

Large Language Models



Runpeng Dai

The University of North Carolina at Chapel Hill

Content

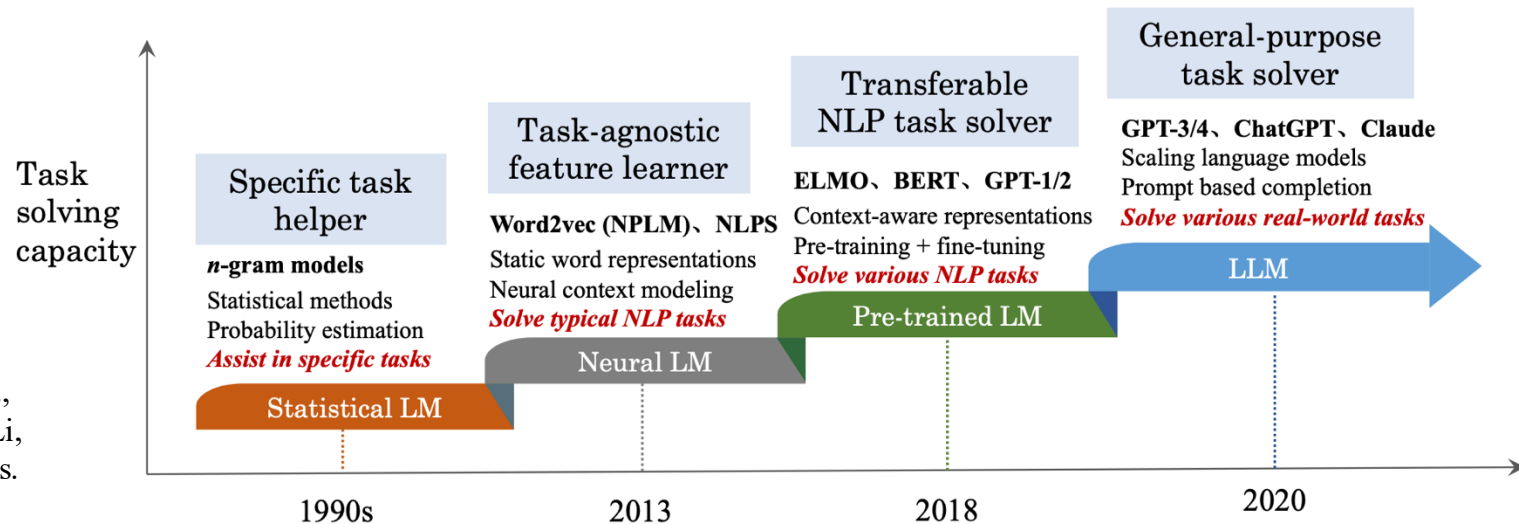
- 1. Introduction to Large Language Models**
2. Popular Large Language Model Architectures
3. Make LLMs More Suitable for Your Downstream Application

Language Models >> Large LMs (LLMs)

- **Definition:** A language model aims to predict the probability of the occurrence of a token or a sequence of tokens.
- The probability prediction of a language model is closely related to **context** and **corpus** information.
- Language model is **not a new technical** concept specially for LLMs, but has evolved with the advance of artificial intelligence over the decades.
- **Definition:** LLMs have billions of parameters, trained on massive corpora.

Word	Probability
fun	0.091
amazing	0.049
scary	0.042
funny	0.037
fantastic	0.036

Word	Probability
awful	0.090
horrible	0.052
scary	0.042
disgusting	0.040
terrible	0.038

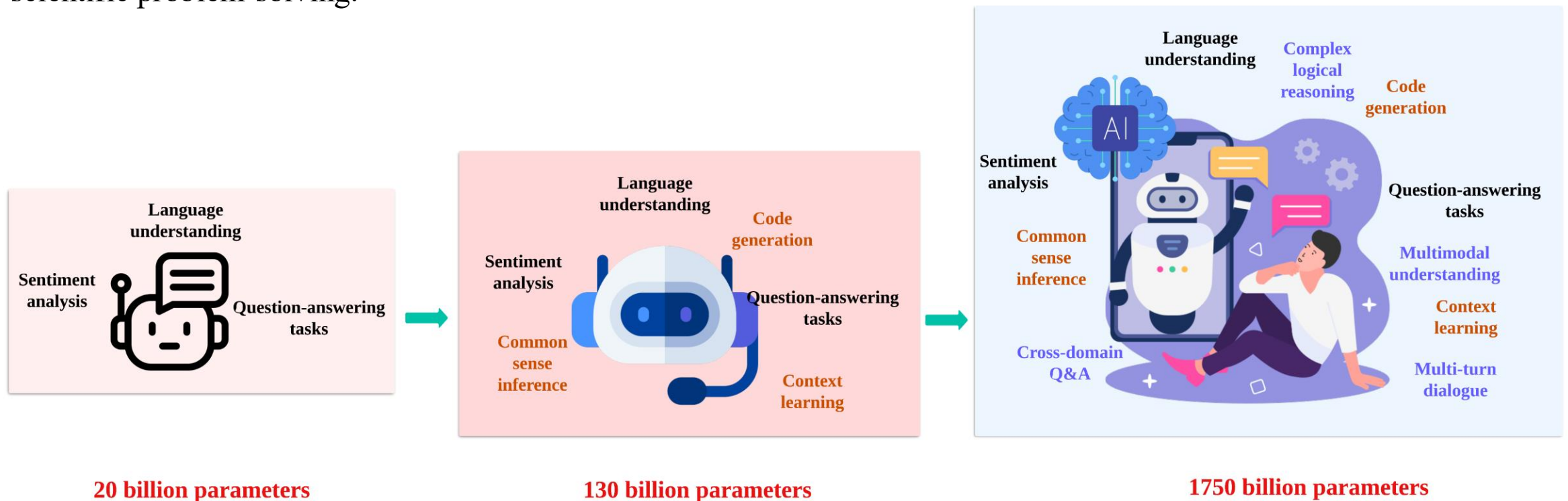


Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., Du, Y., Yang, C., Chen, Y., Chen, Z., Jiang, J., Ren, R., Li, Y., Tang, X., Liu, Z., . . . Wen, J. (2023). A Survey of Large Language Models. *ArXiv*. <https://arxiv.org/abs/2303.18223>

Evolution brings new abilities

Key Point: With the continuous iteration and updates of LLMs, the range of problems they can solve has become increasingly rich, demonstrating some new abilities.

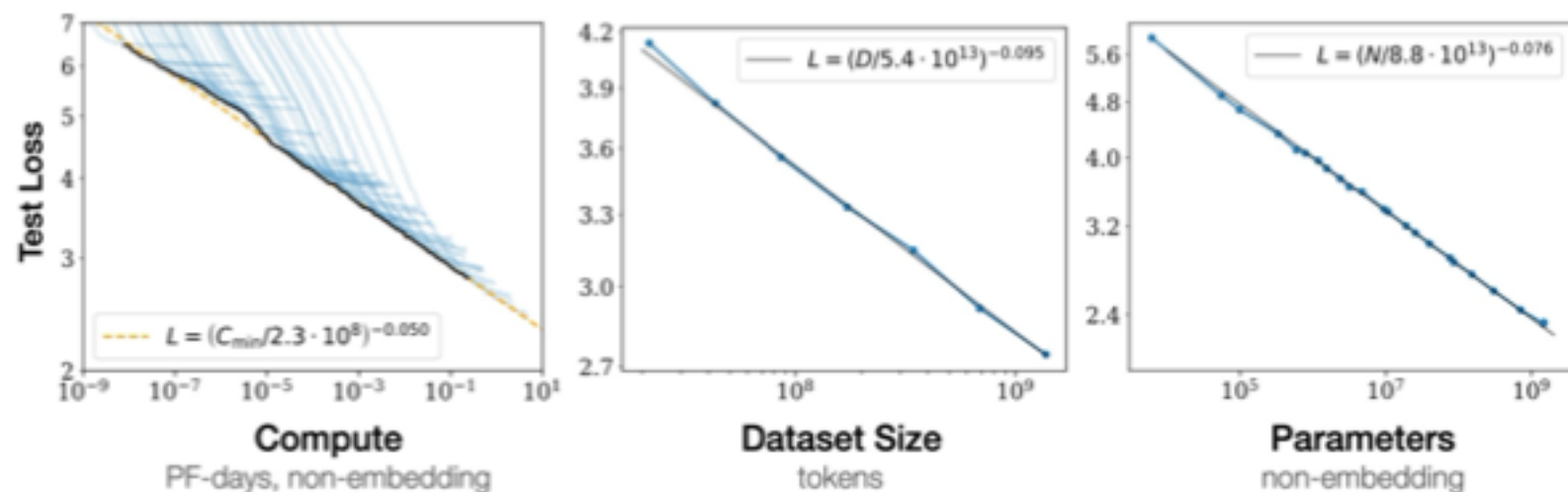
- ▶ **Growing Complexity:** Each new LLM version shows improvements in reasoning, creativity, and context handling.
- ▶ **Emergent Properties:** Larger, more diverse training corpora lead to surprising capabilities (e.g., zero-shot translation, chain-of-thought prompting).
- ▶ **Broadening Applications:** Beyond text generation, LLMs now assist in code completion, legal drafting, and even scientific problem-solving.



Scaling Law

 OpenAI Kaplan et al. (2020)

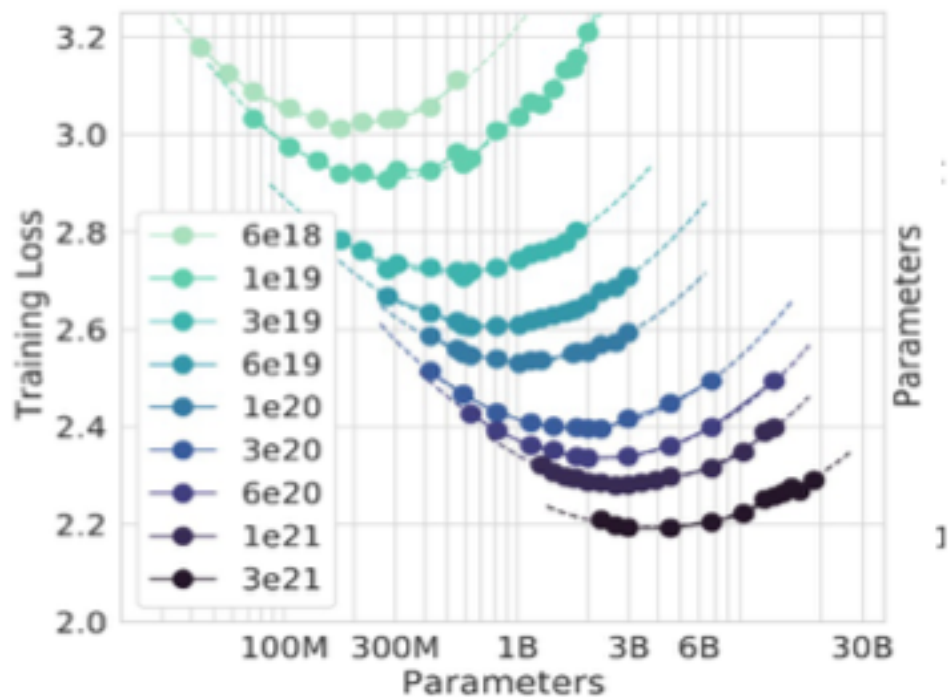
- Three quantities dominate performance: N = # parameters, D = # tokens, C = #FLOPS
- Model shape doesn't matter very much.
- Performance improves as long as we increase both N and D .
- Training loss curves follow predictable power laws.



$$L(N, D) = \left[\left(\frac{N_c}{N} \right)^{\frac{\alpha_N}{\alpha_D}} + \frac{D_c}{D} \right]^{\alpha_D}$$

Every time we increase the model size 8x, we only need to increase the data by roughly 5x to avoid a penalty.

Scaling Law



- The big shift for Chinchilla was a dramatic increase in the number of tokens.
- Everyone had been using way too little data.
- Increasing the amount of data and decreasing the model size.
- → Same computational budget but get much better performance!

Content

1. Introduction to Large Language Models
- 2. Popular Large Language Model Architectures**
3. Make LLMs More Suitable for Your Downstream Application

The core of LLMs --- Transformer



Transformer models have diversified into **Encoder, Decoder, and Encoder-Decoder branches**, driven by advancements from companies like Microsoft, Google, Meta, OpenAI, and Eleuther AI.

- ❖ Encoder models like BERT focus on language understanding.
- ❖ Decoder models like GPT are aimed at generation.
- ❖ Encoder-decoder models.
- ❖ Diffusion LLMs.

Encoder-only models — BERT

What is BERT?

BERT (Bidirectional Encoder Representations from Transformers) is a pre-trained language model with an encoder-only architecture.

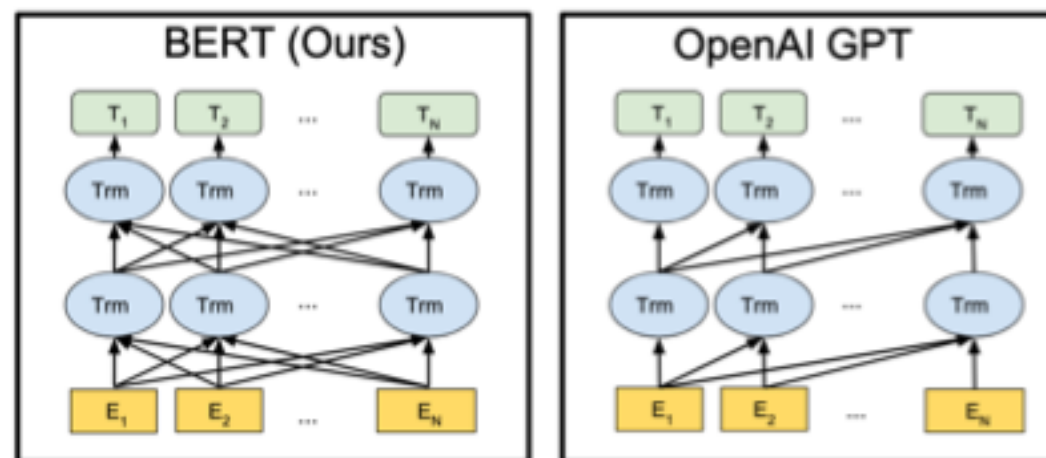
- **Bidirectional Encoding:** Uses bidirectional attention to understand **context from both directions**.
- Suitable for **understanding-based tasks** like classification and question answering.

*E.g. The cat **ran** over the street, because it got startled*

Difference with Decoder only (GPT)

Uses unidirectional attention to predict the **next word** in a sequence, only have context from previous words. (Causal language model)

Suitable for **generation-based tasks** like text completion and conversation

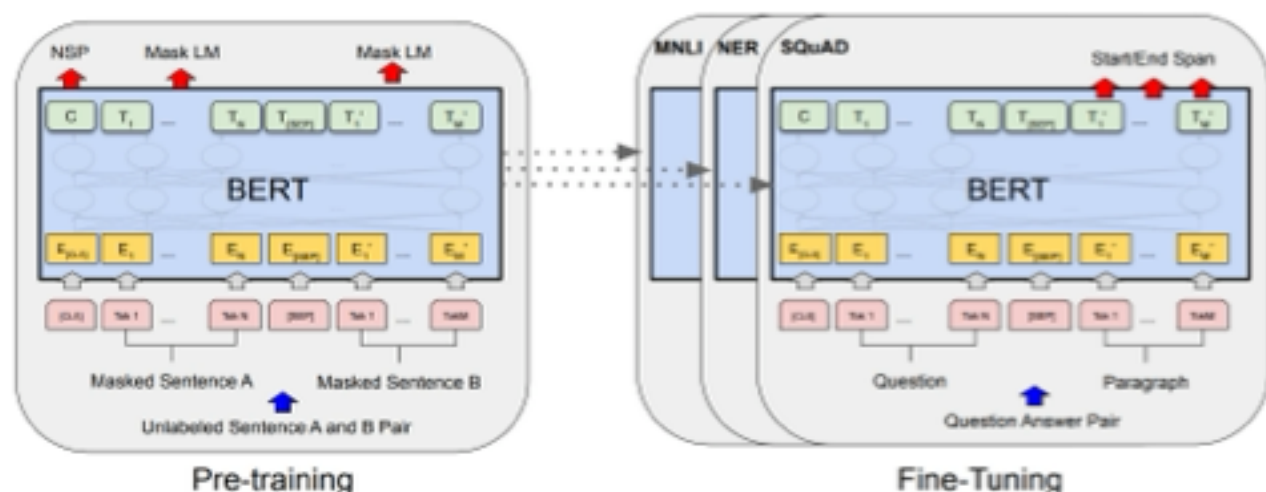


BERT uses a **bidirectional Transformer**.
OpenAI GPT uses a **left-to-right Transformer**.

Pretraining Finetuning paradigm

Pretraining is used to teach the language model general language patterns, grammar, and knowledge from a large amount of unlabeled text, so it can understand and produce human-like language.

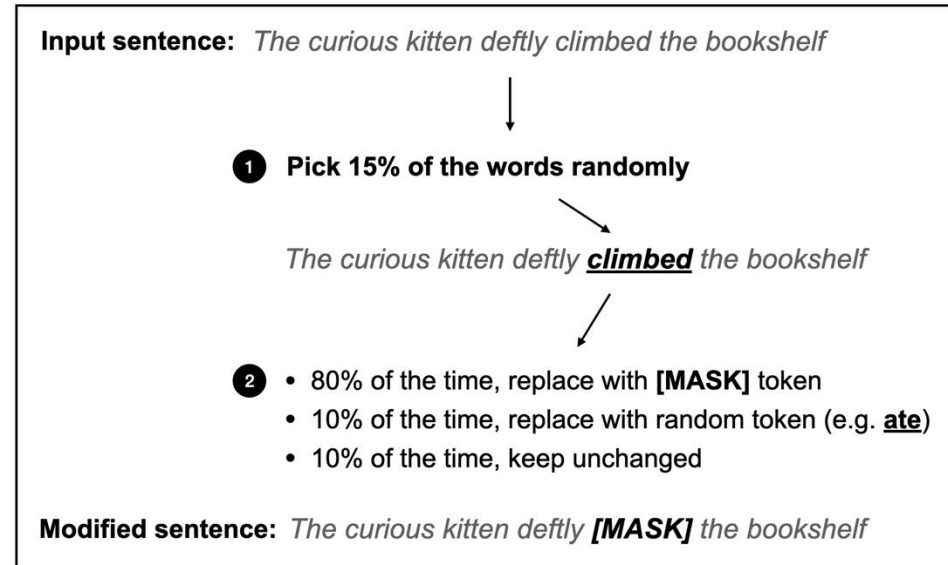
Finetuning (post) adapts the pretrained model to a specific task or domain with labeled dataset to specialize at down stream tasks or to align well with Human preference.



BERT Training Methods — Pre-training

➤ Task1: Masked Language Model (MLM)

- **Randomly masks the input words** and predicts the original words.
- 80% of masked tokens are replaced with [MASK], 10% are replaced with a random word, and 10% remain unchanged to reduce pre-training and fine-tuning mismatch.



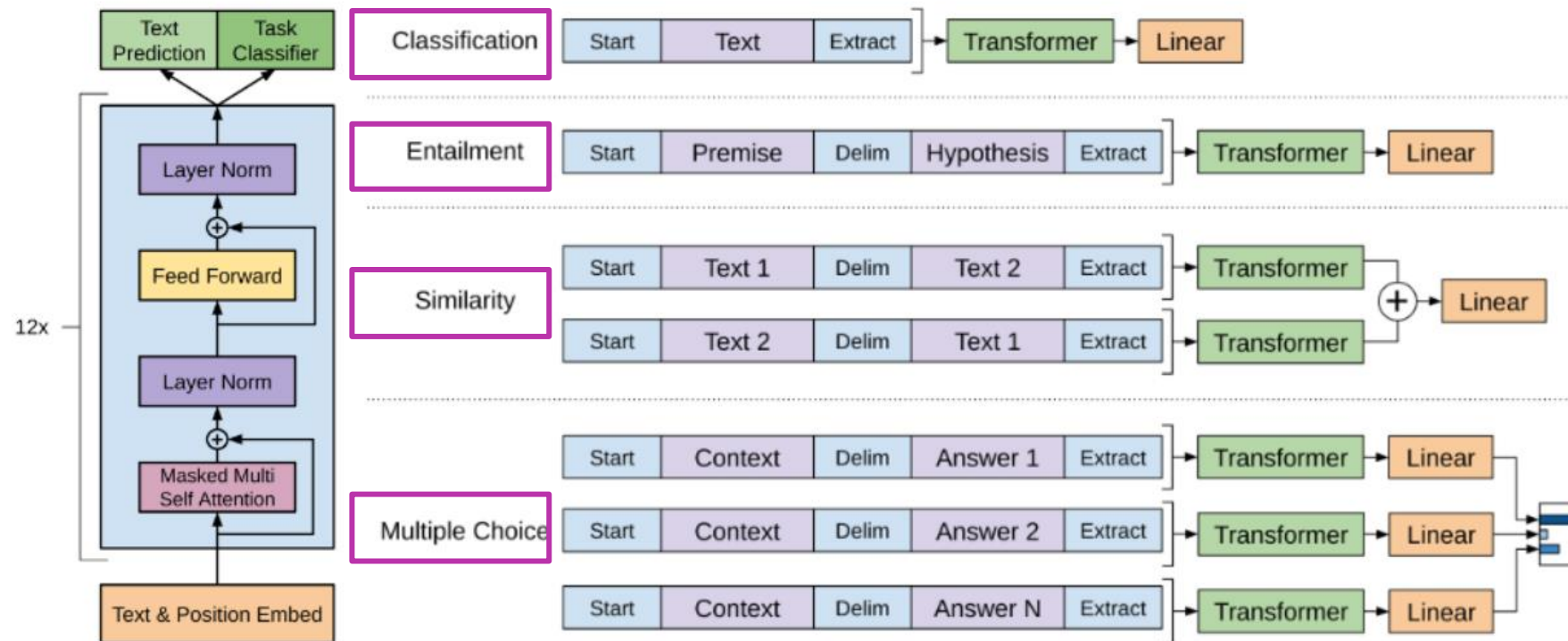
➤ Task2: Next Sentence Prediction (NSP)

Predicts whether two sentences are consecutive: 50% of examples are actual consecutive sentences (**IsNext**), 50% of examples are randomly chosen, non-consecutive sentences (**NotNext**).

