Theories of Pre-Trained Language Models with Practical Implications

Xinyi Wang (Proposal)

Committee: William Wang, Kun Zhang, Shiyu Chang, Xifeng Yan

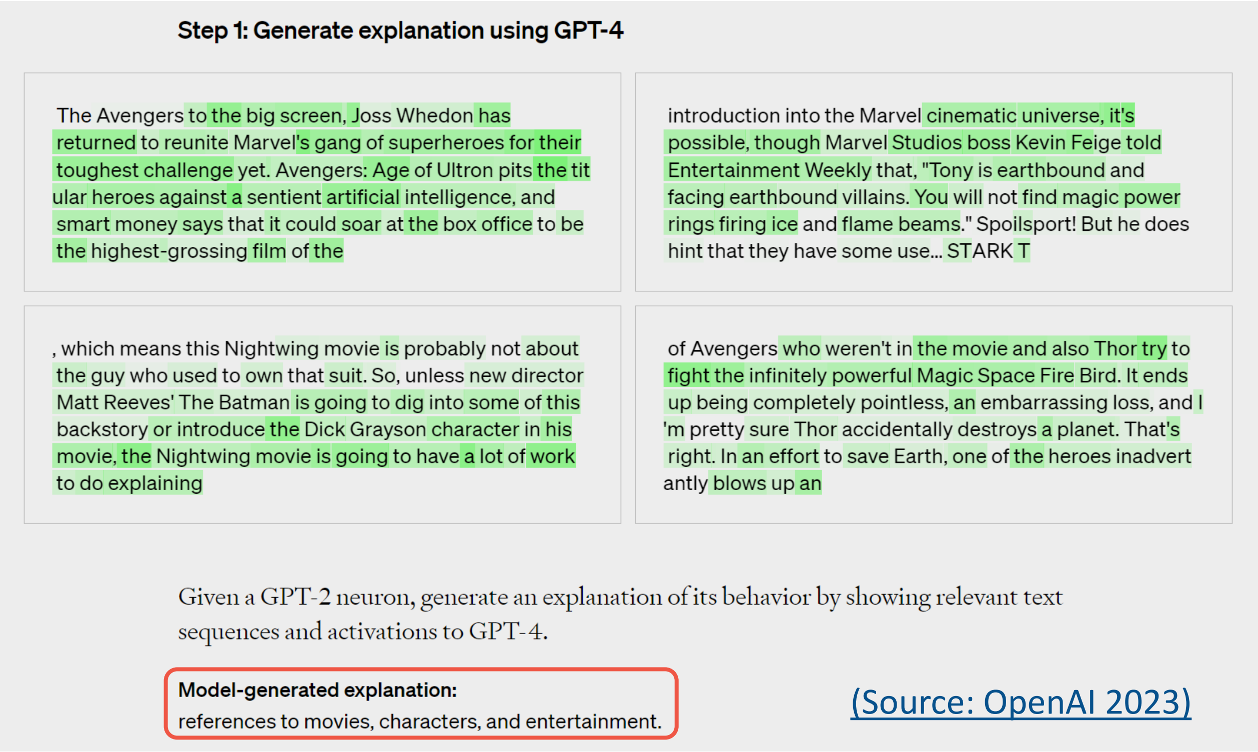
Outline

- Background: the many ways of understanding large language models (LLMs)
  - Interpretability v.s. theory
  - Existing LLM theories
- My progress
  - A latent variable theory
  - A data composition theory
- Future research

Background

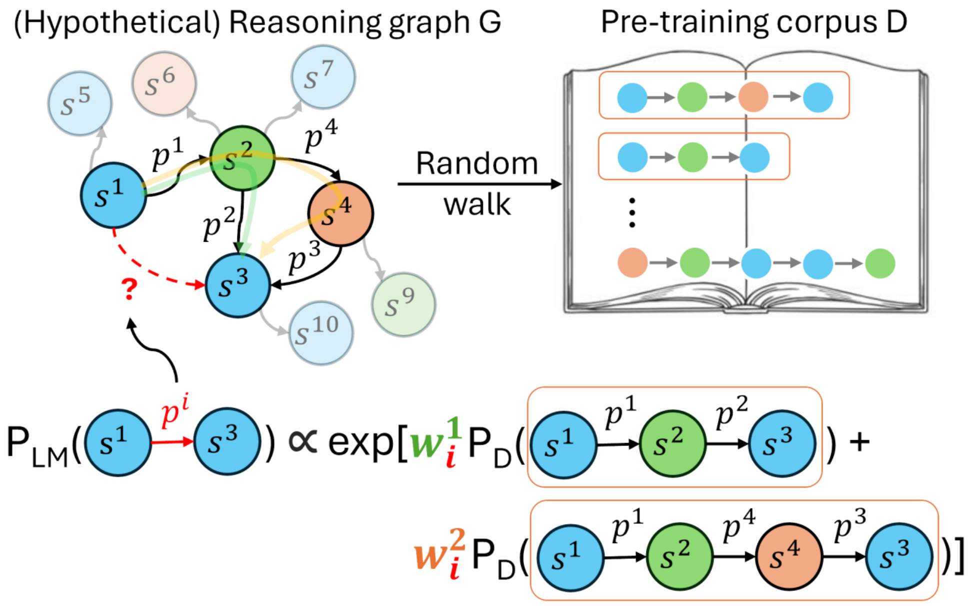
The many ways of understanding large language models (LLMs)

Interpretability



(Source: OpenAI 2023)

Theory

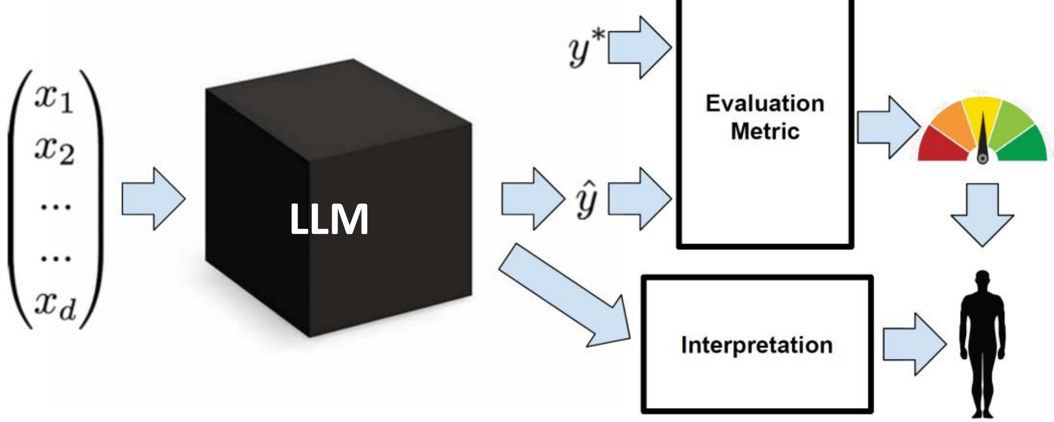


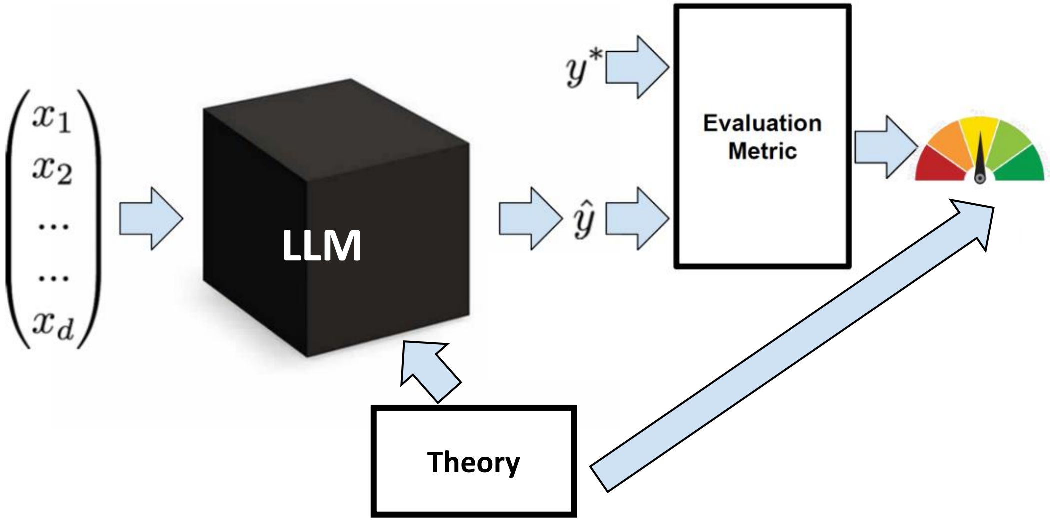
(Source: Ours 2024)

Interpretability v.s. Theory

Interpretability ⊃ Theory

(Source: Lipton 2016)





Goal	Make the LM prediction more transparent to humans, thus assist the final decision making progress or improve user experience.*	Propose a self-consistent theory to explain the LM behavior/learning process as it is, which can be applied to improve the LM performance.
Difference	Not necessarily corresponding to the underlying mechanism of LM learning/inference.	Must revealing the underlying mechanism of LM learning/inference.
Common	Make the LM behavior more predictable. Reduce the risk of unwanted behavior of LMs.	

* Mechanistic interpretability focus on revert LM to human understandable programs. (detailed in next slides.)

Zoom in or zoom out?

