How to think step by step A mechanistic understanding of chain-of-thought reasoning

Subhabrata Dutta, Joykirat Singh, Soumen Chakrabarti, Tanmoy Chakraborty
Accepted at Transactions of Machine Learning Research (TMLR)

Presented by: Parth Mehta, Parmonic

Mechanistic

Relating to theories which explain phenomena in purely physical or deterministic terms.

Problem Definition

Investigates the neural sub-structures within LLMs that manifest CoT reasoning from a mechanistic point of view

Why is it difficult?

1. Hydra Effect

Neural algorithmic components within LLMs are adaptive: once we ‘switch off’ one functional component, others will pitch in to supply the missing functionality

2. LLMs Memorize *"A LOT"*

LLMs tend to memorize factual associations from pre training as key-value caches using the MLP blocks

Due to the large number of parameters in MLP blocks and their implicit polysemanticity, interpretation becomes extremely challenging

Solution

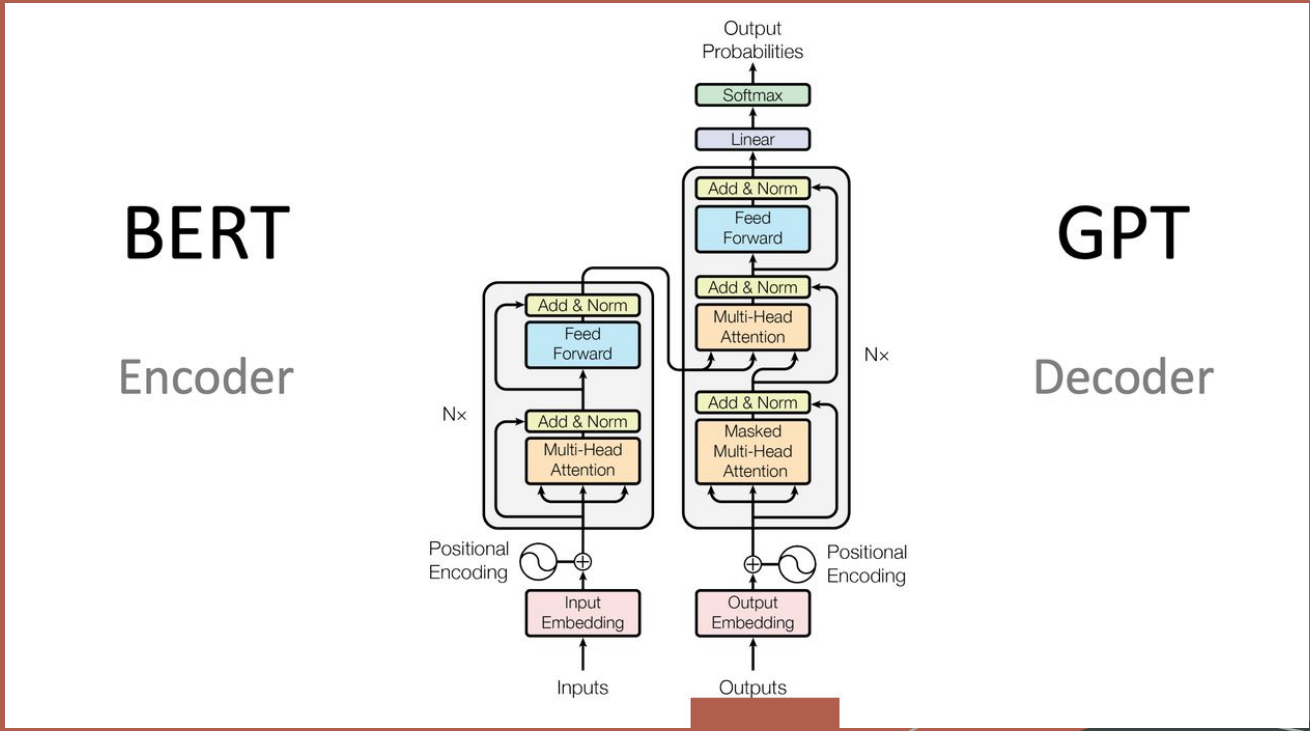
Limit the effect of MLP blocks by using a fictional ontology

Hypothesis: A fictional ontology ensures zero interference between the entity relationships presented in the question and the world-knowledge acquired by the model in the pertaining stage

Key Concepts

- Transformers
- Residual Connections
- Activation Patching
- Knockout
- Circuits
- Probing Classifiers
- Logit Lens

Transformer and Residual Connections



Circuits

- A circuit is a subgraph of the complete computational graph of the model, responsible for a specific set of tasks

- Nodes defined by model components like attention heads and projections and edges defined by interactions between such components in terms of attention, residual streams, etc

Activation Patching

- Activation patching begins with two forward passes of the model, one with the actual input and another with a selectively corrupted one

- Given an input John and Mary went to the park. John passed the bottle to, the model should predict Mary. Further, corrupting the input by replacing Mary with Anne would result in the output changing to Anne.

