages are.
   - Allows us to determine how the degree of locality affects models.

2. We can also analytically calculate the per-token cross-entropy of languages!
   - Gives us a lower bound loss for models so we can confirm learning success in this simple setting.



2-back: b a a b a b b b c a c a a c b a a c

control: b b b a a a a a b c c a a b c a b c

Experimental Design

- Train a **GPT-2** model (early OpenAI language model with very similar architecture)
- Specifically, train it on *k*-back and control languages for *k* from 1 through 8
  - **Vocabulary:** a b c d e f
- If GPT-2 truly has a locality bias, it should **struggle more with higher *k***
  - Both in terms of learning success and speed
  - Gap in learning success/speed on *k*-back vs. control languages should increase with *k*

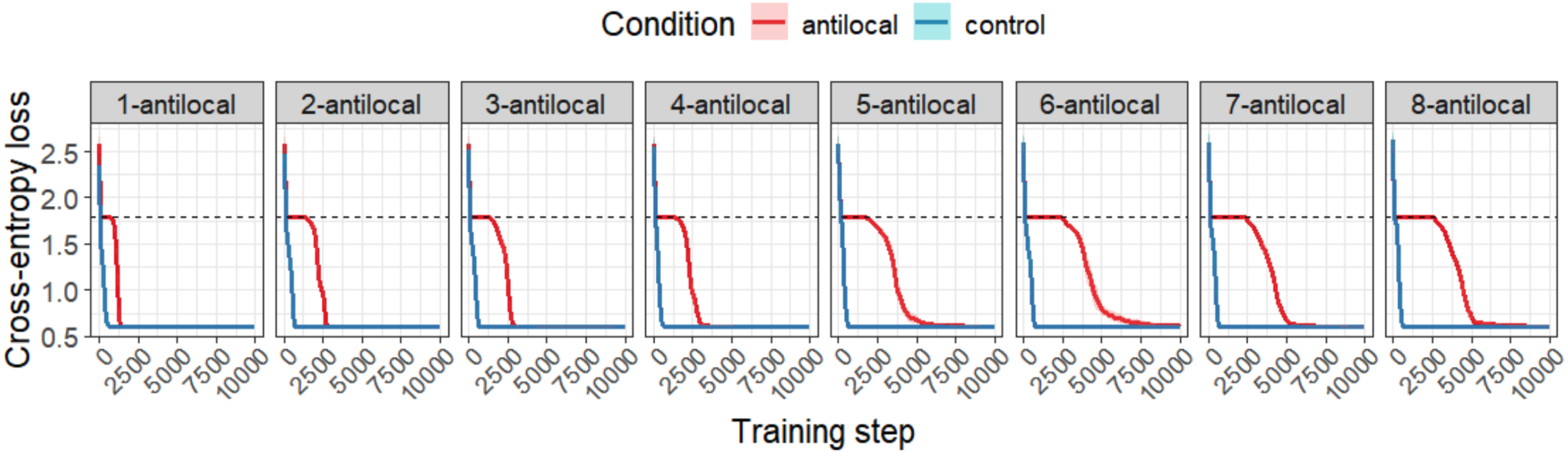
k	m	sentence length	total tokens
1	166	998	9.98×10^7
2	110	992	9.92×10^7
3	83	998	9.98×10^7
4	66	992	9.92×10^7
5	55	992	9.92×10^7
6	47	989	9.89×10^7
7	41	986	9.86×10^7
8	36	974	9.74×10^7

(m: trigrams per generator)

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitasks Learners.