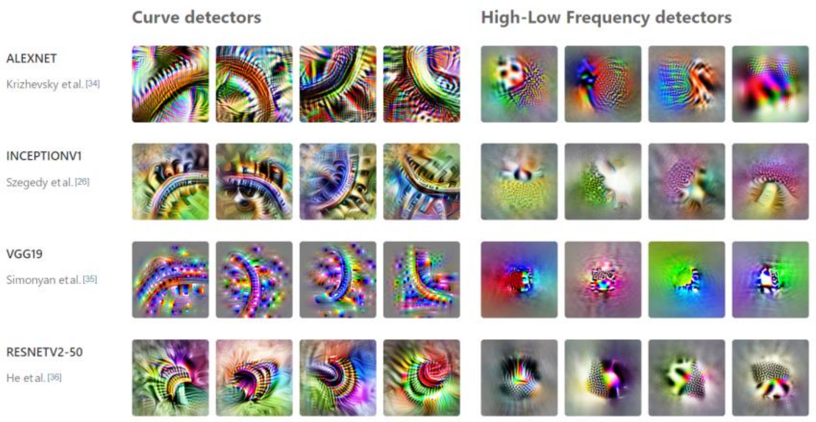
hard

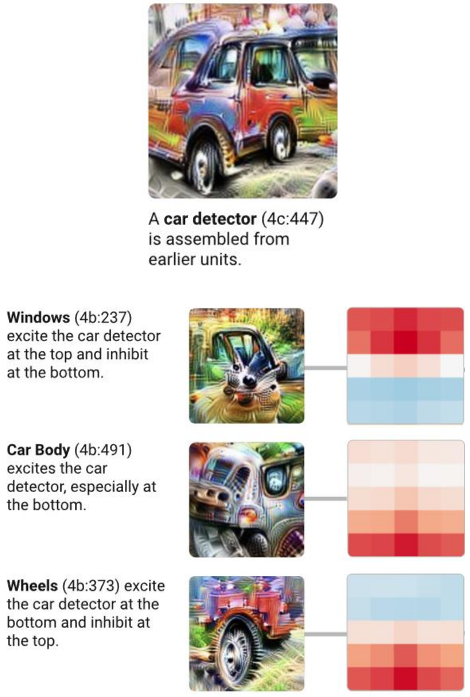
Universality: Analogous features and circuits form across models and tasks.

Circuit: subgraph of connected neurons.

Feature: neuron.

easy





(Source: Olah et al. 2020)

easy

Theory: universal principle governing all models.

Mechanism: how theoretical principles are implemented as algorithm.

Parameter: pinpoint model parameters corresponding to a mechanism.

hard

Zoom in: the *circuit* view from mechanistic interpretability.

Zoom out: verify a hypothesis via theoretical analysis/experiments.

UC SANTA BARBARA

Example In-Context Learning Theories

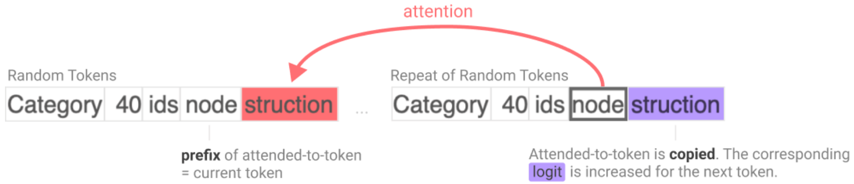
Zoom in: *Induction head*

(Source: Olssen et al. 2022)

hard

Universality: *Induction heads* might constitute the mechanism for the actual majority of all in-context learning (ICL) in large transformer models.

Circuit: *Induction heads "complete the pattern"* by copying and completing sequences that have occurred before.



Feature: An attention head *copies information* from the previous token into each token, which enables the *induction head* to attend to tokens based on what happened before them, rather than their own content.

easy

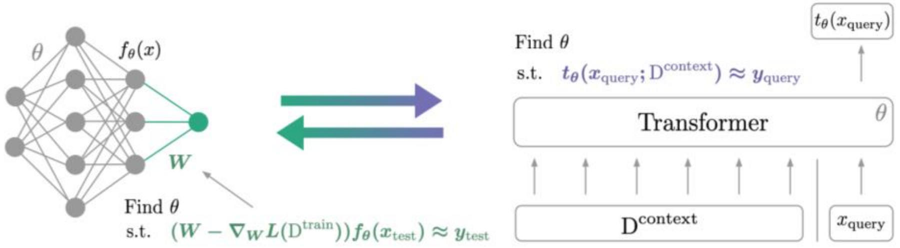
In-context learning score: the loss of the 500th token in the context minus the loss of the 50th token in the context.

Zoom out: *Gradient descent*

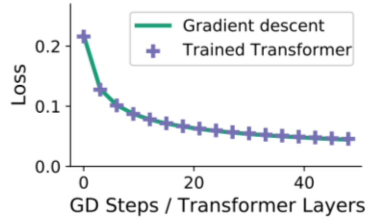
(Source: Akyurek et al. 2022)

easy

Theory: In-context learning (ICL) is implemented by *gradient descent* on given demonstrations in Transformers.



Mechanism: An ordinary gradient-based *least squares algorithm* is implemented for the linear regression task.



Parameter: There is a Transformer construction to exactly implement least square algorithm. It is unknown how it is *actually* implemented in a Transformer.

hard

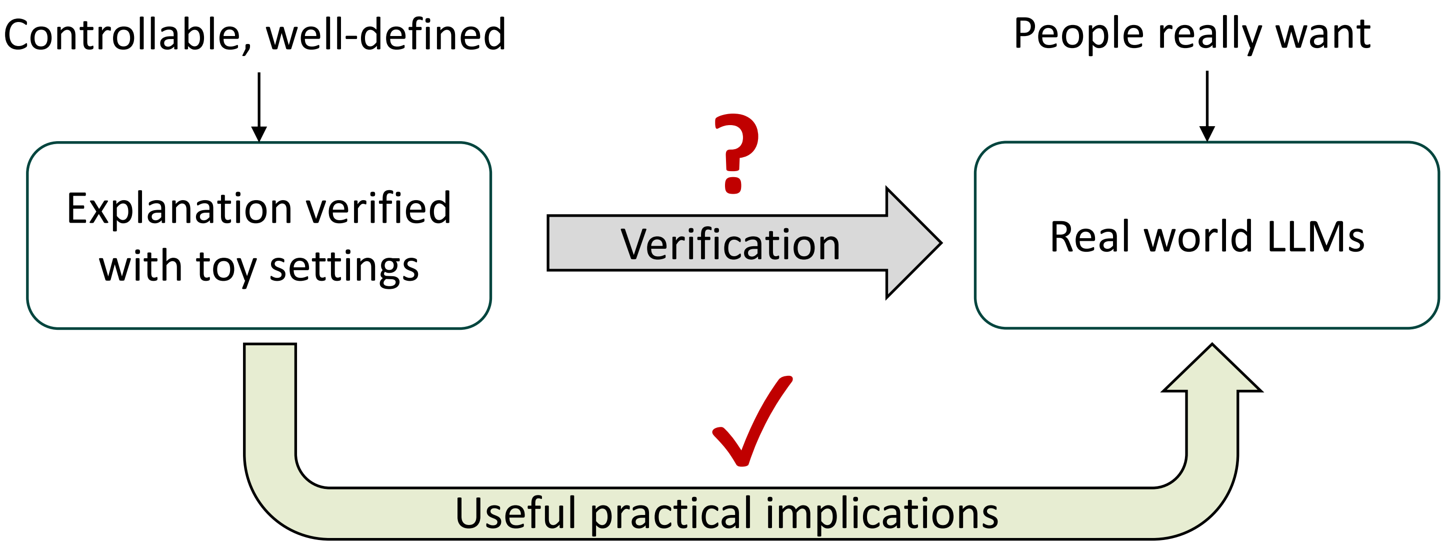
In-context learning task: linear regression

Artificial!

UC SANTA BARBARA

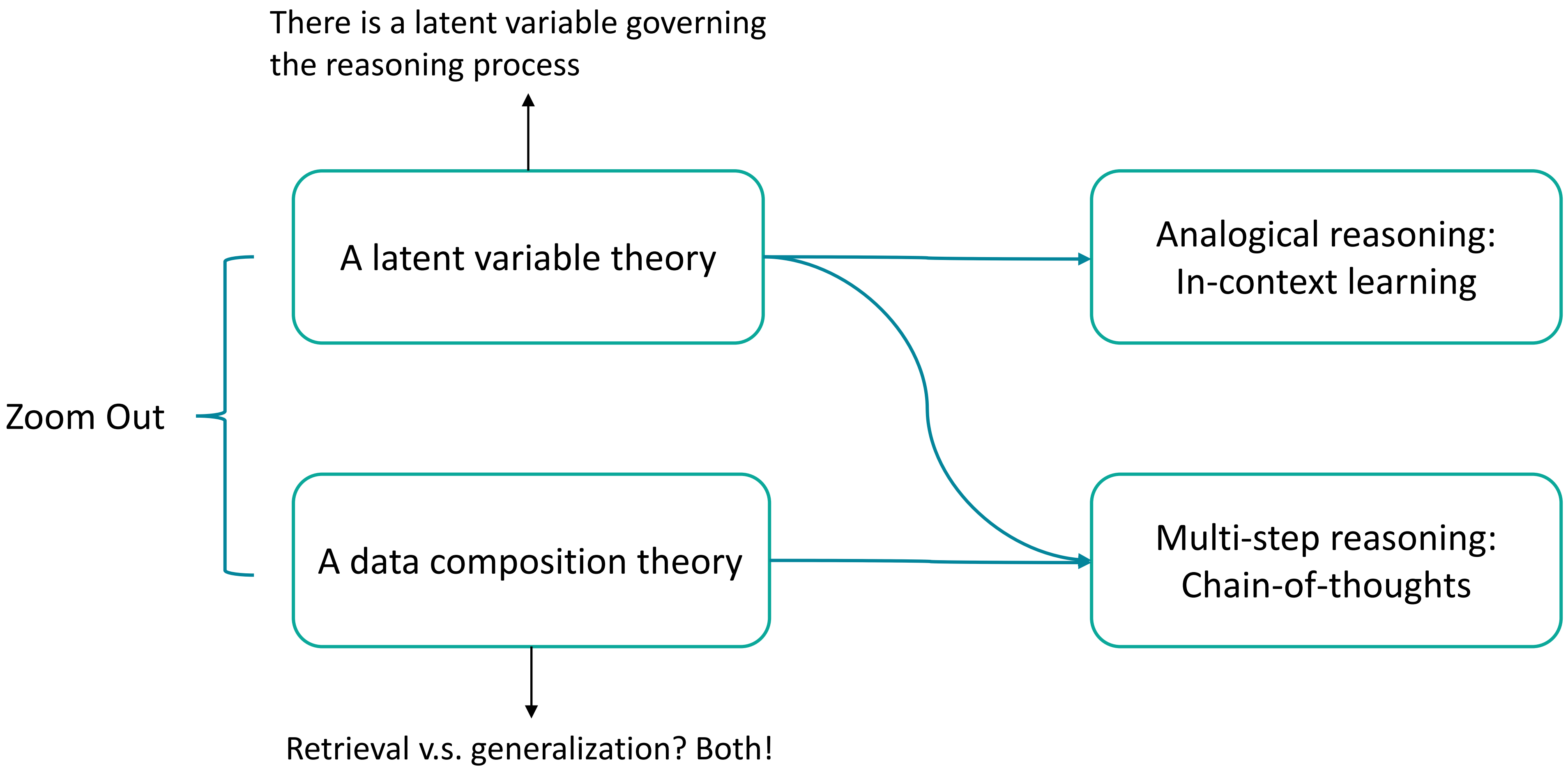
The Common Issue

The big gap between real world LLMs and the proposed explanations.



