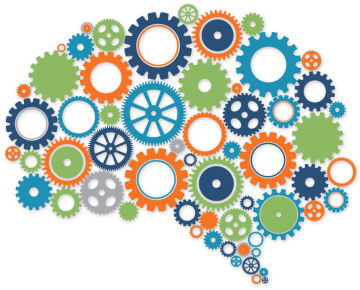


# Critical Thinking & Innovation

Jake Chandler

*Large Language Models, Pt 1*

LA TROBE UNIVERSITY

## Intro

- Search engines:
  - return preexisting documents containing relevant info to answer a question
  - $\Rightarrow$ Users must read info and incorporate it into a reasoned answer
- Large Language Models (LLMs):
  - produce the reasoned answer itself
- LLMs popular since GPT-3.5 (2022)
- Preceded by smaller models: GPT-1/2/3, BERT,…
- Built on important prior developments: embedding, attention,…
- Superseded by newer models: GPT-4, Claude 3.5, …



## Up next

- This lecture:
  - basic architecture and principles
- Next lecture:
  - LLM performance wrt critical thinking
- Presentation owes huge debt to resources cited
- Fast-changing field: much of this will be outdated next year



## Predictive text

- LLMs refine iterated predictive text

**Iterated next-word prediction**

`When I _____`

*was (0.25)*, am (0.20), get (0.15), have (0.10), …

`When I was _____`

*a (0.30)*, young (0.20), in (0.15), at (0.10), …

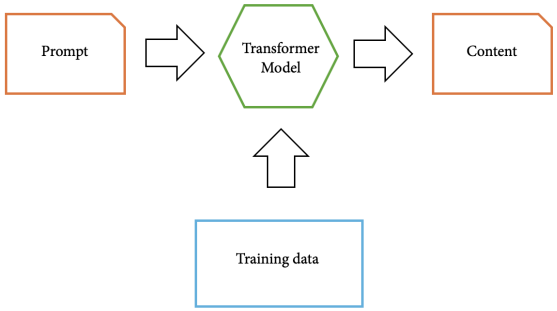
`When I was a _____`

*child (0.25)*, kid (0.20), teenager (0.15), young (0.12), …

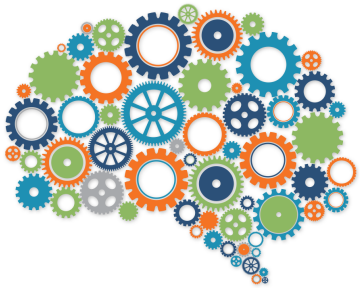
`When I was a child _____`

## GPT models

- Current LLMs are also GPT models:
  - Generative: generate content from user prompts
  - Pretrained: based on prior training,
  - Transformer: using a particular kind of architecture (Vaswani *et al* 2017)





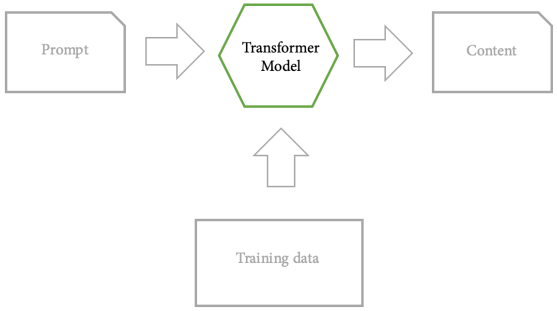
# Critical Thinking & Innovation

Jake Chandler

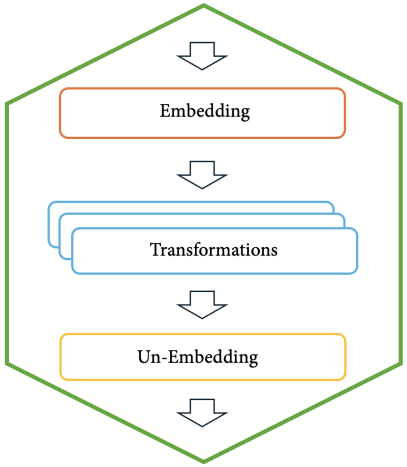
*Large Language Models, Pt 1*

LA TROBE UNIVERSITY

## The Transformer Model



# Inside the model





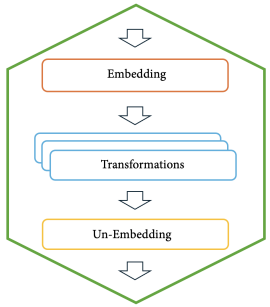

# Inside the model (ctd)

(1) Embedding: Encodes input as sequence of numerical vectors

(2) Transformations: Repeatedly runs vectors in parallel through two steps

   (a) Attention: "Attention heads" adjust each vector based on *other* vectors in sequence