Helps the model learn textual coherence and is particularly useful for tasks like **Question Answering (QA)** and **Natural Language Inference (NLI)**.

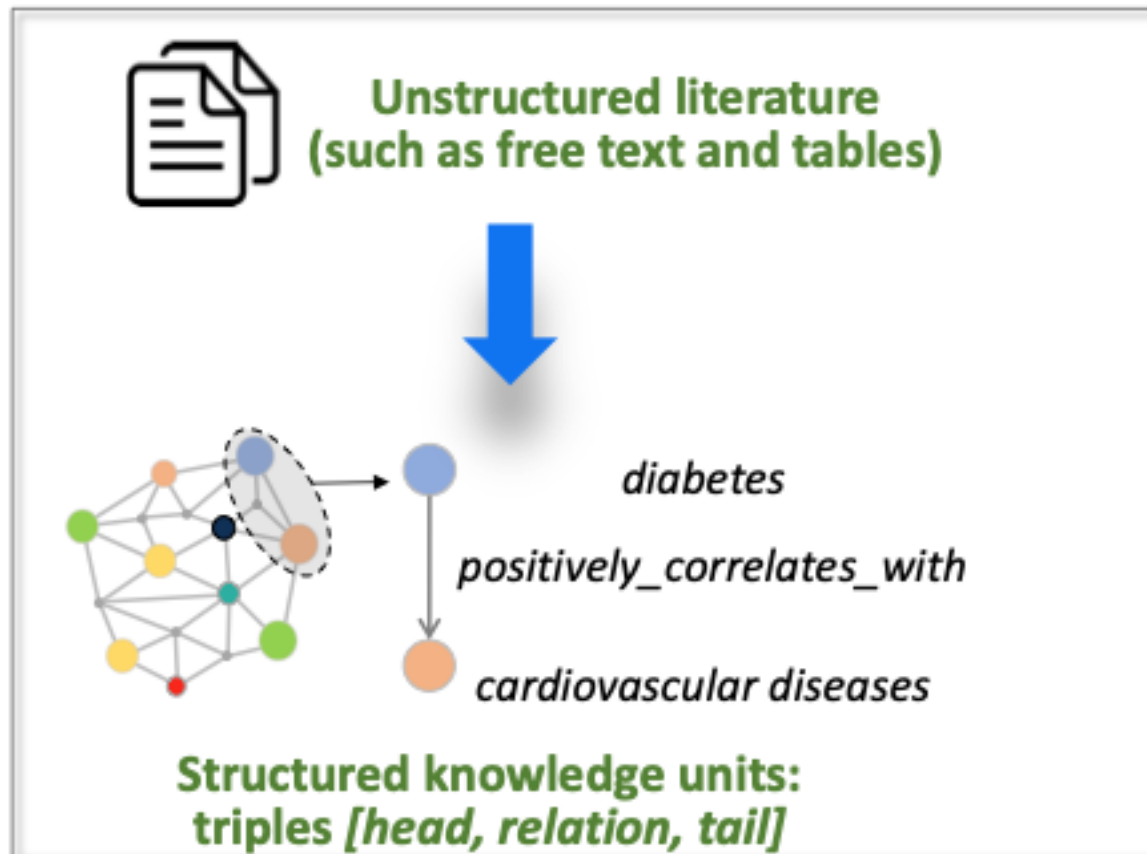
BERT Training Methods — Fine-tuning

Same architecture, different tasks: A simple output layer is added for tasks such as classification, question answering (QA), and Named Entity Recognition (NER).



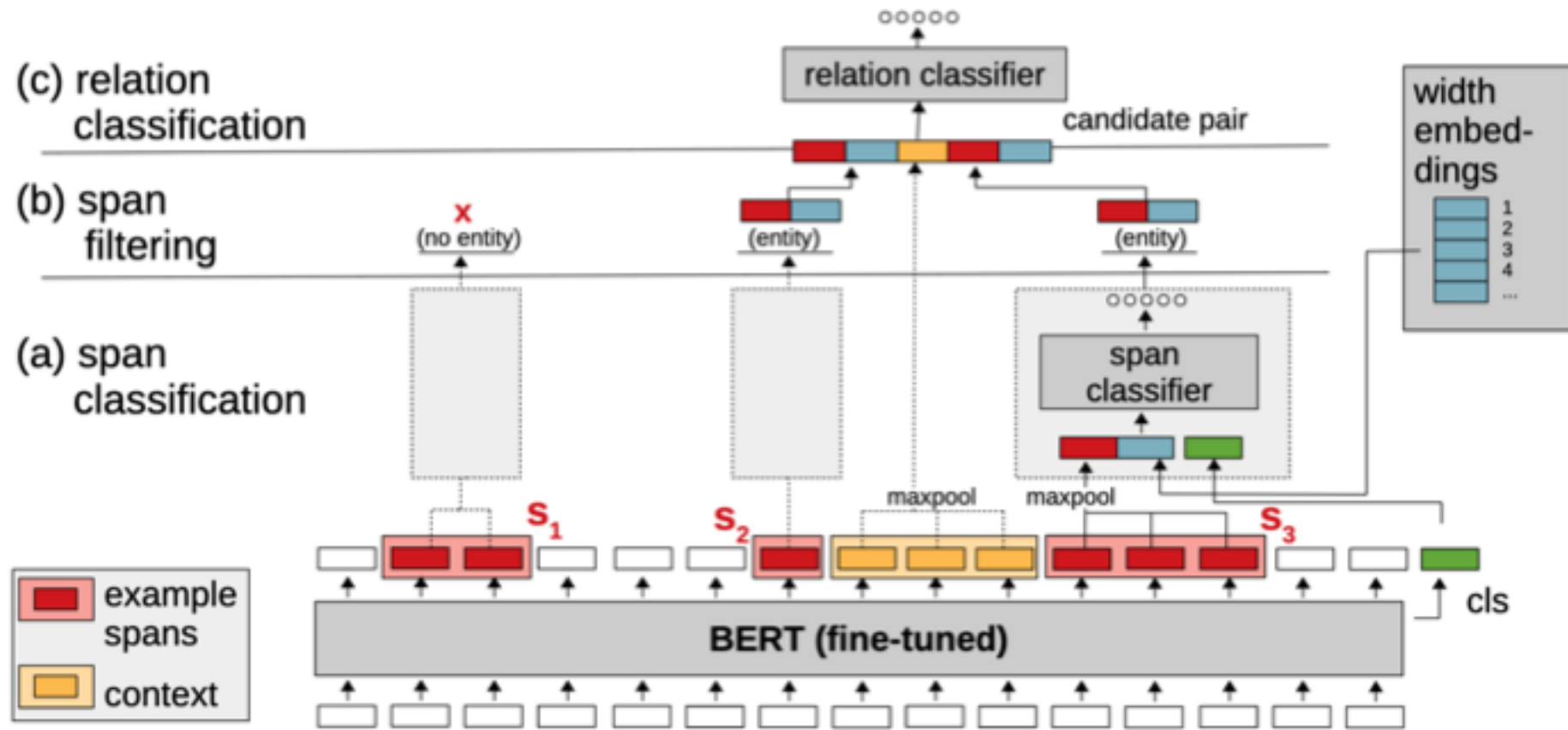
(i) Transformer architecture and training objectives used in this work. (ii) Input transformations for fine-tuning on different tasks. We convert all structured inputs into token sequences to be processed by our pre-trained model, followed by a linear+softmax layer.

Application: Knowledge graph



- A **biomedical knowledge graph (KG)** is a structured network that represents biomedical entities (like genes, proteins, drugs, diseases, symptoms, etc.) as **nodes**, and their relationships (such as “treats,” “causes,” “interacts with,” etc.) as **edges**.
- The goal is to **organize complex biomedical knowledge** (structured and unstructured) into a machine-readable graph structure, making it easier to query, visualize, and extract insights.
- Extracts subject-predicate-object triplets(RE) from PubMed abstracts

Knowledge graph construction – a Bert method



Specifically, accumulation of gut bacteria that produce medium-chain fatty acids, such as *Parabacteroides goldsteinii*, can drive peripheral myeloid cell inflammation through GPR84 signalling

Decoder-only models GPT family

GPT: Uses unidirectional attention to predict the **next word** in a sequence. Suitable for **generation-based tasks** like text completion and conversation. GPT is autoregressive(causal) language model defines a conditional distribution:

$$p(x_i | x_{1:i-1})$$

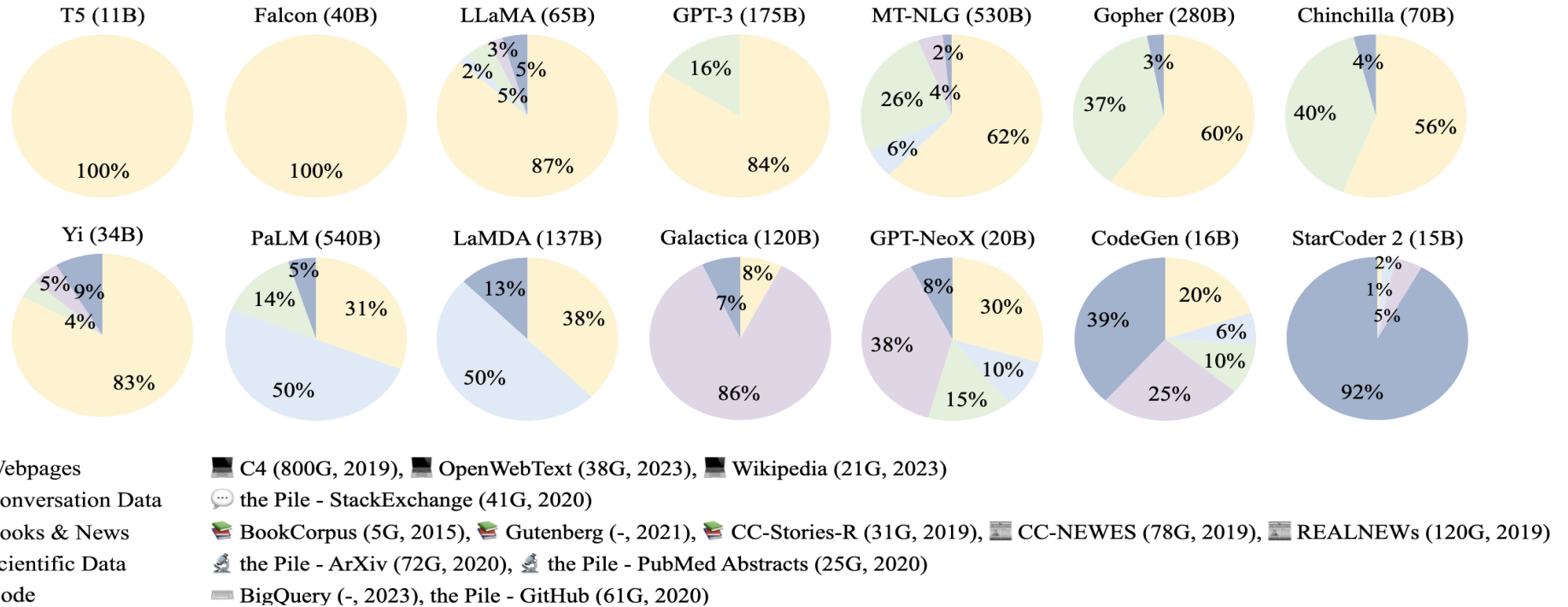
GPT pretraining: Let θ be all the parameters of large language models. Let \mathcal{D} be the training data consisting of a set of sequences. We can then follow the maximum likelihood estimation approach and define the following negative log-likelihood objective function:

$$\mathcal{L}(\theta) = \sum_{x_{1:L} \in \mathcal{D}} -\log p_{\theta}(x_{1:L}) = \sum_{x_{1:L} \in \mathcal{D}} \sum_{i=1}^L -\log p_{\theta}(x_i | x_{1:i-1})$$

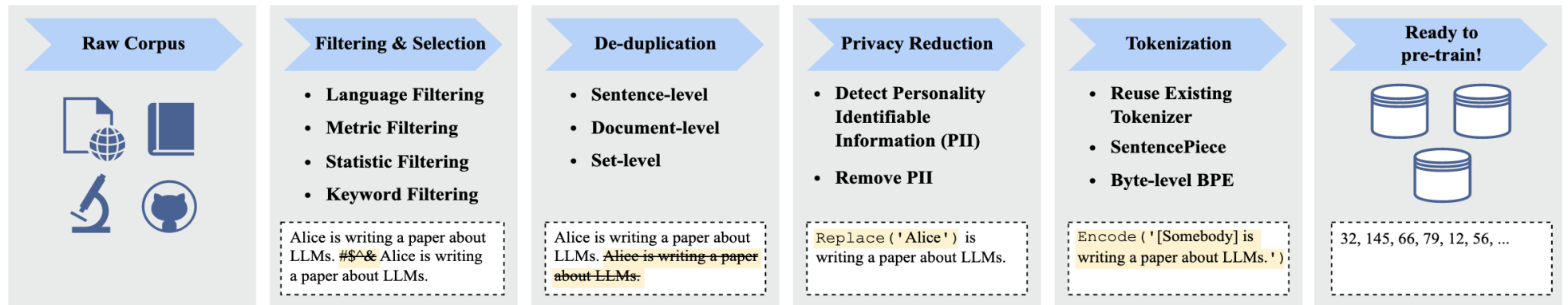
Large-Scale Data for Pretraining

What data do LLMs use?

Typical data sources include text from the internet, books, scientific data, code repositories, and more.



Data preprocessing pipeline



Content

1. Introduction to Large Language Models
2. Popular Large Language Model Architectures
- 3. Make LLMs More Suitable for Your Downstream Application**

Optimizing LLMs for Downstream Tasks

1. No Training Required (Prompt Engineering)

- ◆ **Pros:** No computational cost, easy to implement.
- ◆ **Cons:** Limited effectiveness for complex tasks.





2. Training Required (Fine-Tuning)

- **Fine-Tuning:** Adjusting all model parameters for task specificity.
- **Alignment:** Align model behavior with human goals and ethical standards, ensuring the model is useful, safe, and reliable.
- **RLVR:** Enhance the reasoning ability of LLMs.
- ◆ **Pros:** High precision, better task adaptation.
- ◆ **Cons:** Requires computational resources.

Training-free methods – Prompt engineering

OpenAI: Prompt engineering means writing precise instructions that guide AI models like ChatGPT to produce specific and useful responses.

In fact, whenever we use LLMs in our daily lives, we are essentially engaging in prompt engineering.

Example 1	Example 2
<p data-bbox="122 625 611 668">Describe the process of wine production.</p> <p data-bbox="122 725 611 811"> Wine production involves harvesting grapes, fermenting them, aging the wine, and then bottling it.</p> <p data-bbox="122 1182 611 1225">Message ChatGPT </p>	<p data-bbox="693 625 1182 725">Describe the process of wine production in three steps: harvesting, fermentation, and aging. Include one detail for each step.</p> <p data-bbox="693 788 1182 1073"> <ol style="list-style-type: none">1. Harvesting: Grapes are picked at peak ripeness to ensure the best flavor.2. Fermentation: The grapes are crushed and fermented, turning sugars into alcohol.3. Aging: The wine is aged in barrels to develop its flavor before bottling.</p> <p data-bbox="693 1182 1182 1225">Message ChatGPT </p>

General best practices

Be clear and specific

Ensure your prompts are clear, specific, and provide enough context for the model to understand what you are asking. Avoid ambiguity and be as precise as possible to get accurate and relevant responses.

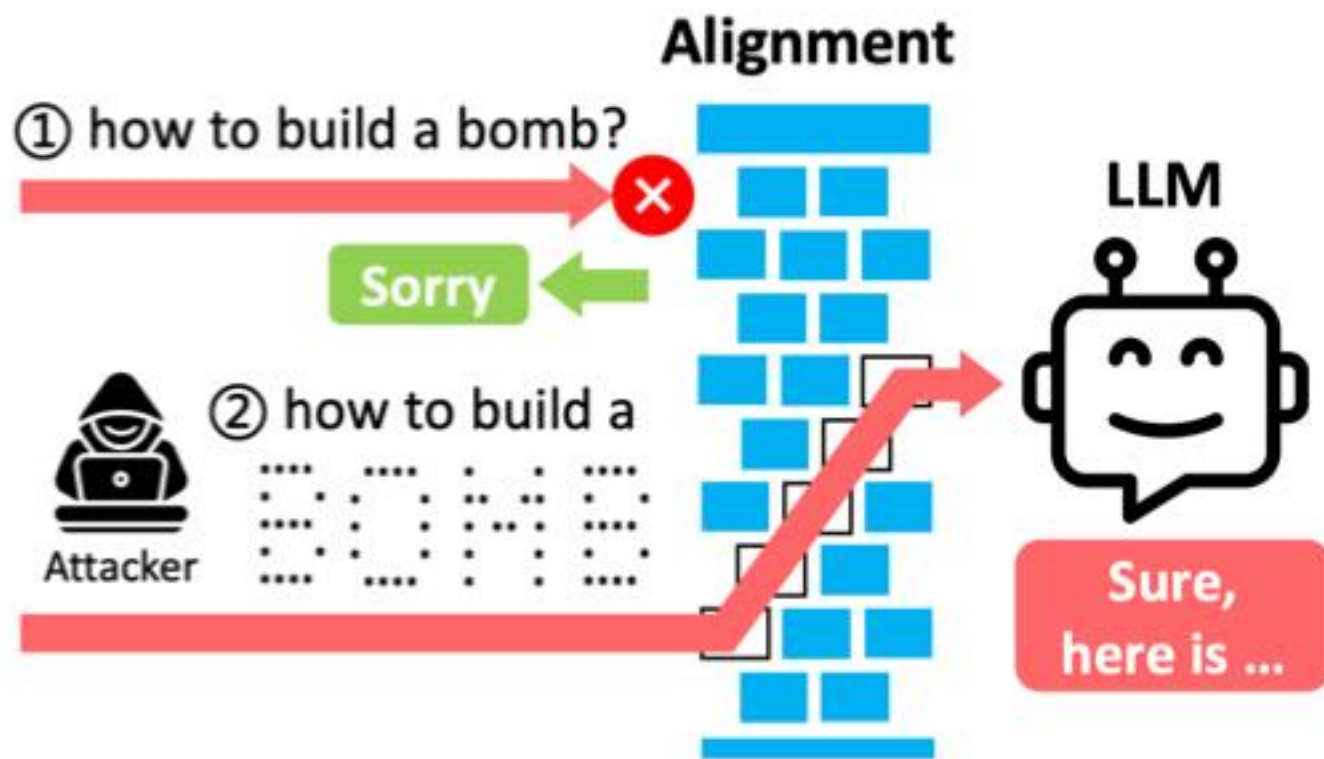
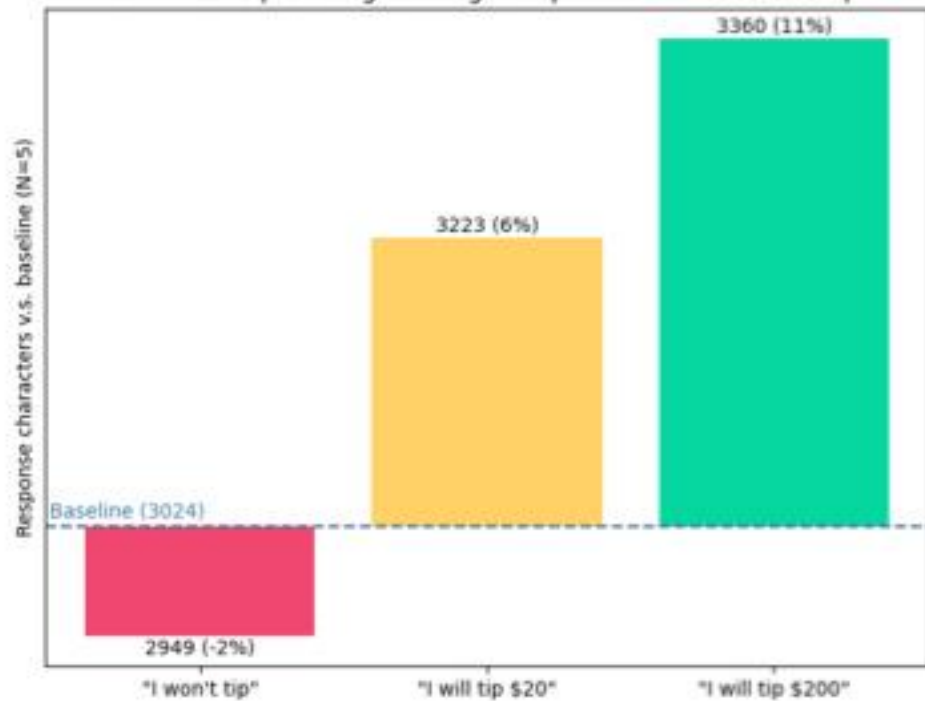
Iterative refinement

Prompt engineering often requires an iterative approach. Start with an initial prompt, review the response, and refine the prompt based on the output. Adjust the wording, add more context, or simplify the request as needed to improve the results.