My progress

Our Proposed LLM Theories



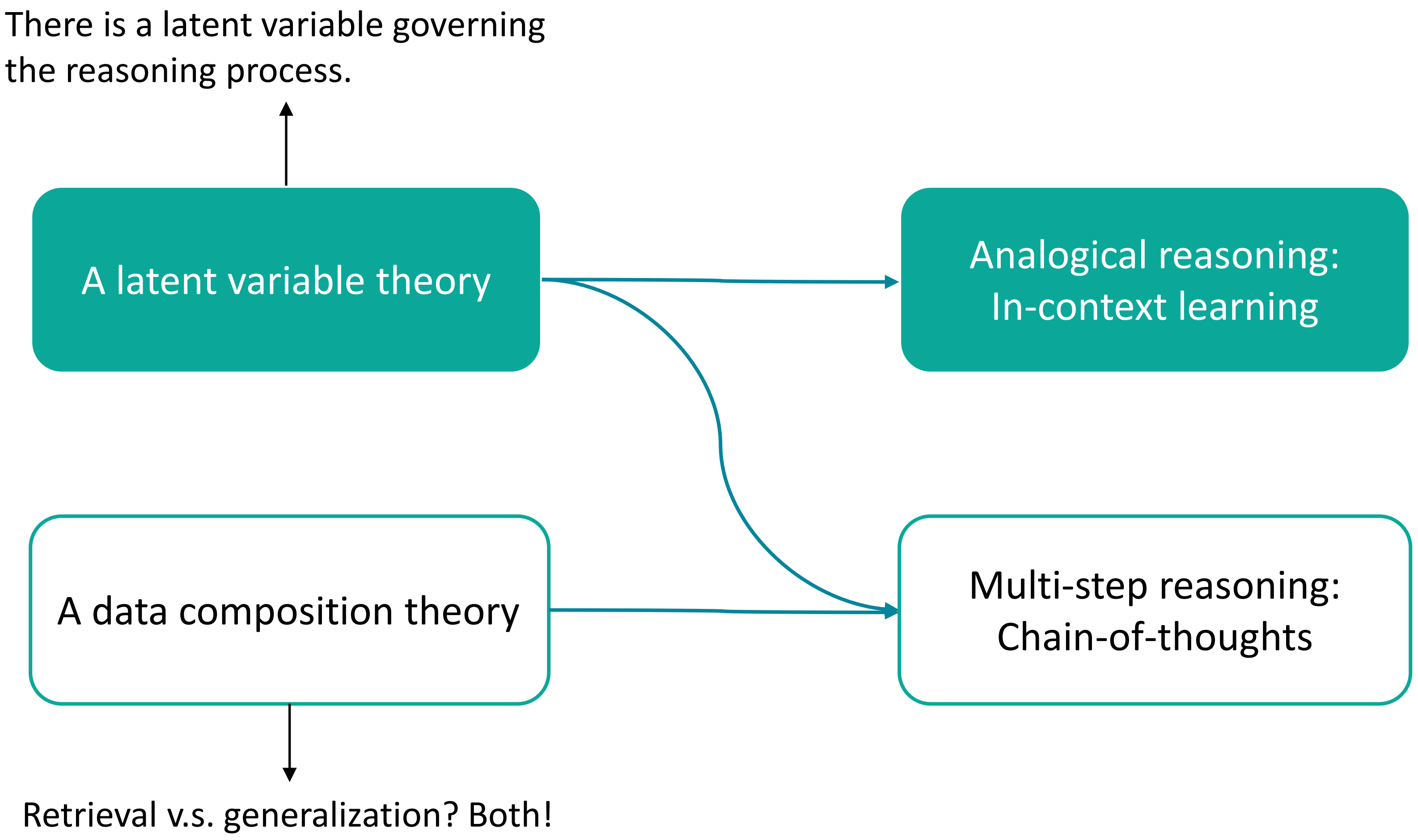
UC SANTA BARBARA

University of California, Irvine

Large Language Models are Latent Variable Models: Explaining and Finding Good Demonstrations for In-Context Learning

Xinyi Wang, Wanrong Zhu, Michael Saxon, Mark Steyvers, William Yang Wang (NeurIPS 2023)

Our Proposed LLM Theories



LLMs are latent variabel models

LLM: $P(\boldsymbol{w}_{1:T}) = \prod_{i=1}^{T} P(\boldsymbol{w}_i | \boldsymbol{w}_{i-1}, ..., \boldsymbol{w}_1)$ **Latent variabel model:** $P(\boldsymbol{w}_{1:T}) = \int_{\Theta} P(\boldsymbol{w}_{1:T} | \boldsymbol{\theta}) P(\boldsymbol{\theta}) d\boldsymbol{\theta}$

Language model probability output by an LLM

Our assumption: $P_M(\boldsymbol{w}_{t+1:T} | \boldsymbol{w}_{1:t}) = \int_{\Theta} P_M(\boldsymbol{w}_{t+1:T} | \boldsymbol{\theta}) P_M(\boldsymbol{\theta} | \boldsymbol{w}_{1:t}) d\boldsymbol{\theta}$

Generated continuation

Prompt

2. Generate the continuation exclusively based on the inferred concept variable θ

1. Implicitly infer a latent concept variable θ from the prompt

Analysis in-context learning classifier

in-context learning classifier

$$P_M^d(Y|X_1^d, Y_1^d, ..., X_k^d, Y_k^d, X)$$

$$= \int_\Theta P_M^d(Y|\boldsymbol{\theta}, X) P_M^d(\boldsymbol{\theta}|X_1^d, Y_1^d, ..., X_k^d, Y_k^d, X) d\boldsymbol{\theta}$$

Goal

Bayes optimal classifier

Approximation

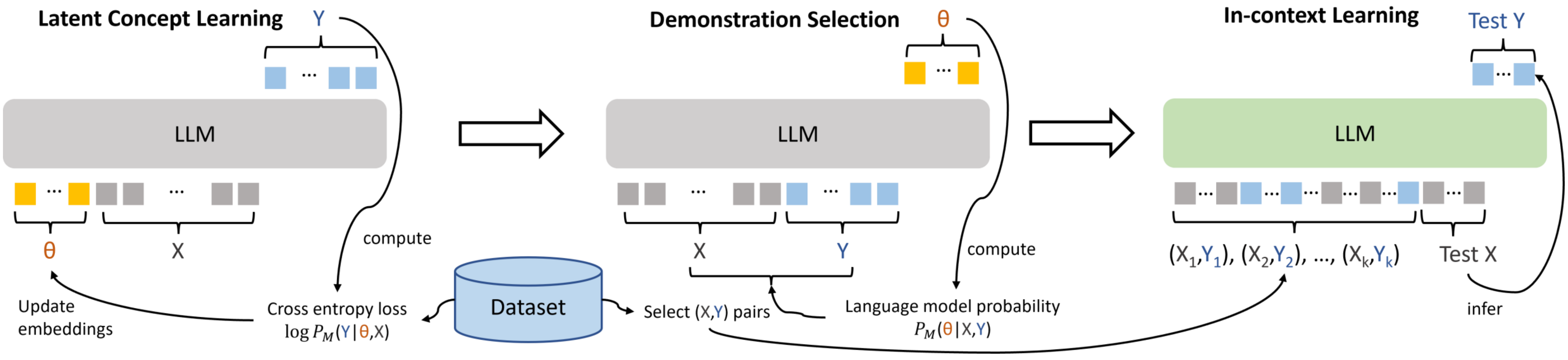
prompt tuning

Exists an optimal value of θ for current task

Infer θ from demonstrations

demonstration selection criteria

Algorithm overview



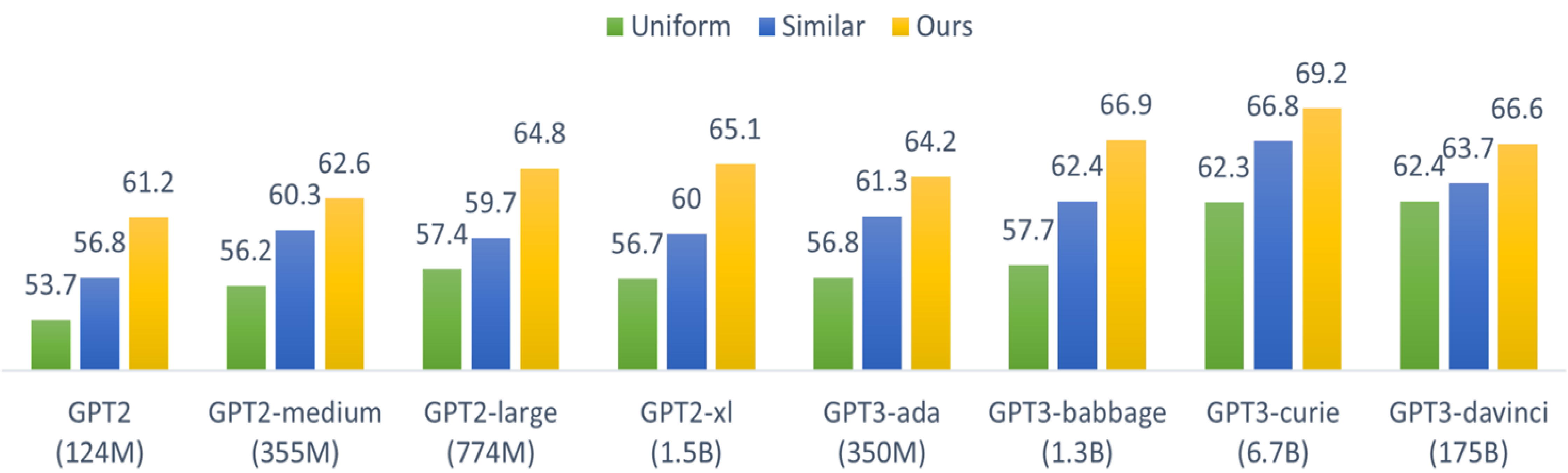
prompt tuning to obtain the latent θ

Scoring examples by $P_M^d(\boldsymbol{\theta}|X_1^d, Y_1^d, ..., X_k^d, Y_k^d, X)$