Activation Patching

- Let x_Mary_j and x_Anne_j represent the original and corrupted residual streams at decoder block j, depending on whether Mary or Anne was injected at the input.

- Now, in a corrupted forward pass, for a given j, if the replacement of x_Anne_j by x_Mary_j results in the restoration of the output token from Anne to Mary, then one can conclude that attention mechanism at decoder block j is responsible for moving the name information.

Knockout

- A method to prune nodes in the full computational graph of the model to identify task-specific circuits.
- Zero Ablation
- Mean Ablation

Probing Classifiers

Training a simple classifier on the hidden states of a model to predict specific linguistic properties or features

- Linear Probes
- Non-linear probes

Logit Lens

Extract Intermediate Logits: During inference, the logits from intermediate layers are extracted alongside the final logits.

Analyze Changes: These intermediate logits are analyzed to understand the evolution of the model's predictions.

Methodology

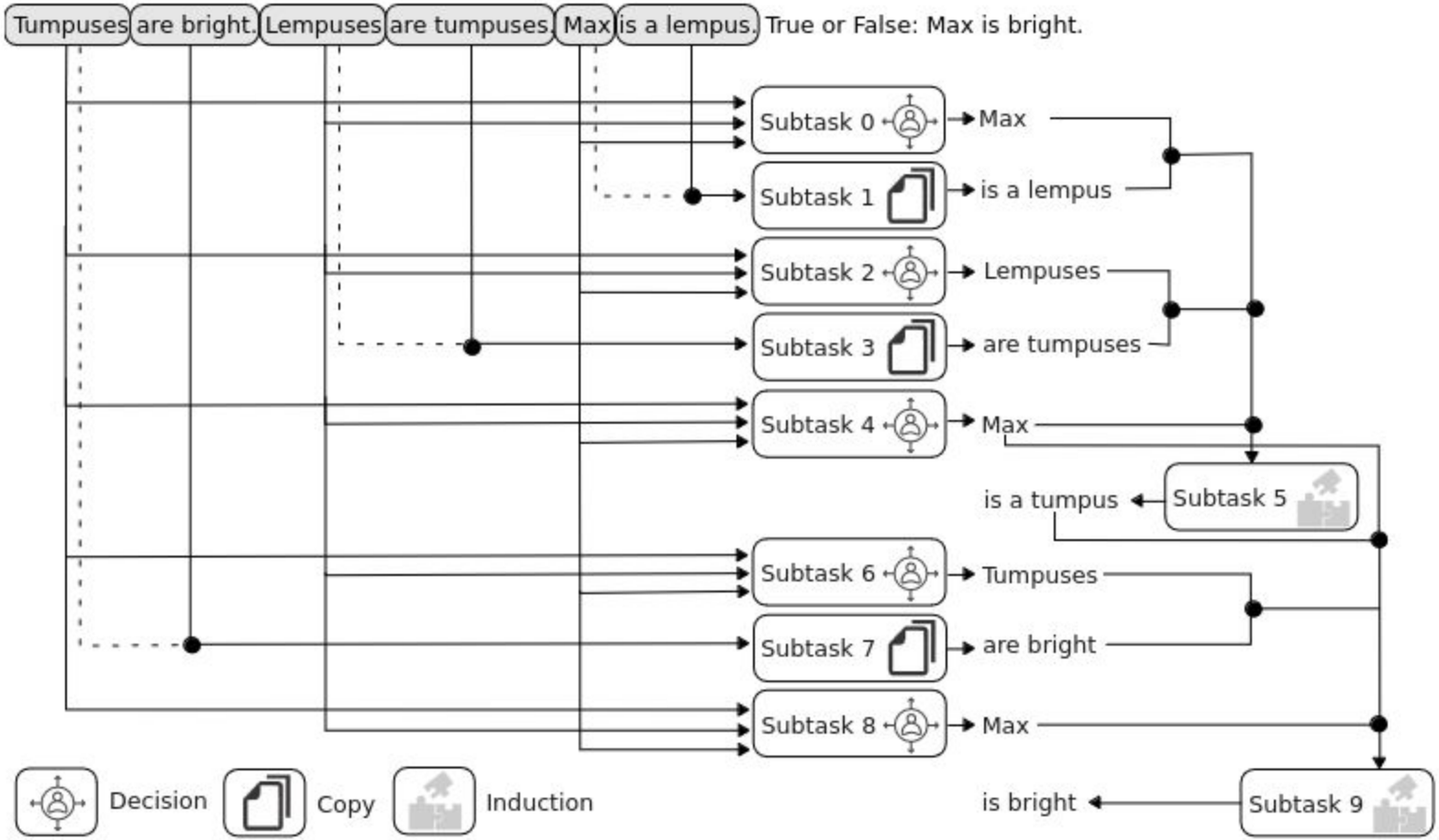
- Make analysis independent of LLMs “Knowledge”
- Uses Fictionial and False ontologies
  - Things that dont exist in real life, things that are incorrect in real life
- Look for task specific components
- Analyze flow of information at each layer