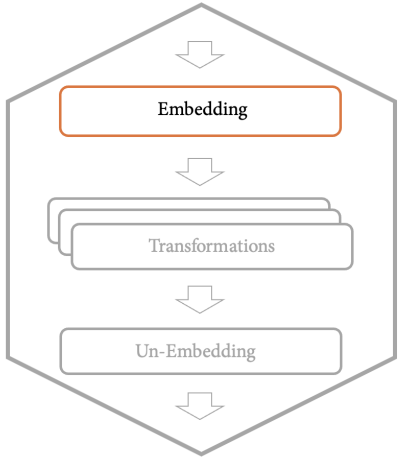
   (b) Feed-forward: "Neural nets" adjust each parameter in each vector based on *other parameters* in same vector




(3) Unembedding: Uses *last* modified vector in sequence to generate next token in sequence



# (1) Embedding





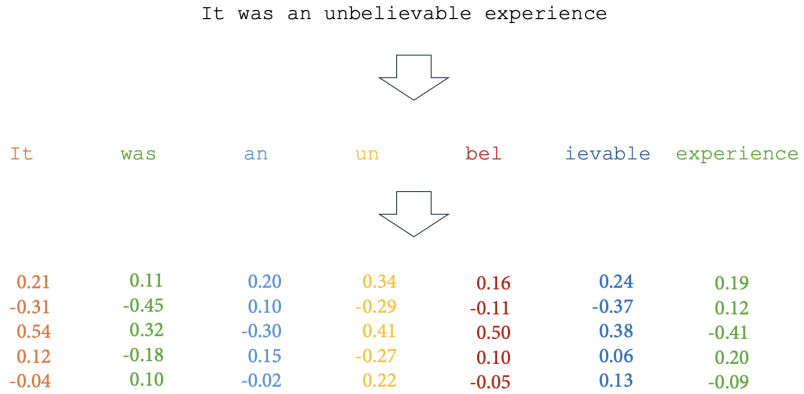

# Embedding

- Input is first broken up into little chunks (tokens)
- Tokens: words or smaller (avg: 1.3 tokens/word)
- Tokens drawn from a predetermined vocabulary
- Each token is assigned an initial list of numbers (vector)

  Vectors typically very large (GPT3: > 12K numbers / vector)
- Max number of tokens that can be handled = context window (model's "short-term memory")

  Current context windows: 128K to 1M tokens (GPT-4o vs Gemini 1.5)

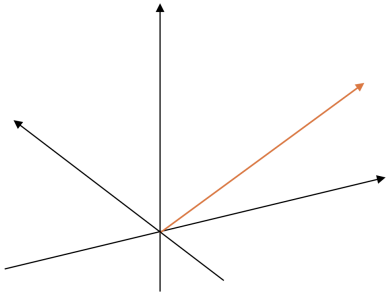
# Embedding (ctd)






# Vectors in space

- Vectors: displacements (with direction + magnitude) in some $n$-dimensional space

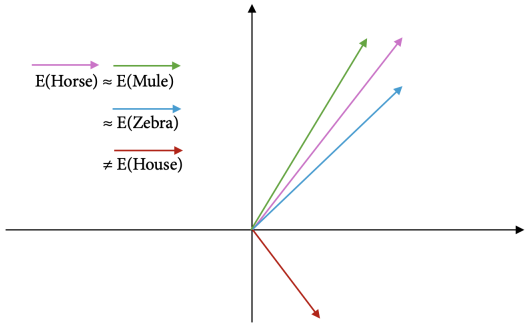


- Typically $n > 3$: hard for us to visualise



# Vectors and meanings

- Vectors seem to encode facts about meaning: Similar meanings $\Leftrightarrow$ similar vectors

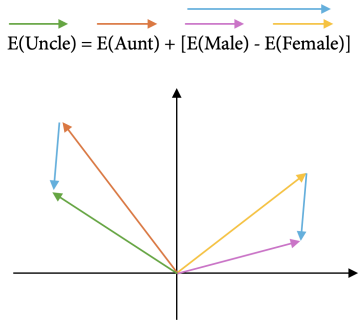



- Initial vector $\approx$ initial guess of meaning (later refined in transformation steps)