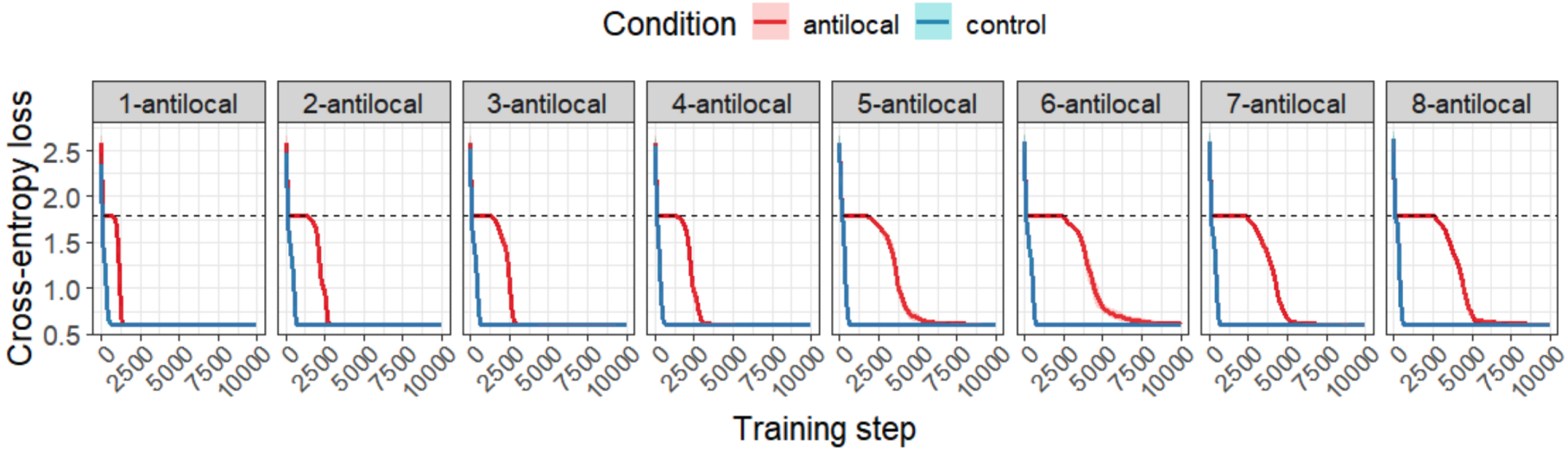
Experimental Results

Finding 1: Learning success (measured by final converged cross-entropy loss) is the same for *k*-back and control languages.



Experimental Results

Finding 1: Learning success (measured by final converged cross-entropy loss) is the same for *k*-back and control languages.

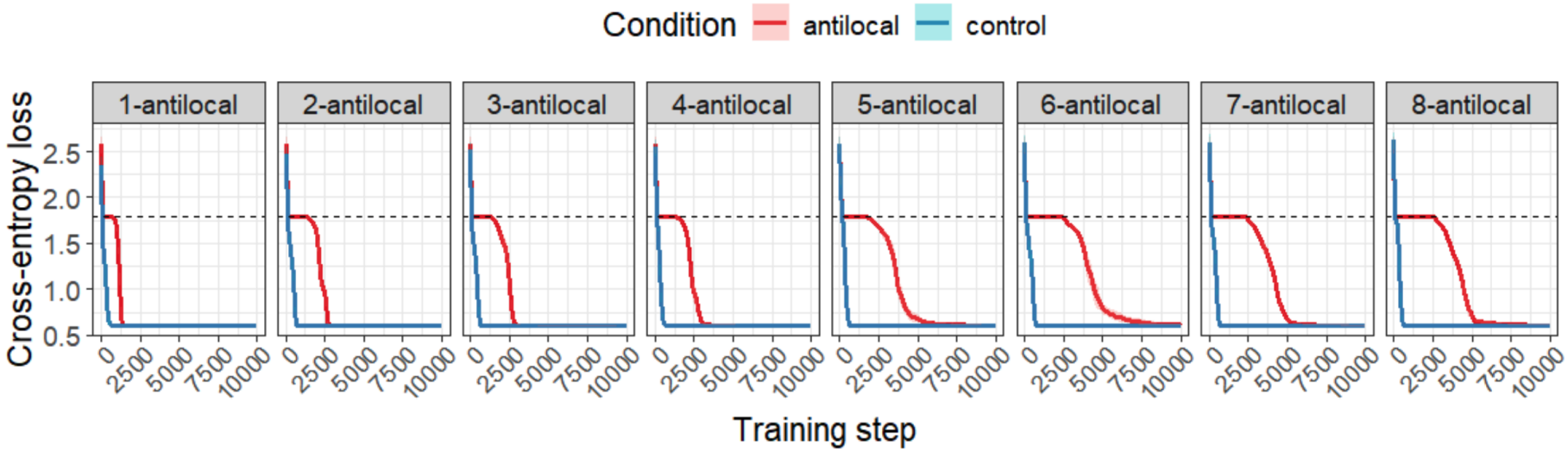


Plateau where model learns to guess randomly between vocabulary tokens.