Use the top examples for in-context learning with any LMs

UC SANTA BARBARA

Text classification results



- Results are averaged over 8 text classification datasets, each experiment is repeated by 5 runs.
- We select the optimal demonstrations by GPT2-large, and use the same set of demonstrations for all other LLMs.

UC SANTA BARBARA

GSM8K results

	Uniform	Similar	Ours w/ Llama 2 (7B)	Ours w/ GPT2-XL (1.5B)
Prompt tuning	-	-	15.2	7.3
Llama 2 (7B)	11.4	13.1	19.3	15.9
Llama 2 (13B)	17.0	18.3	21.6	20.5
Llama 2 (70B)	50.2	53.5	54.3	52.9
ChatGPT (gpt-3.5-turbo)	76.5	78.1	81.2	80.4

Table 1: Prompt tuning and 4-shot in-context learning accuracy on a subset of GSM8K test set. Our demonstrations are selected with either 7B Llama 2 or GPT2-XL

- We select the optimal demonstrations by Llama 2 (7B)/ GPT2-XL, and use the same set of demonstrations for all other LLMs.

Guiding Language Model Math Reasoning with Planning Tokens

Xinyi Wang, Lucas Caccia, Oleksiy Ostapenko, Xingdi Yuan, Alessandro Sordoni (Arxiv)

Our Proposed LLM Theories

There is a latent variable governing the reasoning process.

A latent variable theory

Analogical reasoning:
In-context learning

A data composition theory

Multi-step reasoning:
Chain-of-thoughts

Retrieval v.s. generalization? Both!

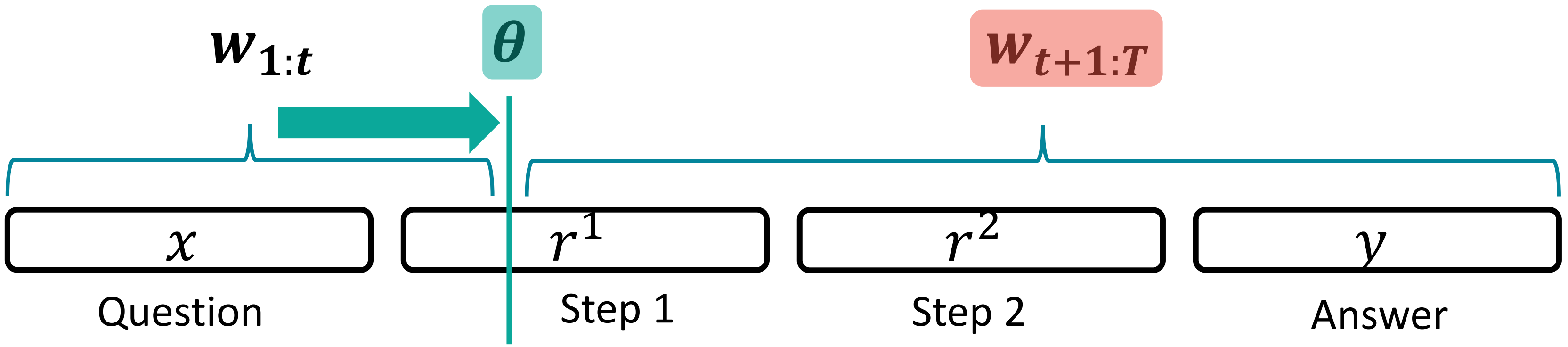
LLM fine-tuned with chain-of-thoughts data

Question: Every day, Wendi feeds each of her chickens three cups of mixed chicken feed, containing seeds, mealworms and vegetables to help keep them healthy. She gives the chickens their feed in three separate meals. In the morning, she gives her flock of chickens 15 cups of feed. In the afternoon, she gives her chickens another 25 cups of feed. **How many cups of feed does she need to give her chickens in the final meal of the day** if the size of Wendi's flock is 20 chickens?

Every day, Wendi gives her chickens 15 cups of feed in the morning + 25 cups of feed in the afternoon = <<15+25=40>>40 cups of feed.
If she has 20 chickens and she feeds them 40 cups of feed every day, then each chicken gets 40/20 = <<40/20=2>>2 cups of feed per chicken.
The answer is: 2

A Bayesian view of chain-of-thoughts (CoTs)

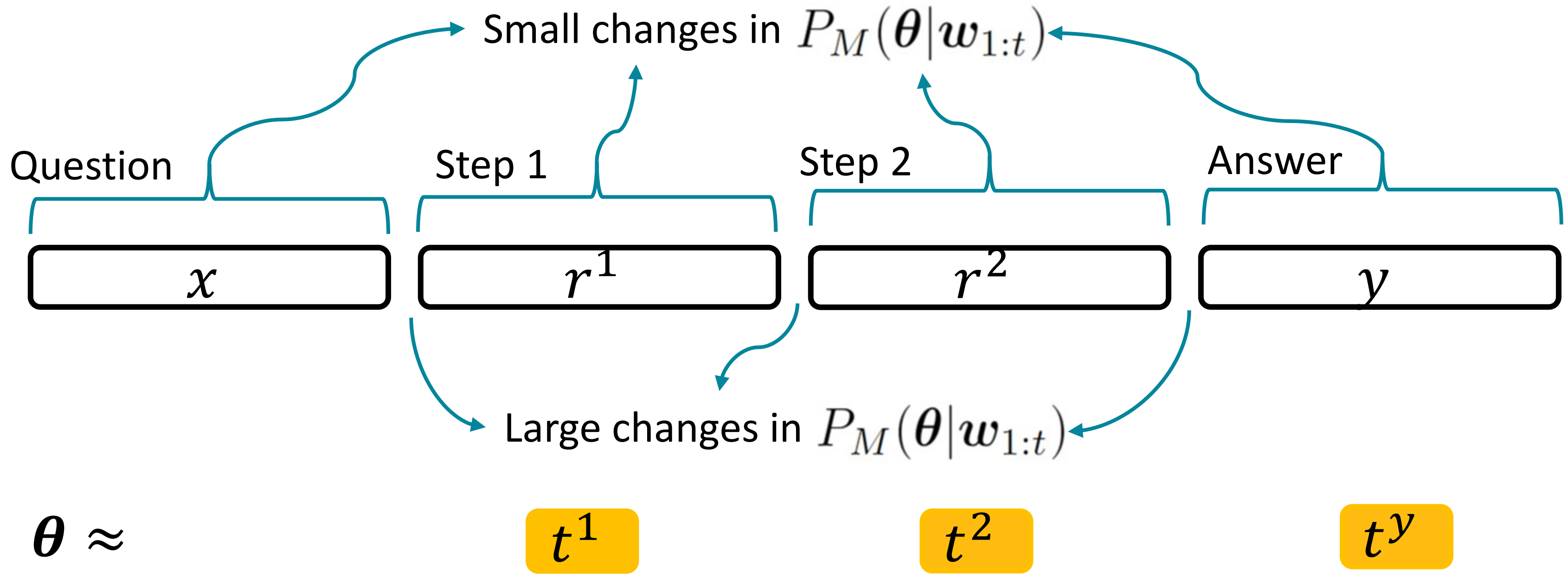
$$P_M(\boldsymbol{w}_{t+1:T}|\boldsymbol{w}_{1:t}) = \int_{\Theta} P_M(\boldsymbol{w}_{t+1:T}|\boldsymbol{\theta}) P_M(\boldsymbol{\theta}|\boldsymbol{w}_{1:t}) d\boldsymbol{\theta}$$



Bayesian assumption: there is a latent variable $\boldsymbol{\theta}$ governing the generation of the whole CoT sequence.

A Bayesian view of chain-of-thoughts (CoTs)

$$P_M(\boldsymbol{w}_{t+1:T}|\boldsymbol{w}_{1:t}) = \int_{\Theta} P_M(\boldsymbol{w}_{t+1:T}|\boldsymbol{\theta})P_M(\boldsymbol{\theta}|\boldsymbol{w}_{1:t})d\boldsymbol{\theta}$$



Simplified Bayesian assumption: there is a discrete planning variable $\boldsymbol{t}$ governing the generation of each chain-of-thoughts step.

LLM fine-tuning with planning tokens

Question: Every day, Wendi feeds each of her chickens three cups of mixed chicken feed, containing seeds, mealworms and vegetables to help keep them healthy. She gives the chickens their feed in three separate meals. In the morning, she gives her flock of chickens 15 cups of feed. In the afternoon, she gives her chickens another 25 cups of feed. **How many cups of feed does she need to give her chickens in the final meal of the day** if the size of Wendi's flock is 20 chickens?

