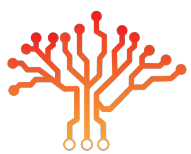
How Many Different Outputs Can a Transformer Generate?

Maxime Meyer*
Department of Mathematics & IPAL, IRL2955
National University of Singapore

Mario Michelessa* (NUS, IPAL), Caroline Chaux (CNRS), Vincent Y. F. Tan (NUS)

ICML
International Conference
On Machine Learning



AI SINGAPORE

NUS
National University
of Singapore

ipal

Out-of-Distribution Sequences are Difficult to Access

Write a sentence of at least ten words, where every word is from a different language.

Bonjour hello hola ciao hallo namaste konnichiwa annyeong salam shalom.

⋮

Out-of-Distribution Sequences are Difficult to Access

Write a sentence of at least ten words, where every word is from a different language.

Bonjour hello hola ciao hallo namaste konnichiwa annyeong salam shalom.

⋮

Je eat pan con queso oggi perché tasty.

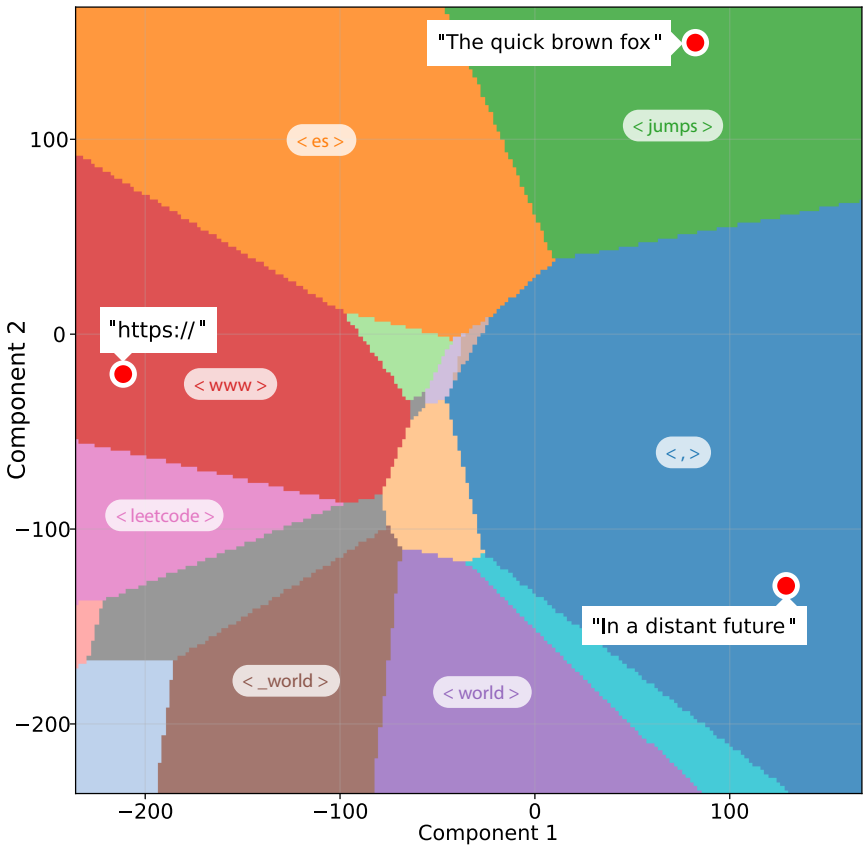
Interpretation: "I eat bread with cheese today because (it is) tasty."

Each word is from a different language (French, English, Spanish, Spanish, Italian, Italian, English), but note the issue: even here we already violate the constraint (two Spanish words, two Italian words). If you enforce the constraint strictly, grammaticality collapses.