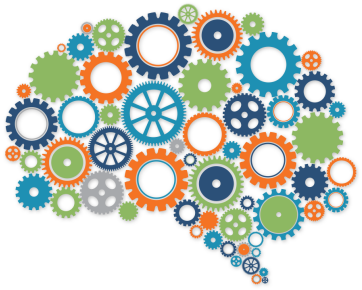


# Vector arithmetic

- Conveniently, vectors can be added, subtracted, multiplied, etc.
- Observed patterns consistent with vectors' encoding meanings




- This facilitates analogical reasoning (an uncle is like a male aunt)



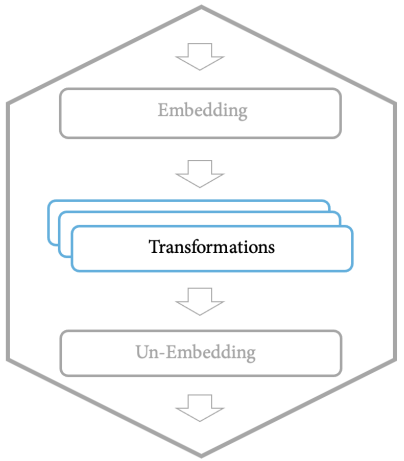
# Critical Thinking & Innovation

Jake Chandler

*Large Language Models, Pt 1*

LA TROBE UNIVERSITY

## (2) Transformation






## Attention

- The ideas that we associate with a word are coloured by the words around it (sentential context)

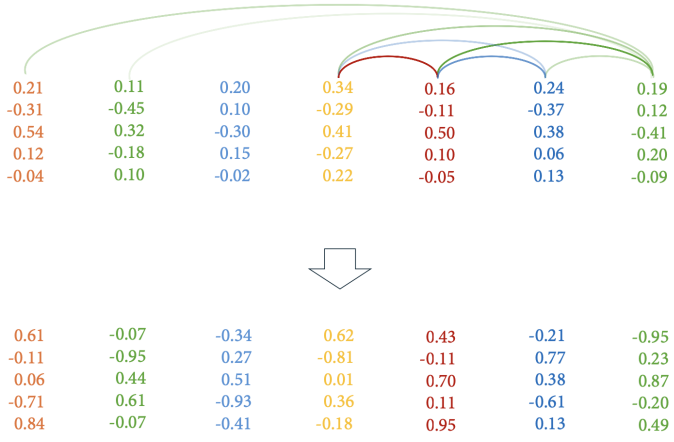
| Context dependence |
| --- |
| John and Mary went shopping. **He** bought a hat. |
| Associated with being called "John", etc. |
| He swung the **bat**. |
| Associated with baseball, etc. (rather than vampires) |
| The small red **car** sped by. |
| Associated with being small and red, etc. |

- The model incorporates, into its understanding of (i.e. vector for) one token, elements of its understanding of surrounding tokens



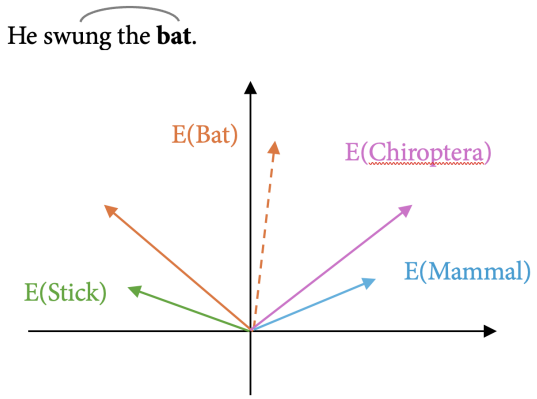
## Attention (ctd)