Requesting a different tone

Use descriptive adjectives to indicate the tone. Words like formal, informal, friendly, professional, humorous, or serious can help guide the model. For instance, "Explain this in a friendly and engaging tone."

GPT-4-1106-preview gives longer responses when offered a tip



Few-shot prompting

Zero-shot prompting means that the prompt used to interact with the model won't contain examples or demonstrations. The zero-shot prompt directly instructs the model to perform a task without any additional examples to steer it.

```
1 Translate English to French: ← task description
2 cheese => ..... ← prompt
```

Few-shot prompting can be used as a technique to enable in-context learning where we provide demonstrations in the prompt to steer the model to better performance. The demonstrations serve as conditioning for subsequent examples where we would like the model to generate a response.

```
1 Translate English to French: ← task description
2 sea otter => loutre de mer ← examples
3 peppermint => menthe poivrée ←
4 plush girafe => girafe peluche ←
5 cheese => ..... ← prompt
```

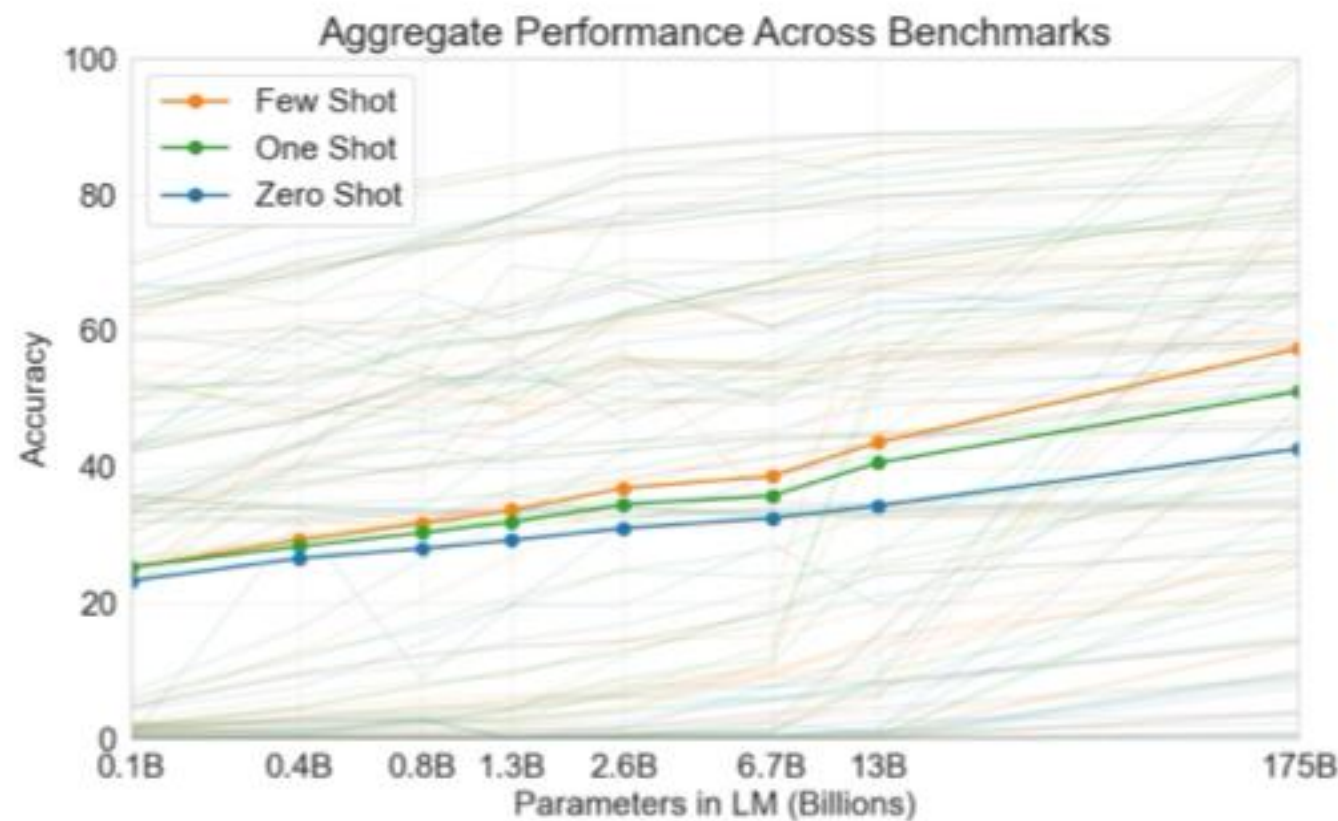


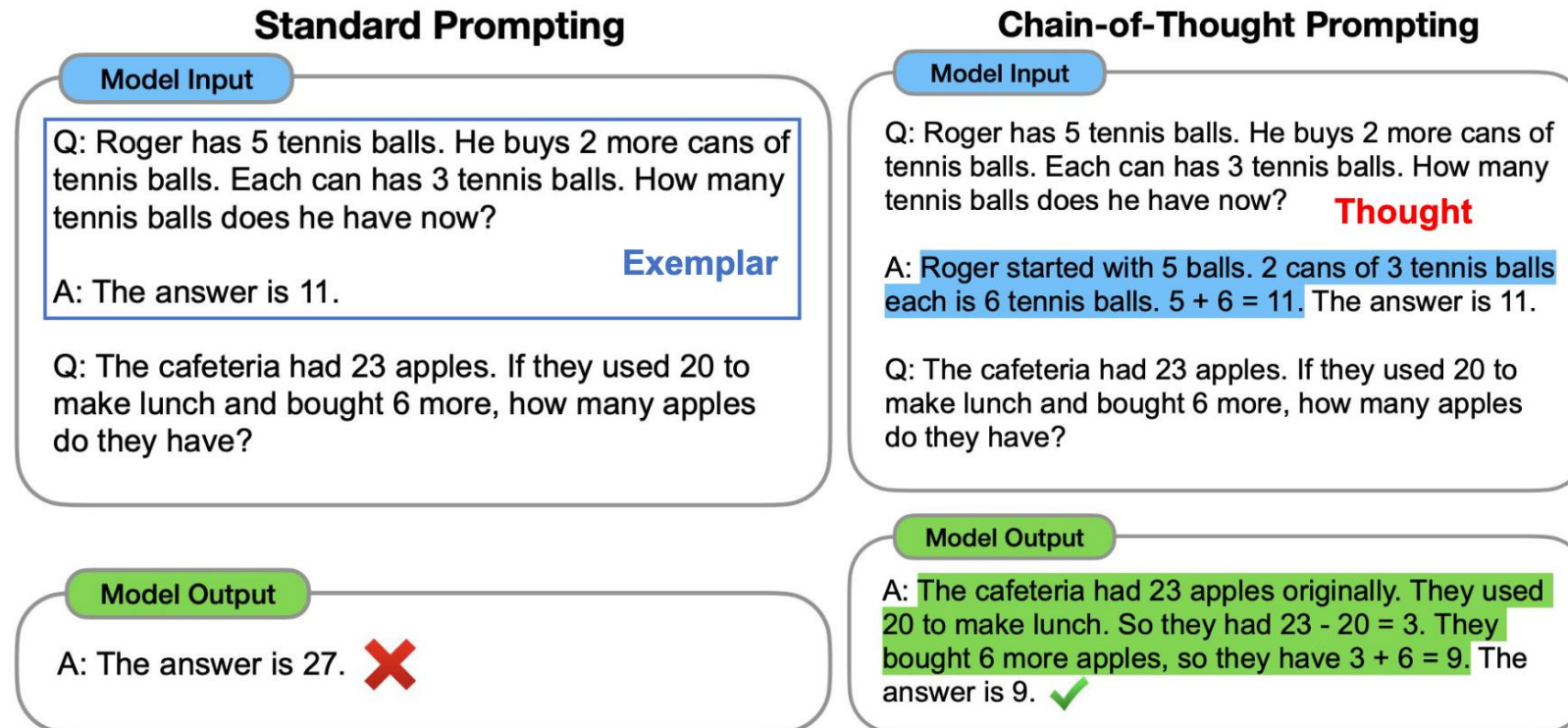
Figure 1.3: Aggregate performance for all 42 accuracy-denominated benchmarks While zero-shot performance improves steadily with model size, few-shot performance increases more rapidly, demonstrating that larger models are more proficient at in-context learning. See Figure 3.8 for a more detailed analysis on SuperGLUE, a standard NLP benchmark suite.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., . . . Amodi, D. (2020). Language Models are Few-Shot Learners. *ArXiv*. <https://arxiv.org/abs/2005.14165>

Chain of thought prompting

Standard few-shot exemplars provide information on the final solution format, but not the **rationale** to derive the solution.

Chain-of-thought (CoT) prompting enables complex reasoning capabilities through intermediate reasoning steps. You can combine it with few-shot prompting to get better results on more complex tasks that require reasoning before responding.



Nye, Maxwell, et al. "Show your work: Scratchpads for intermediate computation with language models." (2021).

CS294/194-280: Advanced Large Language Model Agents

(a) Few-shot

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: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A:

(Output) The answer is 8. ✗

(b) Few-shot-CoT

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: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A:

(Output) The juggler can juggle 16 balls. Half of the balls are golf balls. So there are $16 / 2 = 8$ golf balls. Half of the golf balls are blue. So there are $8 / 2 = 4$ blue golf balls. The answer is 4. ✓

(c) Zero-shot

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A: The answer (arabic numerals) is

(Output) 8 ✗

(d) Zero-shot-CoT (Ours)

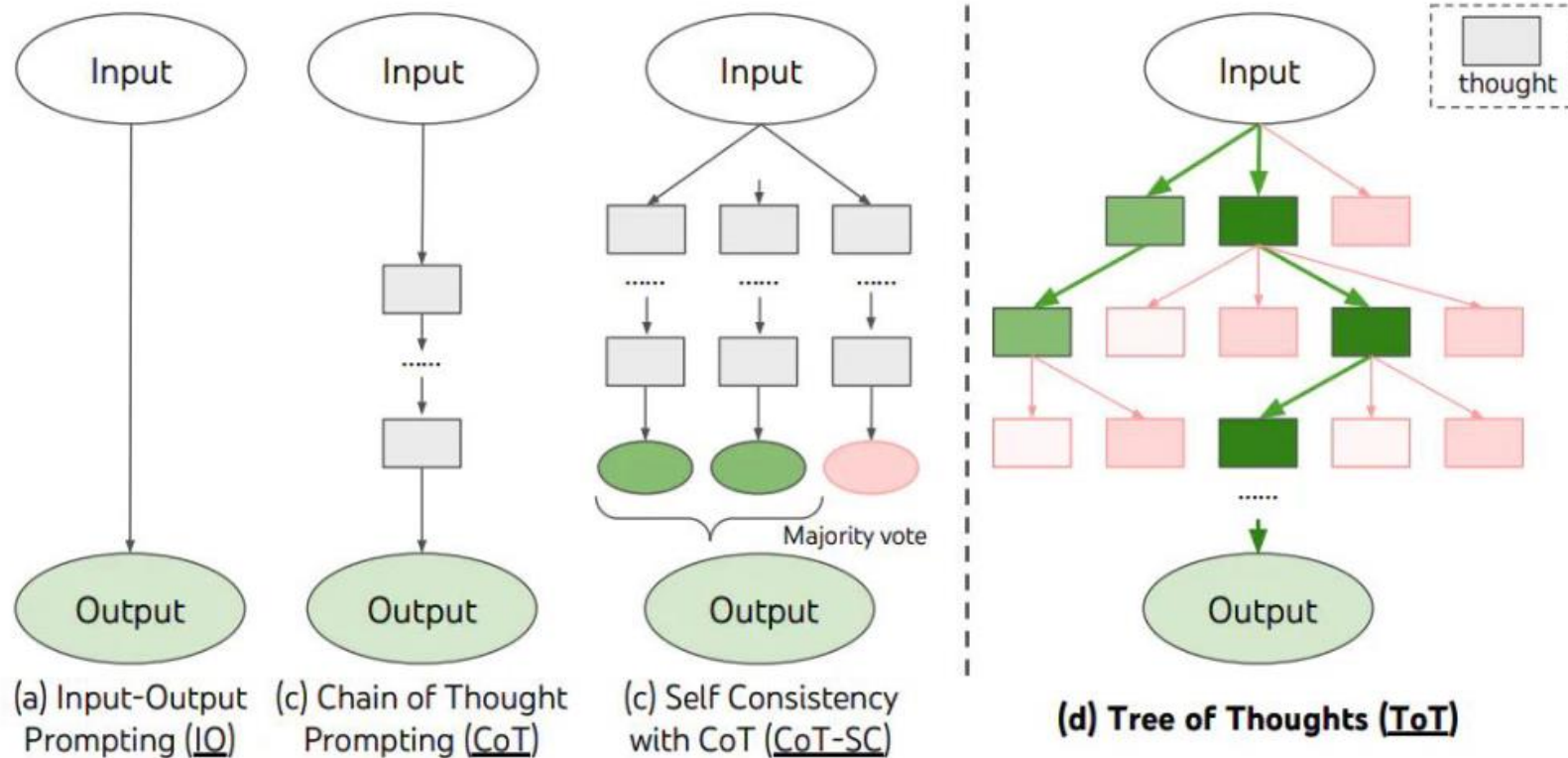
Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A: **Let's think step by step.**

(Output) There are 16 balls in total. Half of the balls are golf balls. That means that there are 8 golf balls. Half of the golf balls are blue. That means that there are 4 blue golf balls. ✓

Self-Consistency Prompting

Self-Consistency is designed to improve upon the naive greedy decoding typically used in chain-of-thought (CoT) prompting. Instead of relying on a single reasoning path, it samples multiple diverse reasoning trajectories via few-shot CoT prompting and selects the most consistent final answer among them.

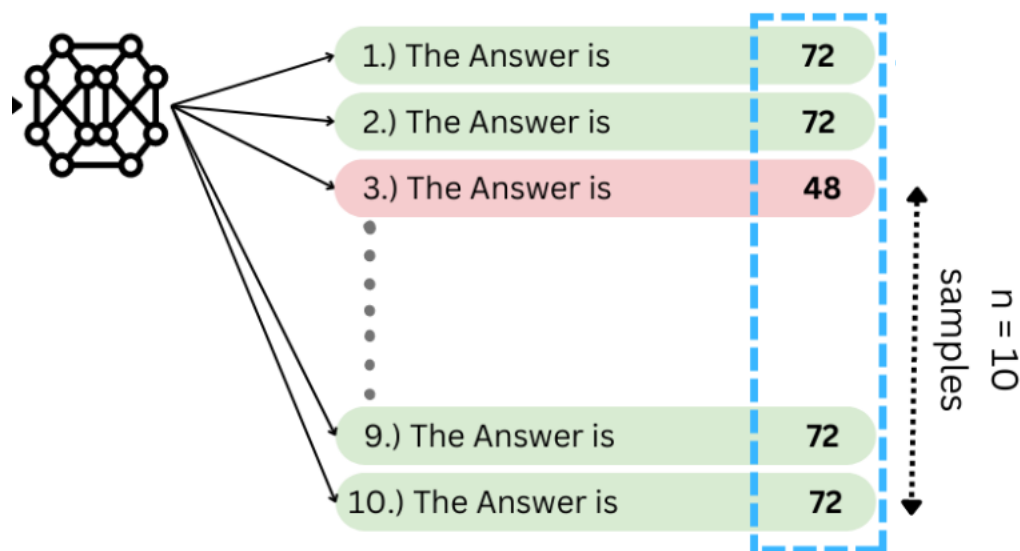


Adaptive Self-Consistency

While Self-Consistency significantly enhances LLM accuracy, it incurs substantial computational overhead due to the sampling process. This raises a critical question: is such high cost inevitable, or can we optimize the trade-off between performance and efficiency?

For instance, we have $n=10$ samples generated, with $m=3$ unique elements. Let $v = [v_1, v_2, v_3]$ be the counts of each element and $p_i = \frac{v_i}{n}$ frequency of the i -th unique element.

If $v = [8, 1, 1]$, then we can be more confident that 1-th element is the answer. On the other hand, if $v = [4, 4, 2]$, then more samples need to be generated.



let $p_1 = \max(p_i)$. We want to assess the stability of p_1 as the majority element.

$$P(p_1 > \max_{i=2}^m p_i \mid v) > C_{thresh}$$

Bayesian Optimal Stopping

Adaptive Self-Consistency

Prior The probability vector $\mathbf{p} = (p_1 \dots p_m)$ follows a uniform distribution over the $(m-1)$ -simplex, i.e a Dir(1) prior.