Experimental Results

Finding 1: Learning success (measured by final converged cross-entropy loss) is the same for *k*-back and control languages.

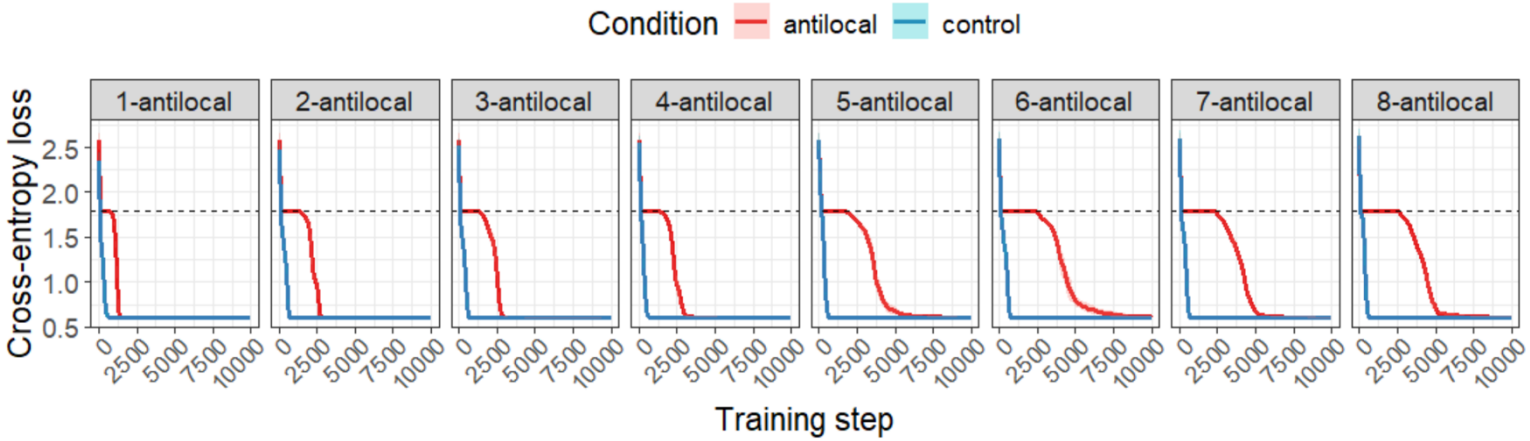
It's also nearly equal to the analytical per-token entropy of the language (<1% higher).

