CSE 493s/599s
Lecture 16.

Sewoong Oh

Lecture notes

- These lecture notes are based on other courses in LLMs, including
  - CSE493S/599S at UW by Ludwig Schmidt: https://mlfoundations.github.io/advancedml-sp23/
  - EE-628 at EPFL by Volkan Cevher: https://www.epfl.ch/labs/lions/teaching/ee-628-training-large-language-models/ee-628-slides-2025/
  - ECE381V Generative Models at UT Austin by Sujay Sanghavi
  - and various papers and blogs cited at the end of the slide deck

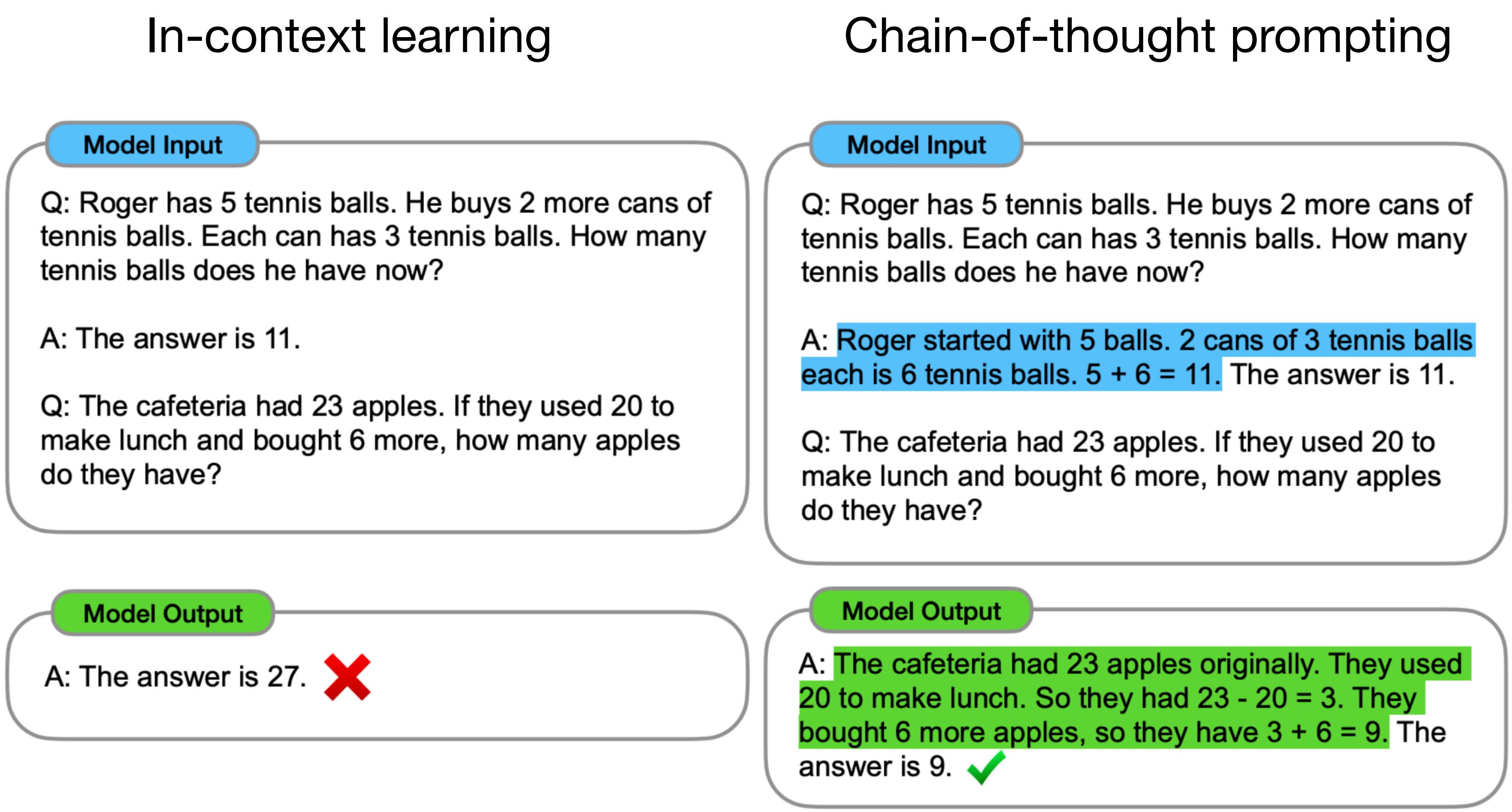
Outline

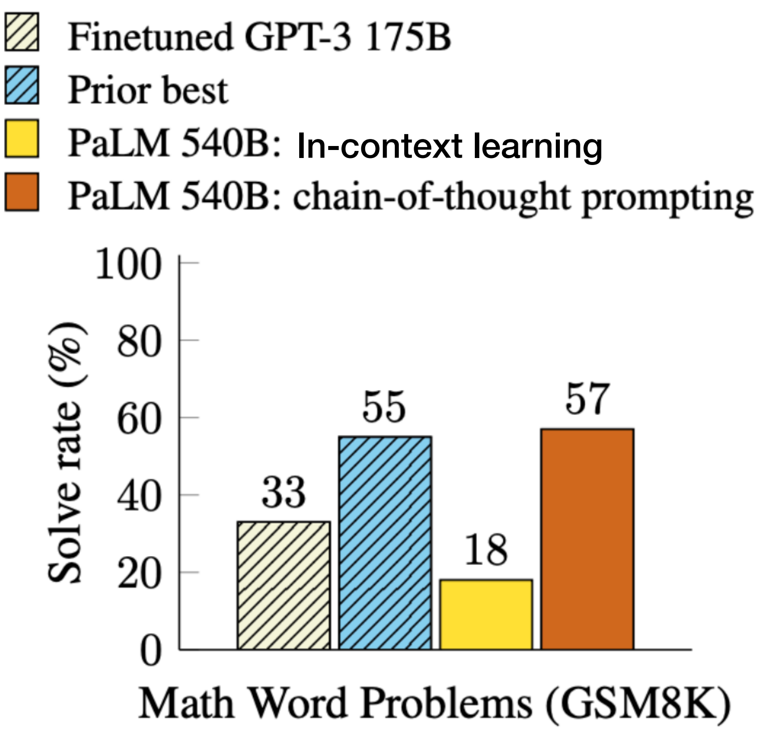
- Language models
- General LLM framework
  - Token processing
  - Sequence mixing
  - Prediction
- Prompting techniques at inference time
  - In-context learning
  - Chain-of-thought prompting
- Fine-tuning

Chain-of-thought prompting [Wei et al. 2022]

Chain-of-thought prompting [Wei et al. 2022]

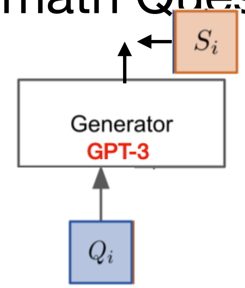
- **In-context learning** provides in the prompt a few paired question and answer examples, which help the LM figure out the context of the given task.
- However, it is not as effective in complex reasoning problems that require logical thinking, e.g., tasks requiring arithmetic/mathematical computation, manipulating symbols, or common sense knowledge.
- **Chain-of-thought** adds intermediate reasoning steps, to attempt to
  - break down the problem into smaller subproblems and
  - provide verifiable or interpretable explanations.
- In-context learning prompts with
  <Question, Answer> + <Question>
- Chain-of thought prompts with
  <Question, **Intermediate results**, Answer> + <Question>


In-context learning
Chain-of-thought prompting
Model Input
Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?
A: The answer is 11.
Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?
Model Output
A: The answer is 27.
Model Input
Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?
A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.
Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?
Model Output
A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had 23 - 20 = 3. They bought 6 more apples, so they have 3 + 6 = 9. The answer is 9.

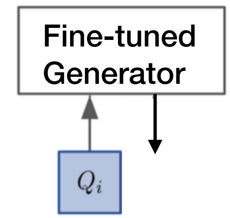


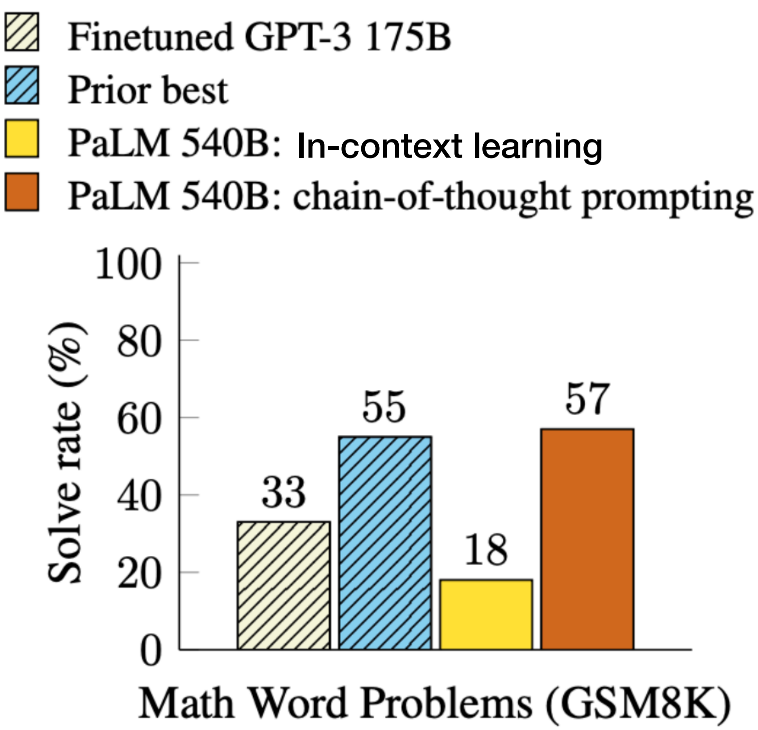
- (Supervised) Fine-tuning: trains the base LM on math Question and Solution pairs.