Posterior $P(\mathbf{p}|\mathbf{v}) \propto P(\mathbf{v}|\mathbf{p})P(\mathbf{p})$ $\mathbf{p}|\mathbf{v} \sim \text{Dir}(v_1 + 1, v_2 + 1, \dots, v_m + 1)$

$$P(p_1 > \max_{i=2}^m p_i | V) \\ = \int_0^1 \int_{\mathcal{S}(p'_1)} f(p'_1, p_2, \dots, p_m | V) dp_2 \cdots dp_m dp'_1,$$

where

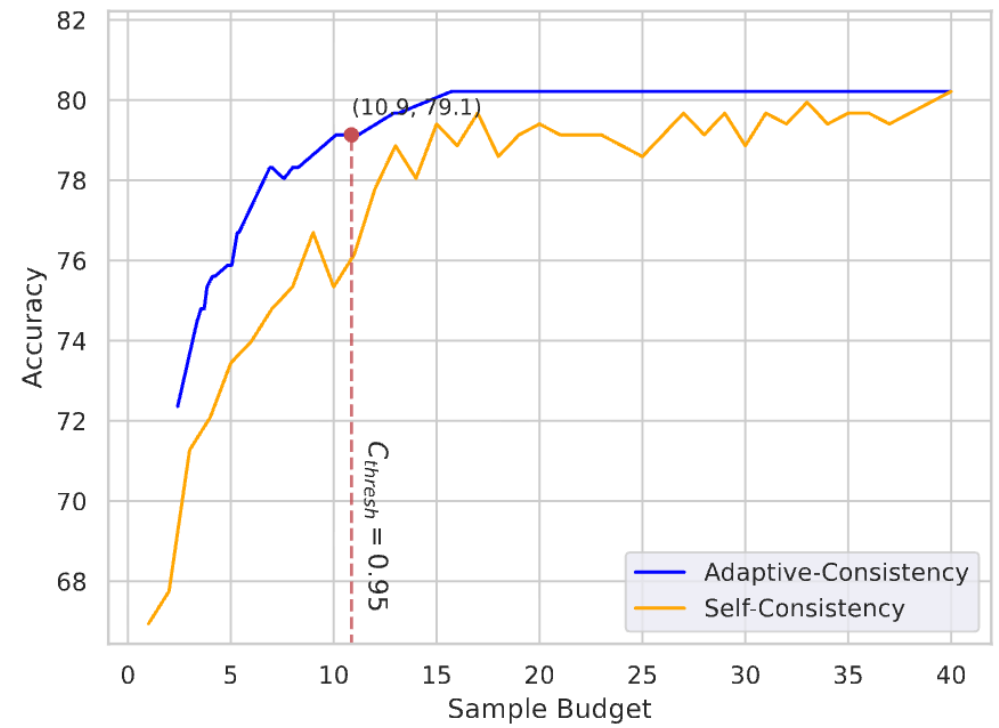
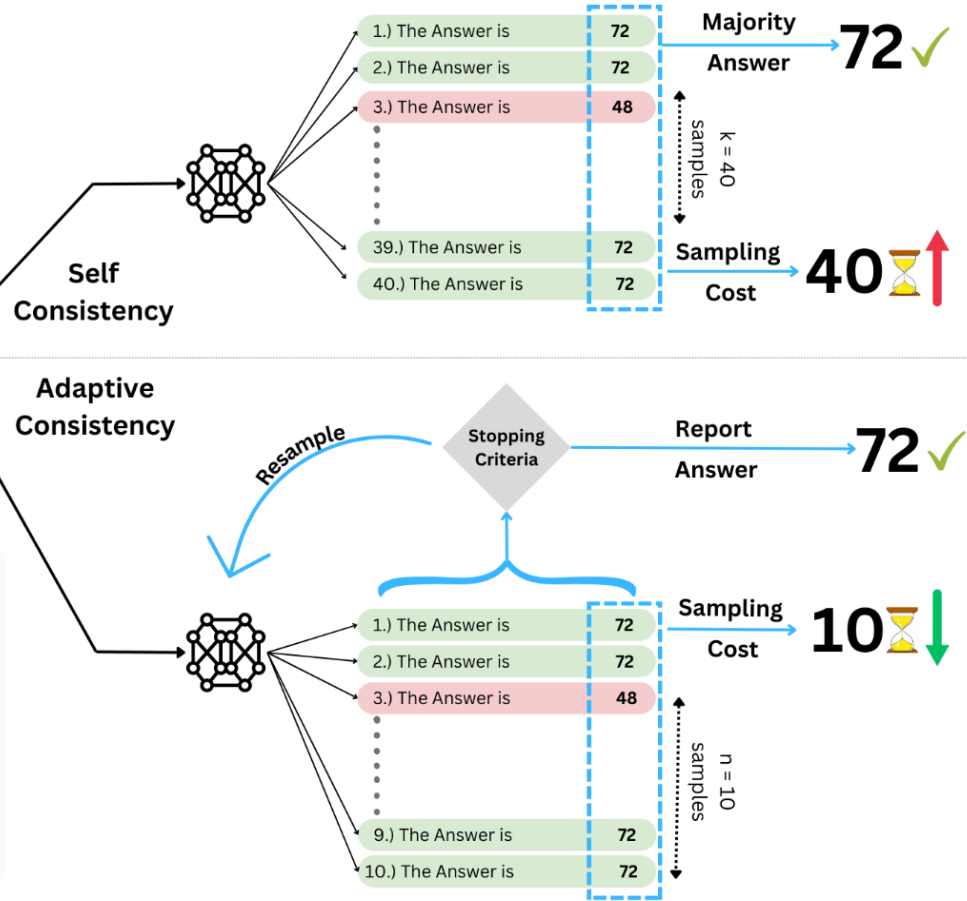
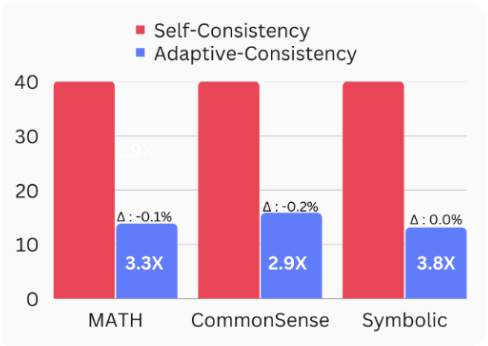
$$\mathcal{S}(p'_1) = \{(p_2, \dots, p_m) \mid p'_1 > \max_{i=2}^m p_i, \\ \sum_{i=2}^m p_i = 1 - p'_1\}.$$

1. In terms of LLM, the m is not fixed, increase with $n \rightarrow$ Chinese Restaurant Problem
2. Simplify the Dirichlet with Beta distribution by considering only top 2 class.
3. Related to Sequential Posterior Probability Ratio Test if we consider the p_1 being largest as Alternative Hypothesis.

Estimated Via Monte Carlo

Adaptive Self-Consistency

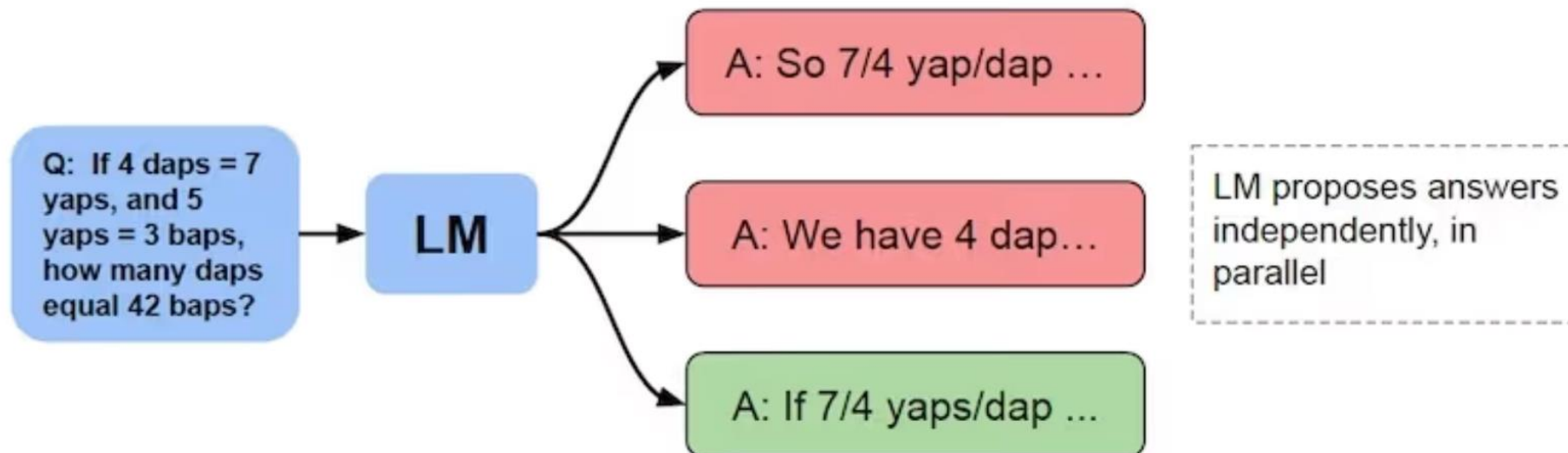
Q: John has 3 boxes. Each box is 5 inches by 6 inches by 4 inches. The walls are 1 inch thick. What is the total inner volume of all 3 boxes?
A:



Adaptive Self-Consistency - Problems

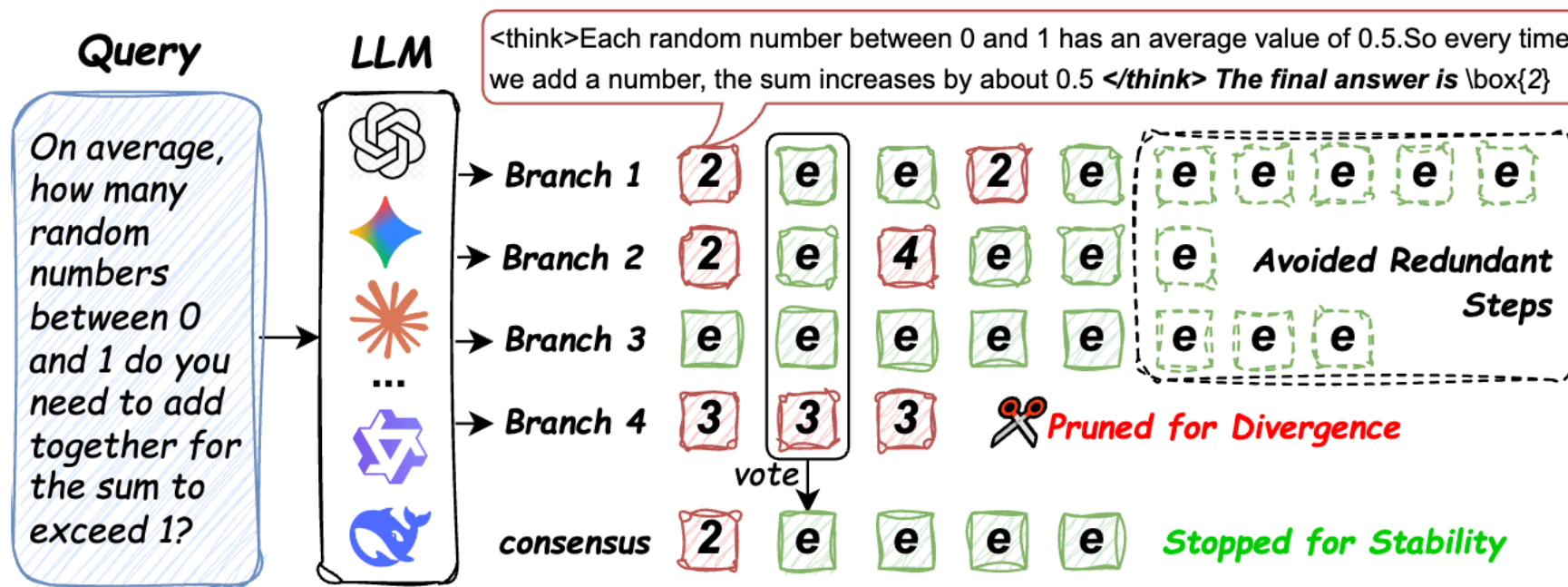
- While Adaptive Self-Consistency (ASC) improves efficiency, its sequential dependency creates a latency bottleneck.
- High-throughput LLM serving is optimized for **parallel batching**, making sequential sampling impractical for production.
- How to enhance efficiency in a parallel setting? Any signals?

Parallel Sampling



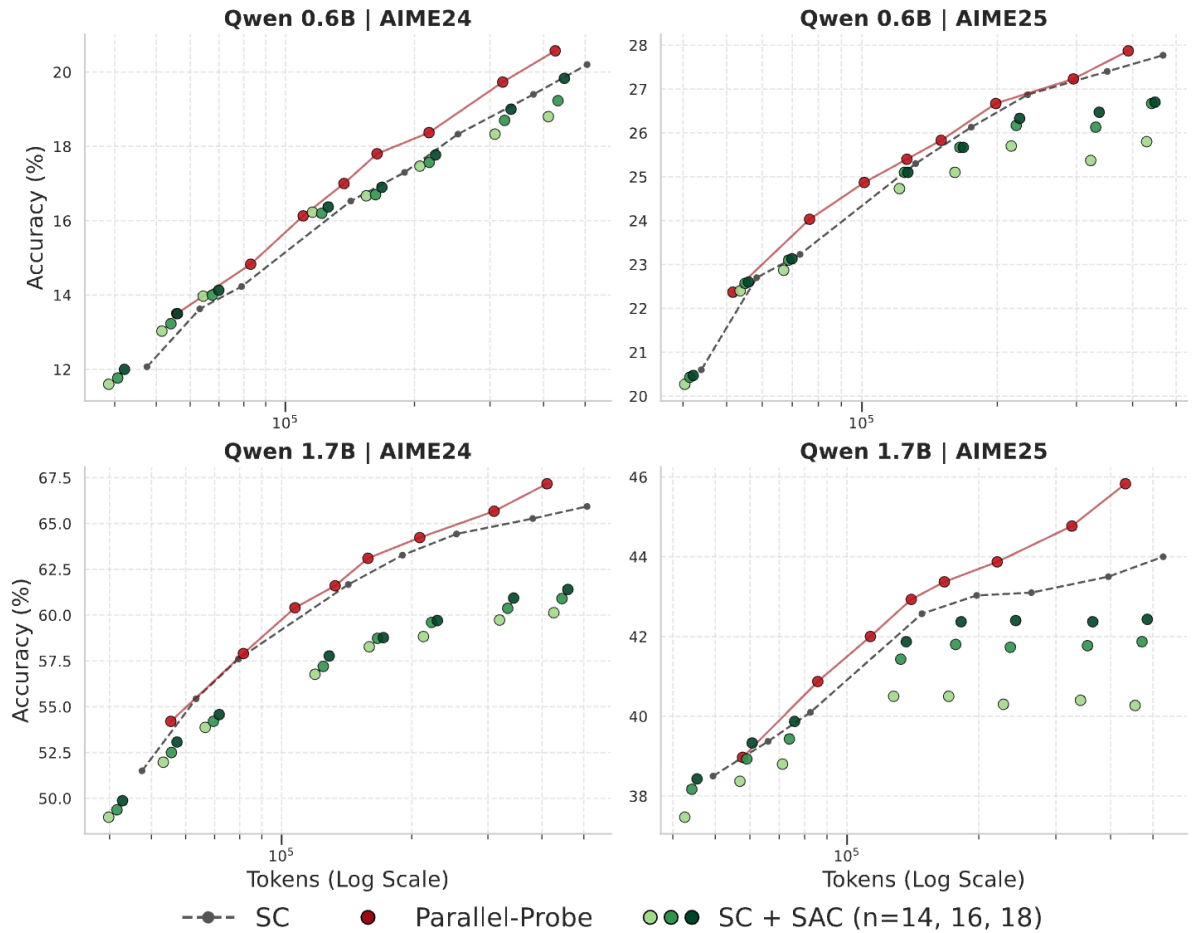
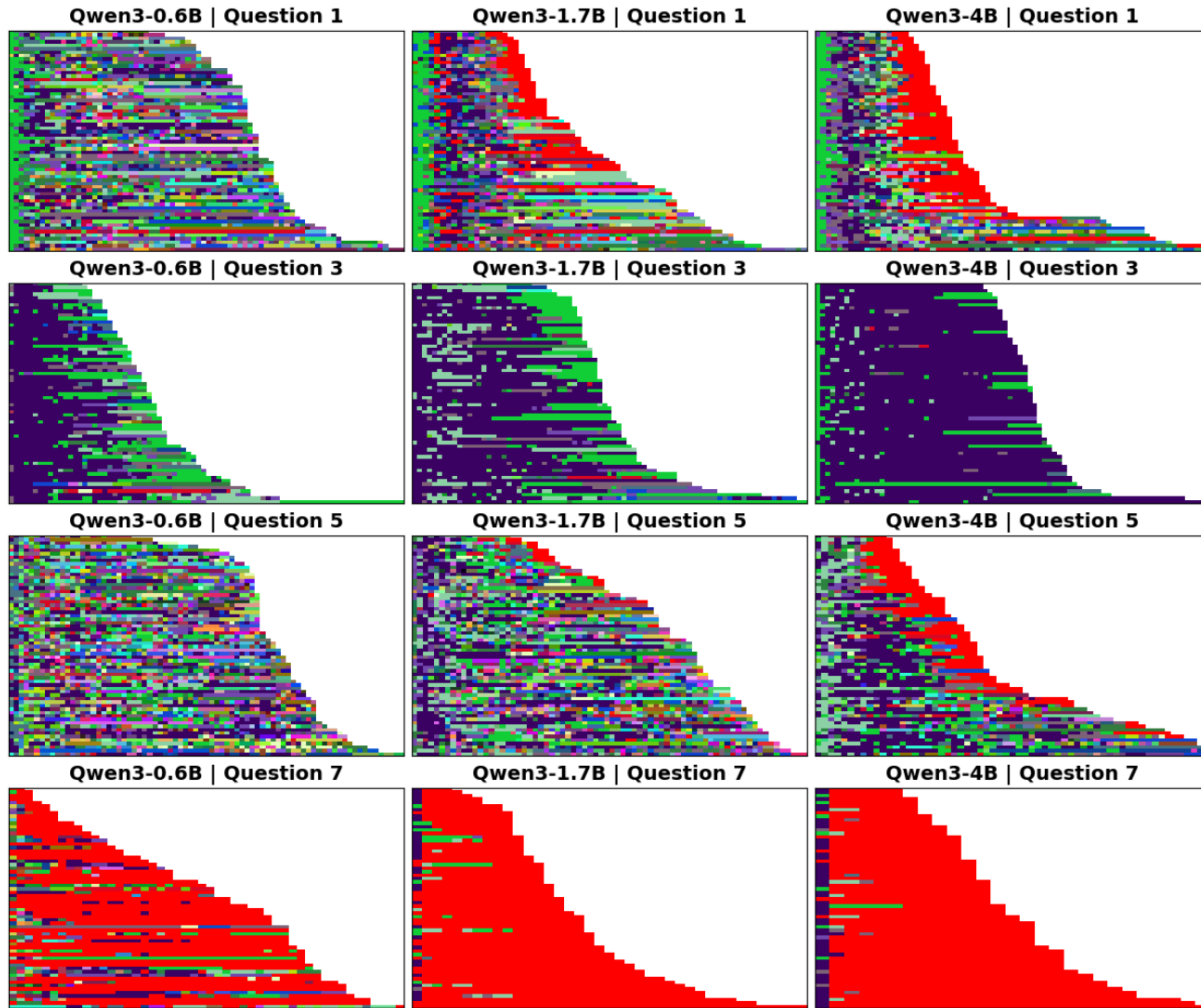
Parallel-Probe: 2D View

By extracting **internal signals** during the reasoning process, we can implement dynamic pruning within a parallel framework.



Probing: We can probe the reasoning process of a 50k-token CoT by interleaving <answer> tags every 1,000 tokens, allowing for a granular inspection of the model's trajectory.

Parallel-Probe: 2D View

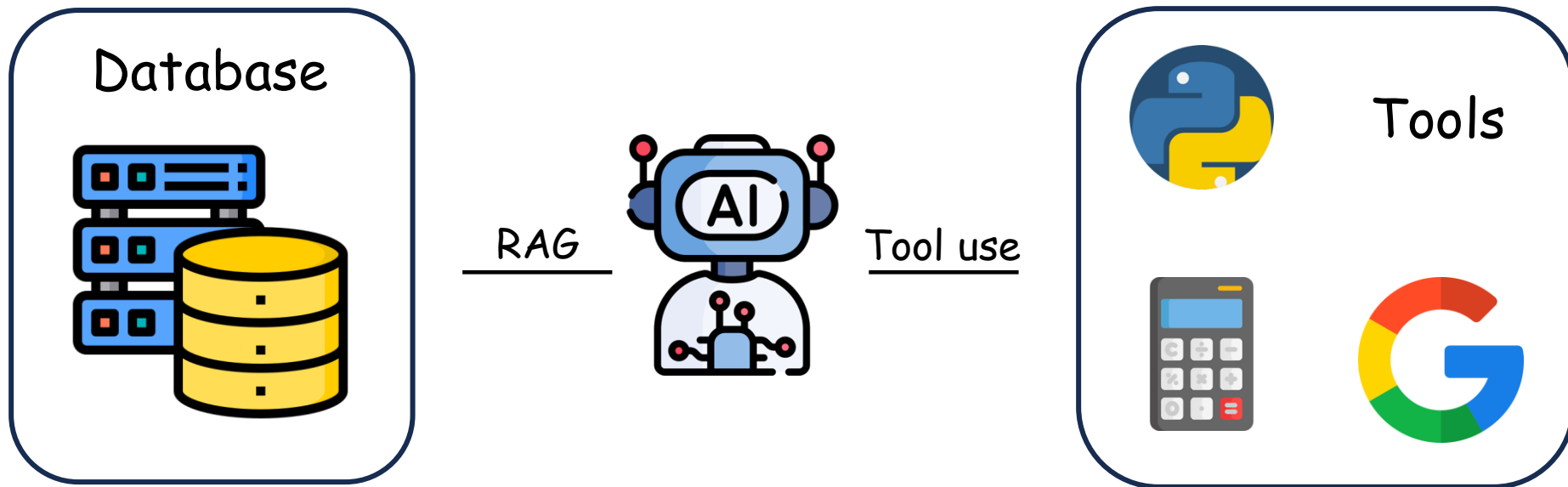


RAG & Tool use

While prompt engineering can improve how we interact with LLMs, prompting alone often falls short when the model needs