- Attention heads adjust each vector based on other vectors



## Attention in vector space

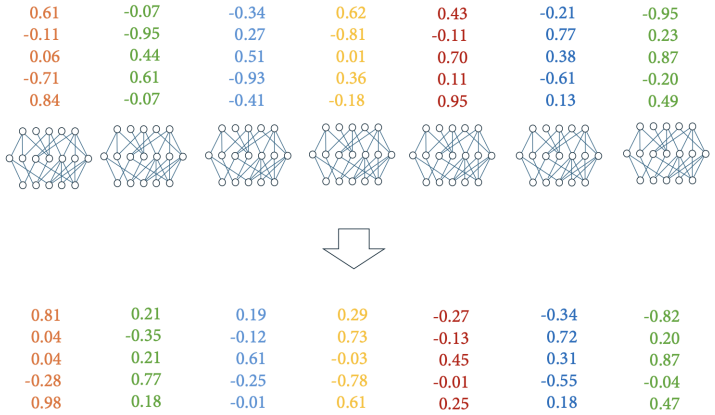
- Spatially, the process nudges the vectors towards more contextually appropriate ones





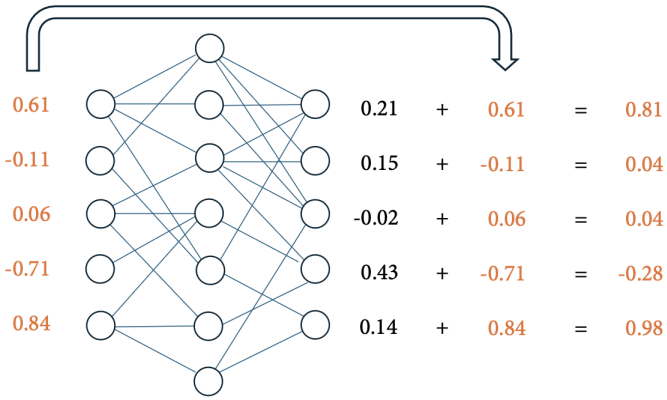

## Feed-forward

- Neural nets adjust each parameter in each vector based on other parameters in same vector






## Feed-forward (ctd)

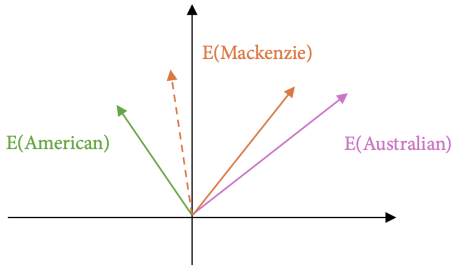





## Feed-forward in vector space

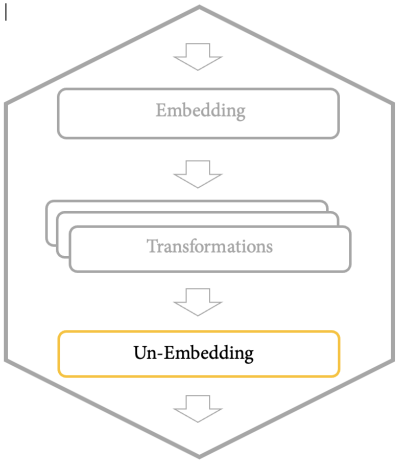
- This also nudges vectors into different positions
- Can draw inferences from info imported during attention phase

As she braced for the penalty kick, **Mackenzie** was shaking.




- Note: Mackenzie Arnold is Goalkeeper for the Matildas

## (3) Unembedding





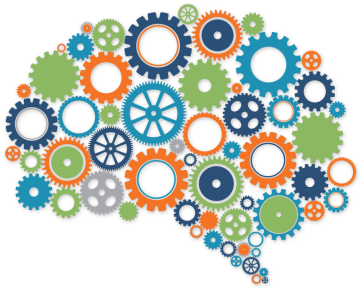

## Unembedding

- Last modified vector in sequence is used to generate a probability distribution over different possible next tokens and one of most probable ones is selected

| | | | | | | |
|---|---|---|---|---|---|---|
| 0.81 | 0.21 | 0.19 | 0.29 | -0.27 | -0.34 | -0.82 |
| 0.04 | -0.35 | -0.12 | 0.73 | -0.13 | 0.72 | 0.20 |
| 0.04 | 0.21 | 0.61 | -0.03 | 0.45 | 0.31 | 0.87 |
| -0.28 | 0.77 | -0.25 | -0.78 | -0.01 | -0.55 | -0.04 |
| 0.98 | 0.18 | -0.01 | 0.61 | 0.25 | 0.18 | 0.47 |