SFT:

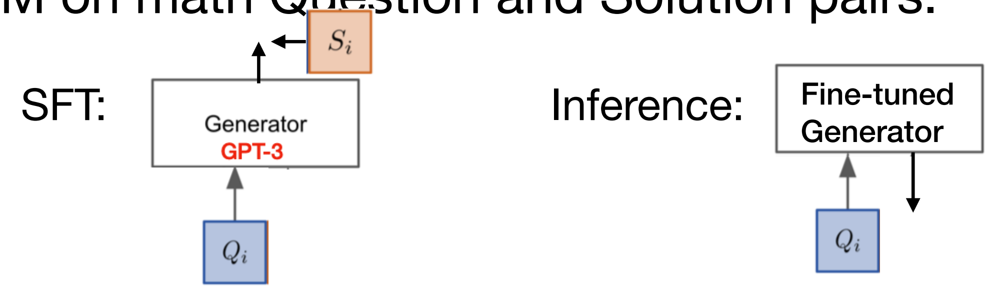


Inference:

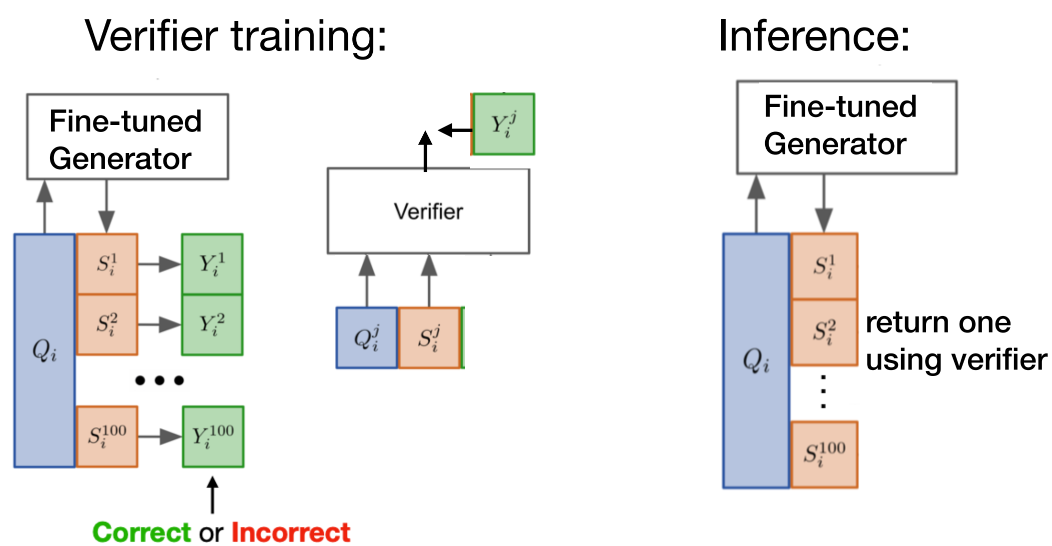




- (Supervised) Fine-tuning: trains the base LM on math Question and Solution pairs.



- Prior Best: is Supervised Fine-Tuning (SFT) followed by a verifier.



- Chain-of-thought prompt examples

Question: Olivia has $23. She bought five bagels for $3 each. How much money does she have left?

Answer: Olivia had 23 dollars. 5 bagels for 3 dollars each will be 5 x 3 = 15 dollars. So she has 23 - 15 dollars left. 23 - 15 is 8. The answer is 8.

Question: If a / b = 3/4 and 8a + 5b = 22,then find the value of a. Answer Choices: (a) 1/2 (b) 3/2 (c) 5/2 (d) 4/2 (e) 7/2

Answer: If a / b = 3/4, then b = 4a / 3. So 8a + 5(4a / 3) = 22. This simplifies to 8a + 20a / 3 = 22, which means 44a / 3 = 22. So a is equal to 3/2. The answer is **(b)**.

- **Arithmetic reasoning** benchmarks

GSM8K

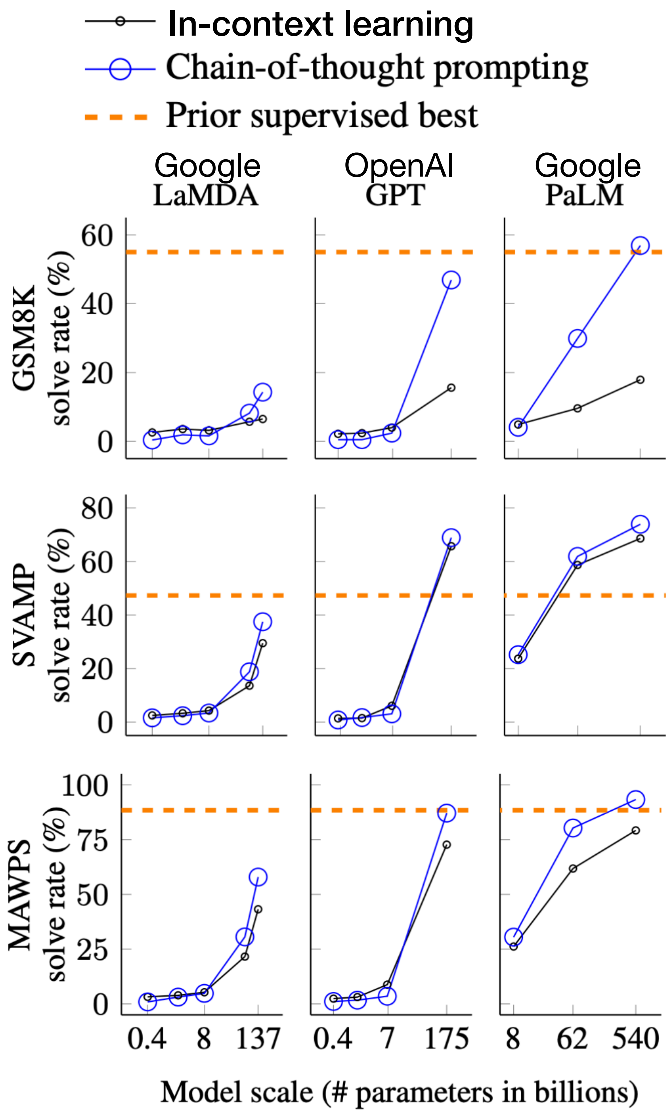
Josh decides to try flipping a house. He buys a house for $80,000 and then puts in $50,00 in repairs. This increased the value of the house by 150%. How much profit did he make?

SVAMP

Each pack of dvds costs 76 dollars. If there is a discount of 25 dollars on each pack. How much do you have to pay to buy each pack?

MAWPS

The school cafeteria ordered 42 red apples and 7 green apples for students lunches. But, if only 9 students wanted fruit, how many extra did the cafeteria end up with?



- **Arithmetic reasoning** benchmarks

GSM8K

Josh decides to try flipping a house. He buys a house for $80,000 and then puts in $50,00 in repairs. This increased the value of the house by 150%. How much profit did he make?

SVAMP

Each pack of dvds costs 76 dollars. If there is a discount of 25 dollars on each pack. How much do you have to pay to buy each pack?

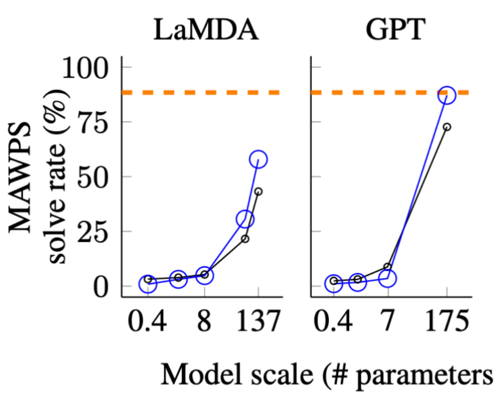
MAWPS

The school cafeteria ordered 42 red apples and 7 green apples for students lunches. But, if only 9 students wanted fruit, how many extra did the cafeteria end up with?