Conclusion:

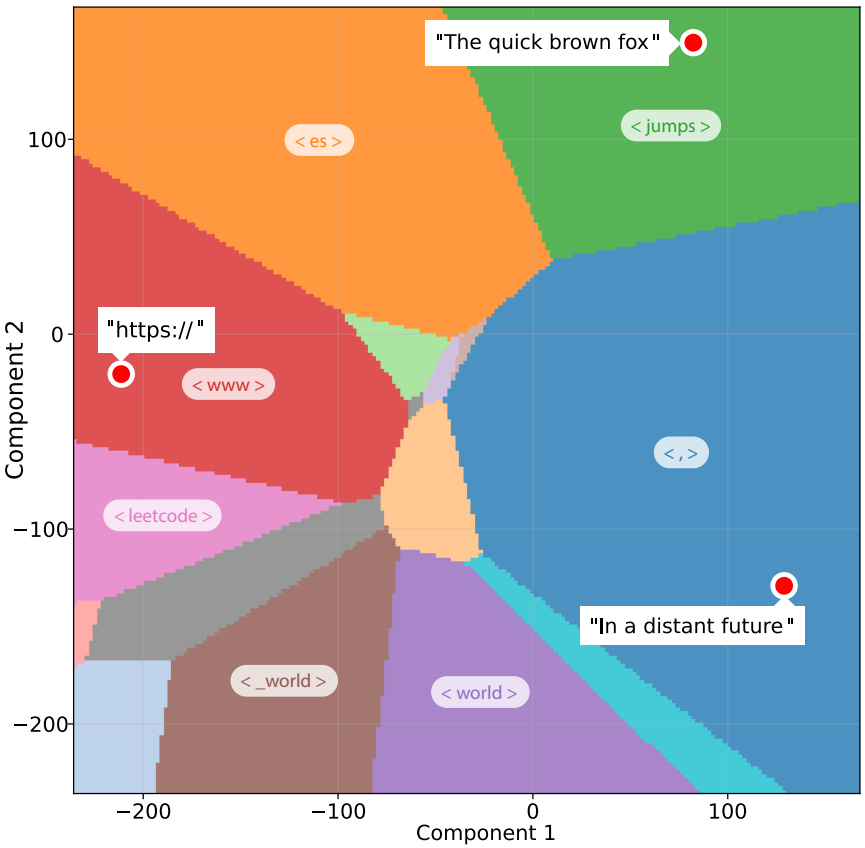
- A *meaningful* sentence: possible (as in the earlier philosophical example).
- A *natural everyday sentence*: not achievable under strict "one word per language" without breaking grammar.

The Embedding Space of a Transformer Carries Meaningful Information

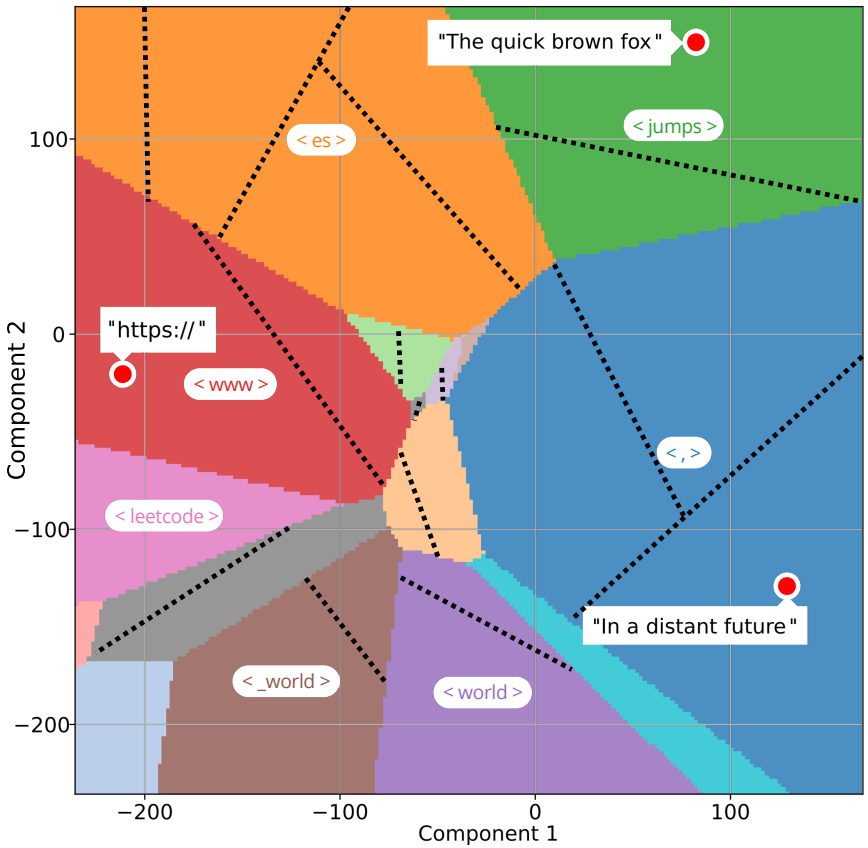


Next token prediction.