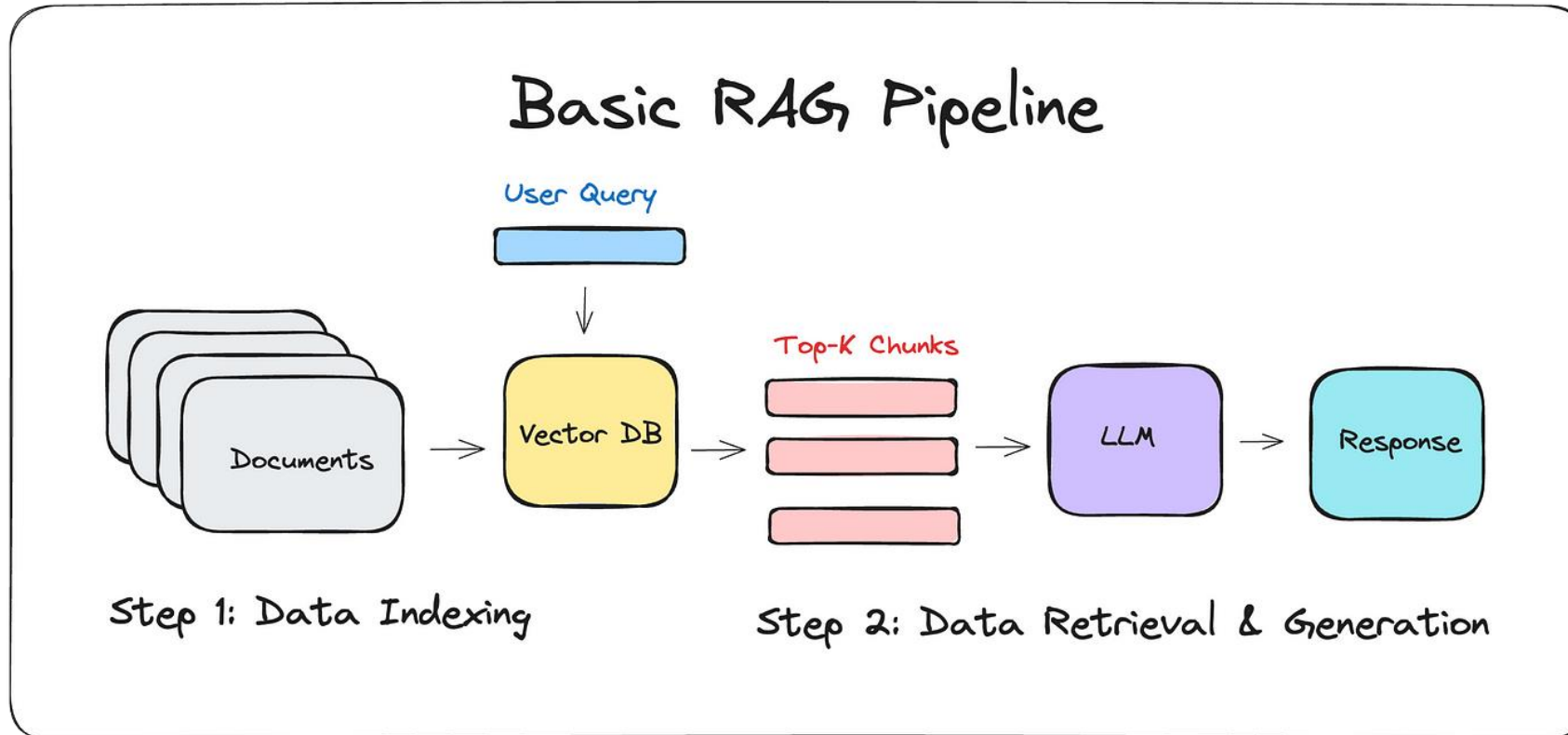
- Up-to-date knowledge, specialized data
- Ability to take real-world actions

To overcome these limitations, techniques like Retrieval-Augmented Generation (RAG) and tool use have been developed, allowing LLMs to access external information and perform tasks that go beyond the scope of simple prompting.



Retrieval-Augmented Generation (RAG)

LLMs showcase impressive capabilities but encounter challenges like **hallucination, outdated knowledge, and non-transparent, untraceable reasoning processes**. RAG has emerged as a promising solution by **incorporating knowledge from external databases**.



1) Indexing. Documents are split into chunks, encoded into vectors, and stored in a vector database.

2) Retrieval. Retrieve the Top k chunks most relevant to the question based on semantic similarity.

3) Generation. Input the original question and the retrieved chunks together into LLM to generate the final answer.

Tool use

Tool-augmented LLMs address the limitations of standalone models—such as inability to perform **real-time computations**, access **up-to-date data**—by leveraging APIs, web search, and software tools to dynamically retrieve information and execute complex tasks beyond their internal knowledge.






Category	Example Tools
 Knowledge access	<code>sql_executor(query: str) -> answer: any</code> <code>search_engine(query: str) -> document: str</code> <code>retriever(query: str) -> document: str</code>
 Computation activities	<code>calculator(formula: str) -> value: int float</code> <code>python_interpreter(program: str) -> result: any</code> <code>worksheet.insert_row(row: list, index: int) -> None</code>
 Interaction w/ the world	<code>get_weather(city_name: str) -> weather: str</code> <code>get_location(ip: str) -> location: str</code> <code>calendar.fetch_events(date: str) -> events: list</code> <code>email.verify(address: str) -> result: bool</code>
 Non-textual modalities	<code>cat_image.delete(image_id: str) -> None</code> <code>spotify.play_music(name: str) -> None</code> <code>visual_qa(query: str, image: Image) -> answer: str</code>
 Special-skilled LLMs	<code>QA(question: str) -> answer: str</code> <code>translation(text: str, language: str) -> text: str</code>

Table 1: Exemplar tools for each category.

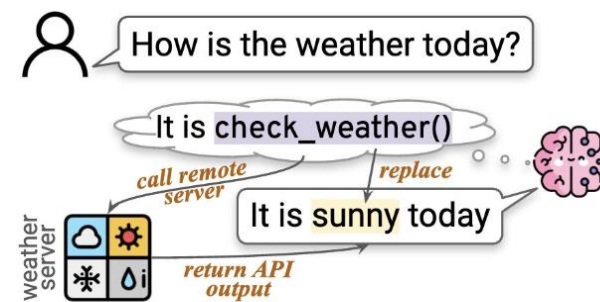
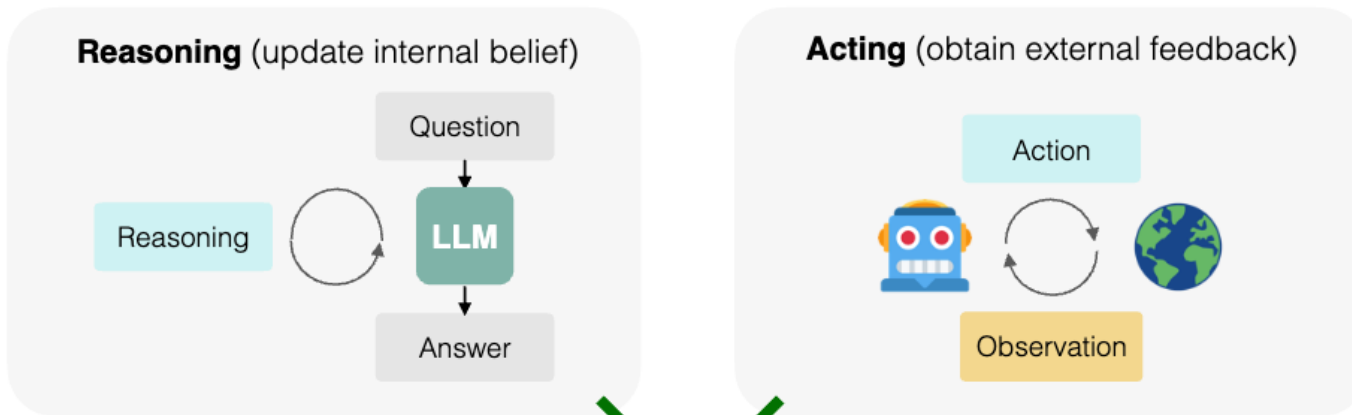


Figure 2: The basic tool use paradigm. LM calls `check_weather` tool by generating text tokens. This call triggers the server to execute the call and return the output `sunny`, using which the LM replaces the API call tokens in the response to the user.

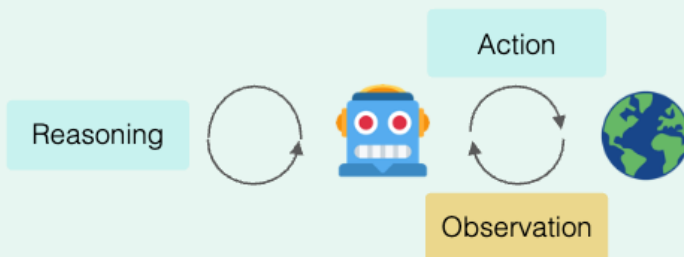
Put everything together – LLM agents

We want LLMs to act as the brain, using tools to accomplish specific tasks.

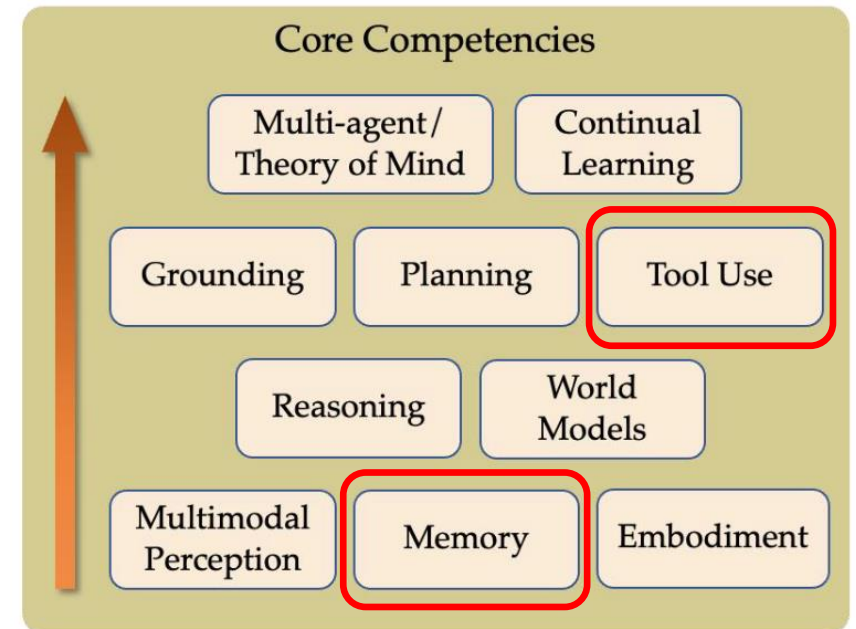
They can **think** ahead, remember past conversations, and use different **tools** to adjust their responses based on the situation and style needed.



ReAct: a new paradigm of agents that **reason and act**



- **Synergy** of reasoning and acting
- **Simple** and intuitive to use
- **General** across domains



LLM agents Examples - STATEval

Every morning Aya goes for a $9s$ -kilometer-long walk and stops at a coffee shop afterwards. When she walks at a constant speed of s kilometers per hour, the walk takes her 4 hours, including t minutes spent in the coffee shop. When she walks $s+2$ kilometers per hour, the walk takes her 2 hours and 24 minutes, including t minutes spent in the coffee shop. Suppose Aya walks at $s+\frac{1}{2}$ kilometers per hour. Find the number of minutes the walk takes her, including the t minutes spent in the coffee shop.

- Large language models excel at mathematical reasoning, and a wide range of math-specific benchmarks have been developed to evaluate this ability.
- How can we curate knowledge from the statistical literature, construct benchmarks, and validate whether LLMs can handle statistical problems?

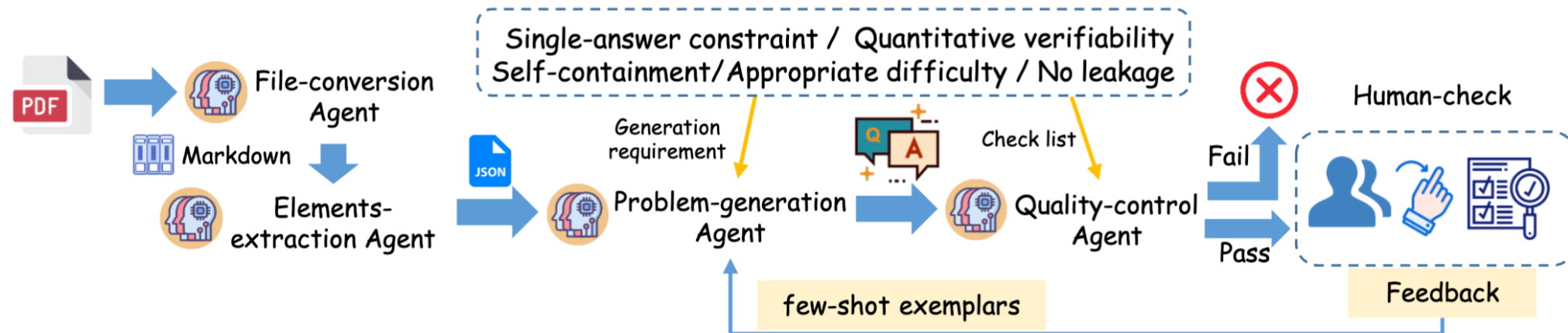
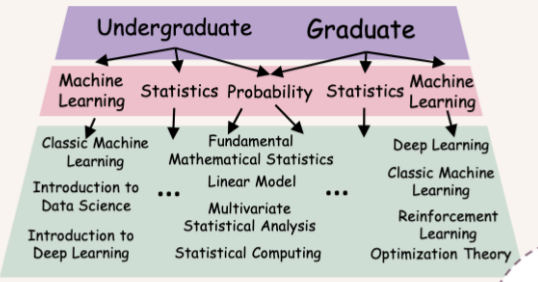


Figure 4: Overview of the StatEval data processing pipeline. Each agent corresponds to a major functional stage in the automated extraction and verification process.

Foundational Knowledge Dataset



Example: Calculation

Question
 In a batch of resistors, 10% are defective. If 10 resistors are randomly selected, what is the probability that exactly 2 are defective?
 (1) Using the binomial distribution;
 (2) Using the Poisson distribution as an approximation to the binomial distribution.

Output(gpt-5-mini) Given $n = 10, p = 0.1$, we want $P(X = 2)$.
 (1) Binomial distribution: $P(X = 2) = \binom{10}{2} p^2 (1 - p)^{10-2} \approx 0.1937$
 (2) Poisson approximation ($\lambda = np = 10 \times 0.1 = 1$):
 $P(X = 2) \approx e^{-\lambda} \lambda^2 / 2! = (e^{-1}) / 2 \approx 0.1839$

Evaluation Thought 1 Process 1 Final Answer 1
 Final Score $1 \times 0.4 + 1 \times 0.3 + 1 \times 0.3 = 1$



Example: Asymptotic Properties

Question
 Consider i.i.d. observations for $i = 1, \dots, N$, where for each individual i we observe covariates $x_i \in \mathbb{R}^p$ (with x_i taking values in a compact set \mathcal{X}) and, for $j = 1, \dots, J$ and $t = 1, \dots, T$, outcomes y_{ijt} that (when observed) follow an exponential-family distribution with density/mass
 $f(y_{ijt} | y_{jt}, a_j, \theta_j, \beta_j, x_i, \phi_j) = \exp(\Phi_j^{-1}\{y_{ijt}, \dots\} + c_j(y_{ijt}, \phi_j))$

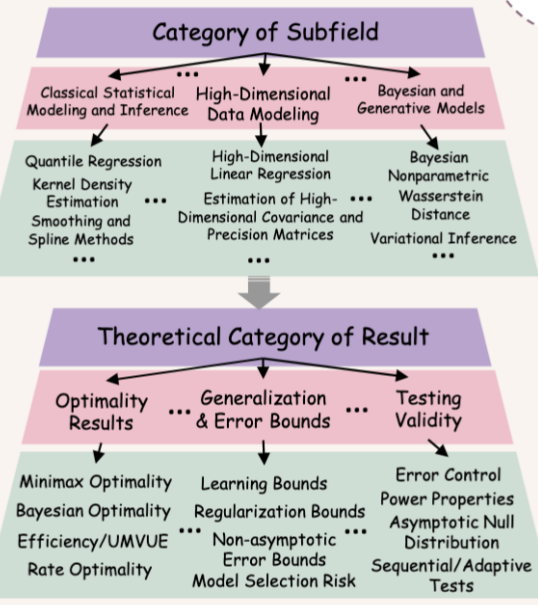
Under the model and the assumptions above, state the asymptotic distribution of the estimator $\hat{\beta}_j$ (the p -vector of regression coefficients for item j) in the large-sample limit $N, J \rightarrow \infty$ (with $N \asymp J$).

Lemmas
Lemma 1: Under the assumptions, the empirical score and Hessian converge uniformly to their population counterparts:
 $\sup_{\Xi \in \mathcal{X}^{\mathcal{E}}} \left\| \frac{1}{N} \nabla_{\alpha_j} l(\Xi) - \mathbb{E} \left[\nabla_{\alpha_j} l(\Xi) \right] \right\| = o_p(1), \sup_{\Xi \in \mathcal{X}^{\mathcal{E}}} \left\| \frac{1}{N} \nabla_{\alpha_j}^2 l(\Xi) - \mathbb{E} \left[\nabla_{\alpha_j}^2 l(\Xi) \right] \right\| = o_p(1)$
 The population Hessian limit equals the nonsingular matrix Φ_j
Lemma 2: The constrained maximizer $\hat{\alpha}$ is consistent with the following rates:
 $\|\hat{\beta}_j - \beta_j^*\| = o_p(1), \frac{1}{\sqrt{J}} \|\hat{\beta}^T - B^*{}^T\|_F = O_p(\min\{\sqrt{N}, \sqrt{J}\}^{-1})$
Lemma 3: The estimator admits the expansion:
 $\sqrt{N}(\hat{\beta}_j - \beta_j^*) = -(\Phi_j^{-1})^{(T+1):(T+p)} \cdot \frac{1}{\sqrt{N}} \sum_{i=1}^N \sum_{t=1}^T r_{it} \phi_j^{-1} b_j'(\cdot) e_{it, (T+1):(T+p)}^* + o_p(1)$

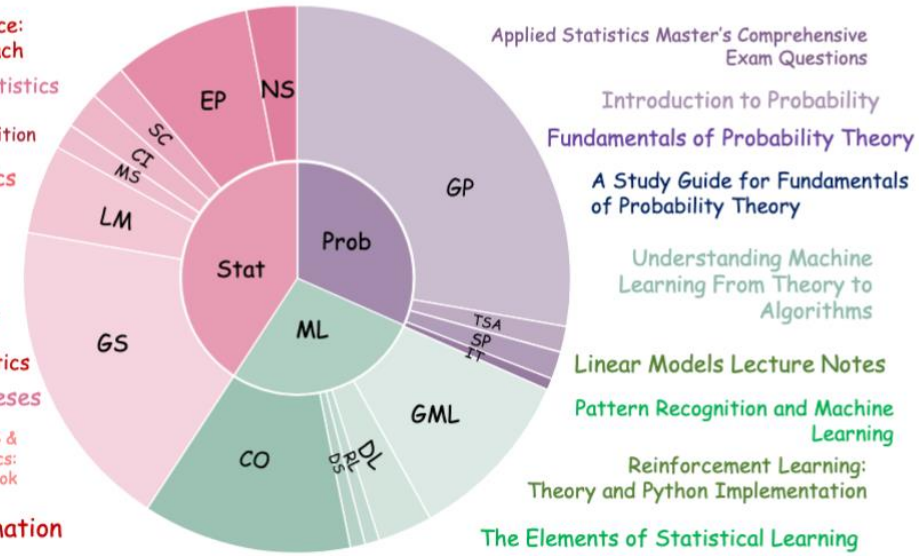
Output(gpt-5-mini) $\sqrt{N}(\hat{\beta}_j - \beta_j^*) \stackrel{d}{\rightarrow} N(0, \Sigma_{\epsilon, j})$ as $N, J \rightarrow \infty$ with $N \asymp J$
 $\Sigma_{\epsilon, j} = -(\Phi_j^{-1})^{(T+1):(T+p), (T+1):(T+p)}$

Answer $\sqrt{N}(\hat{\beta}_j - \beta_j^*) \stackrel{d}{\rightarrow} N(0, (-\Phi_j^{-1})^{(T+1):(T+p), (T+1):(T+p)})$ **CORRECT**