- MAWPS breakdown
    - Chain-of-thought is particularly effective on **complex tasks**

Model		SingleOp		SingleEq		AddSub		MultiArith	
		standard	CoT	standard	CoT	standard	CoT	standard	CoT
UL2	20B	24.9	**27.2**	18.0	**20.2**	18.5	18.2	5.0	**10.7**
LaMDA	420M	2.8	1.0	2.4	0.4	1.9	0.7	5.8	1.5
	2B	4.6	4.1	2.4	3.3	2.7	3.2	5.8	1.8
	8B	8.0	7.0	4.5	4.4	3.4	5.2	5.2	2.4
	68B	36.5	**40.8**	23.9	**26.0**	17.3	**23.2**	8.7	**32.4**
	137B	73.2	**76.2**	48.8	**58.7**	43.0	**51.9**	7.6	**44.9**
GPT	350M	3.2	1.8	2.0	0.2	2.0	1.5	2.3	0.8
	1.3B	5.3	3.0	2.4	1.6	2.3	1.5	2.2	0.5
	6.7B	13.5	3.9	8.7	4.9	8.6	2.5	4.5	2.8
	175B	90.9	88.8	82.7	**86.6**	83.3	81.3	33.8	**91.7**

- MAWPS breakdown
    - Chain-of-thought is particularly effective on **complex tasks**
    - Both In-Context Learning and Chain-of-thought is an **emergent ability** that requires sufficiently large model



Model		SingleOp		SingleEq		AddSub		MultiArith	
		standard	CoT	standard	CoT	standard	CoT	standard	CoT
UL2	20B	24.9	**27.2**	18.0	**20.2**	18.5	18.2	5.0	**10.7**
LaMDA	420M	2.8	1.0	2.4	0.4	1.9	0.7	5.8	1.5
	2B	4.6	4.1	2.4	3.3	2.7	3.2	5.8	1.8
	8B	8.0	7.0	4.5	4.4	3.4	5.2	5.2	2.4
	68B	36.5	**40.8**	23.9	**26.0**	17.3	**23.2**	8.7	**32.4**
	137B	73.2	**76.2**	48.8	**58.7**	43.0	**51.9**	7.6	**44.9**
GPT	350M	3.2	1.8	2.0	0.2	2.0	1.5	2.3	0.8
	1.3B	5.3	3.0	2.4	1.6	2.3	1.5	2.2	0.5
	6.7B	13.5	3.9	8.7	4.9	8.6	2.5	4.5	2.8
	175B	90.9	88.8	82.7	**86.6**	83.3	81.3	33.8	**91.7**

- Chain-of-thought prompt examples

Last letter concatenation

Question: Take the last letters of the words in "Elon Musk" and concatenate them

Answer: The last letter of "Elon" is "n". The last letter of "Musk" is "k". Concatenating them is "nk".

The answer is **nk**.

Coin Flip

Question: A coin is heads up. Jamey flips the coin. Teressa flips the coin. Is the coin still heads up?

Answer: The coin was flipped by Jamey and Teressa. So the coin was flipped 2 times, which is an even number. The coin started heads up, so after an even number of flips, it will still be heads up. So the answer is **yes**.

- **Symbolic reasoning** benchmarks

In-Domain

Take the last letters of the words in "**Elon Musk**" and concatenate them.

Out-of-Domain

Take the last letters of the words in "**Johann Sebastian Bach**" and concatenate them.

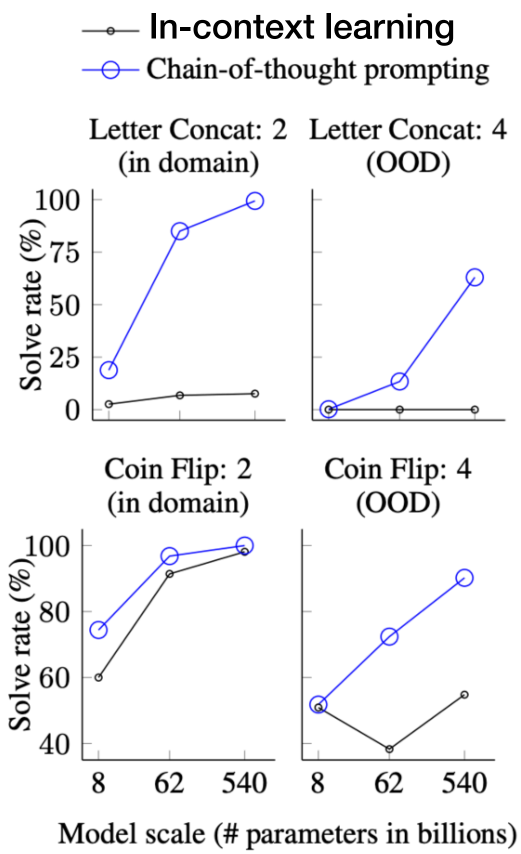
In-Domain

A coin is heads up. **Tom does not flip the coin. *Mike does not flip the coin.*** Is the coin still heads up?

Out-of-Domain

A coin is heads up. **Tom does not flip the coin. *Mike does not flip the coin.* Jake flips the coin.** Is the coin still heads up?

- Chain-of-thought prompting generalizes to longer sequences



- **Symbolic reasoning** benchmarks

In-Domain

Take the last letters of the words in "**Elon Musk**" and concatenate them.

Out-of-Domain

Take the last letters of the words in "**Johann Sebastian Bach**" and concatenate them.

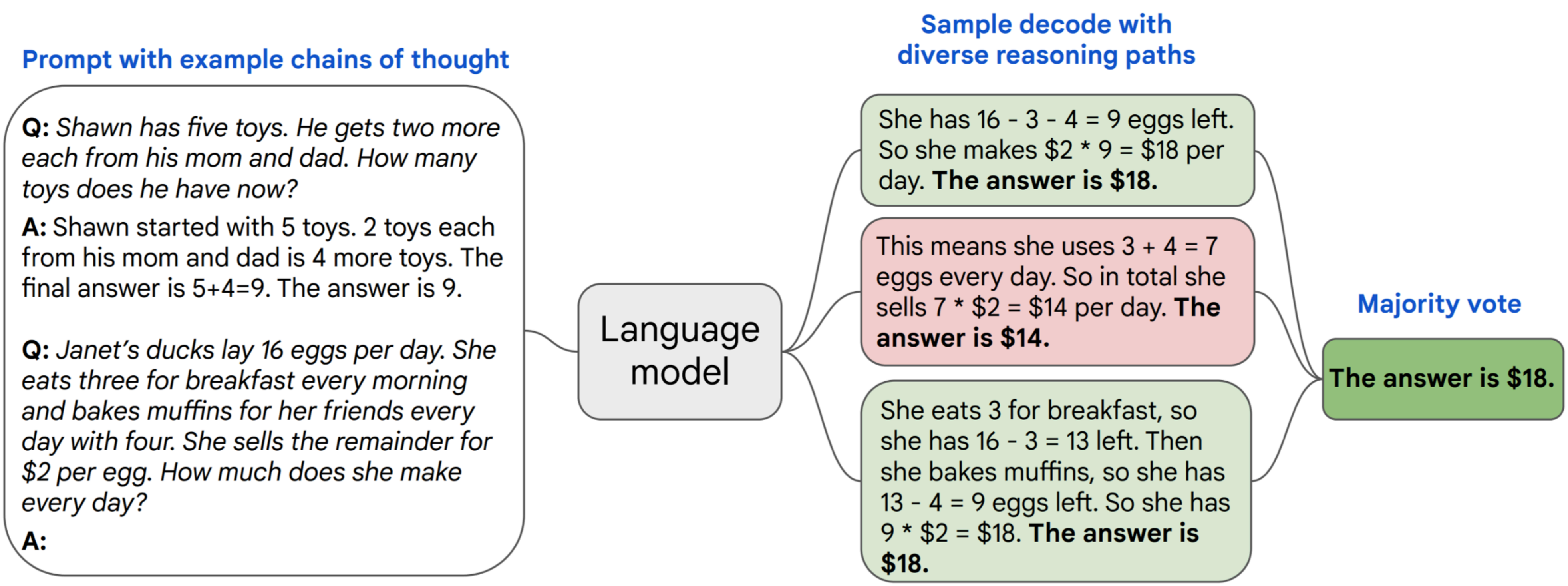
In-Domain

A coin is heads up. **Tom does not flip the coin.** ***Mike does not flip the coin.*** Is the coin still heads up?

Out-of-Domain

A coin is heads up. **Tom does not flip the coin.** ***Mike does not flip the coin.*** **Jake flips the coin.** Is the coin still heads up?

- **Chain-of-thought with Self Consistency** [Wang et al. 2023]
  - borrowing ideas from prior best approach, CoT with self-consistency samples multiple random outputs and chooses the most consistent one.



- **Chain-of-thought with Self Consistency**
    - samples multiple random outputs and chooses the most consistent one.

	Method	GSM8K
	Previous SoTA	35[e] / 57[g]
LaMDA (137B)	Greedy decode (Single-path)	17.1
	Self-Consistency (Multi-path)	27.7 (+10.6)
PaLM (540B)	Greedy decode (Single-path)	56.5
	Self-Consistency (Multi-path)	**74.4** (+17.9)

- How is Chain-of-thought prompting used today?
    - CoT is used quite often as prompt engineering is popular due to its efficiency in many applications.
- Let's consider one popular example of **ReAct** (Reasoning and Acting) framework [Yao et al. 2025] for building **LM agents**: LMs that can plan and perform tasks using external tools
- ReAct prompting is an advanced Chain-of-thought that also uses tools.
- We are interested in challenging factuality and knowledge tasks, e.g., **FRAMES Benchmark** [Krishna et al. 2024]:

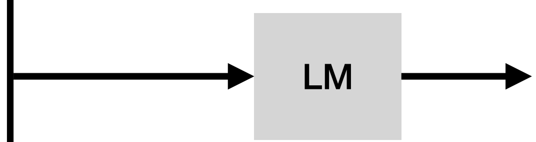
As of August 1, 2024, which country were holders of the FIFA World Cup the last time the UEFA Champions League was won by a club from London?

