

Anthropic's “When Models Manipulate Manifolds”.

Djordje Mihajlovic

November 13, 2025

Today, I am going to cover the recent Anthropic paper:

- Wes Gurnee et al. “When Models Manipulate Manifolds: The Geometry of a Counting Task”. In: *Transformer Circuits Thread* (2025). URL: <https://transformer-circuits.pub/2025/linebreaks/index.html>

Specifically; this follows on from the talks given by Sid 2 weeks ago.

- Why linebreaking?
- Experimental setup.
- Learned geometry.
- Line break mechanisms.
- Perturbations.
- Implications & open questions.

(Biological) Motivation

Intelligent systems (animals/plants) develop sensory capabilities to survive in their environments.

- Bats → Dark conditions → Echolocation
- Arctic Reindeer → Seasonal UV → Vision shifts
- Migratory Birds → Navigate large spans of area → Magnetic field detection

(ML) Motivation

- LLMs → Interpreting ASCII art → ?

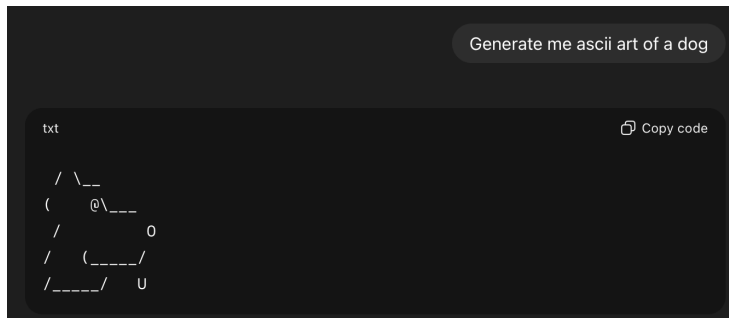


Figure: LLM's adapting 'sensory' capabilities to interpret ASCII art $\approx \exists$ a learned algorithm for producing ASCII art.

The paper aims to **pin down a sensory capability/algorithm learned by LLMs** in detail. Algorithm should be simple → better for mechanistic interpretability.

- LLMs → When to line break → ?

RECAP: Tokenisation

LLMs do not see text, or spaces, or linebreaks → just numbers through a chosen tokenisation.

Naive example:

I'm an LLM who has learned that linebreaking
is essential to generate documents.

...4253 10 20 16 1123...

Here: 10 = ↵

Recap: Sparse Autoencoders

Key idea: SAEs let you pinpoint exact monosemantic features.

Each decoder column, d_i , is a *vector* in the space of the thing being constructed.

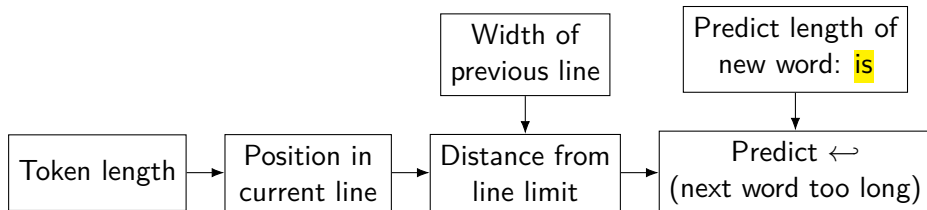
CLTs and Attribution Graphs

Key idea: CLTs let you track how monosemantic features interact with each other to perform a task.

Linebreak Attribution Graph

I'm an LLM who has learned that **linebreaking** **is essential** to generate documents.

... has learned that **linebreaking**



Dual geometric perspectives

The attribution graph provides an **causal wiring diagram** (\square, \rightarrow) of the learned line breaking algorithm

Geometric perspectives can be found on the following questions:

- Will this text fit?
- How are different character counts represented?
- How is the boundary detected?
- Will the next word fit?
- How are representations constructed?

\exists low-dimensional feature manifolds (\square) which interact geometrically (\rightarrow).

Representing character count I

Line character count: total number of characters since the last newline, including characters of current token.

Q: Do models linearly represent character counts as quantitative variables?

A: Yes. Linear probe fit on residual stream has high accuracy → there are features in the SAE that correspond to 'line character count'.