Specialized planning tokens

General planning tokens

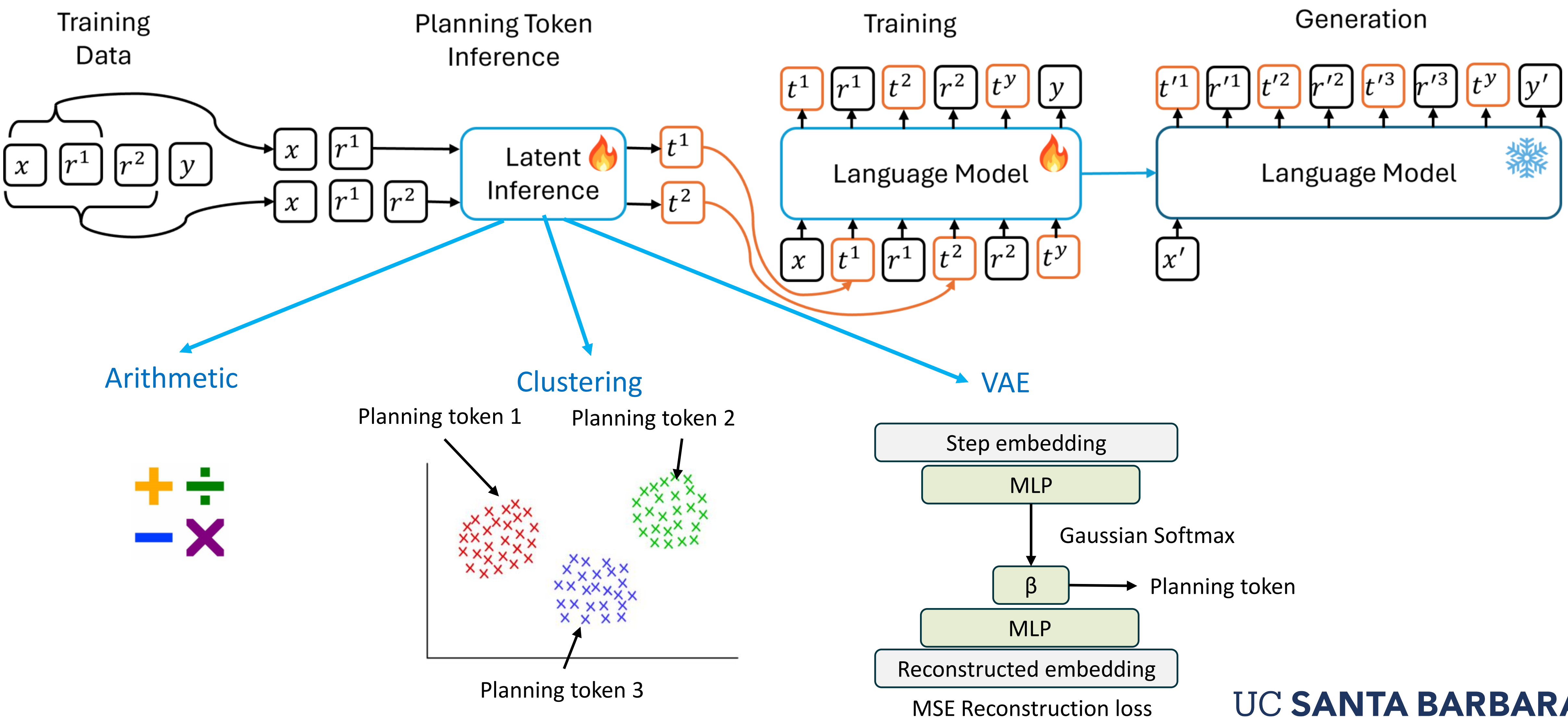
<prefix> <+> Wendi gives her flock 15 cups of feed in the morning and another 25 cups in the afternoon, for a total of 15+25 = <<15+25=40>>40 cups of feed.

<prefix> <*> If Wendi has 20 chickens, then she needs 20*3 = <<20*3=60>>60 cups of feed to feed her flock.

<prefix> <-> If Wendi has already given her flock 40 cups of feed, then she needs to give her flock 60-40 = <<60-40=20>>20 more cups of feed.

<prefix> <answer> The answer is: 20

Algorithm overview



Results on Math Word Datasets

LM	Method	#clusters	#trainable	GSM8K	MATH	AQUA	Avg
Phi 1.5 (1.3B)	Full-FT	0	100%	12.5	1.3	27.2	13.5
	Full-FT + General	1	100%	15.4	2.0	35.4	17.6
	Full-FT + **Arithmetic**	4	100%	15.0	2.3	33.1	16.8
	Full-FT + **K-Means**	5	100%	14.5	2.7	**36.5**	17.7
	Full-FT + **SQ-VAE**	5	100%	**15.8**	**3.3**	34.3	**17.8**
Llama2 (7B)	LoRA	0	0.343%	38.2	6.5	36.6	27.1
	LoRA + General	1	0.344%	38.5	6.7	37.8	27.7
	LoRA + **Arithmetic**	4	0.344%	39.5	5.6	38.2	27.8
	LoRA + **K-Means**	5	0.344%	39.1	6.7	40.5	28.8
	LoRA + **SQ-VAE**	5	0.344%	**40.0**	**7.0**	**41.3**	**29.4**
Llama2 (13B)	LoRA	0	0.279%	44.6	7.2	41.3	31.0
	LoRA + General	1	0.280%	47.9	7.9	42.5	32.8
	LoRA + **Arithmetic**	4	0.280%	41.9	4.6	35.8	27.4
	LoRA + **K-Means**	5	0.280%	49.6	8.4	**44.1**	34.0
	LoRA + **SQ-VAE**	5	0.280%	**50.6**	**8.5**	43.9	**34.3**

Reasoning length effect

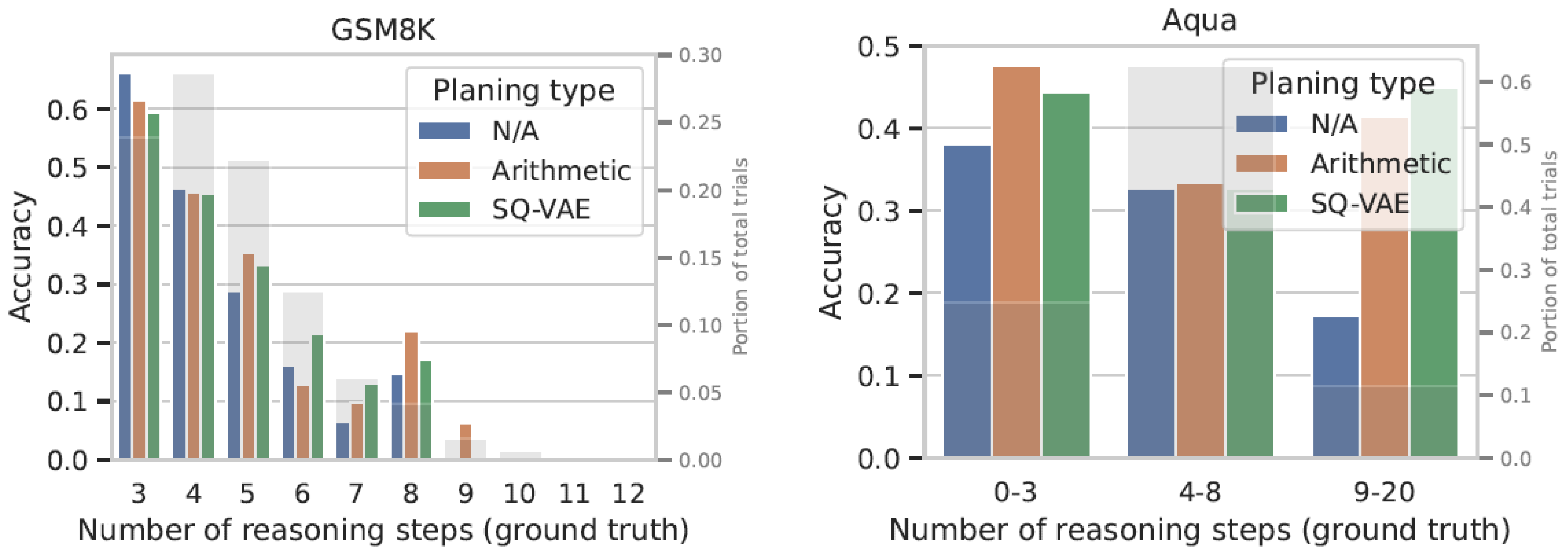


Figure 2: Accuracy on GSM8K (**left**) and Aqua (**right**) on test examples by their number of ground-truth reasoning steps. SQ-VAE consistently increases performance for test examples that require more steps of reasoning to be solved.