The Basibasy mine is located in Madagascar. This mine is abundant in a specific chemical element that was discovered for the first time in 1791. The person who discovered this element was born on what is now known as a major US holiday - what holiday is this?

- If no access to a search engine is allowed, one might use **Chain-of-thought prompting**:

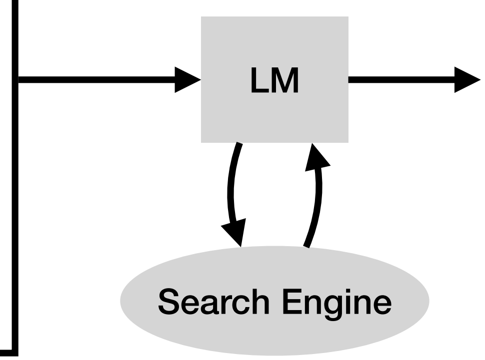
Question: Which magazine was started first, Arthur’s Magazine or First for Women?
Thought: Arthur’s Magazine was started in 1844. First for Women was started in 1989. 1844 (Arthur’s Magazine) < 1989 (First for Women), so Arthur’s Magazine was started first.
Answer: Arthur’s Magazine

Question: As of August 1, 2024, which country were holders of the FIFA World Cup the last time the UEFA Champions League was won by a club from London?



- **ReAct prompting** uses the following template that calls **external “tools”** in a multi-turn interaction of an LM and its tools:

Question: Which magazine was started first, Arthur’s Magazine or First for Women?
Thought 1: Need to search Arthur’s Magazine and First for Women, and find which was started first.
Action 1: Search[Arthur’s Magazine]
Observation 1: Arthur’s Magazine (1844-1846) was an American literary periodical in Philadelphia …
Thought 2: Arthur’s Magazine was started in 1844. I need to search First for Women next.
Action 2: Search[First for Women]
Observation 2: First for Women is a woman’s magazine published by Bauer Media group - was started in 1989 …
Thought 3: First for Women was started in 1989. 1844 (Arthur’s Magazine) < 1989 (First for women), so Arthur’s Magazine was started first.
Action 3: Finish[Arthur’s Magazine]
Question: As of August 1, 2024, which country were holders of the FIFA World Cup the last time the UEFA Champions League was won by a club from London?



- OpenDeepSearch [Alzubi et al. 2025] is an open-source agent framework with search tools built upon ReAct prompting.
- OpenDeepSearch with ReAct prompting significantly improves performance of open-source base model of DeepSeek-R1.

	State-of-the-art	FRAMES accuracy (%)
Closed-source	Perplexity	42.4
	Perplexity Sonar Reasoning Pro	44.4
	GPT-4o Search	65.6
Open-source	DeepSeek-R1	30.1
	LLaMA-3.1-70B	34.3
	OpenDeepSearch with ReAct	**56.7**

Question: What year was the band leader of the group who originally performed the song sampled in Kanye West's song "Power" born?

Answer: 1946

Perplexity response:

The song "Power" by Kanye West samples several tracks, including "21st Century Schizoid Man" by King Crimson. The lead vocalist and bassist of King Crimson during the time of that song was Greg Lake. Greg Lake, the band leader and original lead vocalist of King Crimson for the song "21st Century Schizoid Man," was born on November 10, 1947[2][4].

- OpenDeepSearch [Alzubi et al. 2025] is an open-source agent framework with search tools built upon ReAct prompting.

	State-of-the-art	FRAMES accuracy (%)
Closed-source	Perplexity	42.4
	Perplexity Sonar Reasoning Pro	44.4
	GPT-4o Search	65.6
Open-source	DeepSeek-R1	30.1
	LLaMA-3.1-70B	34.3
	OpenDeepSearch with ReAct	**56.7**

Question: What year was the band leader of the group who originally performed the song sampled in Kanye West's song "Power" born?

Answer: 1946

OpenDeepSearch with LLaMA3.1-70B response:
'Initial_Thought':
To answer this question, I\'ll need to follow these steps:
1. Identify the song sampled in Kanye West\'s song "Power".
2. Identify the band that originally performed the sampled song.
3. Identify the band leader of the original band.
4. Find the birth year of the band leader.

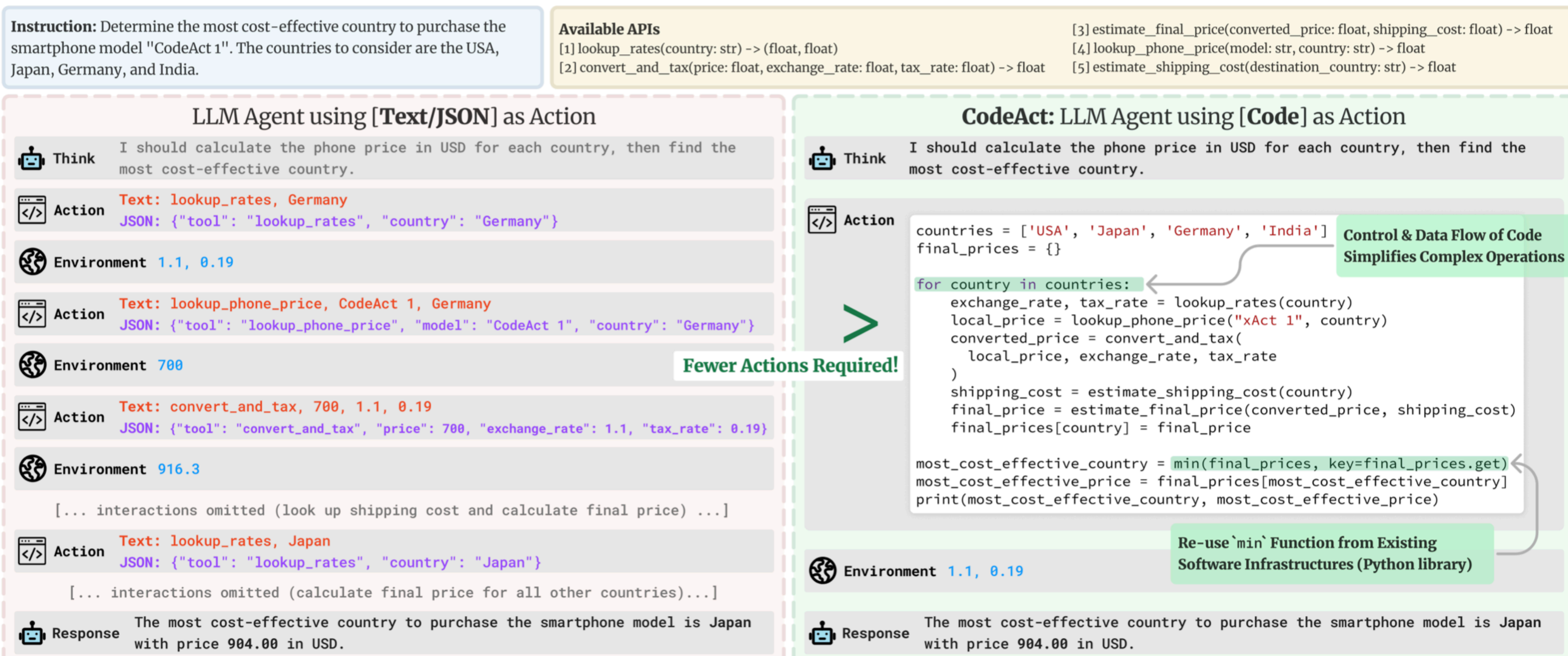
…. according to the context, the song sampled is "21st Century Schizoid Man" by King Crimson. … According to the context, the founder of King Crimson is Emilio Castillo, but this is not correct. After re-reading the context, I found that the founder of King Crimson is not explicitly mentioned, ... However, I can infer that the band leader of King Crimson is likely Robert Fripp,.... But I couldn\'t find any information about Robert Fripp\'s birth year in the given context.

'Action': 'search_internet', 'Action_Input': 'Robert Fripp birth year'
'Thought': … the band leader of King Crimson, was born on May 16, 1946.',

- By switching from **ReAct** prompting to **CodeAct** prompting, OpenDeepSearch surpasses the SOTA closed-source counterparts.