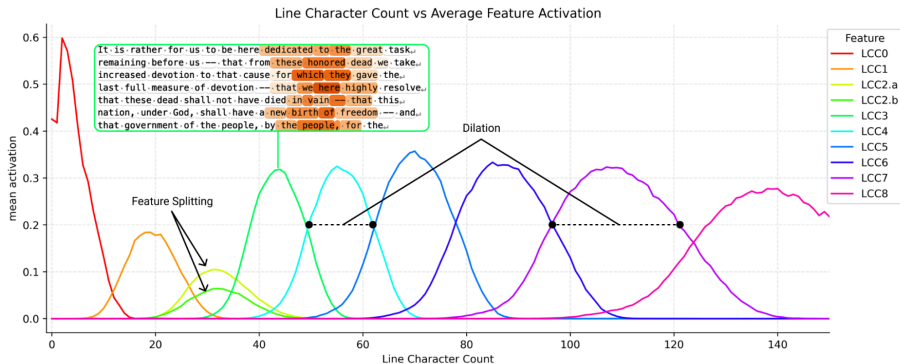
Q: How does the model represent character count?

M: Study the dictionary vectors of the corresponding features.

Representing character count II

Character count features

There exist features whose interpretation corresponds to position in k .



Representing character count III

Run a PCA on the N -dimensional vectors (d_i corresponding to line character counts)

→ There are 6 PC's (everything else is effectively noise).

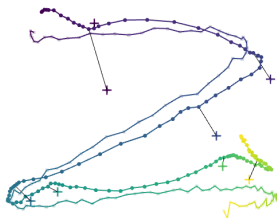
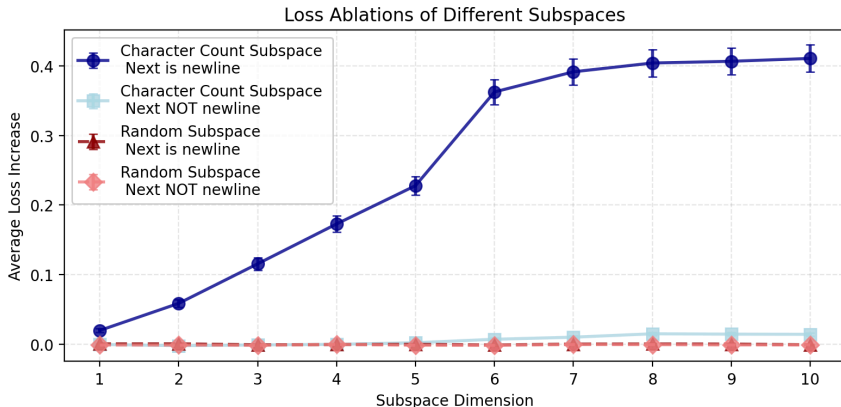


Figure: 1-3 PCs of features corresponding to max activation at a given line character count (in residual stream)

Representing character count IV

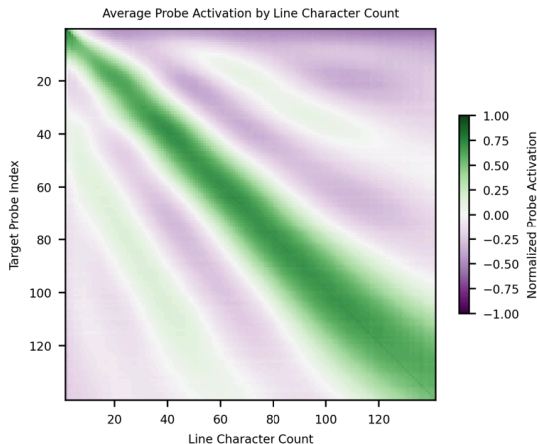
Ablation on space



Representing character count V

Probe perspectives (on PCA'd low-d data)