UNIVERSITY OF WATERLOO

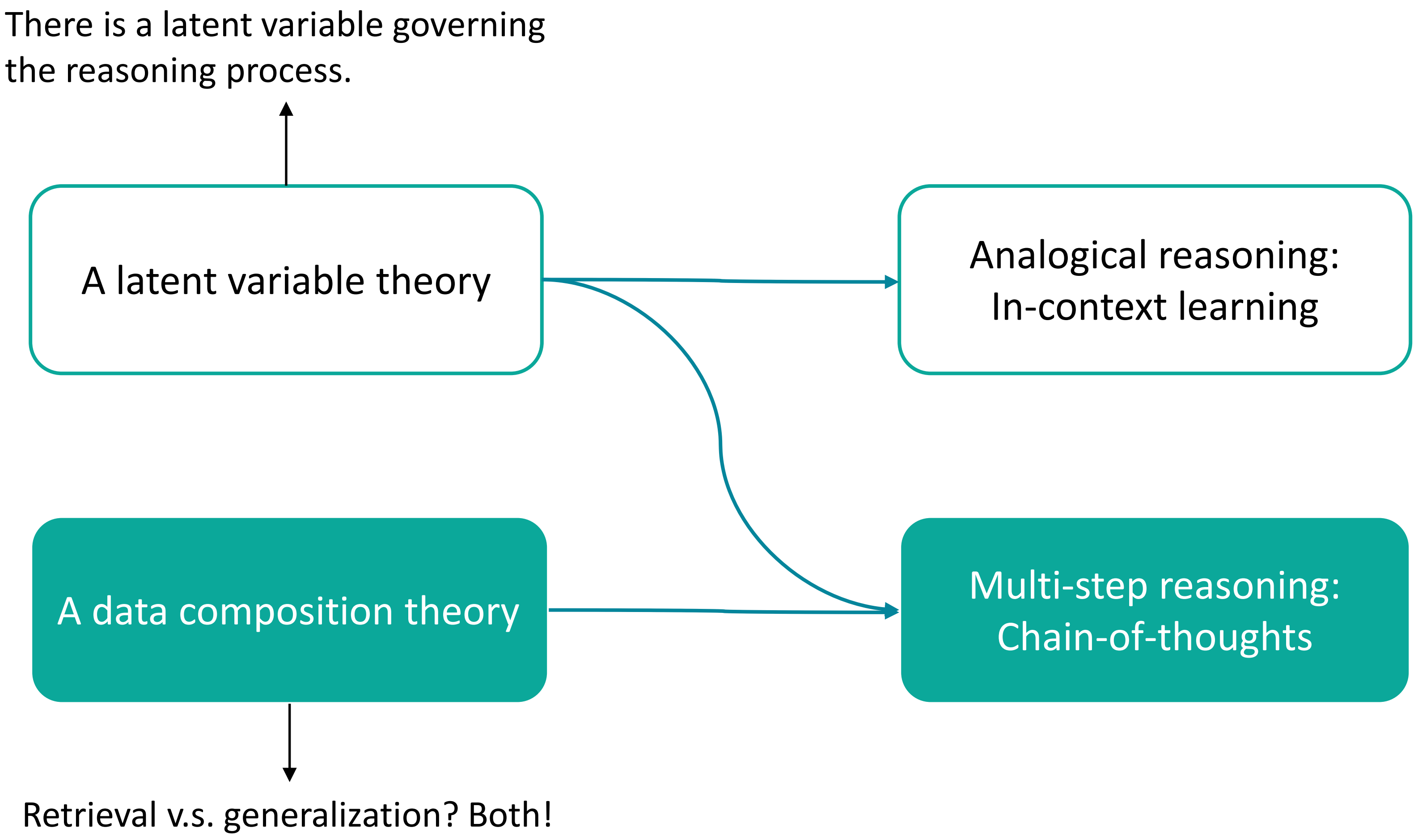
Carnegie Mellon University

UC SANTA BARBARA

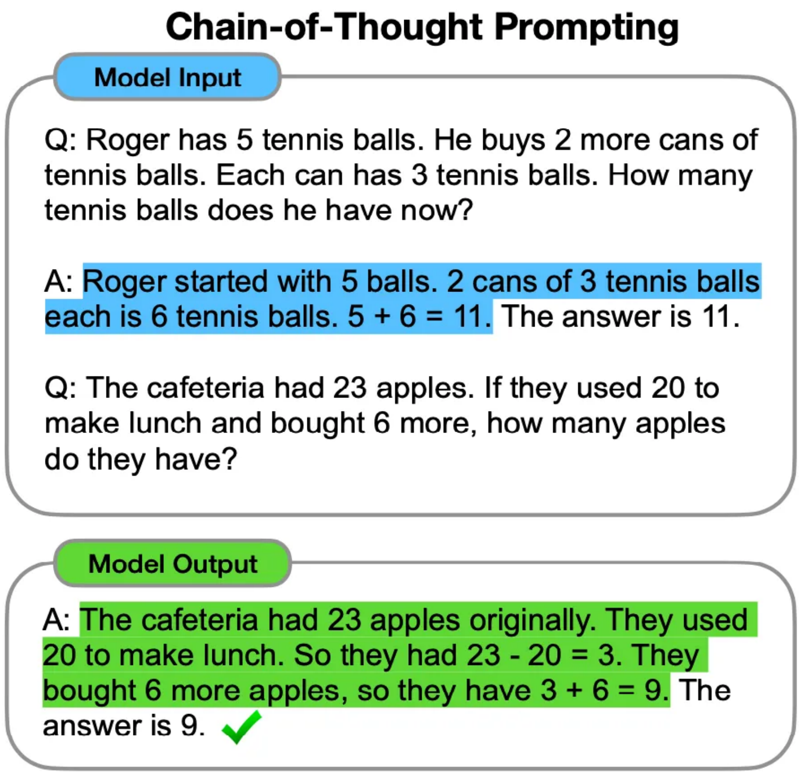
Understanding the Reasoning Ability of Language Models From the Perspective of Reasoning Paths Aggregation

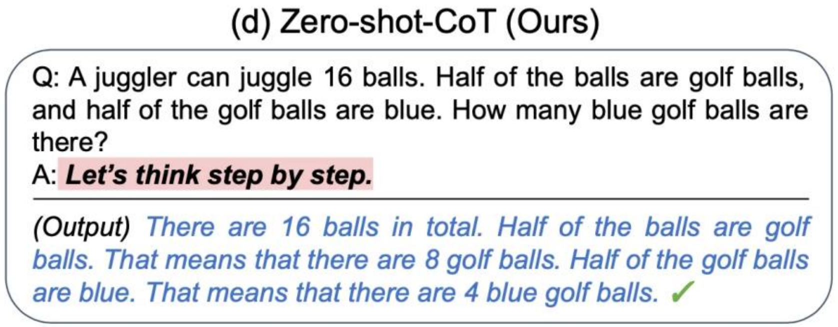
Xinyi Wang, Alfonso Amayuelas, Kexun Zhang, Liangmin Pan, Wenhu Chen, William Yang Wang (Arxiv)

Our Proposed LLM Theories



Reasoning with LLMs



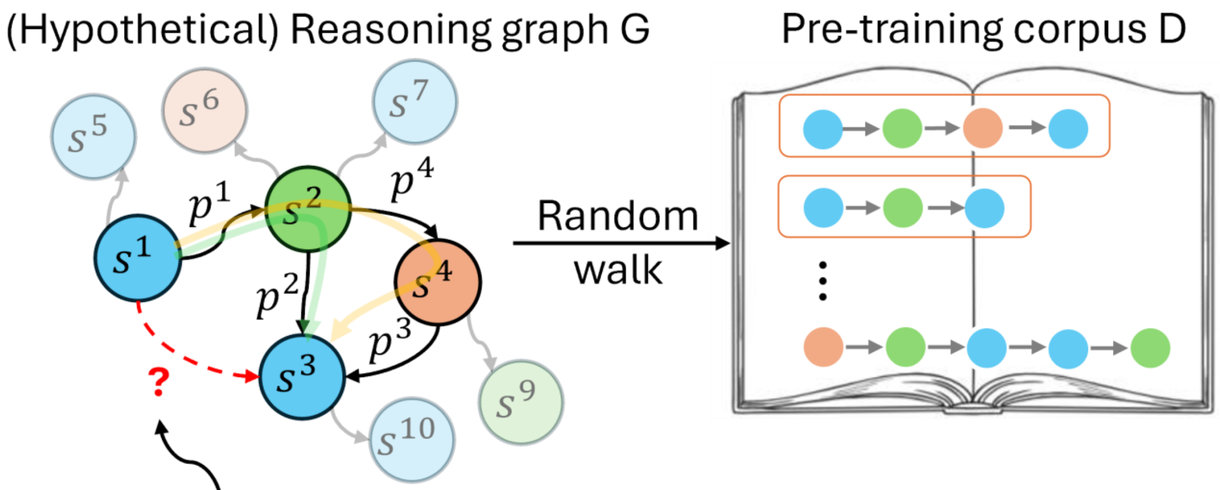


(Source: Wei et al. 2022; Kojima et al. 2022)

- **Definition** of reasoning: deriving new conclusions with novel conditions from the known facts.
- **Observation**: Pre-trained-only base LLMs exhibit impressive reasoning capability without any fine-tuning.

Understand reasoning ability of LLMs

- **Hypothesis**: LLMs can retrieve and aggregate (random walk) reasoning paths seen at pre-training time to do complex reasonings at inference time.
- **Approach**: We study two specific cases of reasoning:
    - **logical**/knowledge graph (KG) reasoning: pre-train a toy transformer on KGs
    - **mathematical** reasoning: continue (pre-)train a pre-trained LM on more unlabeled augmented reasoning paths.

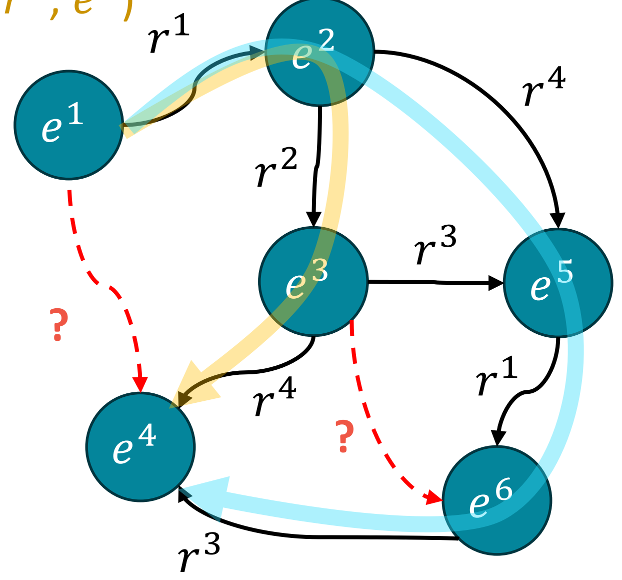


$$P_{LM}(s^1 \xrightarrow{p^i} s^3) \propto \exp[w_i^1 P_D(s^1 \xrightarrow{p^1} s^2 \xrightarrow{p^2} s^3) + w_i^2 P_D(s^1 \xrightarrow{p^1} s^2 \xrightarrow{p^4} s^4 \xrightarrow{p^3} s^3)]$$