The Embedding Space of a Transformer Carries Meaningful Information



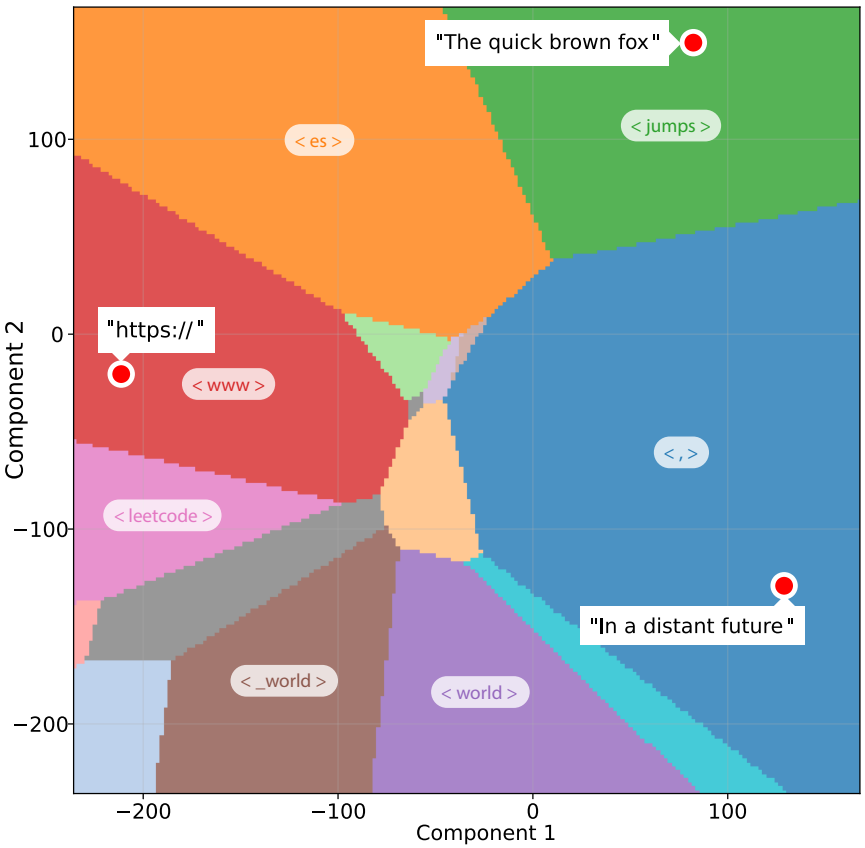
Next token prediction.



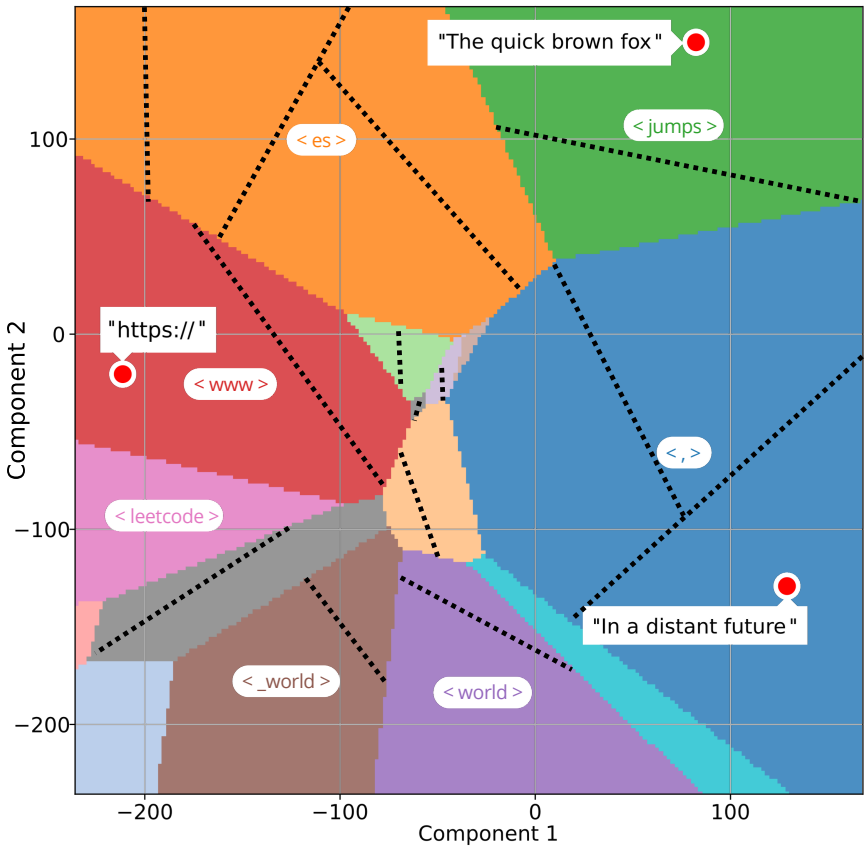
Next-next token prediction.