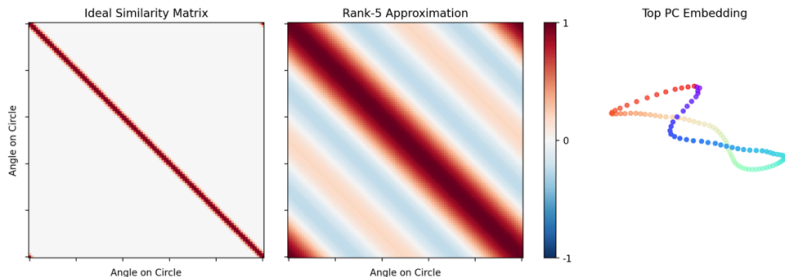
Notice: strong diagonal + rippling



Representing character count VI

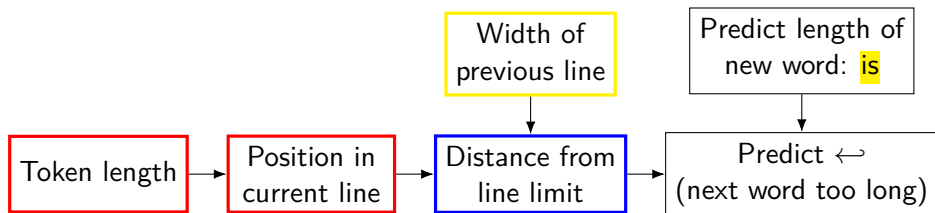
Toy example

Symmetric set of unit vectors in 150D. 5D embedding retains as much of the curvature of the 150D embedding.



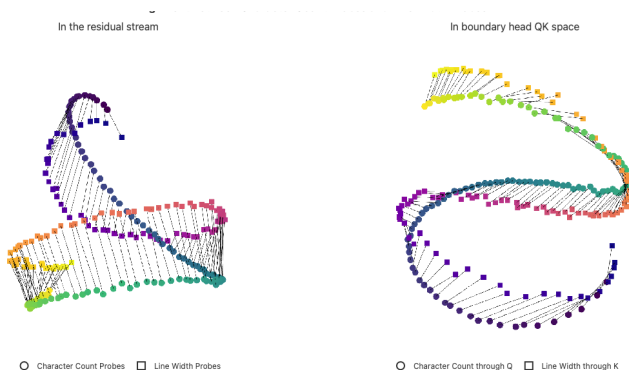
Geometric Perspective 2: Sensing the line boundary

Q: How are character counting representations used to determine if the current line is approaching the line boundary?



Sensing the line boundary II

Alignment between character count probes and line width probes

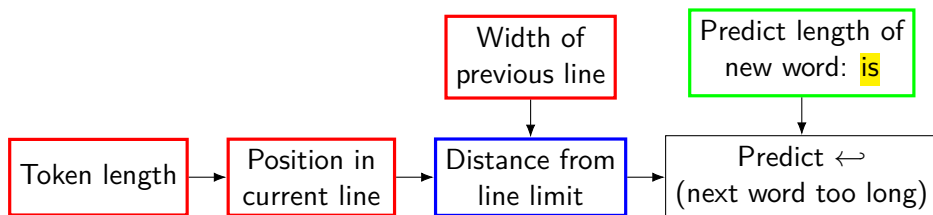


It turns out each head gives a feature which activates highly according to *a specific number* of characters remaining.

Multiple heads and better line prediction

Predicting the newline

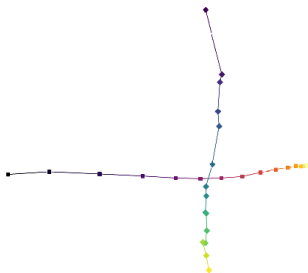
Combine the estimation of line boundary with the prediction of the next word to predict line break



Predicting the newline II

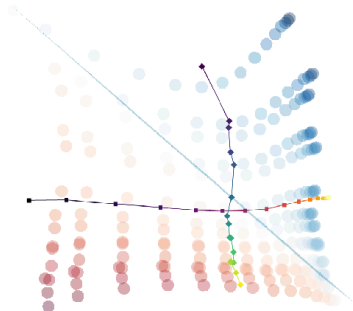
Orthogonal Representations Create a Linear Decision Boundary for Linebreaking

Next Word Length vs Characters Remaining



◇ Next Token Length □ Characters Remaining

The Sum Makes Linebreaking Linearly Separable



◇ Next Token Length □ Characters Remaining ○ Margin After Next Word

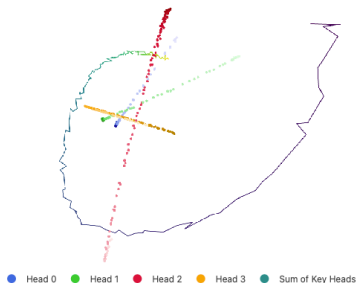
Q: We know how character counting representations are used but how does a model compute them: i.e. there is a feature indicating character count of line 'X' but how did the model get there in the first place?