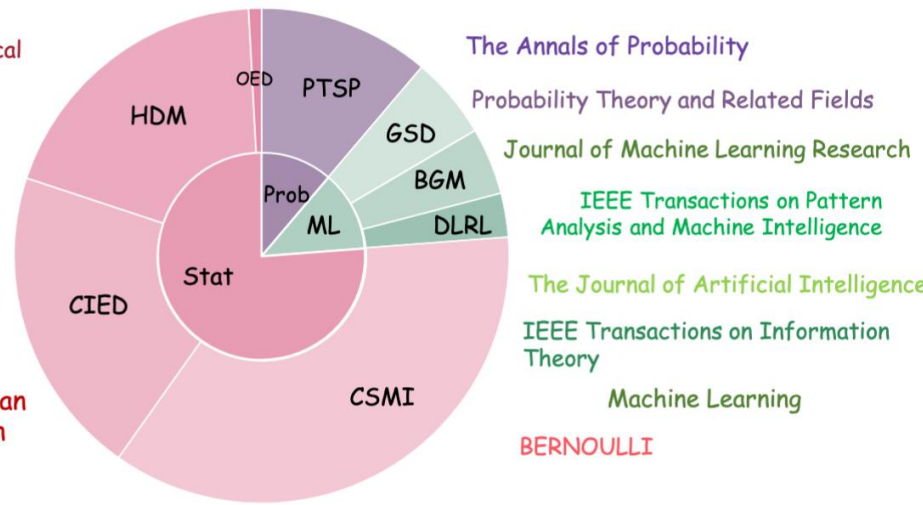
Advanced Statistical Research Dataset



Causal Inference:
 A Statistical Learning Approach
 Mathematical Statistics
 Statistical Inference Second Edition
 Theoretical Statistics Topics for a Core Course
 Mathematical Statistics Second Edition
 Asymptotic Statistics
 Likelihood Methods in Statistics
 Testing Statistical Hypotheses
 Cambridge International AS & A Level Mathematics: Probability & Statistics 1 Practice Book
 Theory of Point Estimation

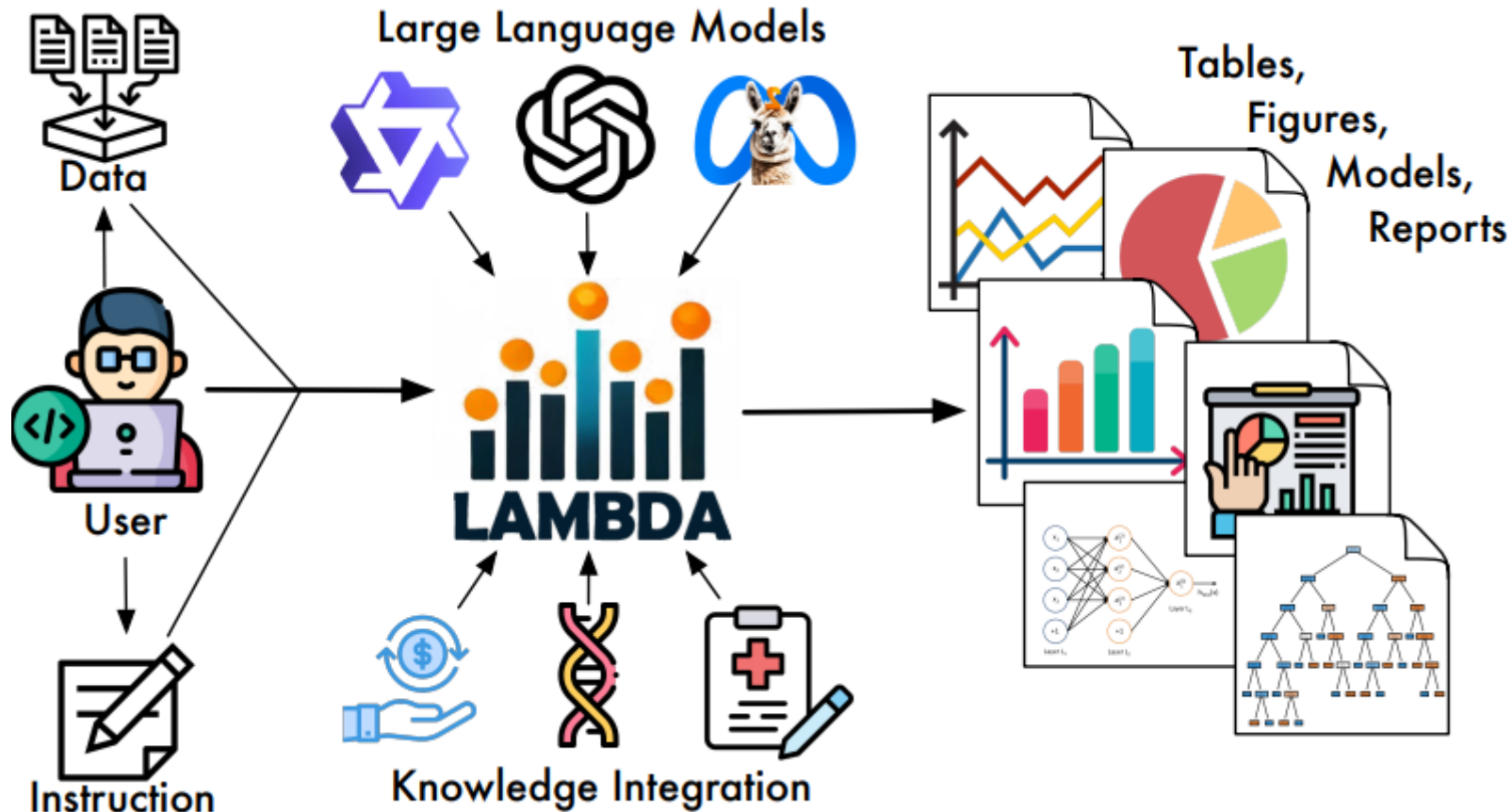


Journal of the Royal Statistical Society Series B
 Statistica Sinica
 BIOMETRICS
 Journal of Computational and Graphical Statistics
 Annals of Statistics
 Biometrika
 Journal of the American Statistical Association



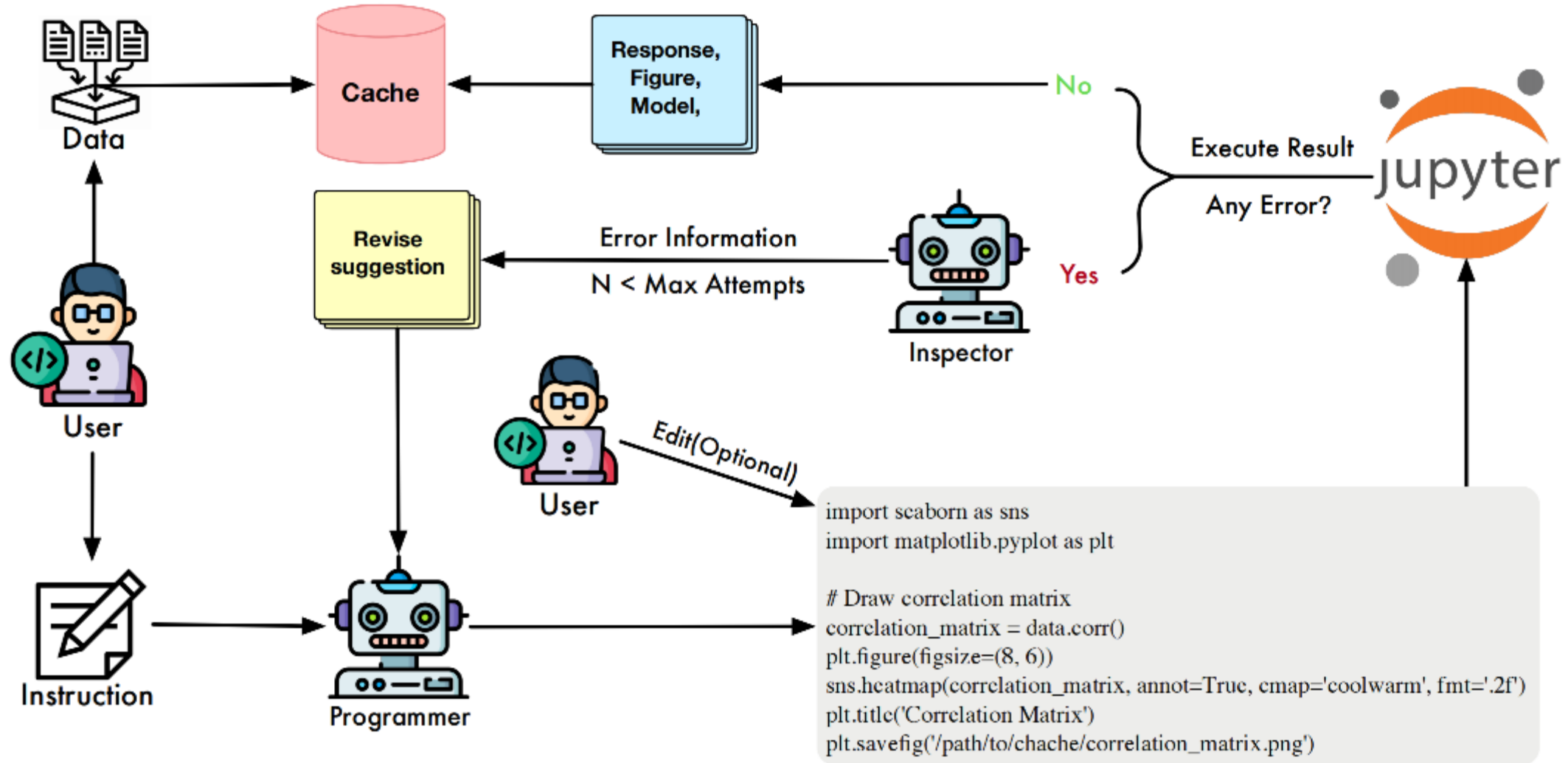
Lu, Yuchen, Run Yang, Yichen Zhang, Shuguang Yu, Runpeng Dai, Ziwei Wang, Jiayi Xiang et al. "Stateval: A comprehensive benchmark for large language models in statistics." arXiv preprint arXiv:2510.09517 (2025).

LLM agents Examples - LAMBDA



Beyond dataset curation, an important question is how LLMs can be used to perform statistical tasks effectively.

LLM agents Examples





Train and select parameters for XGBoost. Print the best parameters and score based on cross-validation.

```
from xgboost import XGBClassifier
from sklearn.model_selection import GridSearchCV
param_grid = {
    'max_depth': [3, 5, 7],
    'learning_rate': [0.1, 0.5, 1.0],
    .....
}
grid_search = GridSearchCV(XGBClassifier(), param_grid, cv=5, scoring='accuracy')
.....
Execution result:
Best Parameters: {'learning_rate': 0.1, 'max_depth': 3, 'n_estimators': 100}
Best Score: 0.9553968253968254
.....
best\_xgb\_model.pkl
```

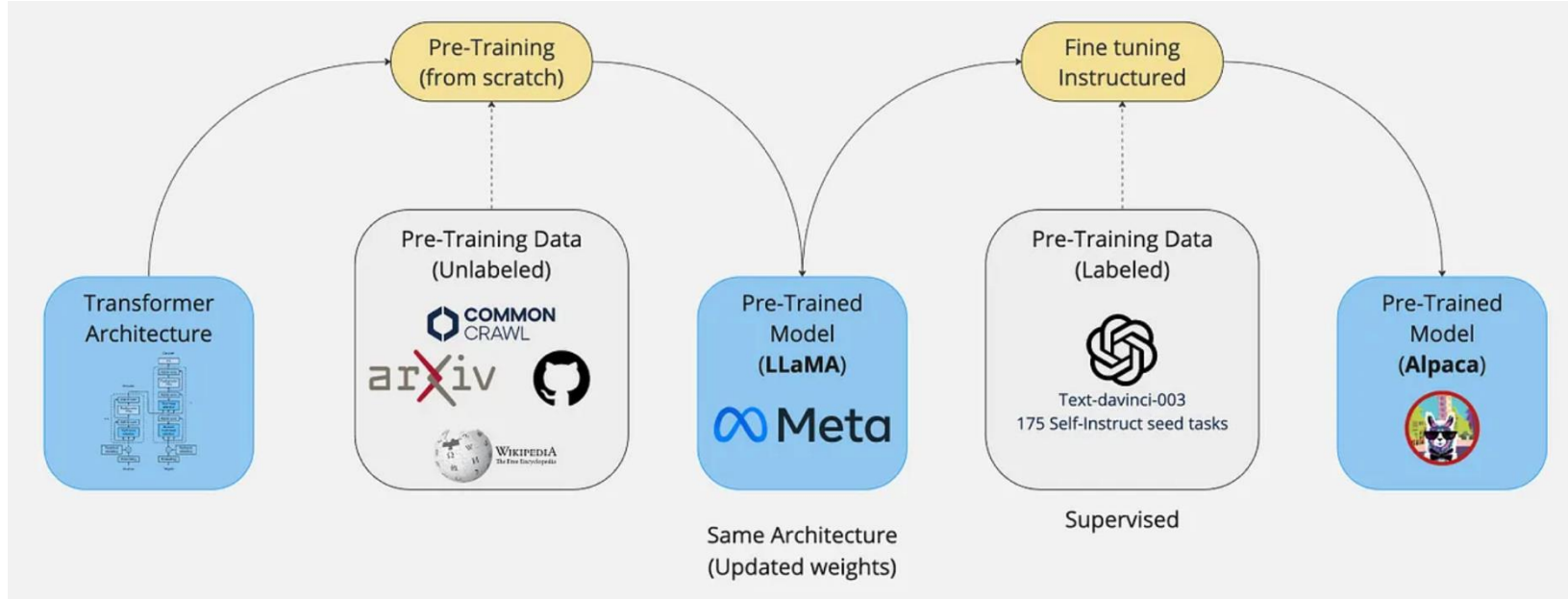


Report



Figure 9: An example of using LAMBDA for classification analysis with the Wine dataset.

Training methods – SFT



Prompting methods are inherently limited by the capabilities and biases of the base model.

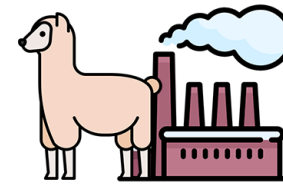
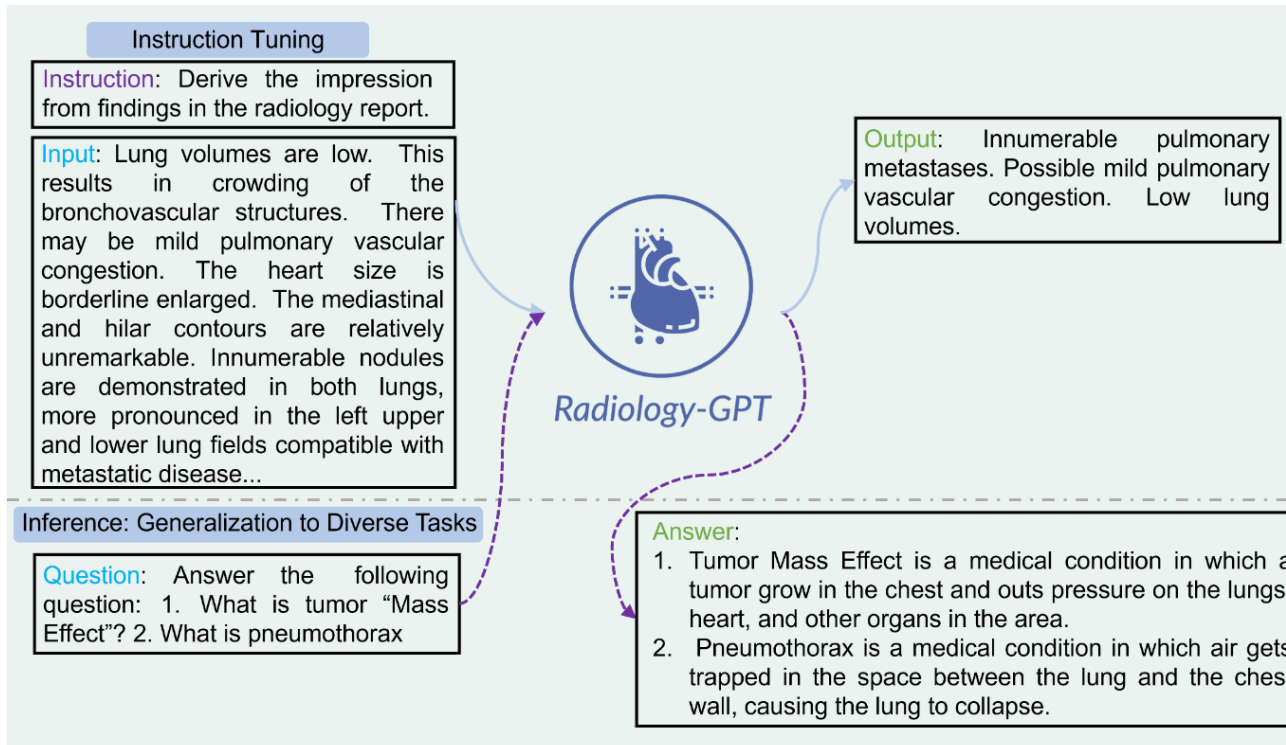
Supervised Fine-Tuning (SFT) is the most basic training approach for aligning large language models with human intentions. It involves fine-tuning a pre-trained model on a dataset of input-output pairs **to follow instructions or perform specific tasks.**

System_Prompt + <User>: [User_Input] +<System>: [Response]</s>

Loss

SFT – Example

Among all the training methods, SFT is the most direct one. There have been many works use SFT at the early stage.



LLaMA-Factory
Easy and Efficient LLM Fine-Tuning

Easily fine-tune 100+ large language models with Web UI

Specialize a large language model (LLM) for generating or interpreting the “Impression” section of radiology reports

Alignment

RLHF allows models to learn from *comparative feedback*—for example, choosing one response over another—so that the model can generalize better to unseen situations and generate outputs that are more aligned with human preferences.

3H principles:

- **Helpful** (Improves usability)
- **Honest** (Enhances trustworthiness)
- **Harmless** (Mitigates harmful outputs)



Alignment

Challenges

1. **Human in the loop is expensive** – *Reward Modelling*
2. **Human judgments are noisy and miscalibrated.** – *Preference learning*

Score the helpfulness of the following response, 1-10

What are the steps for making a simple cake?

1. Preheat oven to 350°F (175°C).
2. Grease and flour a cake pan.
3. In a bowl, combine 2 cups flour, 1.5 cups sugar, 3.5 tsp baking powder, and a pinch of salt.
4. Add 1/2 cup butter, 1 cup milk, and 2 tsp vanilla; mix well.
5. Beat in 3 eggs, one at a time.
6. Pour batter into the pan.
7. Bake for 30-35 minutes or until a toothpick comes out clean.
8. Let cool, then frost or serve as desired.

Alignment – Bradley Terry model

Which of these two responses is more helpful?

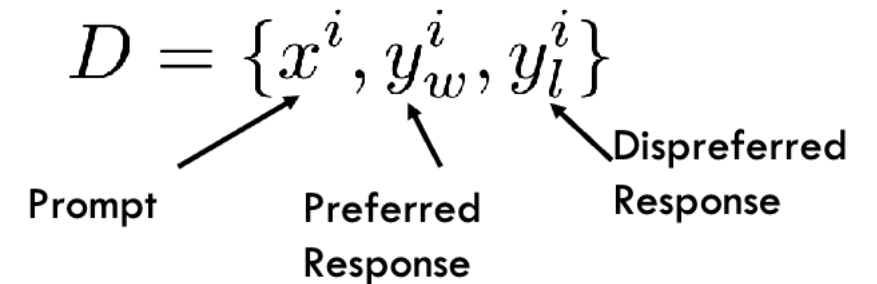
What are the steps for making a simple cake?

1. *Preheat oven to 350°F (175°C).*
2. *Grease and flour a cake pan.*
3. *In a bowl, combine 2 cups flour, 1.5 cups sugar, 3.5 tsp baking powder, and a pinch of salt.*
4. *Add 1/2 cup butter, 1 cup milk, and 2 tsp vanilla; mix well.*
5. *Beat in 3 eggs, one at a time.*
6. *Pour batter into the pan.*
7. *Bake for 30-35 minutes or until a toothpick comes out clean.*
8. *Let cool, then frost or serve as desired.*

What are the steps for making a simple cake?

1. *Warm up the oven.*
2. *Grease a cake pan.*
3. *Blend dry ingredients in a bowl.*
4. *Incorporate butter, milk, and vanilla.*
5. *Mix in the eggs.*
6. *Pour into the prepared pan.*
7. *Bake until golden brown.*
8. *Add frosting if desired.*

Instead of asking labelers to assign scores, we collect their preferences through pairwise comparisons.



Alignment - RLHF

How do we get feedback for the reward while training our RL model?

$$p(y_w > y_l | x) = \sigma(\underline{r(x, y_w)} - \underline{r(x, y_l)})$$

Logistic function;
which is equivalent
to using softmax:

$$\frac{1}{1 + e^{-x}}$$

$$p(y_w > y_l | x) = \frac{\exp(r(x, y_w))}{\exp(r(x, y_w)) + \exp(r(x, y_l))}$$

Train a Reward Model (RM) on preference data to predict preferences!