Reasoning over fictional ontology

“Tumpuses are bright. Lempuses are tumpuses. Max is a lempus. True or False: Max is bright”

Breakdown into subtasks

- Decision-making: Where to start?
- Copying: Which information to copy from input to output
- Induction: The LM uses a set of statements to infer new relations


Tumpuses
are bright.
Lempuses
are tumpuses.
Max
is a lempus.
True or False: Max is bright.
Subtask 0
Max
Subtask 1
is a lempus
Subtask 2
Lempuses
Subtask 3
are tumpuses
Subtask 4
Max
Subtask 5
is a tumpus
Subtask 6
Tumpuses
Subtask 7
are bright
Subtask 8
Max
Subtask 9
is bright
Decision
Copy
Induction

Task Specific Heads?

- Decision Subtask
- Activation patching on individual heads over decision-making subtasks

$$\mu_{\text{Decision}}(h_{j,k}) = \frac{P_{\text{org}}(x = s_{\text{ans}}) - P_{\text{corrupt}}(x = s_{\text{ans}})}{P_{\text{org}}(x = s_{\text{ans}}) - P_{\text{patched}}(x = s_{\text{ans}})}$$

- Other subtasks - Appendix C

Subtask index	Accuracy	Heads removed	Threshold range
0	1	475	0.30487806 - 0.31463414
1	0.93	554	0.30243903,0.31707317
2	0.96	617	0.3 - 0.3195122
3	1	617	0.3 - 0.3195122
4	0.99	554	0.30243903 - 0.31707317
5	0.93	475	0.30487806 - 0.31463414
6	0.94	475	0.30487806 - 0.31463414
7	0.88	617	0.3 - 0.3195122
8	0.95	617	0.3 - 0.3195122
9	0.91	663	0.297561 - 0.3219512

Table 1: Statistics of subtask-wise attention head removal according to their importance in inductive reasoning.