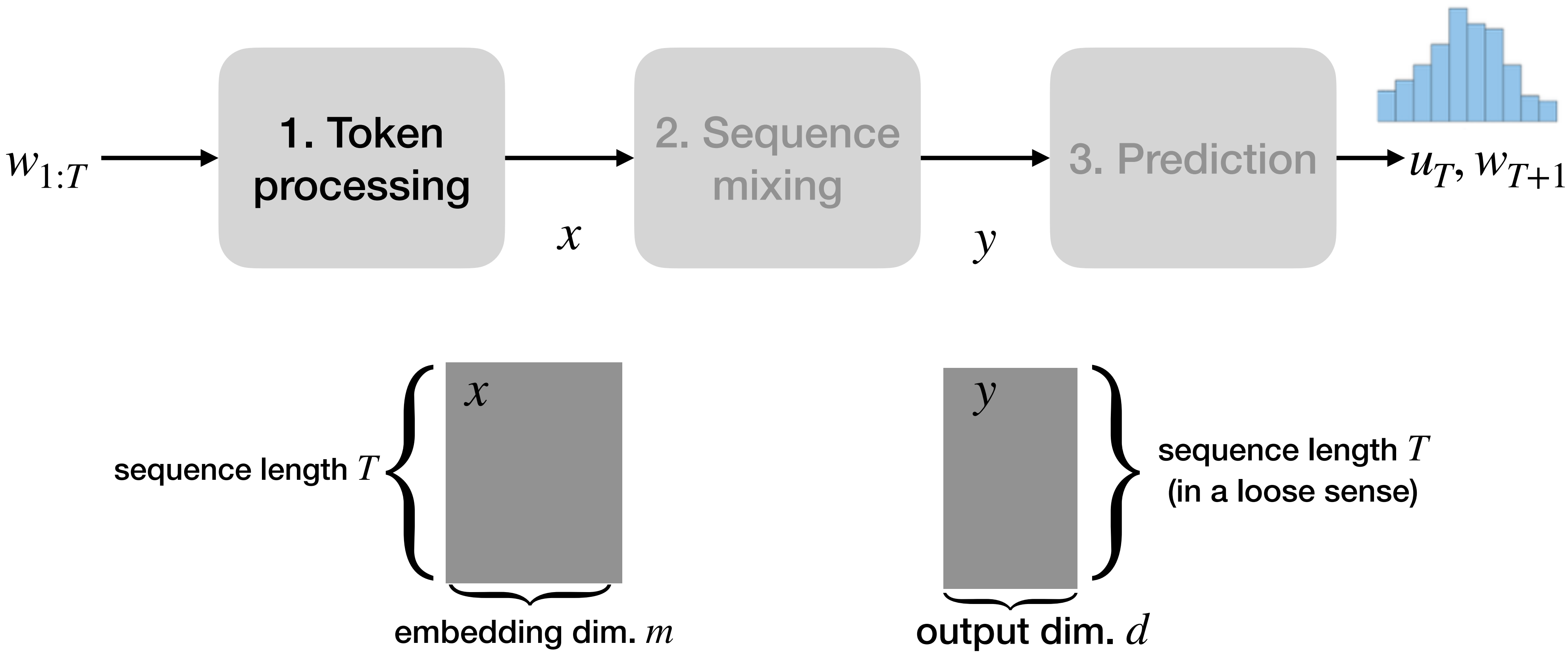
	State-of-the-art	FRAMES accuracy (%)
Closed-source	Perplexity	42.4
	Perplexity Sonar Reasoning Pro	44.4
	GPT-4o Search	65.6
Open-source	DeepSeek-R1	30.1
	LLaMA-3.1-70B	34.3
	OpenDeepSearch with ReAct	56.7
	OpenDeepSearch with CodeAct	**75.3**

- **CodeAct** prompting [Wang et al. 2024] reasons in Python executable code:



- Chain-of-thought prompting is commonly used in various applications of LMs.
- It breaks down the reasoning into multiple steps and provides explanation that is interpretable and verifiable.
- Various advances has been made including, Tree-of-thought, ReAct Prompting, and CodeAct prompting.

Prediction



$w_{1:T}$
1. Token processing
x
2. Sequence mixing
y
3. Prediction
u_T, w_{T+1}
x
sequence length T
embedding dim. m
y
sequence length T
(in a loose sense)
output dim. d

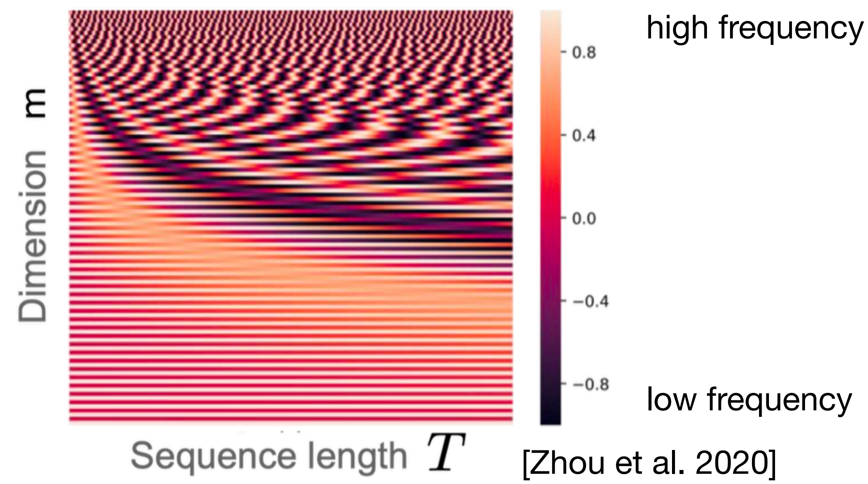
- **Positional embeddings** are designed to capture the positions of the input sequence of words to a transformer.
- Without positional encoding, the original transformer treats the input as a set, in which case the following two inputs are treated the same (more to follow when we learn architectures):

“I am happy”   vs.   “Am I happy”

- To take into account the order of the input, **absolute positional embedding** represents a word by concatenating its learned semantic embedding with an absolute position of the word.

- Absolute positional embedding **solution 1**: original transformer paper [Vaswani et al. 2017] proposes using alternating sin() and cos() functions of decreasing frequencies at position index t, **added** to the vector word embedding:

$$x_t = \mathrm{WordEmb}(w_t) + \underbrace{\begin{bmatrix} \sin(t/10000^{0/m}) \\ \cos(t/10000^{0/m}) \\ \sin(t/10000^{2/m}) \\ \cos(t/10000^{2/m}) \\ \vdots \\ \sin(t/10000^{1/2}) \\ \cos(t/10000^{1/2}) \end{bmatrix}}_{\text{position embedding}}$$



[Zhou et al. 2020]

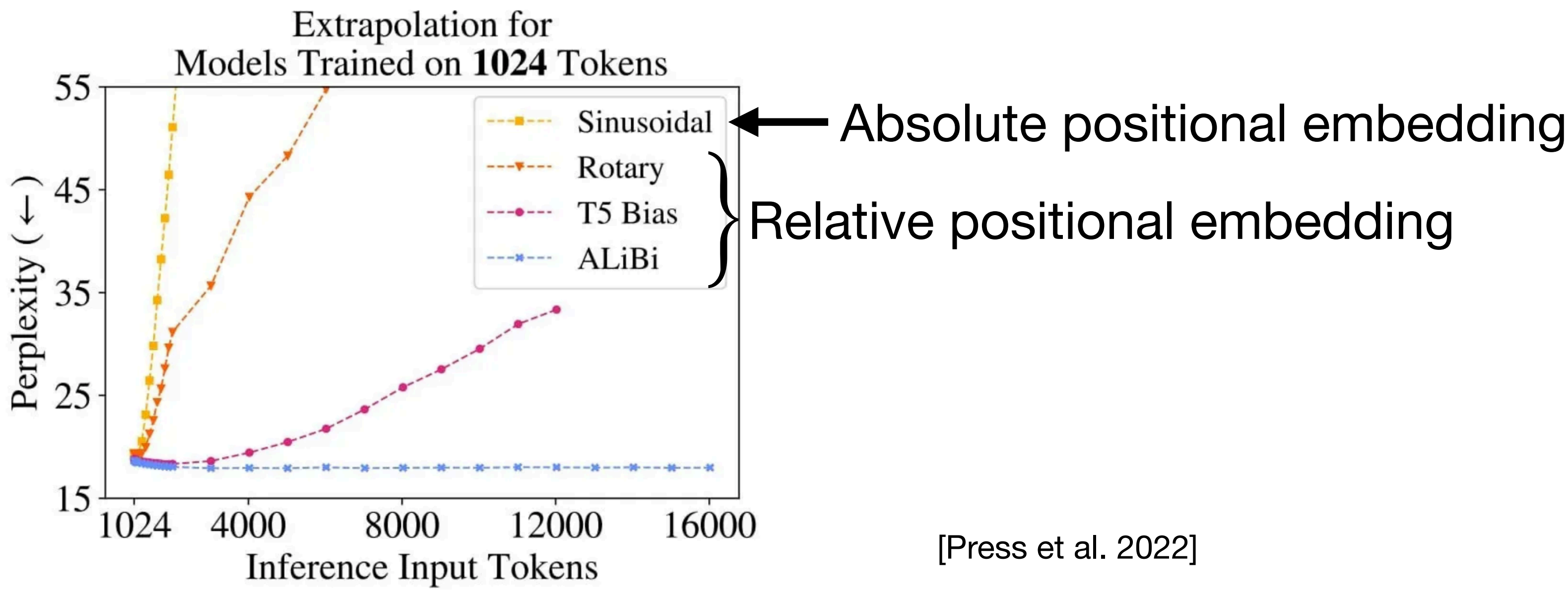
- Absolute positional embedding **solution 2**: learned positional embedding.

$$x_t = \mathrm{WordEmb}(w_t) + \mathrm{PosEmb}(t)$$

- Empirical performance is similar for the two absolute positional embeddings

- Learned positional embedding is popular in vision transformers

- Absolute positional embeddings encode the absolute position of the word in the sequence, which has two problems:
    - it is hard to extrapolate to sequence lengths unseen during training,



[Press et al. 2022]

    - and relative position is as important as absolute position, for example, “happy new ?” appearing in positions (1,2,3) have similar meaning as appearing in positions (500,501,502).