Logical reasoning with knowledge graph

- **Knowledge graph (seen triples):** (e^1, r^1, e^2), (e^2, r^2, e^3), (e^2, r^4, e^5), (e^3, r^3, e^5), (e^3, r^4, e^4), (e^5, r^1, e^6), (e^6, r^3, e^4)
- **Unseen triples**: (e^1, r^3, e^4), (e^3, r^4, e^6)
- **Task:** how to infer unseen triples from the seen ones?
  - We propose to look at P(tail | head, relation).
  - i.e. $P(e^4 \mid e^1, r^3)$, $P(e^6 \mid e^3, r^4)$.
  - The sample space is all entities in the graph.
- **Language model training**:
  - Translate each entity and relation into a new token.
  - Sample random walk paths from the knowledge graph to form the pre-training data.
  - Use the next-token-prediction objective to pre-train a small transformer based LM from scratch.
- **Language model inference**:
  - prediction of the tail entity: prompt the LM with head entity and relation.

path1 = $(e^1, r^1, e^2), (e^2, r^2, e^3), (e^3, r^4, e^4)$



path2 = $(e^1, r^1, e^2), (e^2, r^4, e^5), (e^5, r^1, e^6), (e^6, r^3, e^4)$

Distributions

- **Language model:**

logits

$$P_{\mathrm{LM}}(e_2|e_1,r) = \frac{\exp\left(f_\theta(e_2|e_1,r)\right)}{\sum_{e\in\mathcal{E}}\exp\left(f_\theta(e|e_1,r)\right)} \quad (2)$$

All Entities

- **Unweighted aggregation:**

A simplified version of P_w would be letting $w_r(h) = 1$ for all h and r. And **we define this unweighted aggregation distribution to be P_s:**

$$P_s(e_2|e_1,r) = \frac{\exp(\sum_{h\in\mathcal{H}_r} P(e_2|e_1,h)/T)}{\sum_{e\in\mathcal{E}}\exp(\sum_{h\in\mathcal{H}_r} P(e|e_1,h)/T)} \quad (4)$$

Rules related to relation r

Sum of Random walk paths probability

- **Weighted aggregation:**

Path ranking algorithm (PRA) (Lao et. al. 2011)

$$P_w(e_2|e_1,r) = \frac{\exp(S_w(e_2|e_1,r)/T)}{\sum_{e\in\mathcal{E}}\exp(S_w(e|e_1,r)/T)} \quad (3)$$

$$S_w(e_2|e_1,r) = \sum_{h\in\mathcal{H}} w_r(h) P(e_2|e_1,h)$$

All possible rules

Rule weight learned by logistic regression

Sum of Random walk paths probability

$$P(e_n|e_0,h) = \sum_{(e_0,r_1,e_1)\ldots(e_{n-1},r_n,e_n)\in\mathcal{P}_h} \prod_{i=1}^{n} P(e_i|e_{i-1},r_i)$$