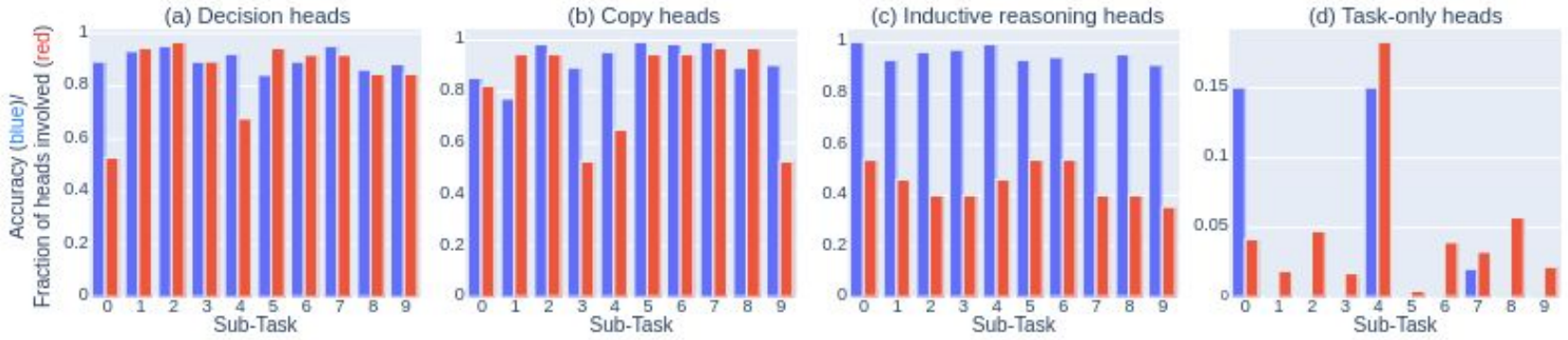
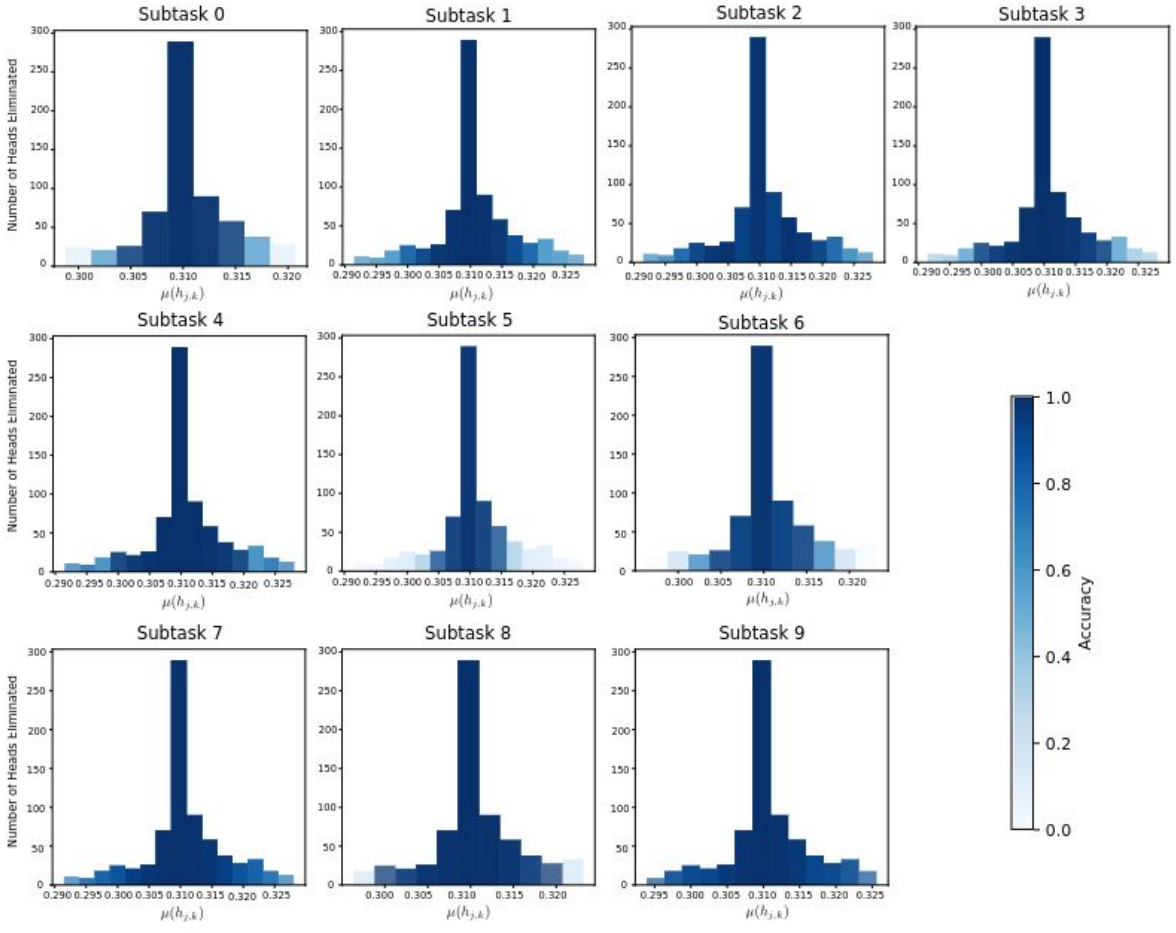


Figure 2: **Performance of attention heads identified for each different subtask type across different subtasks.** We show the performance of (a) decision-making, (b) copy, and (c) inductive reasoning heads for each subtask 0 to 9 (blue bars show accuracy when the rest of the heads are knocked out; red bars denote the fraction of heads involved, see Figure 1 for subtask annotation). (d) Task-only heads are only those that are not shared with other tasks, e.g., only those copying heads for subtask 4 that are not decision-making or inductive reasoning heads. Inductive reasoning heads are consistently functional across all the subtasks with the least number of heads involved.