- Relative positional encoding addresses both: generalize to sequences of unseen lengths by relying on the pairwise distances between two words.

Key

Query

To solve machine translation Google introduced <u>transformer</u> …

	x_1	x_2	x_3	x_4	x_5	x_6	x_7
To	<q1,k1>						
solve	<q2,k1>	<q2,k2>					
machine	<q3,k1>	<q3,k2>	<q3,k3>				
translation	<q4,k1>	<q4,k2>	<q4,k3>	<q4,k4>			
Google	<q5,k1>	<q5,k2>	<q5,k3>	<q5,k4>	<q5,k5>		
introduced	<q6,k1>	<q6,k2>	<q6,k3>	<q6,k4>	<q6,k5>	<q6,k6>	
<u>transformer</u>	<q7,k1>	<q7,k2>	<q7,k3>	<q7,k4>	<q7,k5>	<q7,k6>	<q7,k7>
…							

- Recall the Relevance matrix from self-attention, defined by inner-products of keys and queries.

Query \ Key	To x_1	solve x_2	machine x_3	translation x_4	Google x_5	introduced x_6	<u>transformer</u> x_7 …
To	<q1,k1> + b0						
solve	<q2,k1> + b1	<q2,k2> + b0					
machine	<q3,k1> + b2	<q3,k2> + b1	<q3,k3> + b0				
translation	<q4,k1> + b3	<q4,k2> + b2	<q4,k3> + b1	<q4,k4> + b0			
Google	<q5,k1> + b4	<q5,k2> + b3	<q5,k3> + b2	<q5,k4> + b1	<q5,k5> + b0		
introduced	<q6,k1> + b5	<q6,k2> + b4	<q6,k3> + b3	<q6,k4> + b2	<q6,k5> + b1	<q6,k6> + b0	
<u>transformer</u>	<q7,k1> + b6	<q7,k2> + b5	<q7,k3> + b4	<q7,k4> + b3	<q7,k5> + b2	<q7,k6> + b1	<q7,k7> + b0
…							

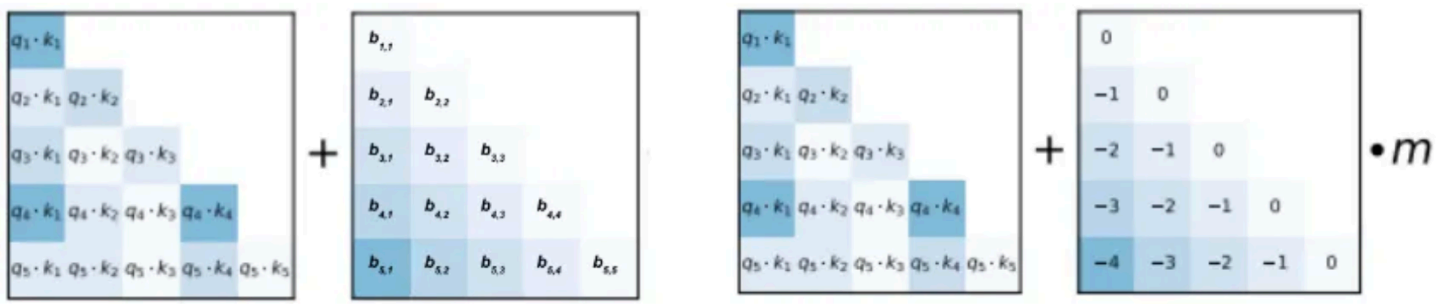
- Recall the Relevance matrix from self-attention, defined by inner-products of keys and queries.
- Learned Relative Position Bias (e.g., T5-bias) adds bias to this matrix that only depends on relative positions, and the biases are learnable parameters.

- **Rotary Positional Embeddings (RoPE)** [Shaw et al. 2018]
  - Recall that in self-attention, each token i is associated with query q_i, key k_i, and value v_i such that the output embedding of the t-th token is

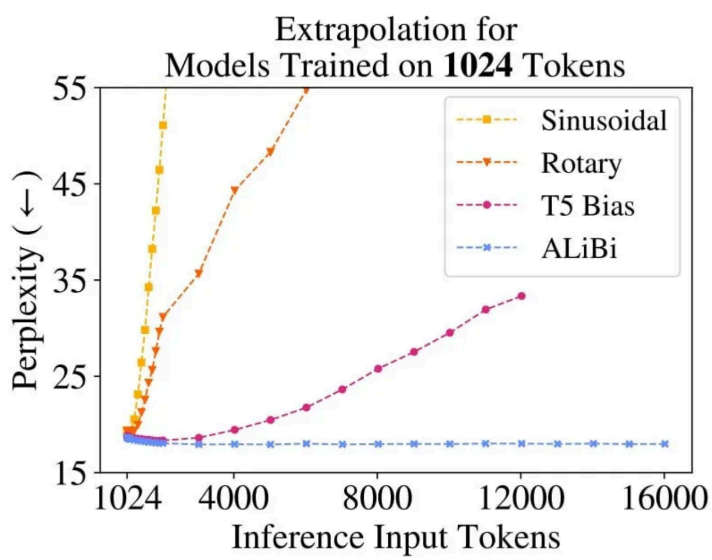
$$y_t = \sum_{i=1}^{t} \frac{e^{\langle q_t, k_i \rangle}}{\sum_{j=1}^{t} e^{\langle q_t, k_j \rangle}} v_i$$

  - **RoPE** encodes positional embedding not in the input x_i, but the key and query.
    - Let $R_{\theta i}$ denote a matrix that rotates a vector by angle θi for some integer i.
    - RoPE rotates the key and the query by θi for some θ, i.e.,
      $\tilde{k}_i \leftarrow R_{\theta i} k_i$ and $\tilde{q}_i \leftarrow R_{\theta i} q_i$ for all $i \in [T]$.
    - Then, the inner product $\langle \tilde{q}_i, \tilde{k}_j \rangle$ only depends on the difference of the positions: $j - i$
    - This follows from the fact that
      - $R_{\theta i}^T = R_{-\theta i}$, and $\langle \tilde{q}_i, \tilde{k}_j \rangle = \tilde{q}_i^T \tilde{k}_j = q_i^T R_{\theta i}^T R_{\theta j} k_j = q_i^T R_{\theta (j-i)} k_j$

- ALiBi



T5 bias (left 2) and ALiBi (right 2) | Source (T5 bias drawn off of description)



Parameter Efficient Fine-Tuning (PEFT)

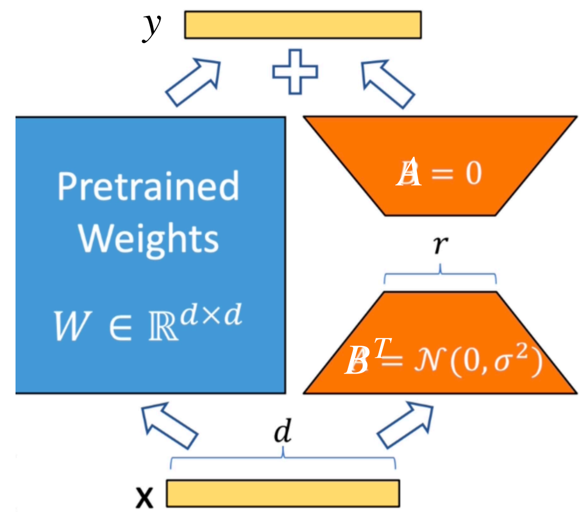
Fine-tuning

- Supervised Fine-Tuning (SFT) is a common practice to adapt a given base LM to the target domain of interest, given labeled fine-tuning samples.
- With the increasing scale of LLMs, oftentimes full=scale fine-tuning of the model weights is prohibitively expensive.
- **Parameter Efficient Fine-Tuning** (PEFT) corresponds to a family of approaches that freezes most of the parameters of the original pertained network and only trains a small subset of parameters.

