DEEP SEQUENCE MODELS TEND TO MEMORIZE GEOMETRICALLY; IT IS UNCLEAR WHY.

Shahriar Noroozizadeh* Vaishnavh Nagarajan † Elan Rosenfeld † Sanjiv Kumar †

*Carnegie Mellon University †Google DeepMind (work done when in Google Research)

Machine Learning Department, Carnegie Mellon University, School of Computer Science

Google Research

DeepMind

Paper Code

OUR BROAD QUESTION:

How do deep sequence models memorize "atomic facts" in its parameters?

Consider a dataset with no high-level rules; just arbitrary associations. Nothing to "generalize", "learn" or "represent".

Input → Output:

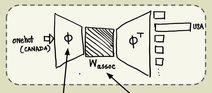
GUATEMALA → HONDURAS
CANADA → USA
MEXICO → GUATEMALA
USA → MEXICO

We identify and contrast two *competing* forms of parametric memory in sequence models

ASSOCIATIVE MEMORY

(The predominant view of memory. E.g., "Birth of a Transformer: A memory viewpoint" Bietti et al., 2023)

Model stores local associations from training data as is.



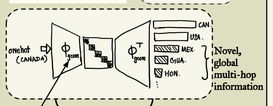
embedder : matrix of *random* "keys"

lookup table: adjacency matrix

GEOMETRIC MEMORY

(What we formalize)

Model infers a "global map" from data



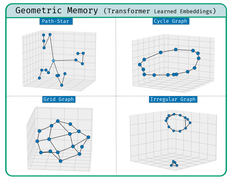
Highly organized embeddings

(low-rank) factorization of adjacency matrix

Novel, global multi-hop information

FINDING 1:

Deep sequence models tend to memorize *geometrically*, not *associatively* .



EMPIRICAL EVIDENCE:

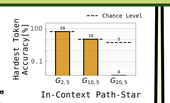
Token embeddings of a Transformer on small graph datasets.

(true of Mamba and Neural networks — it's *not* the magic of attention!)

FINDING 2: Unlike associative memory, geometric memory has synthesized novel *global* information *not* explicit in training data. This aids reasoning.

Consider the notoriously hard task of path-finding on path-star graphs given in-context

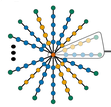
[from "Pitfalls of next-token prediction" Bachmann and Nagarajan' ICML 2024]

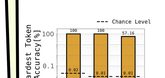


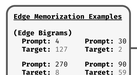
PROMPT: adjacency list of random path-star graph

TARGET: start →goal path

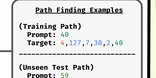
The same task becomes possible when the graph is memorized in-weights! *Why?*







Memorize the edges of a *fixed* large graph into the weights



Train on 50% of paths and test on the rest.

INSIGHT:

Geometry makes exponentially-hard composition task

$$f\text{-}first\text{-}token = W_{assoc}(W_{assoc} \cdot (goal))$$

into a spatial-navigation task

$$f\text{-}first\text{-}token = \frac{\Phi_{geom}(goal) - \Phi_{geom}(start)}{length}$$

We identify and contrast two *competing* forms of parametric memory in sequence models

ASSOCIATIVE (a *local* lookup table)

VS.

GEOMETRIC (a *global* map)

We then pose foundational open questions:

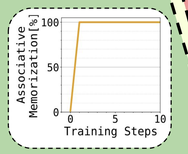
- Why does the model memorize geometrically rather than associatively?
- How do we explicitly control the model memory to choose a mode of storage?

OPEN QUESTION 1: THE MEMORIZATION PUZZLE

Why does the model memorize geometrically rather than associatively, even when there's no obvious training pressure to do so?

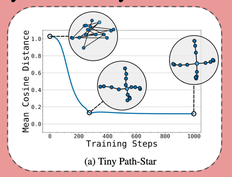
IS THERE PRESSURE FROM GLOBAL SUPERVISION?

No. Geometry is learned even from memorizing local edges!



IS THERE OPTIMIZER PRESSURE?

No. Associative memory is just 2 steps away! Geometry is farther away.



(a) Tiny Path-Star

IS THERE ARCHITECTURAL PRESSURE?

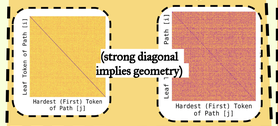
No. There *is* sufficient capacity to memorize associatively!

Frozen-embedding model fits training data as it can store width² associations.

Why? More compute needed to "unearth" latent information!

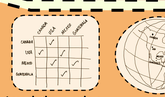
Embedding cosines ... with path-finding

... with just local edge data.



IS IT A PRESSURE TO COMPRESS?

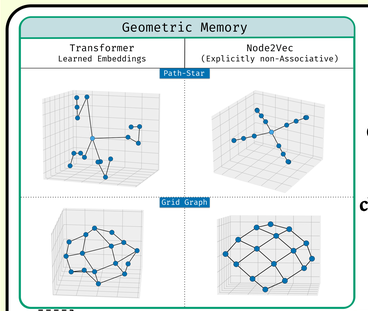
No. Counterintuitively, geometric memory is *not* necessarily more succinct!



Associative requires |edge| bits

Geometry requires |vertices| * dim bits

Graphs like path-star & cycle have |edges| ≈ |vertices|. Hence same bit complexity.

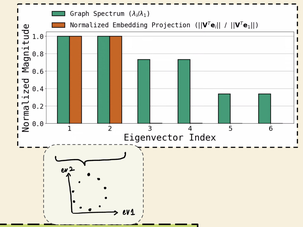


EMPIRICAL OBSERVATION: 2-layer model (right) gives us cleaner geometry than Transformer.

OPEN QUESTION 2: How to make Transformer memory more geometric?

INSIGHT:

"Global" geometry is the top spectrum of adjacency matrix.



OPEN QUESTION 3: *Wide* 2-layer models *under cross-entropy loss* implicitly has a low-rank spectral bias. Why? What is the precise solution?

Φ_{geom} $\Phi^\top_{geom}$