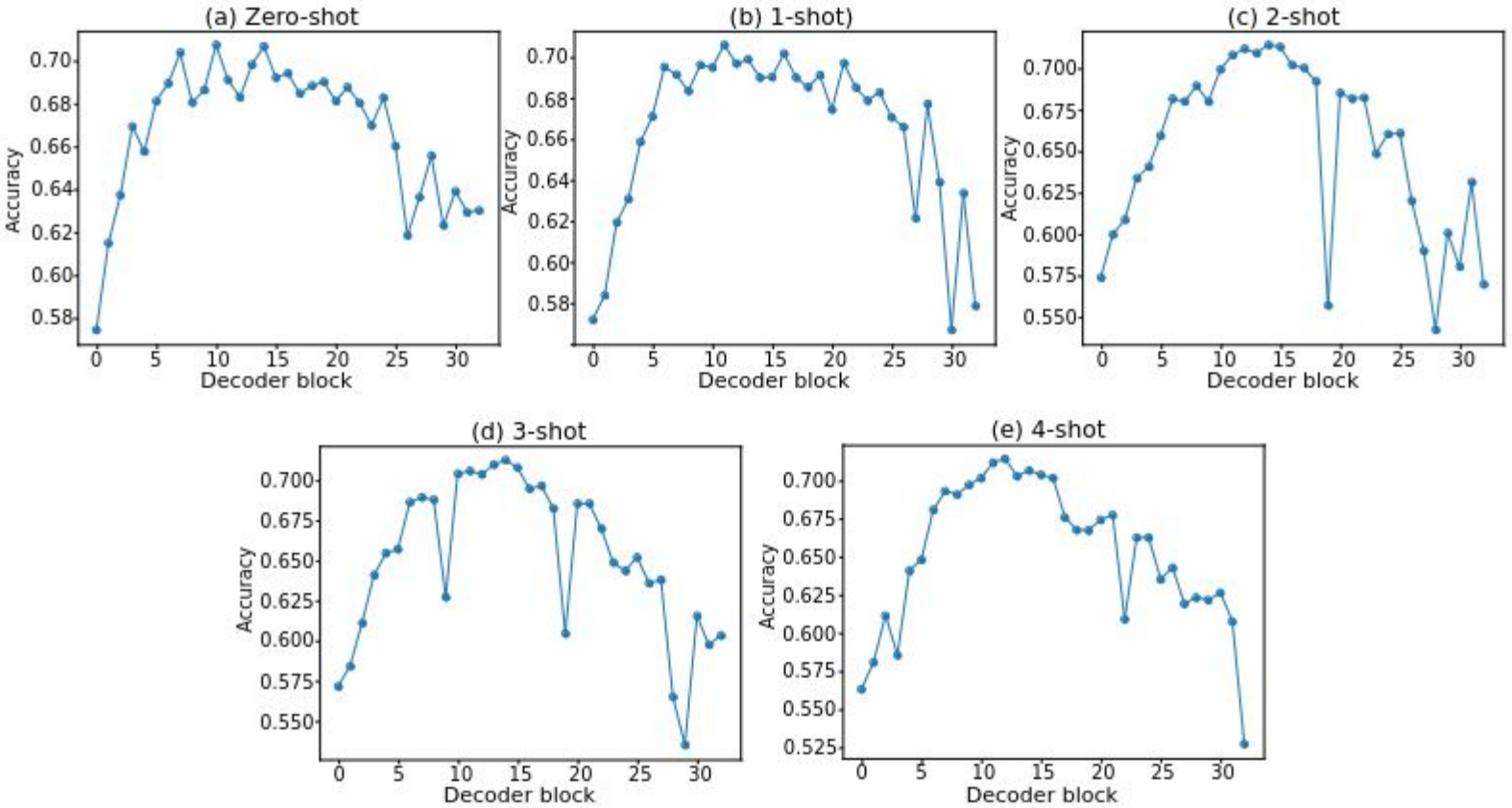
Majority of heads not task specific

- It is evident that these tasks are not structurally well differentiated in the language model
- A good majority of heads share the importance of all three subtasks


Subtask 0
Subtask 1
Subtask 2
Subtask 3
Subtask 4
Subtask 5
Subtask 6
Subtask 7
Subtask 8
Subtask 9
Number of Heads Eliminated
$\mu(h_{j,k})$
Accuracy
1.0
0.8
0.6
0.4
0.2
0.0