`It was an unbelievable experience` **`that`**





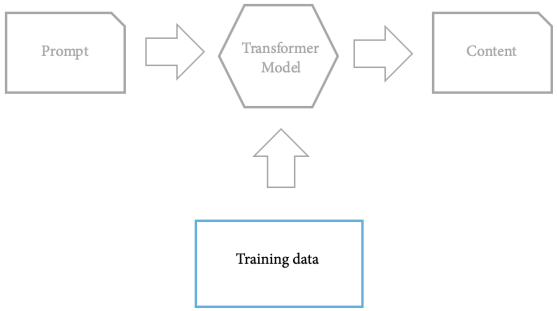
# Critical Thinking & Innovation

Jake Chandler

*Large Language Models, Pt 1*



## Training





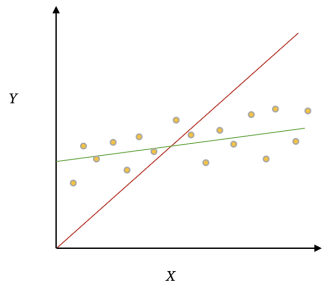

# Training

- GPT models are based on machine learning:
  - Responses aren't initially explicitly encoded
  - Responses are learned from data by tuning parameter values that determine these responses (training phase)
- After training, parameter values remain fixed during use
- *Large* LMs have a *large* number of parameters (GPT-4: > 1 tn)
- They are trained on huge amounts of data (GPT-4: > 1,000 TB)



# Example: gradient descent for linear regression

- Problem: output value of variable $Y$, when we input value of $X$
- Assumption: $X$, $Y$ linearly related ($Y = aX + b$) w. random noise
- 2 parameters: $a$ (slope) and $b$ (intercept)
- Start with random values for $a$ and $b$ (red line)
- Gradually nudge these to minimise error wrt data (green line)

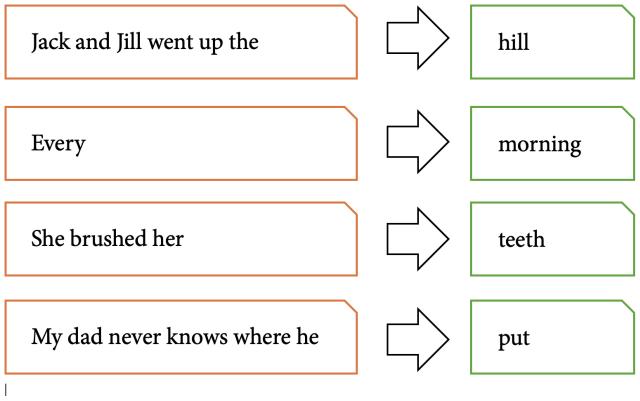





# A two-step process: pre-training

- Step 1: pre-training
  - Provides basic general capability (general language patterns, grammar, some factual knowledge and elementary reasoning)
  - Requires minimal human input
  - Method: Next-word prediction on naturally occurring text (e.g. web pages)

## A two-step process: pre-training (ctd)



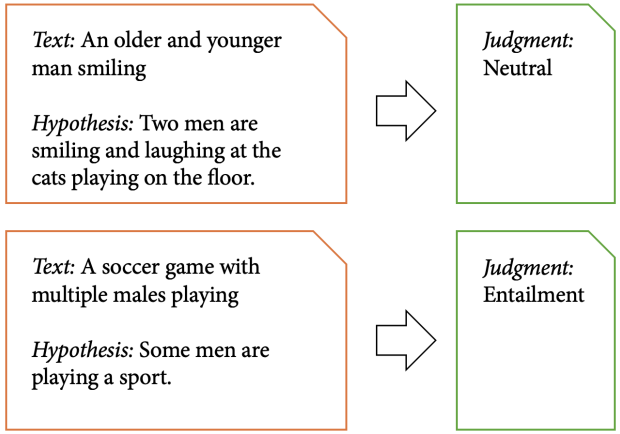



## A two-step process: fine-tuning

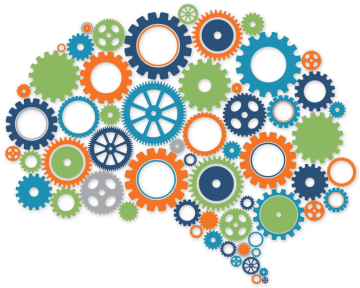
- Step 2: fine-tuning
  - Provides more advanced, task specific capabilities (generating text with specific styles or satisfying specific constraints)
  - Requires substantial human input
  - Methods include prediction on human-labeled data sets (e.g Stanford Natural Language Inference (SNLI) Corpus)
- Note: *You* can fine-tune the latest models yourself!