Uniform for random walk

Comparing LM with path aggregation

KL Divergence

Decrease

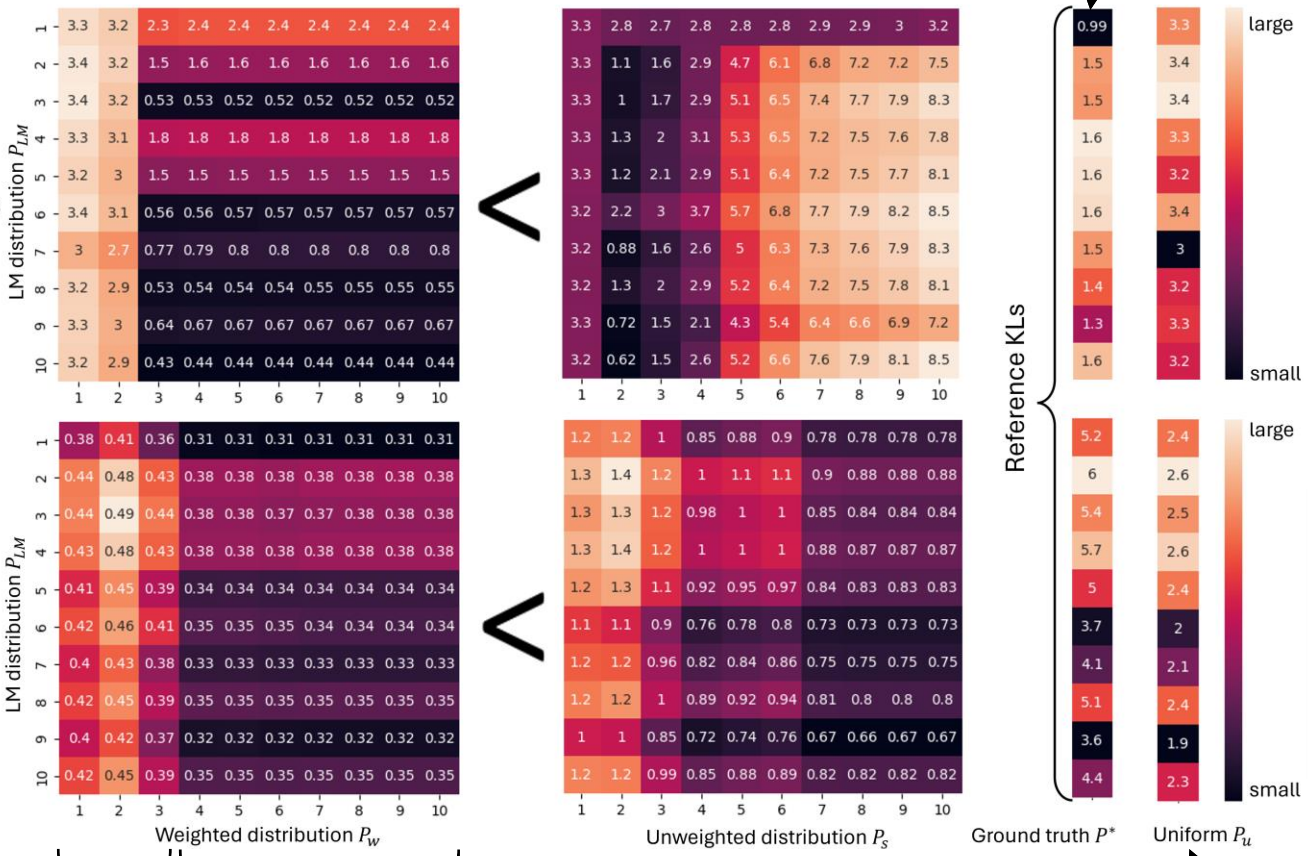
easy

Countries

Decrease

hard

UMLS



unstable

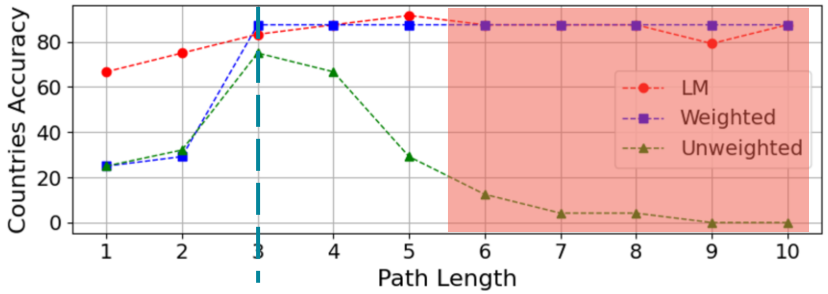
Stable

More uniform

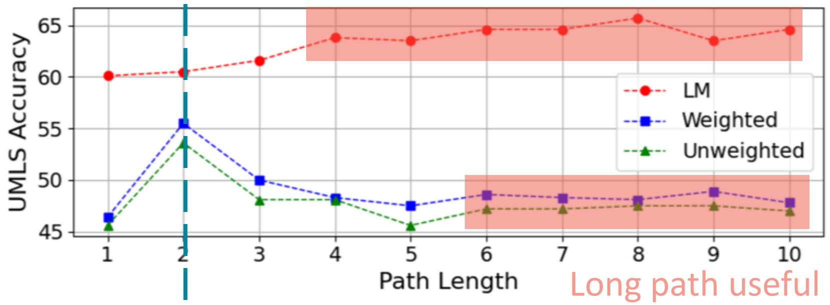
Prediction Accuracy

Ground truth path length

Long path not useful



LM learn better path weight

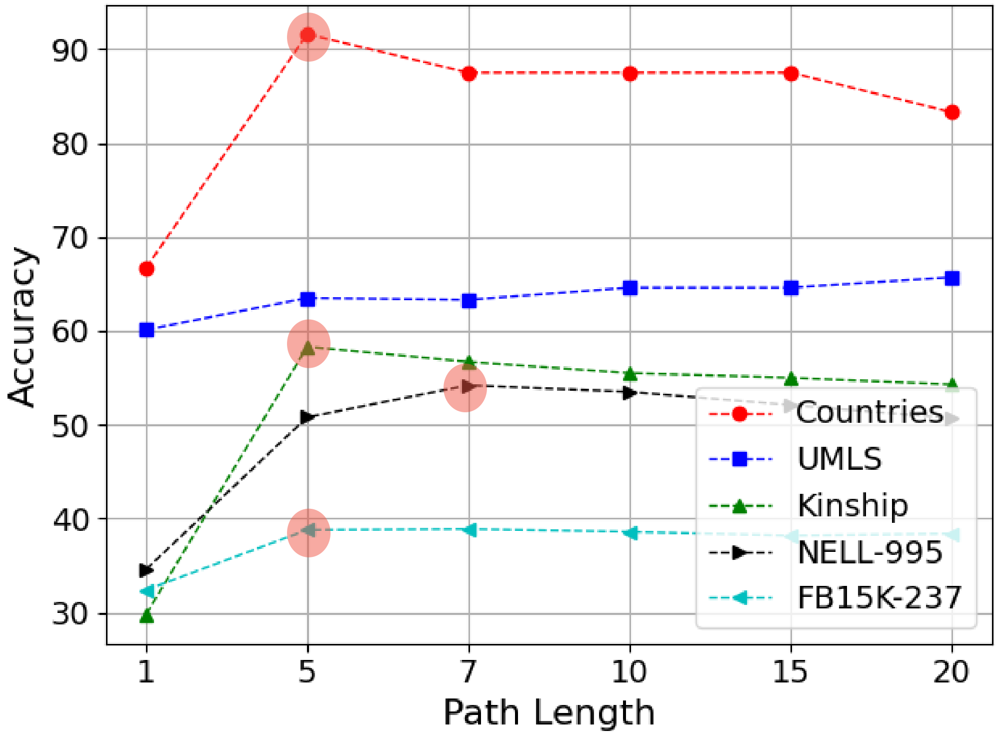


Long path useful

Path length with the largest number of correct paths

UC SANTA BARBARA

Ablation on random walk path length

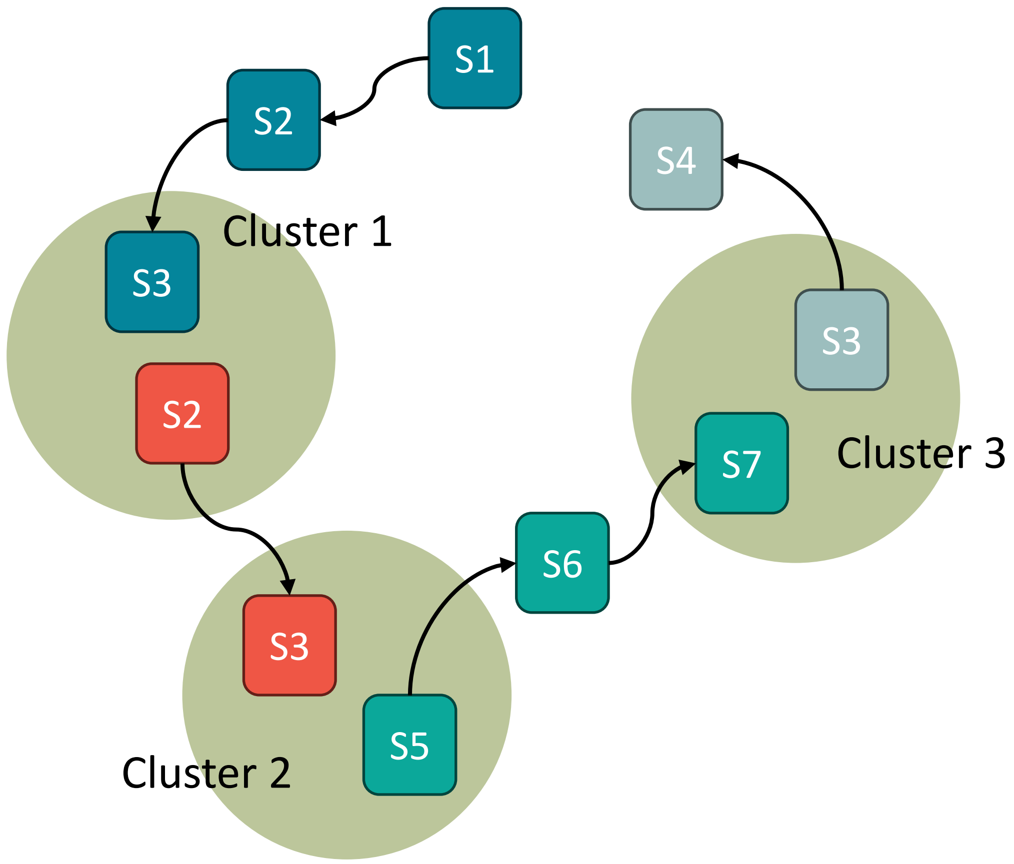


* Exists an optimal path length

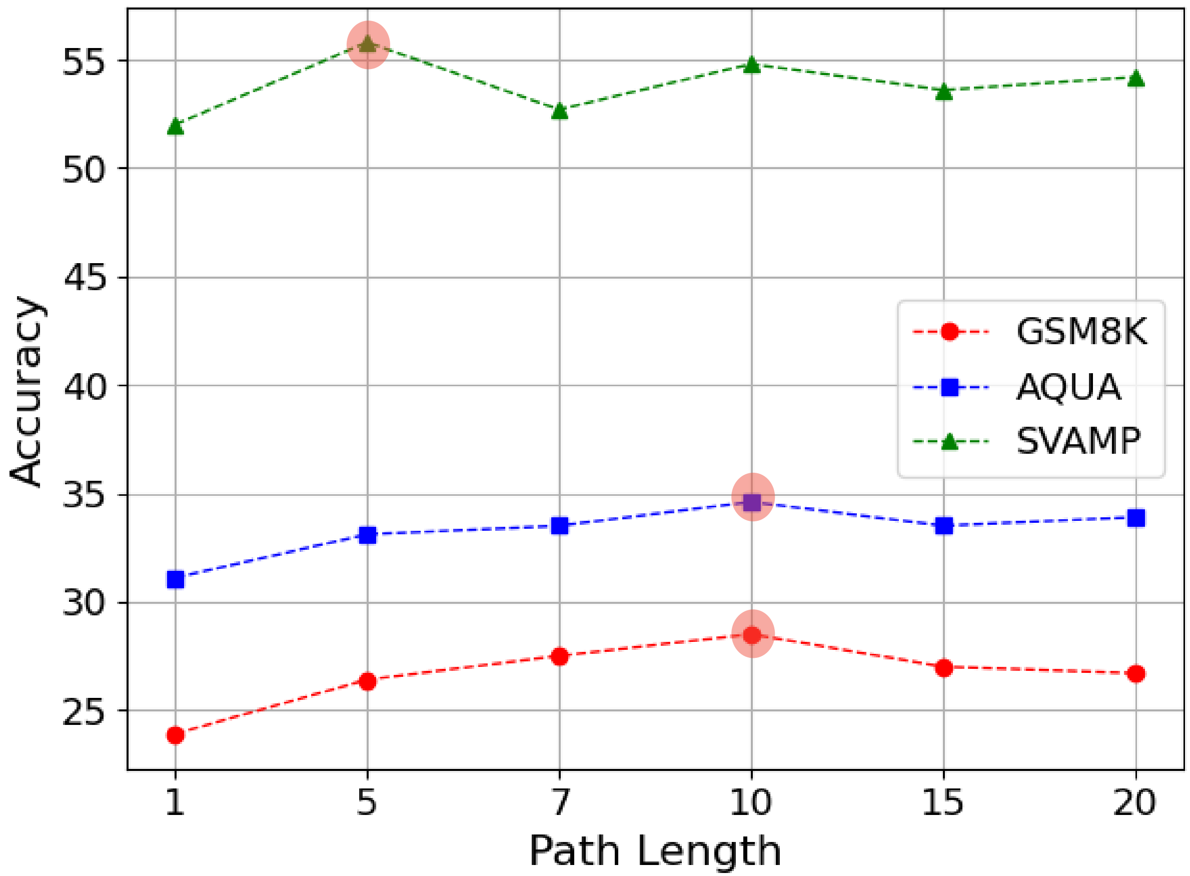
- Test accuracy (%) of GPT2 pre-trained on different length random walk paths.
- All entities and relationships are translated as new tokens. i.e. no natural language involved.

Math reasoning with Chain-of-thoughts (CoT)

- **Reasoning graph**: CoTs can be regarded as walking on a graph whose nodes represent the current reasoning state.
- **Encode reasoning state**: cumulatively embed each CoT step with a pre-trained LM.
- **Construct nodes**: cluster the CoT steps. Each cluster represents a node in the reasoning graph.
- **Random walk** on this graph:
  - Step 1. Randomly select a starting step
  - Step 2. Follow the original CoT for m steps
  - Step 3. In the cluster of the end step, randomly select another step as the next step.
  - Step 4. Go back to step 2.
- This can be regarded as a light-weight **data augmentation** method for CoT reasoning.



Ablation on random walk path length



* Exists an optimal path length

- GSM8K, AQUA, SVAMP are three math word datasets. We LORA fine-tune a Llama 2 (7B) model on CoT data.
- We first do 500 steps of random walk training then 2000 steps of regular supervised fine-tuning.

More results & ablations

Model	Method	GSM8K	AQUA	SVAMP	Avg.
7B	SFT	26.8	30.0	53.3	36.7
	Ours	**28.5**	**34.6**	**55.8**	**39.6**
13B	SFT	37.1	35.0	66.4	46.2
	Ours	**41.2**	**37.4**	**69.0**	**49.2**

Table 1. Testing accuracy of different size Llama 2 models continue pre-trained with our random walk paths and then supervised fine-tuned. The supervised fine-tuning baseline (SFT) is fine-tuned by the same number of total steps. Results are reported on three math word problem (MWP) datasets.

#Nodes	GSM8K	AQUA	SVAMP	Avg.
0	26.8	30.0	53.3	36.7
10	26.8	30.3	54.8	37.3
50	26.6	29.9	54.7	37.1
100	**28.5**	**34.6**	**55.8**	**39.6**
200	26.6	31.1	52.5	36.7

Table 3. Ablation on the number of clusters/nodes K.

#Steps	GSM8K	AQUA	SVAMP	Avg.
0	26.8	30.0	53.3	36.7
200	27.5	30.1	53.6	37.1
500	**28.5**	**34.6**	**55.8**	**39.6**
1000	24.9	32.3	51.6	36.3

Table 2. Ablation on the number of random walk training steps M.

- Ablation on model size, number of clusters, and number of training steps.
- #Nodes = 0 and #Steps = 0 means we don't do any random walk training.

Future Research

Research plan

- Future research directions:
  - Zoom in more: how model parameters correspond to proposed theory.
  - Exploring better ways to verify the proposed theory with real world LLMs.
  - Connections between foundation models of different modalities. E.g. language model and diffusion

Thank you!

Questions?