Token mixing over fictional ontology

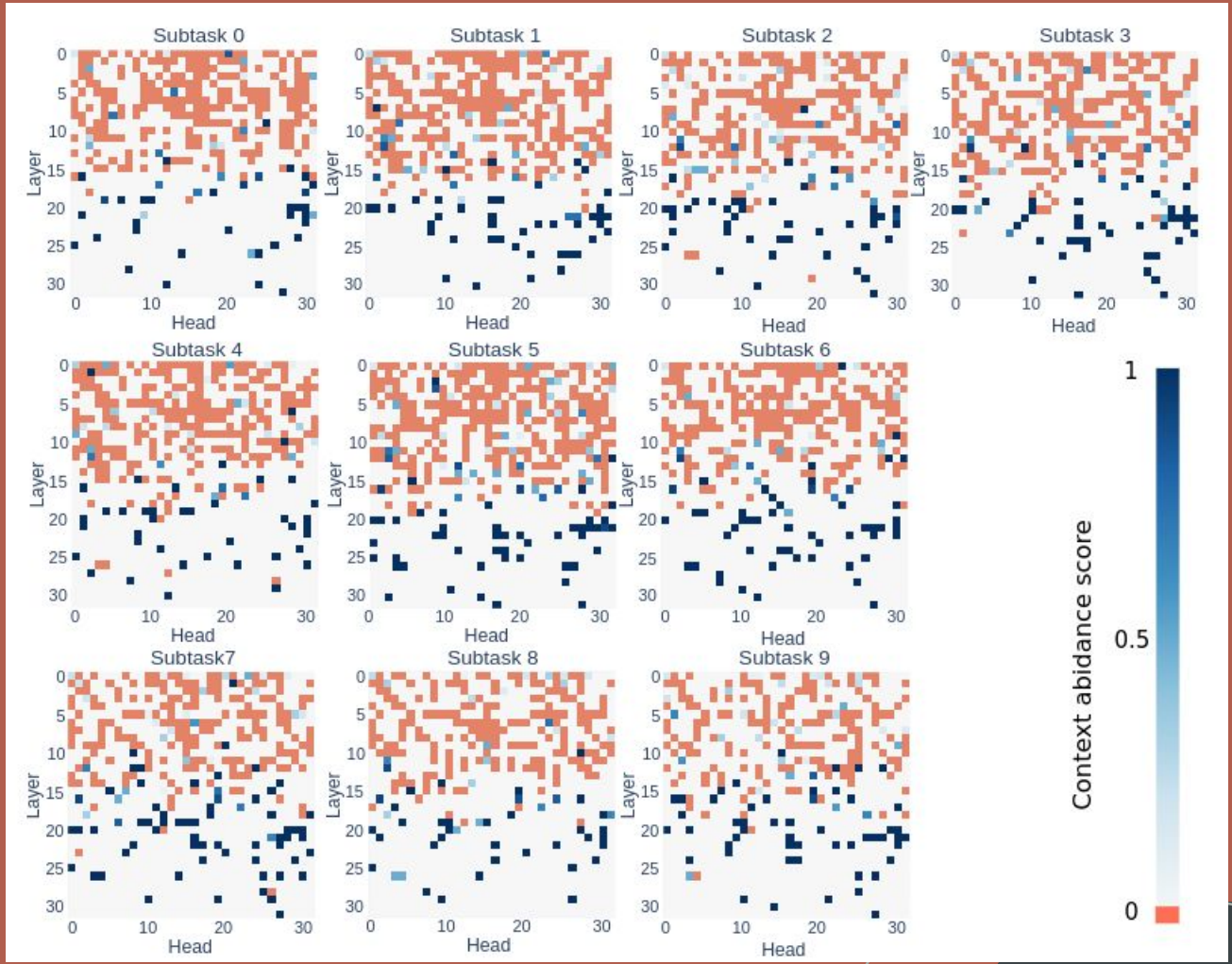
- Given a statement A is B, A is not C
- Concatenate residual streams of A and B and train a classifier
- Linear classifiers - Not good
- Nonlinear Classifiers - Good
- Accuracy increases with depth before it starts falling
- In-context examples could be speeding up the process


(a) Zero-shot
(b) 1-shot)
(c) 2-shot
(d) 3-shot
(e) 4-shot
Accuracy
Decoder block