## A two-step process: fine-tuning (ctd)





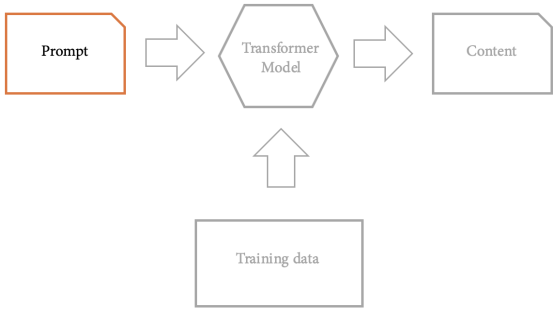



# Critical Thinking & Innovation

Jake Chandler

*Large Language Models, Pt 1*

# Prompting






## Intro

- Output quality hinges massively on prompt quality
- Current understanding of best practise is very limited
- "Prompt engineering": emerging field of study of prompt optimisation (see Schulhof, S. *et al* (2024))
- Prompts comprise
  - directive
  - exemples (optional)
  - "context" (optional)
- Thanks to large context windows, LLMs can refer back to earlier prompts and outputs: enables iterative prompting



## Directives and examples

- Directives: question answering, text summarisation, coding, editing, argument evaluation, creative writing,…
- Zero-shot prompting: directive alone

  Entirely relies on understanding acquired during training

Zero-shot prompting

```
Classify the following sentences as expressing positive,
negative, or neutral sentiments: [insert sentences].
```

## Directives and examples (ctd)

- Few-shot prompting: directive + examples
  Adds info on task, expected format and level of detail of response
- This is known as In-Context Learning (Dong *et al* 2024)
- Note: examples only influence model within context window

**Few-shot**

```
Here are some examples of sentiment classification:

    'I love this movie!' - Positive
    'The weather is cloudy today.' - Neutral
    'This is a complete waste of money.' - Negative

Now, classify the following sentences as expressing
positive, negative, or neutral sentiments: [insert
sentences].
```



## "Context"

- LLMs are only trained on a limited amount of data and up to a certain date (knowledge cutoff; e.g. 2023 for GPT-4o)
- Some further data ("contexts") can be provided to them alongside the prompt
- With larger context windows (e.g. Gemini's 1M tokens), easy to upload a whole book (avg token count 80K-130K)



## Prompt patterns

- Prompt pattern: customisable template for applying same kind of procedure to different situations
- Recommendation: create a prompt pattern library for your most common tasks

**Some prompt patterns**

```
Contrast [context A] with [context B], pinpointing their
major commonalities and distinctions.
Clarify [context] by simplifying its components into easily
understandable terms.
Rewrite [context] in such as way as to be easily
understandable by [insert audience]
```



# Epilogue: Understanding LLMs

## How and what

- Question:

  How exactly do LLMs do what they do?

- The field of mechanistic interpretability is devoted to this
- Similar to neuroscience and only slightly more manageable!
- Wang *et al* (2022): tracing GPT-2's prediction of last word in

  "When Mary and John went to the store, John gave a drink to **Mary**."

  Only explain why "Mary" is chosen rather than "John"
- But another crucial question is

  What exactly can LLMs do in the first place?

- More on this in the next lecture



## Resources

- Lee, T.B. and S. Trott (2023) 'Large language models, explained with a minimum of math and jargon'. Available at `https://www.understandingai.org/p/large-language-models-explained-with`
- Three Blue One Brown illustrated video series on neural networks and GPT models (esp. Chs. 5–7). Available at `https://www.3blue1brown.com/topics/neural-networks`



## References


- Dong, Q. *et al* (2024): 'A Survey on In-context Learning'. Available at: `https://arxiv.org/abs/2301.00234`
- Schulhof, S. *et al* (2024): 'The Prompt Report: A Systematic Survey of Prompting Techniques'. Available at: `https://arxiv.org/abs/2406.06608`
- Vaswani, A. *et al* (2017): 'Attention Is All You Need'. Available at: `https://arxiv.org/abs/1706.03762`.
- Wang, K. *et al* (2022): 'Interpretability in the Wild: a Circuit for Indirect Object Identification in GPT-2 small'. Available at: `https://arxiv.org/abs/2211.00593`