The Embedding Space of a Transformer Carries Meaningful Information

But transformers have finite precision! Indeed, they operate on a finite number of bits.

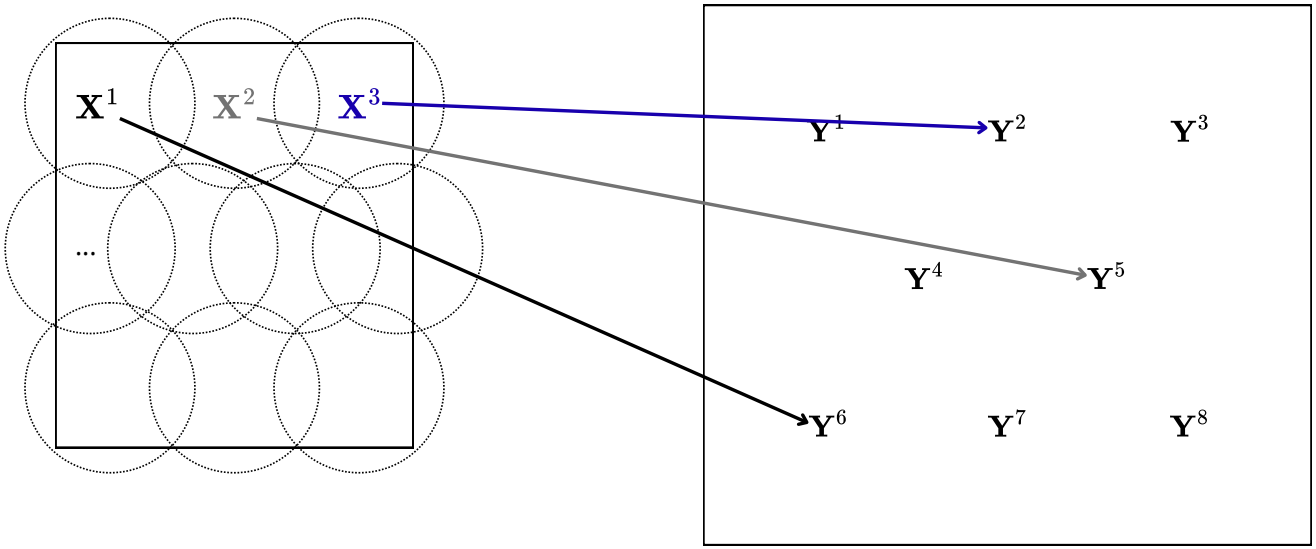


Next token prediction.



Next-next token prediction.