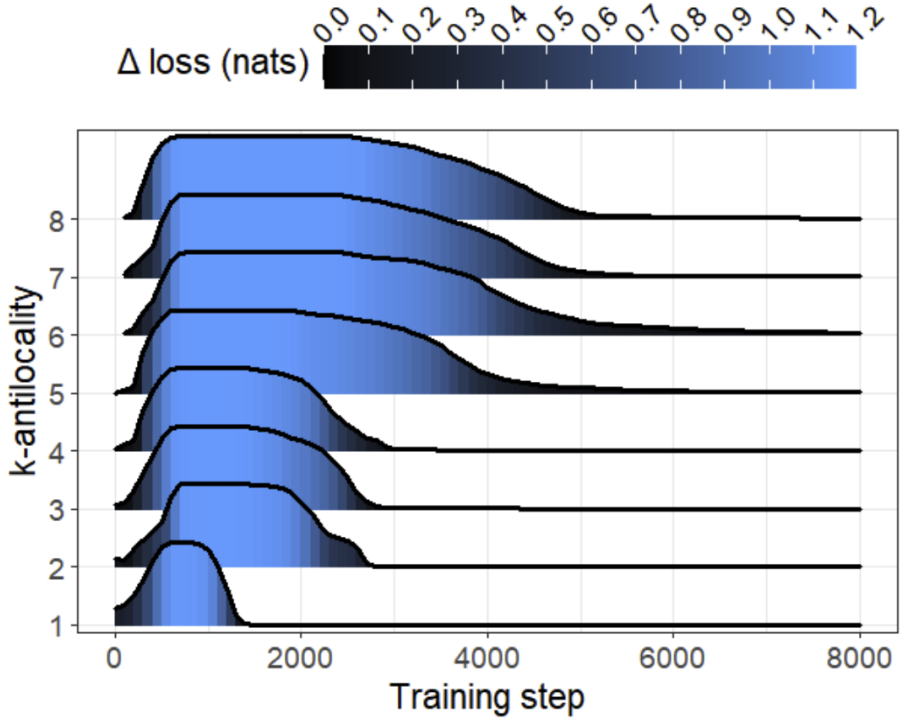


<1% away from the analytical cross-entropy of languages (lower bound).

Experimental Results

Finding 2: Models learn *k*-back languages much slower than controls, with the disparity growing with higher *k*.





So what does this mean?

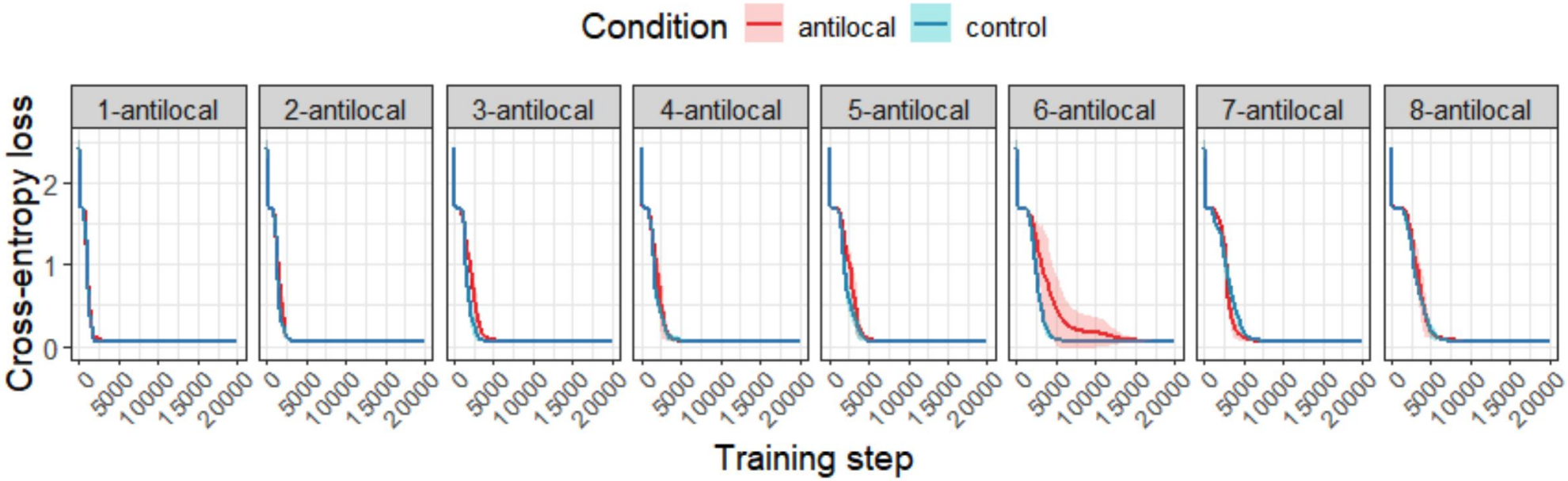
- Based on these results, we conclude that transformer-based LLMs do apparently have a **learning bias against information antilocality**
  - Therefore, they have a bias toward information locality

- But this learning bias **only affects learning speed, not learning success**

- New information, but not quite what we expected!