Low Rank Adaptation (LoRA)

- Given a layer of pretained weight matrix W, fine-tuning results in an updated weight matrix
$$W' \leftarrow W + \Delta W\,.$$

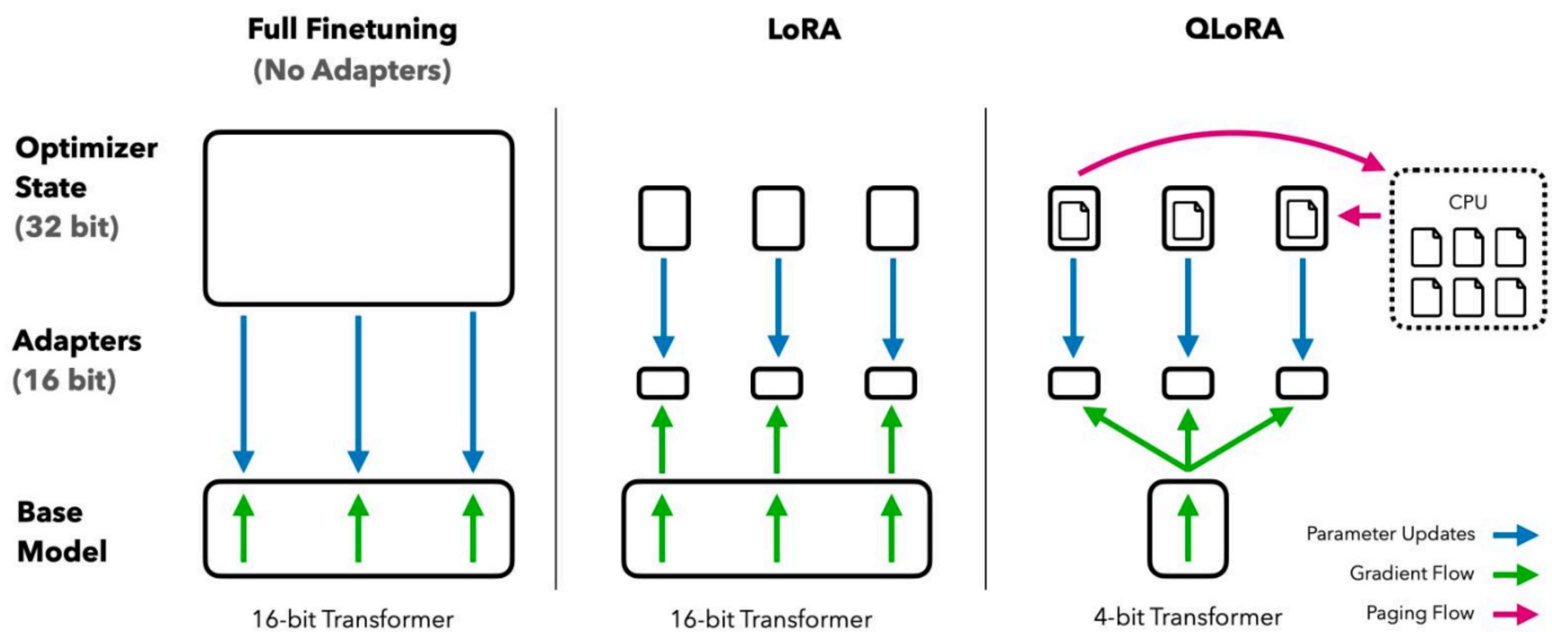
- Instead of a unrestricted, full-rank update ΔW, LoRA parametrizes the update to be low-rank using a bi-linear form:
$$W = AB^T \text{ such that}$$
$$W' \leftarrow W + AB^T,$$
where $A \in \mathbb{R}^{d_{\text{out}} \times r}, B \in \mathbb{R}^{d_{\text{in}} \times r}$ are trainable parameters during fine-tuning.



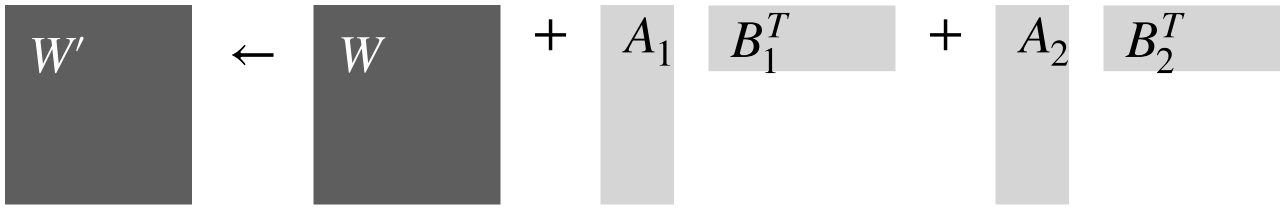
- LoRA Fine-tuning has the following advantages:
  - LoRA requires significantly fewer trainable parameters, requiring reduced **memory** usage: $r \cdot (d_{\text{in}} + d_{\text{out}}) \ll d_{\text{in}} \times d_{\text{out}}$.
  - LoRA avoids making multiple copies of the full parameter, since the base model is frozen.
  - LoRA updates can be applied selectively to specific layers, reducing **computational** overhead.
  - LoRA updates are modular, and can be plugged in and out at inference time.

- **QLoRA** [Dettmers et al. 2023] significantly reduces the memory requirement further, by quantizing the frozen base model weights to 4bit precision, while maintaining the fine-tuned performance.

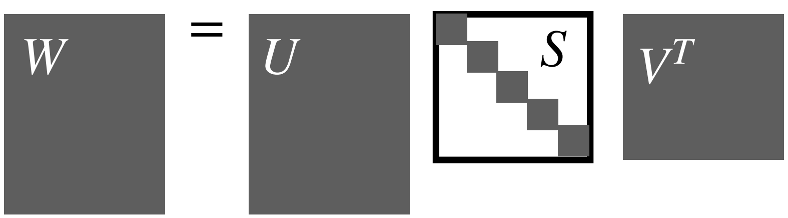
- Average memory requirement for fine-tuning 65B LLM reduces from 780GB of memory to 48GB, which enables fine-tuning on a single GPU.

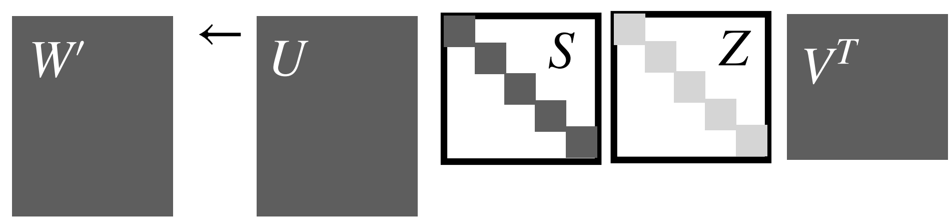


- One down side, is that when we train multiple LoRA adapters on different tasks, say code and math, and merge them to get both skills, the **merged adapter** use twice the memory.

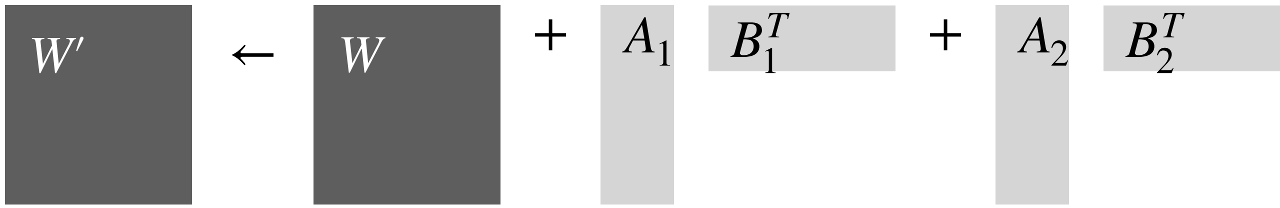


- This parameter increase when merging can be avoided, if the subspaces spanned by A_1 and A_2 are the same and the subspaces spanned by B_1 and B_2 are the same. But, how do we enforce that?
    - SVF: Use the subspace of the SVD(W), and train a diagonal Z, initialized at I

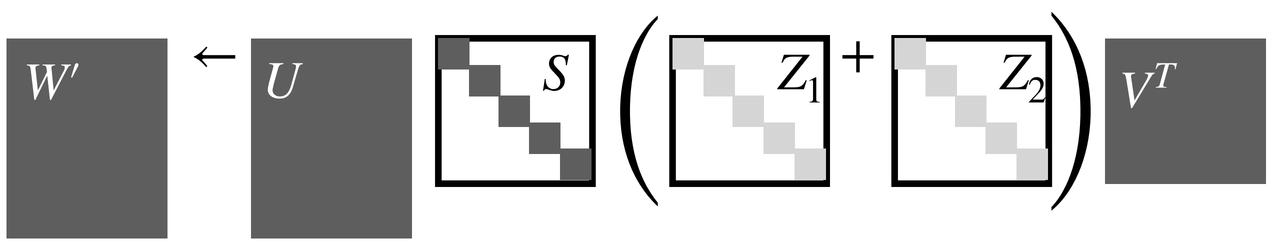




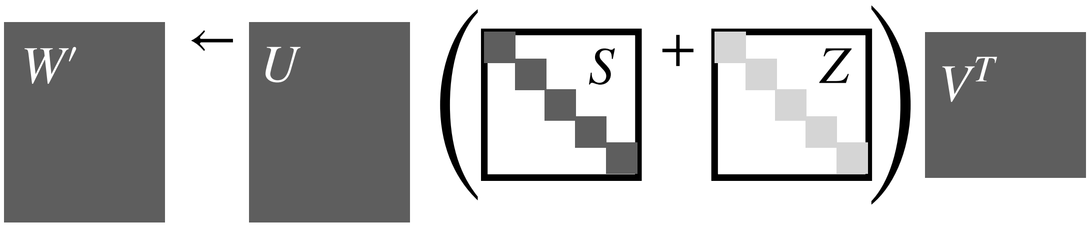
- One down side, is that when we train multiple LoRA adapters on different tasks, say code and math, and merge them to get both skills, the **merged adapter** use twice the memory.