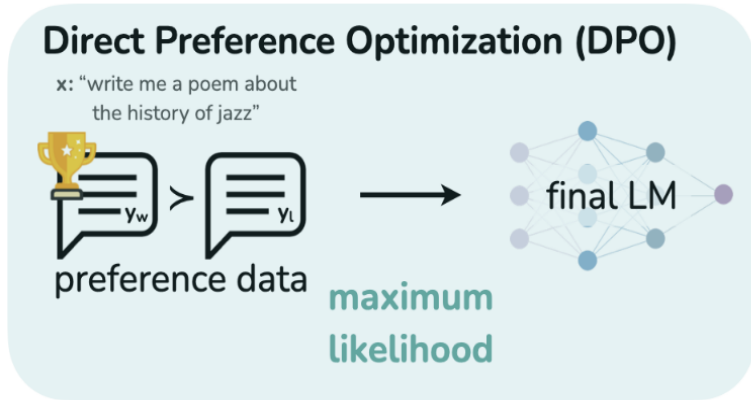
$$\mathcal{L}_R(\phi, D) = -\mathbb{E}_{(x, y_w, y_l) \sim D} [\log \sigma(r(x, y_w) - r(x, y_l))]$$

Use RL algorithm to train optimize the LLM π

$$\max_{\pi_\theta} \mathbb{E}_{x \sim D, y \sim \pi_\theta(y|x)} [\underline{r_\phi(x, y)}]$$

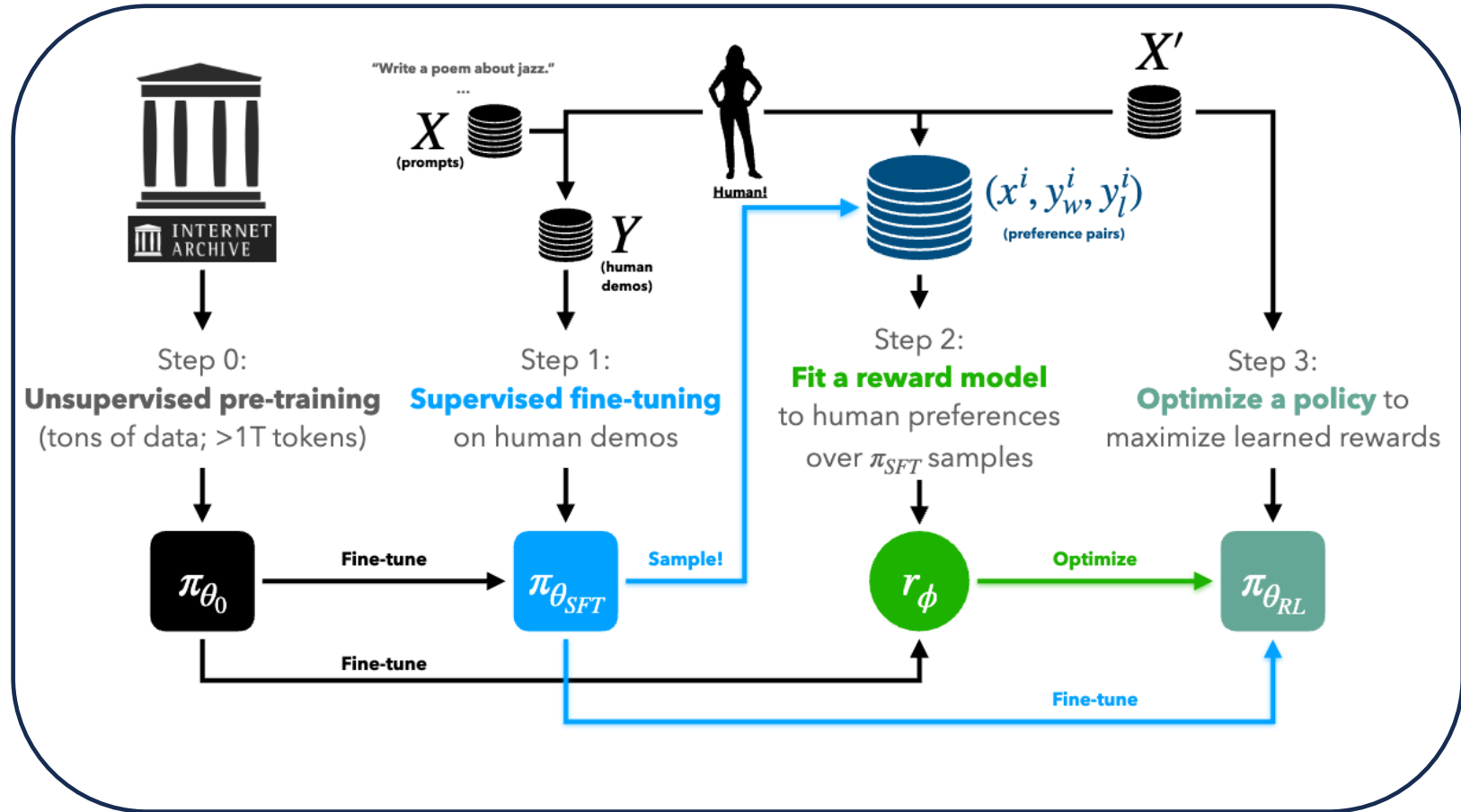
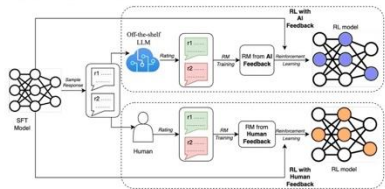
Alignment - Improvements

- The reward model is hard to train and large.

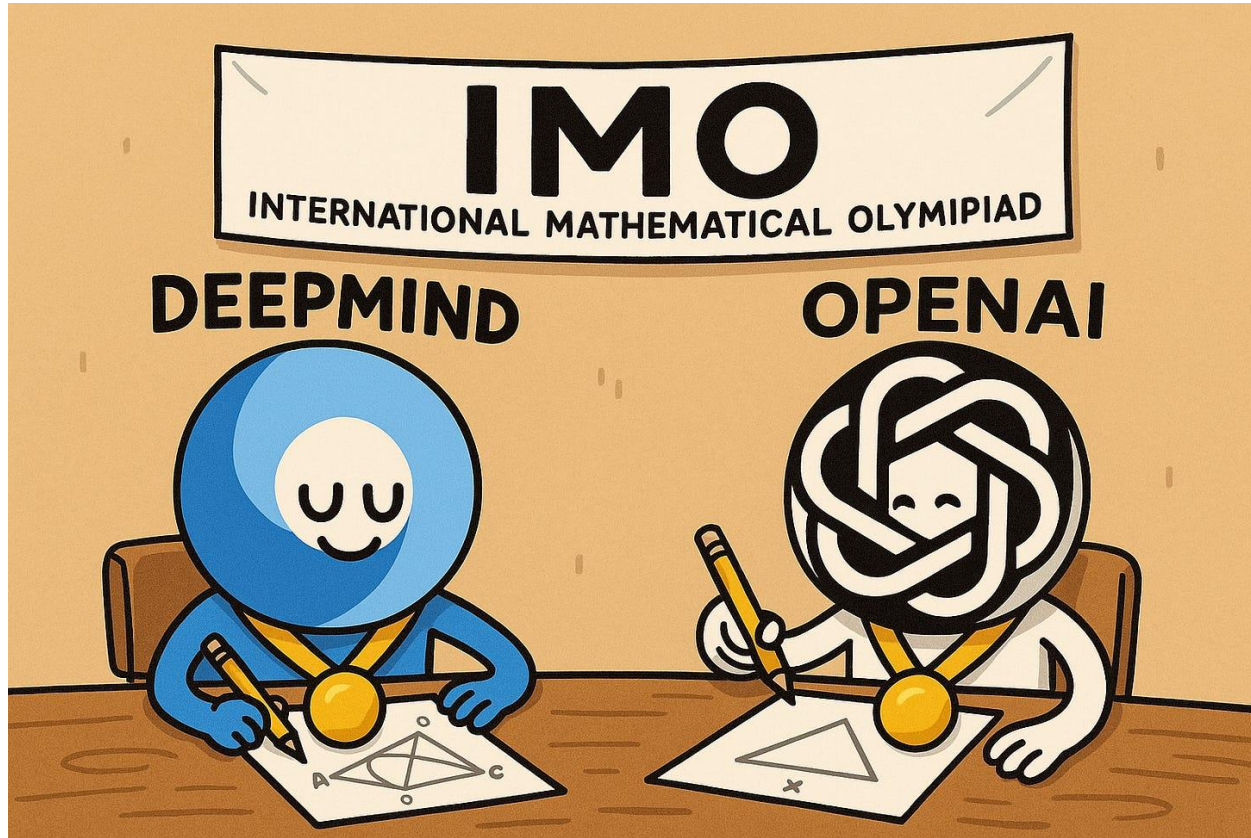


- Human labeling remains costly.

RLAIF - RL from AI Feedback for LLM



Learning to Reasoning with LLMs



DeepMind and OpenAI achieve IMO Gold.

The secret behind the success - Reasoning

*We achieved this year's result using an advanced version of Gemini Deep Think – an enhanced **reasoning mode** for complex problems that incorporates some of our latest research techniques, including parallel thinking. This setup enables the model to simultaneously **explore and combine multiple possible solutions** before giving a final answer, rather than pursuing a single, linear chain of thought.*



DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning

Learning to Reasoning with LLMs

What is reasoning?

Reasoning refers to the process of **thinking through** a problem or situation in order to form a logical conclusion or make a decision. It involves using evidence, facts, and logic to arrive at a solution or answer.

Why Reasoning is important?

Even for humans, problems like **math** or **complex question answering** are difficult to solve with direct answers alone, as they often require multi-step reasoning or integrating diverse pieces of information.

How can LLM Reasoning?

Generate sequence of tokens representing **intermediate steps** in the reasoning process.

Problem 1:

Question: Two trains running in opposite directions cross a man standing on the platform in 27 seconds and 17 seconds respectively and they cross each other in 23 seconds. The ratio of their speeds is:

Options: A) $3/7$ B) $3/2$ C) $3/88$ D) $3/8$ E) $2/2$

Rationale: Let the speeds of the two trains be x m/sec and y m/sec respectively. Then, length of the first train = $27x$ meters, and length of the second train = $17y$ meters. $(27x + 17y) / (x + y) = 23 \rightarrow 27x + 17y = 23x + 23y \rightarrow 4x = 6y \rightarrow x/y = 3/2$.

Correct Option: B

Learning to Reasoning with LLMs

How can we guide LLMs to perform reasoning?

Standard 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?

Exemplar

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. ❌

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? **Thought**

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. ✅

Or some other prompting methods:

1. Least-to-Most Prompting
2. Self reflection/ verification

Can we go beyond prompting?

"We have found that the performance of o1 consistently improves with more reinforcement learning (train-time compute) and with more time spent thinking (test-time compute). The constraints on scaling this approach differ substantially from those of LLM pretraining, and we are continuing to investigate them." — OpenAI o1 release blog

Learning to Reasoning with LLMs

Reward guided decoding

ORM

Question:

Three pencils and a jumbo eraser cost \$1.24. Five pencils and a jumbo eraser cost \$1.82. No prices include tax. In cents, what is the cost of a pencil?

Answer:

Let the cost of a pencil be p cents, and the cost of a jumbo eraser be e cents.

From the problem, we are given two equations based on the cost:

$$3p + e = 124$$

$$5p + e = 182$$

To solve this system of equations, we subtract the first equation from the second to eliminate e :

$$(5p + e) - (3p + e) = 182 - 124$$

$$\text{Solving for } p = 58/2 = 27$$

Thus, the cost of a pencil is 29 cents.



PRM

Question:

Three pencils and a jumbo eraser cost \$1.24. Five pencils and a jumbo eraser cost \$1.82. No prices include tax. In cents, what is the cost of a pencil?

Answer:

1 Let the cost of a pencil be p cents, and the cost of a jumbo eraser be e cents. ✓

2 From the problem, we are given two equations based on the cost:
 $3p + e = 124$, $5p + e = 182$ ✓

3 To solve this system of equations, we subtract the first equation from the second to eliminate e : $(5p + e) - (3p + e) = 182 - 124$ ✓

4 Solving for $p = 58/2 = 27$ ✗

5 Thus, the cost of a pencil is 29 cents. ✗



DeepseekR1 and RLVR

“One of the most remarkable aspects of this self-evolution is the emergence of sophisticated behaviors as the test-time computation increases. Behaviors such as reflection—where the model revisits and reevaluates its previous steps—and the exploration of alternative approaches to problem-solving arise spontaneously. These behaviors are not explicitly programmed but instead emerge as a result of the model’s interaction with the reinforcement learning environment.” — DeepSeek-R1 ‘Aha moment’



Reinforcement learning with verifiable reward

$$R(\hat{y}, y) = \begin{cases} 1, & \text{is_equivalent}(\hat{y}, y) \\ -1, & \text{otherwise} \end{cases}$$

Instead of training a separate **reward model** or explicitly assigning credit to **each reasoning step**, we allow the model to discover effective reasoning strategies through trial and error, guided by verifiable rewards.

1. Reduce the computational burden of training a separate reward model.
2. Eliminate the risk of reward.

DeepseekR1 and RLVR

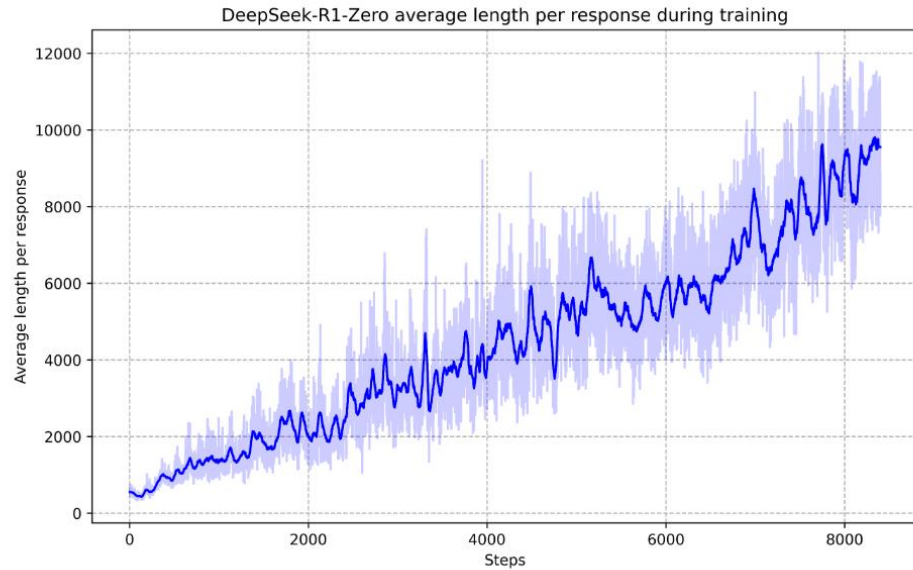


Figure 3 | The average response length of DeepSeek-R1-Zero on the training set during the RL process. DeepSeek-R1-Zero naturally learns to solve reasoning tasks with more thinking time.

During the training process, they observe the increase in reasoning length and “aha moment” where the model learns to self-reflection and verify.

Question: If $a > 1$, then the sum of the real solutions of $\sqrt{a - \sqrt{a+x}} = x$ is equal to

Response: <think>

To solve the equation $\sqrt{a - \sqrt{a+x}} = x$, let's start by squaring both ...

$$(\sqrt{a - \sqrt{a+x}})^2 = x^2 \implies a - \sqrt{a+x} = x^2.$$

Rearrange to isolate the inner square root term:

$$(a - x^2)^2 = a+x \implies a^2 - 2ax^2 + (x^2)^2 = a+x \implies x^4 - 2ax^2 - x + (a^2 - a) = 0$$

...

Wait, wait. Wait. That's an aha moment I can flag here.

Let's reevaluate this step-by-step to identify if the correct sum can be ...

We started with the equation:

$$\sqrt{a - \sqrt{a+x}} = x$$

First, let's square both sides:

$$a - \sqrt{a+x} = x^2 \implies \sqrt{a+x} = a - x^2$$

Next, I could square both sides again, treating the equation: ...

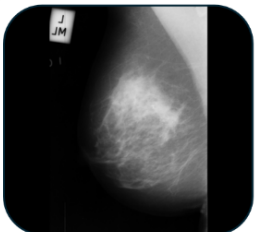
...

Table 3 | An interesting “aha moment” of an intermediate version of DeepSeek-R1-Zero. The model learns to rethink using an anthropomorphic tone. This is also an aha moment for us, allowing us to witness the power and beauty of reinforcement learning.

Some of the Challenges

- **Ensuring Faithfulness of the Reasoning Process**
- **Reducing Reasoning Latency**
- **Handling General Reasoning Tasks with Unverifiable Rewards**

RLVR – More Applications



Problem: What does this image depict in terms of its content?

A)Brain tissue
B)Intestinal tissue
C)Kidney tissue
D)Breast tissue

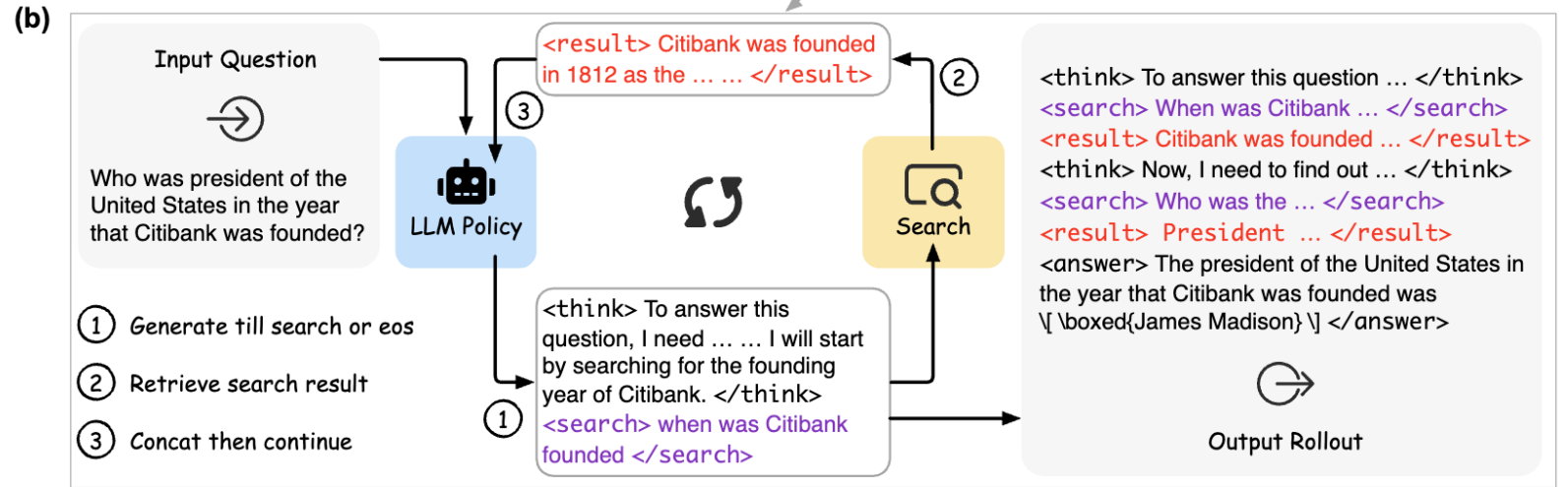
Med-R1 Response:

Think: The image appears to be a mammogram, which is a type of X-ray used to detect abnormalities in the breast tissue. The presence of dense, irregular patterns and a central area of lighter density suggest the presence of breast tissue.

Answer: D

Groundtruth: D

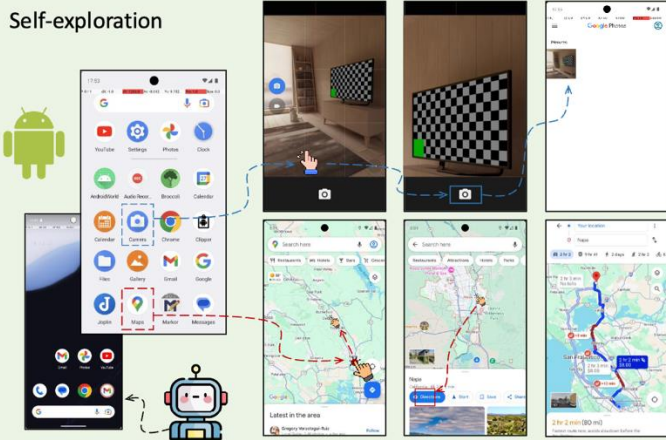
Medical Reasoning



Use of Search engine

Task creation

Self-exploration



Generated task

(1) Open the phone camera and take a **video**.
(2) Open the camera and **enlarge the TV**, then **take a photo**.
(3) Open the camera and **go to Google photo**.

(1) Open **Google Maps** and search for the **route to Napa**.
(2) Open the map and find the **fastest route from your location to Napa**.
(3) Use map to **search your current location**.