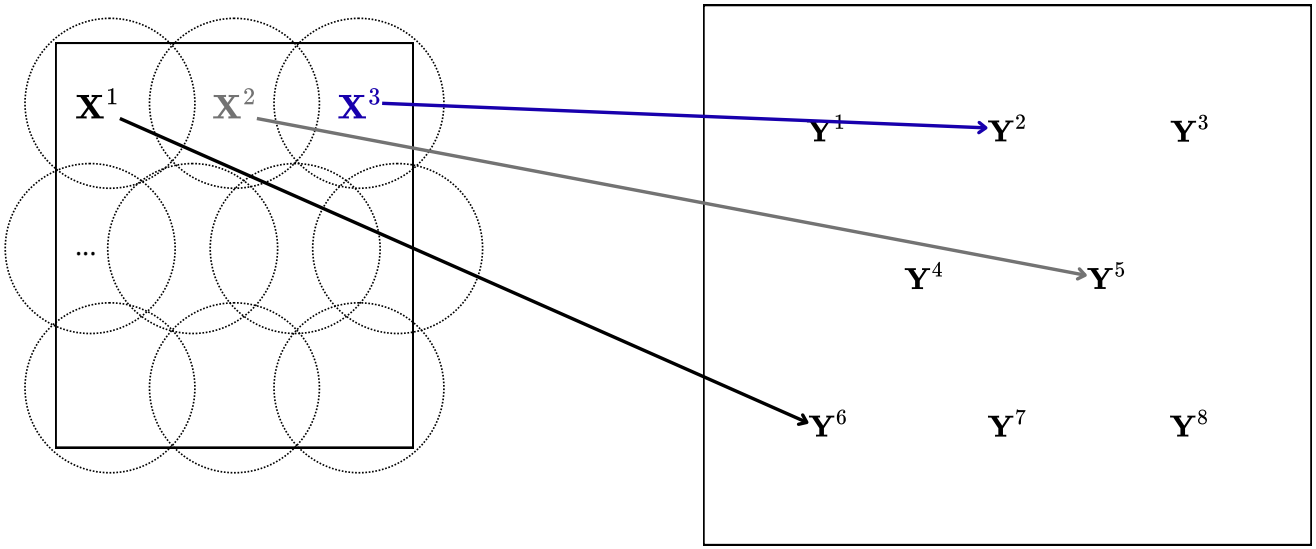
Formalizing the Intuition: There Is Only a Finite Number of Possible Inputs



Prompts of length m
$\left(C(\varepsilon)^m \text{ many}\right)$

Outputs of length n
$\left(|\mathcal{V}|^n \text{ many}\right)$

Formalizing the Intuition: There Is Only a Finite Number of Possible Inputs



Prompts of length m
$\left(C(\varepsilon)^m \text{ many}\right)$

Outputs of length n
$\left(|\mathcal{V}|^n \text{ many}\right)$

$\rightarrow$ When $n > \left(\frac{\ln C(\varepsilon)}{\ln |\mathcal{V}|}\right) m$, some sentences of length n are inaccessible.

$\rightarrow$ The proportion of accessible sequences decreases exponentially fast with n past this capacity limit.

Validating our Results through the Cramming Task

Goal: compress a long sequence into a shorter prompt.

Given a text $\mathbf{X} \in \mathbb{R}^{d \times n}$, find a prompt $[\mathbf{mem}] \in \mathbb{R}^{d \times m}$ with $m \ll n$ such that

$$\tau([\mathbf{mem}]) \longrightarrow \mathbf{X}.$$

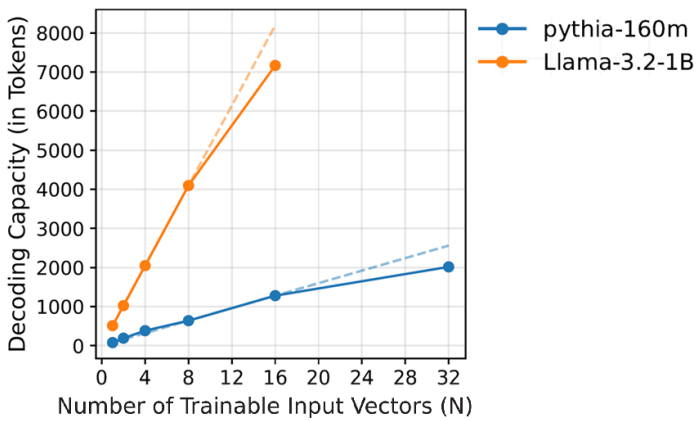


Figure 1: Scaling between the memory size and the number of encodable tokens.

Y. Kuratov et al. (2025). "Cramming 1568 Tokens into a Single Vector and Back Again: Exploring the Limits of Embedding Space Capacity". In: *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*

What the Theory Predicts

$\rightarrow$ For a fixed prompt length m, the proportion of accessible sequences of length n decreases exponentially fast with n past a given threshold.

$\rightarrow$ This threshold scales linearly with m.