Context Abidance

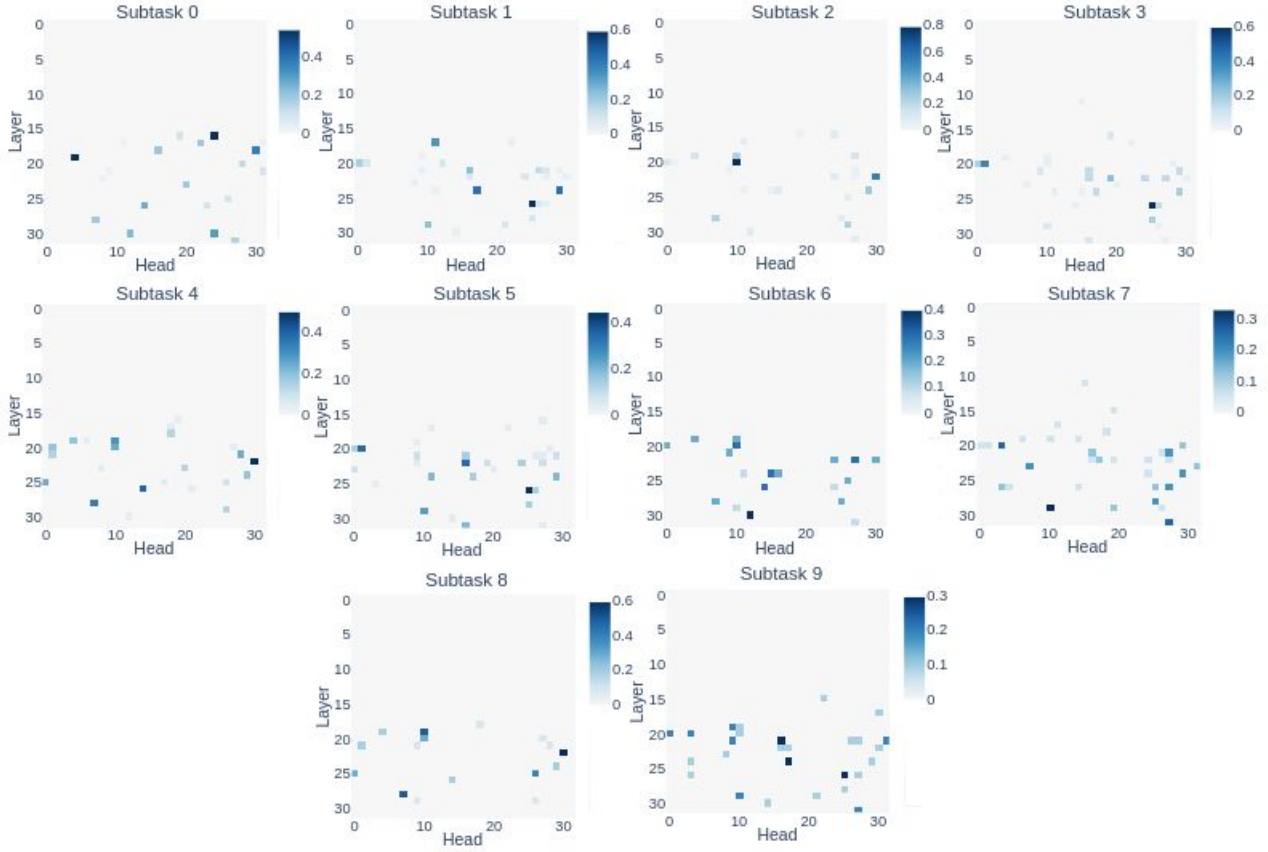
Check for a given token s_i in the sequence S, if s_i_jk, the token projected by the attention head hj,k is such that s_i, s_i_jk is a bigram present in S.

context-abidance score is fraction of attention heads for which above condition is true


Subtask 0
Subtask 1
Subtask 2
Subtask 3
Subtask 4
Subtask 5
Subtask 6
Subtask7
Subtask 8
Subtask 9
Layer
Head
0
5
10
15
20
25
30
Context abidance score
1
0.5
0

Subtask-wise answer generation

- For each attention head, apply unembedding matrix U to its output
- Compute probability of the answer in the attention head output
- LMs exploit multiple pathways to generate the same answer


Subtask 0
Subtask 1
Subtask 2
Subtask 3
Subtask 4
Subtask 5
Subtask 6
Subtask 7
Subtask 8
Subtask 9
Layer
Head

Parallel Pathways for Information Processing

- Do the multiple pathways for answer generation process the answer from the input using the same mechanism?