Distributed character count algorithms II

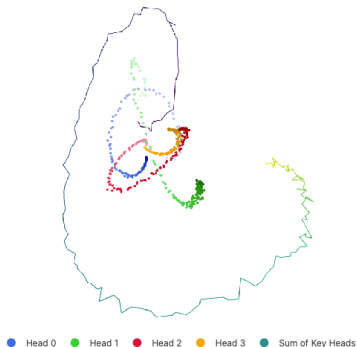
Interacting heads \rightarrow encoded in their geometry is information about character counts.

Individual Head Outputs Tile the Joint Output Space

Layer 0 Heads

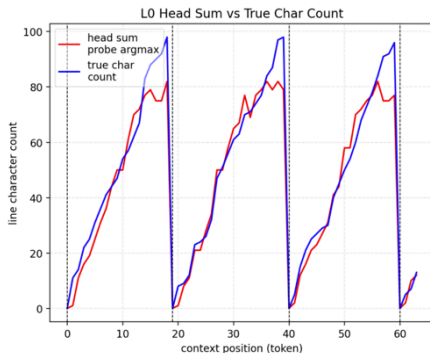
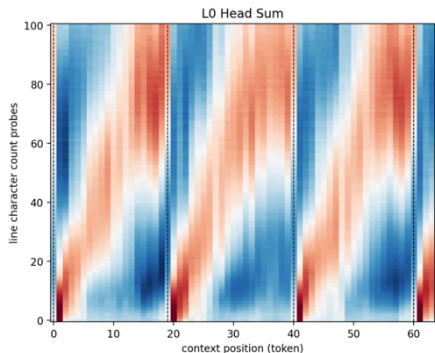


Layer 1 Heads



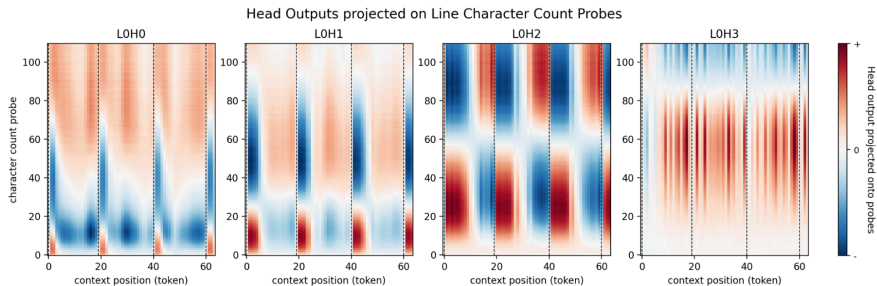
Distributed character count algorithms III

A: Attention head outputs sum to produce the character count



Distributed character count algorithms IV

Q: What does each head capture?



Q: Having understood the character counting mechanism, can we construct *visual illusions*.

To do this → check effect of data (from large corpus of text) on attention heads for character counting.

@@-Visual Illusions


@@ was a found two-character string which changed the behaviour of the newline predictions.

e.g.

Original Prompt

After gallium was ruled out due to melting point, the engineers on the project chose the chemical element with atomic number 13, also called


Original Prediction

TOKEN	PROBABILITY
	0.79
aluminum	0.12

Insert @@

After gallium was ruled out due @@ melting point, the engineers @@ the project chose the chemical element with atomic number 13, also called

New Prediction

TOKEN	PROBABILITY
	0.22
aluminum	0.66

Intelligent systems (animals/plants/LLMs) develop sensory capabilities to survive in their environments.

- Bats → Dark conditions → Echolocation
- LLMs → Linebreaking → Manifold manipulation

Table of contents

1 Bibliography

Bibliography I



Gurnee, Wes et al. “When Models Manipulate Manifolds: The Geometry of a Counting Task”. In: *Transformer Circuits Thread* (2025). URL: <https://transformer-circuits.pub/2025/linebreaks/index.html>.