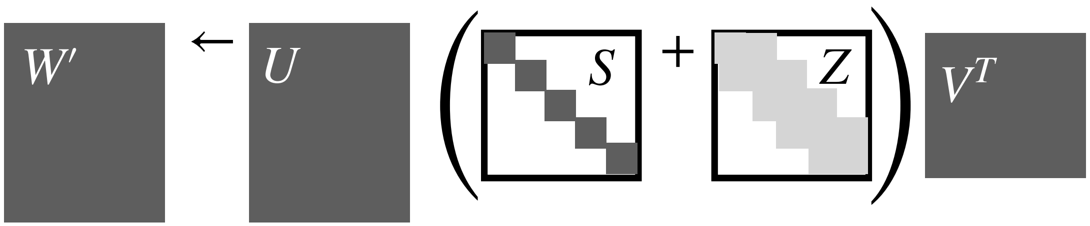
- This parameter increase when merging can be avoided, if the subspaces spanned by A_1 and A_2 are the same and the subspaces spanned by B_1 and B_2 are the same. But, how do we enforce that?
  - SVF: Use the subspace of the SVD(W), and train a diagonal Z, initialized at I
  - When merged, the number of parameters stay the same.



- One can traverse the parameter size vs. downstream accuracy trade-off keeping the SVD subspace frozen:
  - Plain:



  - Banded:



  - Random:

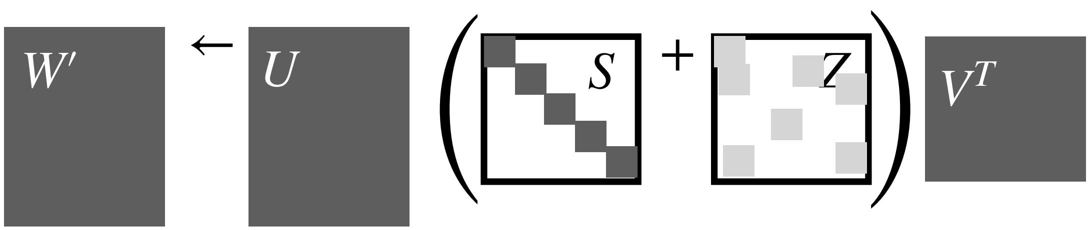


Table 5: Results on fine-tuning Gemma-2B with SVFT using different M parameterizations.

Structure	#Params	GSM-8K	MATH
Plain (diagonal)	0.2M	40.34	14.38
Banded	3.3M	46.47	**16.04**
	6.4M	47.84	15.68
Random	3.3M	47.76	15.98
	6.4M	**50.03**	15.56

Sources

- Other courses in LLMs that the lecture slides are based on
  - CSE493S/599S at UW by Ludwig Schmidt: https://mlfoundations.github.io/advancedml-sp23/
  - EE-628 at EPFL by Volkan Cevher: https://www.epfl.ch/labs/lions/teaching/ee-628-training-large-language-models/ee-628-slides-2025/
  - https://sharif-llm.ir/assets/lectures/Chain-of-Thought-Prompting.pdf
  - https://www.cs.princeton.edu/courses/archive/fall22/cos597G/lectures/lec09.pdf
- Useful blog posts
  - https://azizbelaweid.substack.com/p/complete-summary-of-absolute-relative
  - https://blog.dust.tt/speculative-sampling-llms-writing-a-lot-faster-using-other-llms/
  - https://gordicaleksa.medium.com/eli5-flash-attention-5c44017022ad
  - https://medium.com/@dillipprasad60/qlora-explained-a-deep-dive-into-parametric-efficient-fine-tuning-in-large-language-models-llms-c1a4794b1766
- Dan Jurafsky and James H. Martin. Speech and Language Processing (3rd ed. draft). draft, third edition, 2023.
- Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. “Efficient estimation of word representations in vector space”, In International Conference on Learning Representations, 2013.
- Jeffrey Pennington, Richard Socher, and Christopher D. Manning. "Glove: Global vectors for word representation”, Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014.
- Ofir Press, Noah A. Smith1,3 Mike Lewis2 , “Train Short, Test Long: Attention with Linear Biases Enables Input Length Extrapolation”, In International Conference on Learning Representations, 2022
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin, “Attention Is All You Need”, In Neural Information Processing Systems, 2017
- Beitong Zhou, Cheng Cheng, Guijun Ma, and Yong Zhang. “Remaining useful life prediction of lithium-ion battery based on attention mechanism with positional encoding”, In IOP Conference Series: Materials Science and Engineering, 2020.
- Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. “On the difficulty of training recurrent neural networks." In International Conference on Machine Learning, 2013

- Sepp Hochreiter and Jürgen Schmidhuber. “Long short-term memory.” In Neural Computation, 9(8):1735–1780, 11 1997.
- Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. “Learning phrase representations using rnn encoder-decoder for statistical machine translation”, In ACL 2014
- Andrey Andreyevich Markov. “Essai d’une recherche statistique sur le texte du roman. ‘Eugene Onegin’ illustrant la liaison des epreuve en chain”. In: Izvistia Imperatorskoi Akademii Nauk (Bulletin de l’Académie Impériale des Sciences de St.-Pétersbourg). 6th ser, 7:153–162, 1913.
- Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, Yejin Choi, “The Curious Case of Neural Text Degeneration”, In *International Conference on Learning Representations*, 2020
- Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre and John Jumper, “Accelerating Large Language Model Decoding with Speculative Sampling”In, *ACL-findings*, 2024
- Sergey Ioffe, Christian Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift", In *International Conference on Machine Learning*, 2015
- Shibani Santurkar* MIT shibani@mit.edu Dimitris Tsipras* MIT tsipras@mit.edu Andrew Ilyas* MIT ailyas@mit.edu Aleksander Madry, “How Does Batch Normalization Help Optimization?”, In *Advances in Neural Information Processing Systems* 31 (NeurIPS 2018)
- Jimmy Lei Ba, Jamie Ryan Kiros, Geoffrey E. Hinton, “Layer Normalization “, In 2016
- Tianyu Gao, Adam Fisch, Danqi Chen, “Making Pre-trained Language Models Better Few-shot Learners”, In *ACL*, 2021
- Sewon Min1,2 Xinxi Lyu1 Ari Holtzman1 Mikel Artetxe2 Mike Lewis2 Hannaneh Hajishirzi1,3 Luke Zettlemoyer, “rethinking the role of demonstrations what makes in conte…”
- Hila Gonen1,2 Srini Iyer2 Terra Blevins1 Noah A. Smith1,3 Luke Zettlemoyer1, “Demystifying Prompts in Language Models via Perplexity Estimation”
- E Akyürek, B Wang, Y Kim, J Andreas , “In-context language learning: Architectures and algorithms”, 2024
- What learning algorithm is in-context learning? Investigations with linear models Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, Denny Zhou, 2022
- Ziqian Lin, Kangwook Lee, “Dual Operating Modes of In-Context Learning”, 2024
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, “Chain-of-Thought Prompting Elicits Reasoning in Large Language Models”, In NeurIPS 2022
- Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, Yusuke Iwasawa, “Large Language Models are Zero-Shot Reasoners”, In NeurIPS 2022
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, Denny Zhou, “Self-Consistency Improves Chain of Thought Reasoning in Language Models”, In ICLR 2023

- Shunyu Yao · Jeffrey Zhao · Dian Yu · Nan Du · Izhak Shafran · Karthik Narasimhan, Yuan Cao, “ReAct: Synergizing Reasoning and Acting in Language Models”, In ICLR 2025
- Satyapriya Krishna1, Kalpesh Krishna2, Anhad Mohananey†2, Steven Schwarcz2, Adam Stambler2, Shyam Upadhyay2, Manaal Faruqui*3 , “Fact, Fetch, and Reason: A Unified Evaluation of Retrieval-Augmented Generation“
- Salaheddin Alzubi, Creston Brooks, Purva Chiniya, Edoardo Contente, Chiara von Gerlach, Lucas Irwin, Yihan Jiang, Arda Kaz, Windsor Nguyen, Sewoong Oh, Himanshu Tyagi, Pramod Viswanath, “Open Deep Search: Democratizing Search with Open-source Reasoning Agents“, https://arxiv.org/abs/2503.20201
- Xingyao Wang, Yangyi Chen, Lifan Yuan, Yizhe Zhang, Yunzhu Li, Hao Peng, Heng Ji, “Executable Code Actions Elicit Better LLM Agents”, In ICML 2024
- Peter Shaw, Jakob Uszkoreit Ashish Vaswani, “Self-Attention with Relative Position Representations”, 2018
- Ofir Press1,2 Noah A. Smith1,3 Mike Lewis2, “TRAIN SHORT, TEST LONG: ATTENTION WITH LINEAR BIASES ENABLES INPUT LENGTH EXTRAPOLATION”, 2022
- Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, Luke Zettlemoyer, “QLoRA: Efficient Finetuning of Quantized LLMs”, In NeurIPS 2023