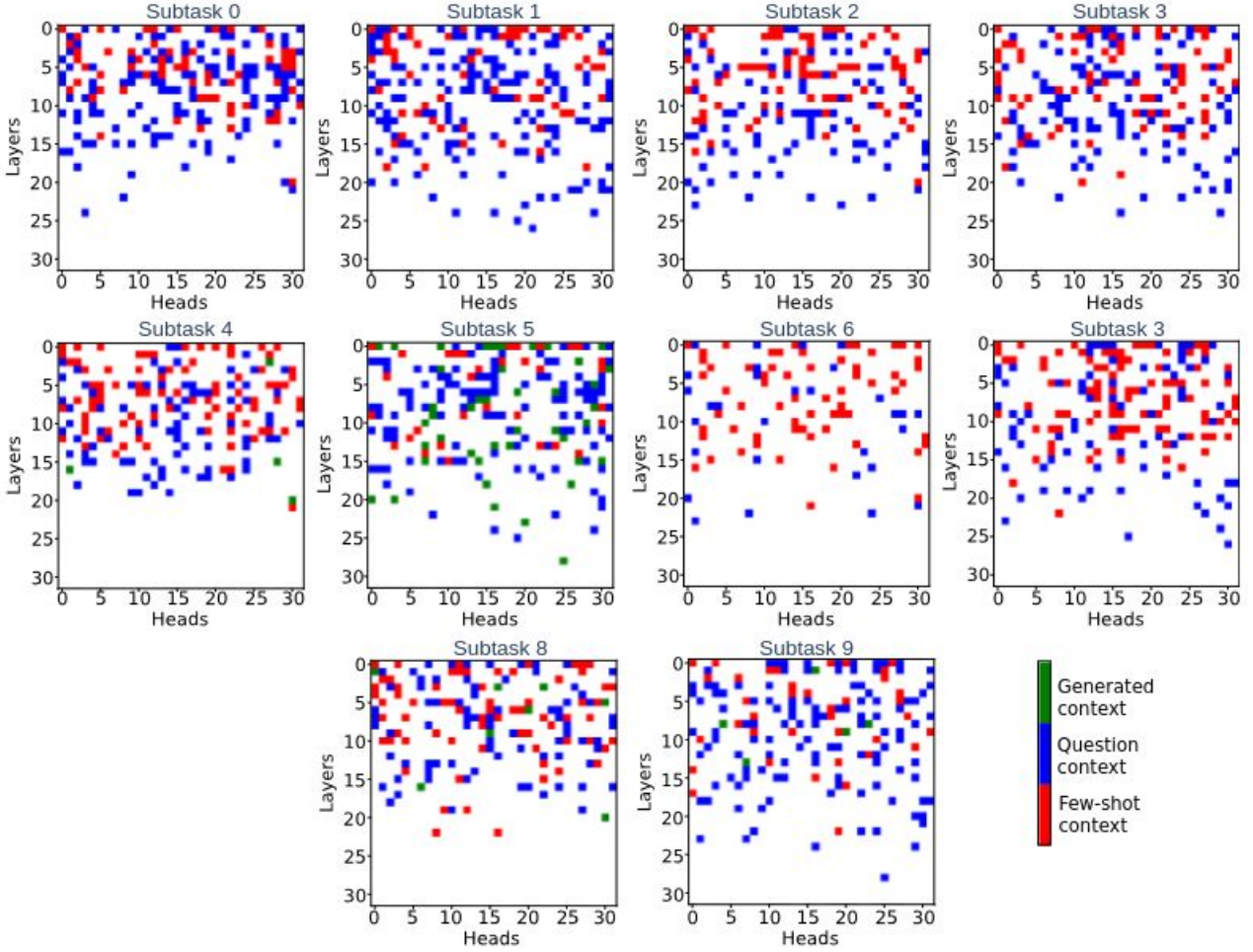
Where do the answer-writing heads collect their answers from?

- Start from the answer-writing heads
- Follow which residual streams that are being attended by these heads
- Identify the content of these residual streams via unembedding projection,
- Identify the heads in the previous layers that are writing that content into those residual streams
- Continue till one of the two conditions is met:

  (i) Reach a head in the first decoder block

  (ii) Reach a residual stream corresponding to the first token in the input token sequence.

  Construct trees of attention heads rooted at the answer writing heads


Subtask 0
Subtask 1
Subtask 2
Subtask 3
Subtask 4
Subtask 5
Subtask 6
Subtask 3
Subtask 8
Subtask 9
Layers
Heads
Generated context
Question context
Few-shot context

Wrapping Up

- Despite different reasoning requirements across different stages of CoT generation, the functional components of the model remain almost the same
- Attention heads perform information movement between ontologically related (or negatively related) tokens
- Multiple different neural pathways are deployed to compute the answer, that too in parallel
- These parallel answer generation pathways collect answers from different segments of the input
- A functional rift at the very middle of the LLM (16th decoder block) when LLM moves from bigram associations memorized via pretraining to following the in-context prior

Questions?