One possibility: **next-word prediction**

- GPT-2 always predicts the next word from the embedding of the previous word
- Could this privilege the previous word and create the locality bias?
- **Follow up experiment:** Train DeBERTa-v3, a transformer-based *masked* LM that learns language by "filling in the blank" instead of predicting the next word.

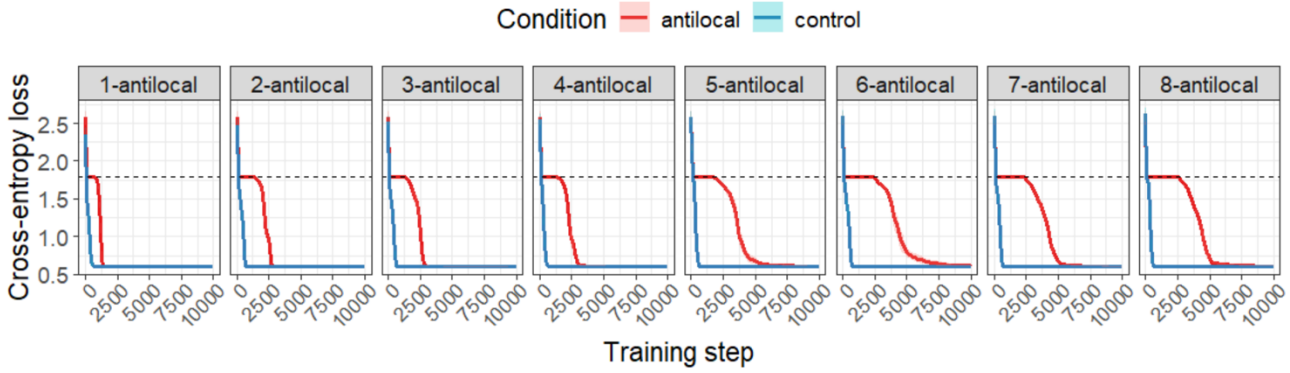


Locality bias nearly disappears!

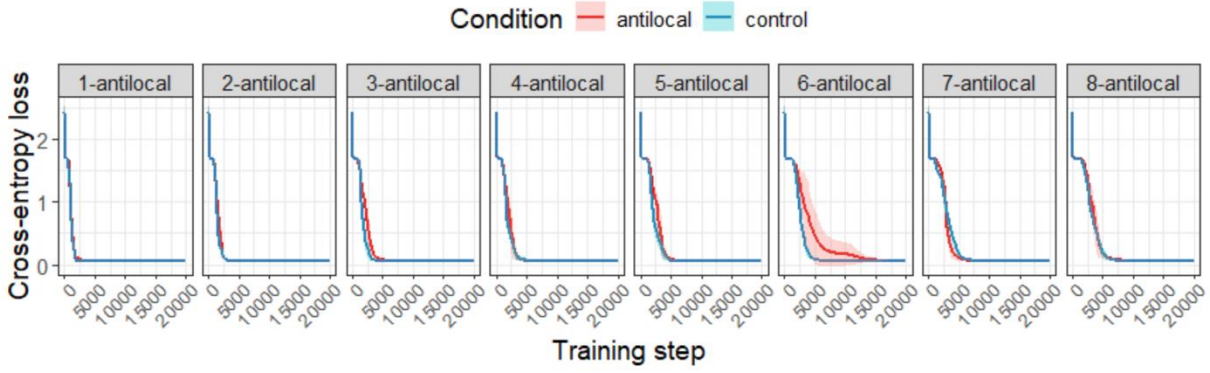
He, Gao, and Chen. 2021. DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing. *ICLR 2023.*

Conclusion

- Transformer-based LLMs **do** have a bias toward information locality which makes them learn languages faster… But not necessarily better!
  - What this implies about the theory of universal grammar… still up in the air!
- Seems to have something to do with the next-word prediction formulation (autoregression) of most contemporary LLMs
  - But there's still a lot more to understand (the fight continues)!



GPT-2



DeBERTa-v3

McInnerney*, Storks*, Abney, & Lewis. (2026). **Information Antilocality and the Locality Bias in LLMs.** *Decision pending.*