$\rightarrow$ We provide an upper bound on the linear scaling factor.

Cramming Tokens: Theory vs Practice (Qualitative)

Left: exponential decrease in accessibility past a given threshold.

Right: linear scaling between threshold and m.

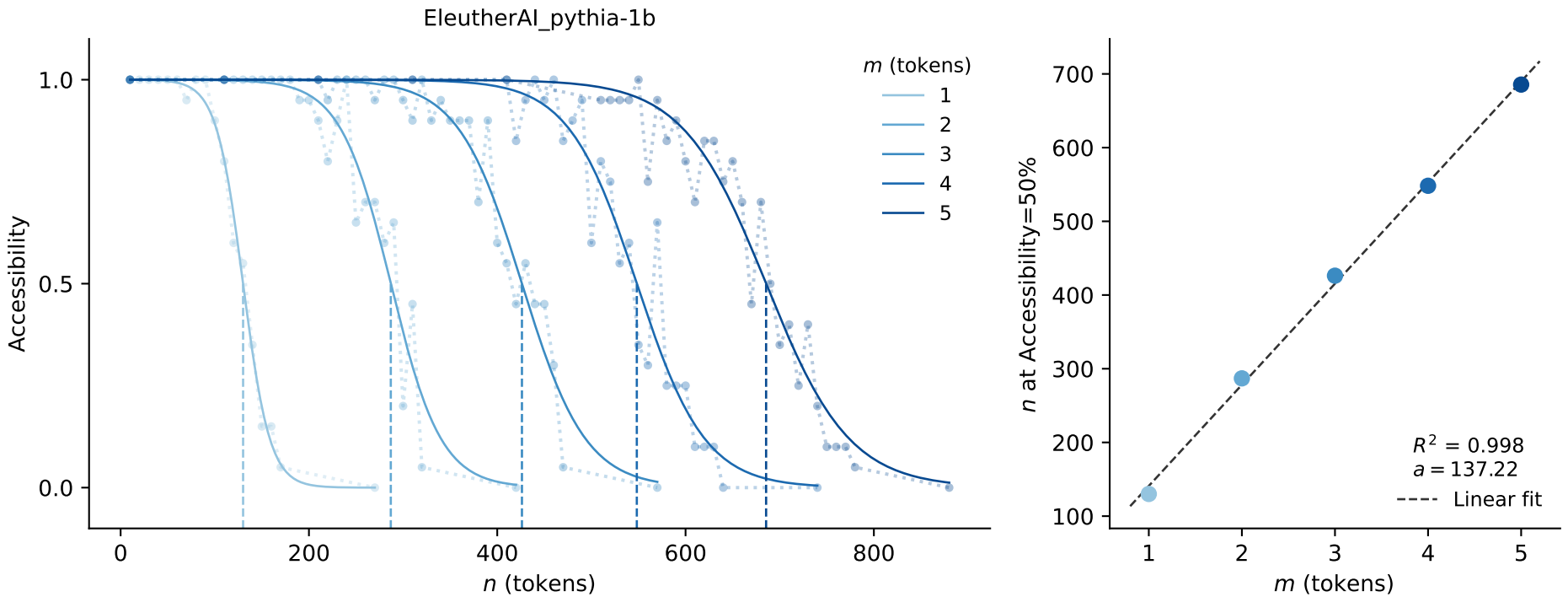


Figure 2: n: output length
m: input length

Cramming Tokens: Theory vs Practice (Quantitative)

We analyse how tight the theoretical upper bound is by computing the ratio with the empirical slope. More complex theoretical analysis yields better bounds.

Pythia			Qwen-2.5		Llama-3.2	Gemma-3
160M	410M	1B	0.5B	1.5B	1B	270M
9.24	9.79	7.77	14.1	20.4	14.3	11.52
9.10	9.60	7.70	14.01	20.34	13.98	11.24
7.92	8.15	6.12	10.96	15.30	11.86	11.12
6.66	**5.99**	**4.56**	**7.92**	**10.82**	**10.71**	**8.79**
8.65	9.83	7.71	12.32	18.81	14.63	13.42

Conclusion

Main Takeaway

$\rightarrow$ Despite representing only a fraction of a transformer's parameters, its embedding space closely characterizes its performance on some tasks.