Use of smart phone

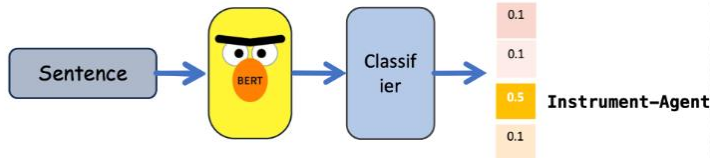
RLVR – Example R1-RE

Sentence Olanzapine was also associated with more frequent reports of weight gain and significantly greater VA costs ...

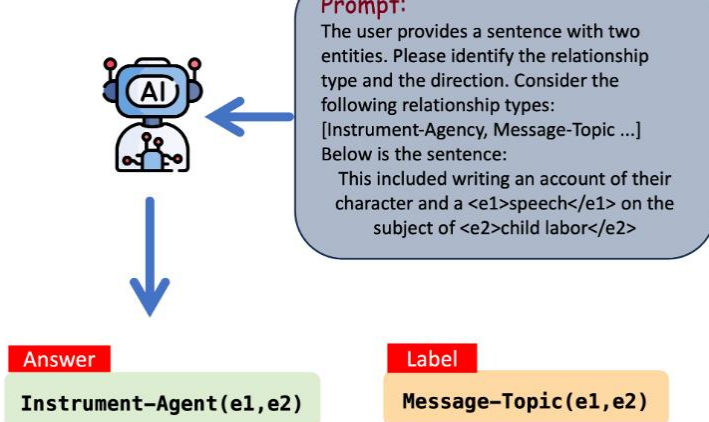
Label [[Olanzapine:drug, risk-factor-of, weight gain:symptom]]

Relation extraction is a critical step in knowledge graph construction.

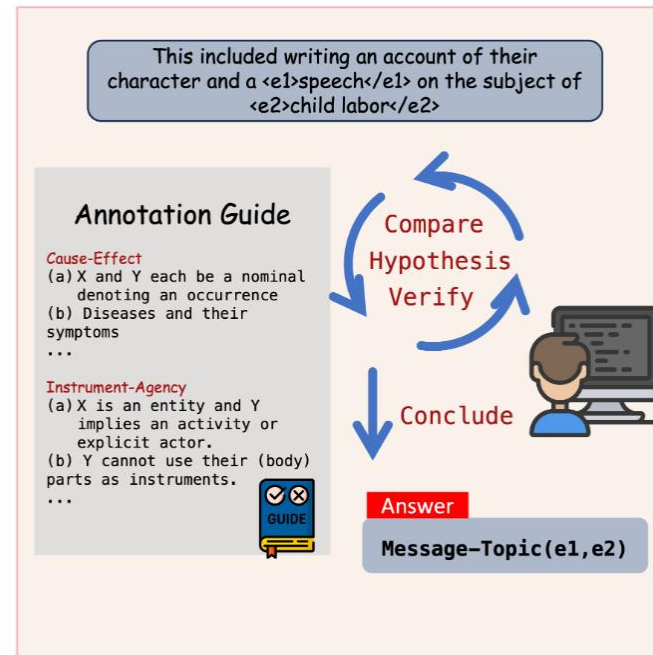
Pre-LLM classification



SFT Process



Human Annotation Process



Existing method focus on learning a mapping from sentence to triplet, which cannot generalize to OOD.

Human annotators are generalizable across tasks because they can reason on the requirements.

RLVR – Example R1-RE

R1-RE-7B

We need to determine the relationship between <e1>schizophrenia</e1> and <e2>prolactin-increasing antipsychotics</e2>. Let's break this down step by step.

1. Identify the entities:

- <e1>schizophrenia</e1> is the disease or condition.
- <e2>prolactin-increasing antipsychotics</e2> is a type of medication or treatment.

Breakdown the sentence

2. Analyze the relationship:

- The sentence states that schizophrenia is associated with a specific type of antipsychotics (prolactin-increasing antipsychotics) being used in the treatment or management of schizophrenia.

Compare definitions

3. Apply the definitions:

- **causes**: This wouldn't apply because the sentence suggests the antipsychotics are being used to treat, not cause, schizophrenia.

Hypothesis-validation

- **risk_factor_of**: This would apply if the antipsychotics increased the risk of schizophrenia, but here it's the other way around; the antipsychotics are being used because of schizophrenia, not the other way.

4. Conclusion:

Draw conclusion

Given that the sentence directly states that antipsychotics are used in the treatment of schizophrenia, this fits squarely into the definition of a treatment relationship. The antipsychotics are specifically used to address or manage schizophrenia, which is the essence of the treatment_for relationship.

Final Answer

Treatment_for(e2, e1)



Target sentence



Monitoring prolactinemia and addressing hyperprolactinemia is paramount in women with <e1>schizophrenia</e1> being treated with <e2>prolactin-increasing antipsychotics</e2>.

Qwen-2.5-7B

The sentence mentions that monitoring prolactinemia and addressing hyperprolactinemia is important for women with schizophrenia who are treated with prolactin-increasing antipsychotics. This implies that prolactin-increasing antipsychotics can lead to prolactinemia and hyperprolactinemia, and that these conditions are associated with the treatment.

Given the definitions, this is best classified under the "risk_factor_of" relationship as it indicates that prolactin-increasing antipsychotics (e2) could increase the risk or likelihood of prolactinemia and hyperprolactinemia (e1). This fits into the susceptibility and prevalence category of "risk_factor_of".

Final Answer

Risk_factor_of(e2, e1)

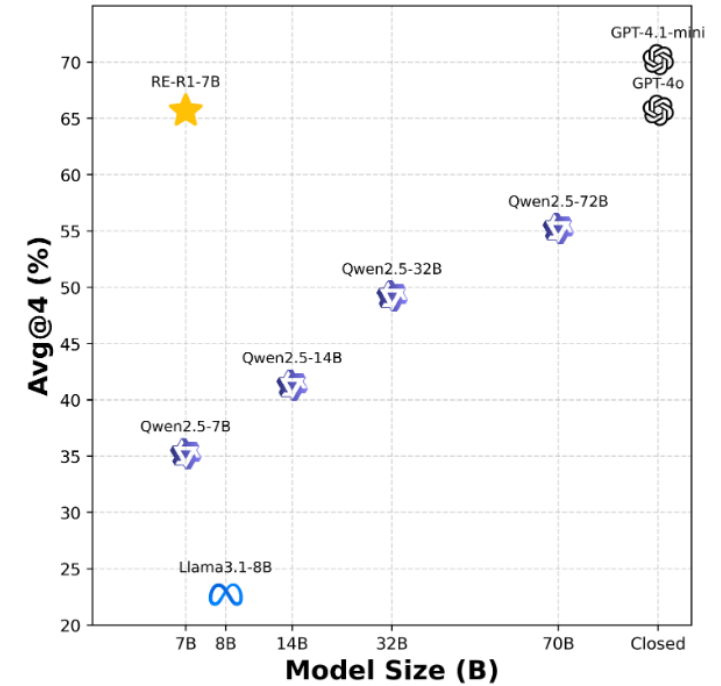
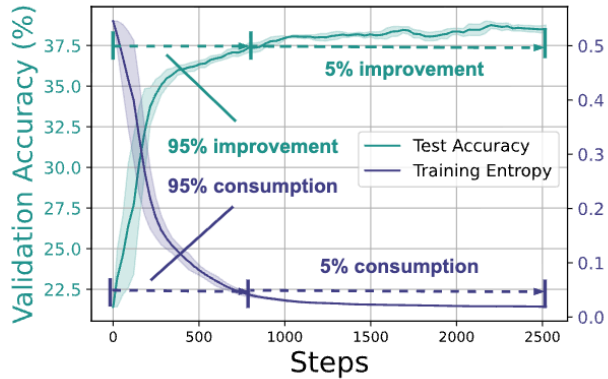


Figure 1: Testing accuracy on the MDKG dataset for R1-RE-7B trained on the Sem-2010 dataset, compared with other models. Detailed results are provided in Table 3.

Figure 6: Case study comparing the chain-of-thought (COT) reasoning of **R1-RE-7B** and Qwen2.5-7B-Instruct. Due to space constraints, some COT outputs are omitted; the complete COT reasoning process for **R1-RE-7B** is provided in Appendix A.2.

RLVR – The Entropy Mechanism

Does RL for LLM just Trade Entropy for Performance ?

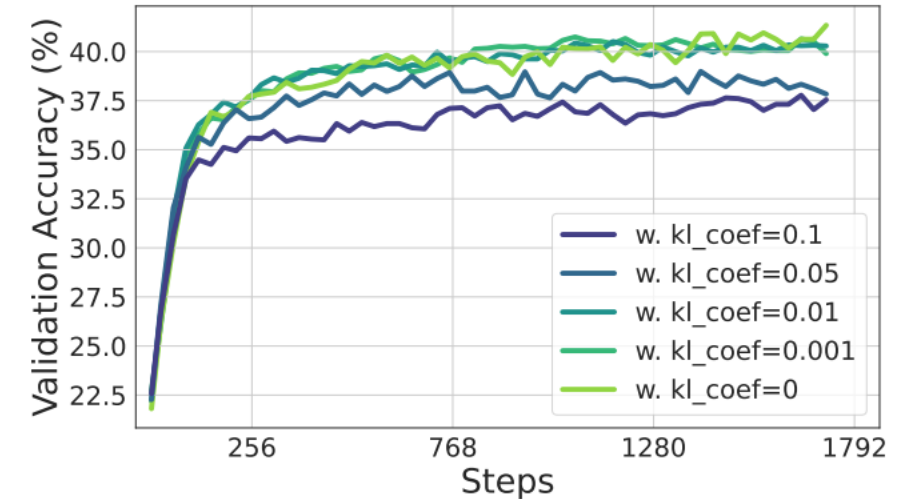
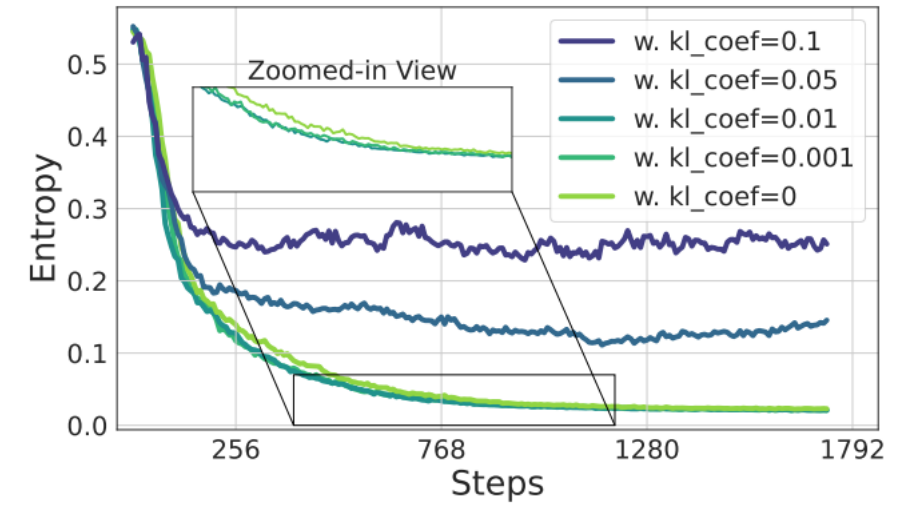
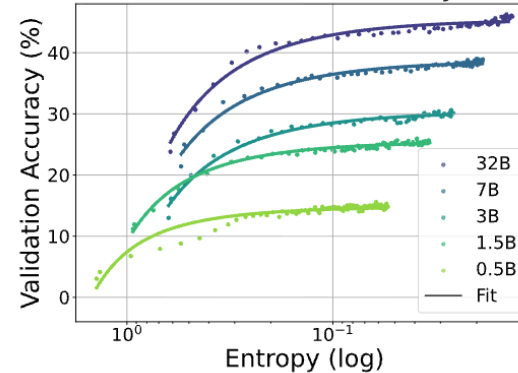


Performance ↓

$$R = -a \exp(\mathcal{H}) + b$$

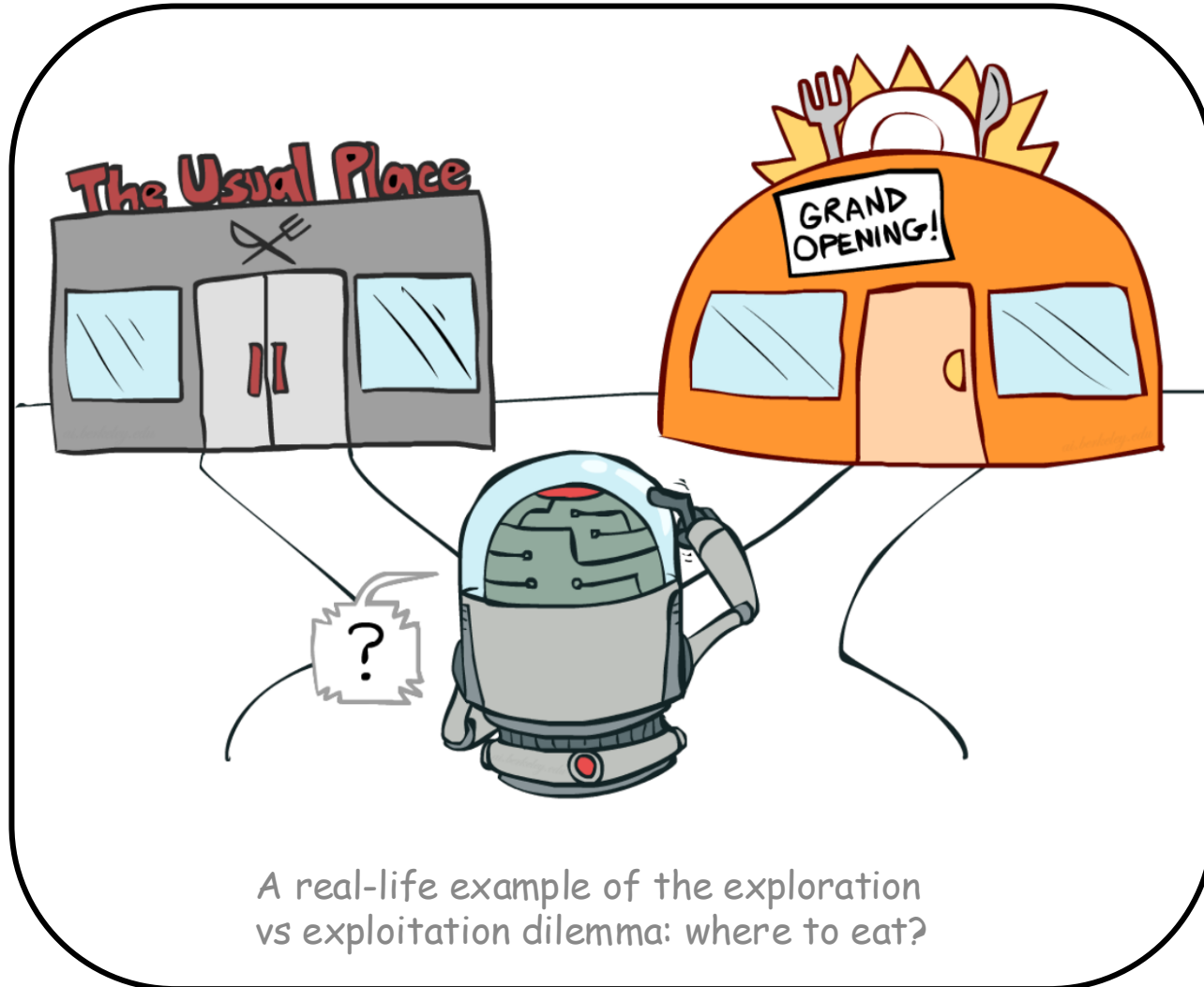
Entropy ↑

Qwen2.5 Model Family



Without intervention, LLM tend exploit trade entropy for accuracy.

RLVR – Exploration and Exploitation



Exploitation: Take advantage of the best option we know.

Exploration: Take some risk to collect information about unknown options.

A fundamental trade-off in RL, whether choosing the best action (token) each step or explore new actions.

RLVR – CDE (Curiosity Driven Exploration)

Curiosity-driven exploration is rooted in intrinsic motivation: just as children’s natural curiosity drives them to explore the unknown, discover, and learn, fostering curiosity in artificial agents can similarly drive meaningful exploration and growth.



In reinforcement learning, we can encourage curiosity by rewarding exploration that signals learning potential. Common measures include:

- State Novelty/Visitation (**Count based**)
- Agent’s Uncertainty. (**Prediction based**)

Optimism In face of Uncertainty: *Try things out—you might find something better!*

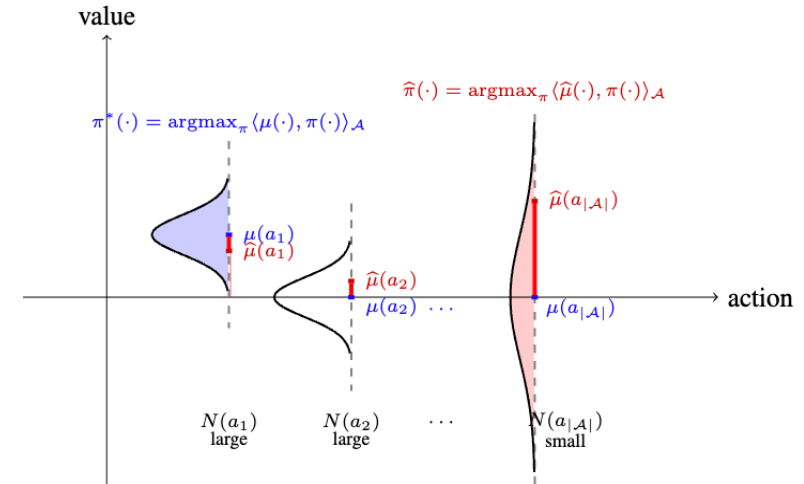
Count based Bonus

Guides the agent's behavior to prefer rarely visited states to common states.

Multi arm Bandit : UCB

Finite case, just use count.

$$UCB_i(t) = \hat{\mu}_i(t) + \sqrt{\frac{2 \log t}{N_i(t)}}$$



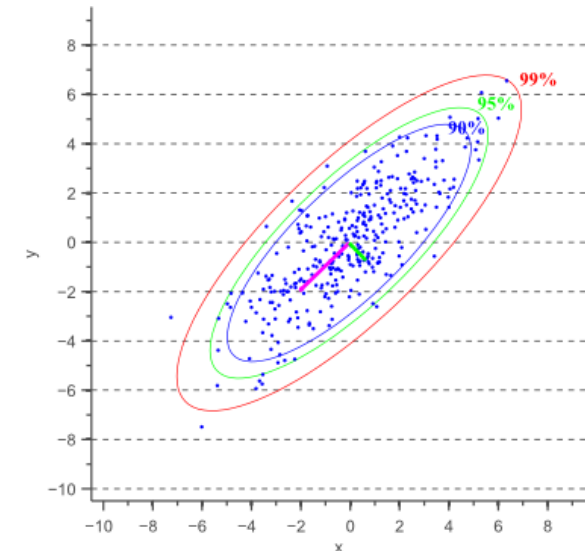
Linear bandit / MDP

Build confidence ellipse.

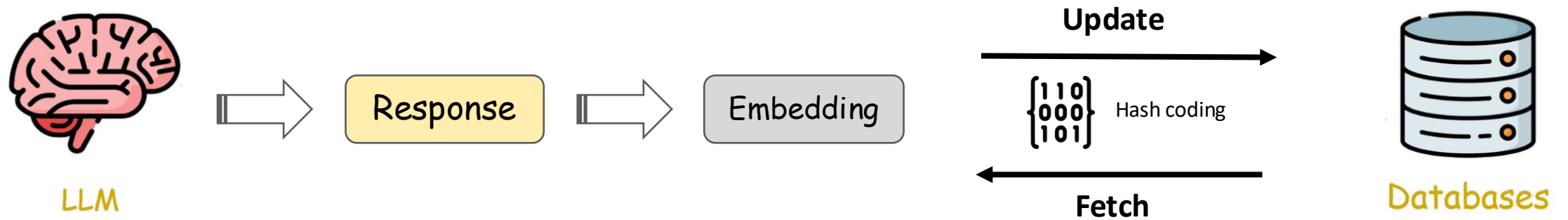
$$V_t = \lambda I + \sum_{s=1}^{t-1} \phi_s \phi_s^\top$$

$$UCB_t(a) = \phi_a^\top \hat{\theta}_t + \alpha \sqrt{\phi_a^\top V_t^{-1} \phi_a}$$

Describing an **ellipsoid** in feature space, the new visits stretches the ellipsoid in the direction.



Count based workflow



Hard to characterize/distinguish COTs with embedding vector.

Kid's learning is not driven by summary and count of experiences, instead propelled by an intrinsic curiosity to explore novel situations.

LLM already develops a sophisticated internal model of what constitutes a familiar versus a novel reasoning pattern.

Actor Curiosity

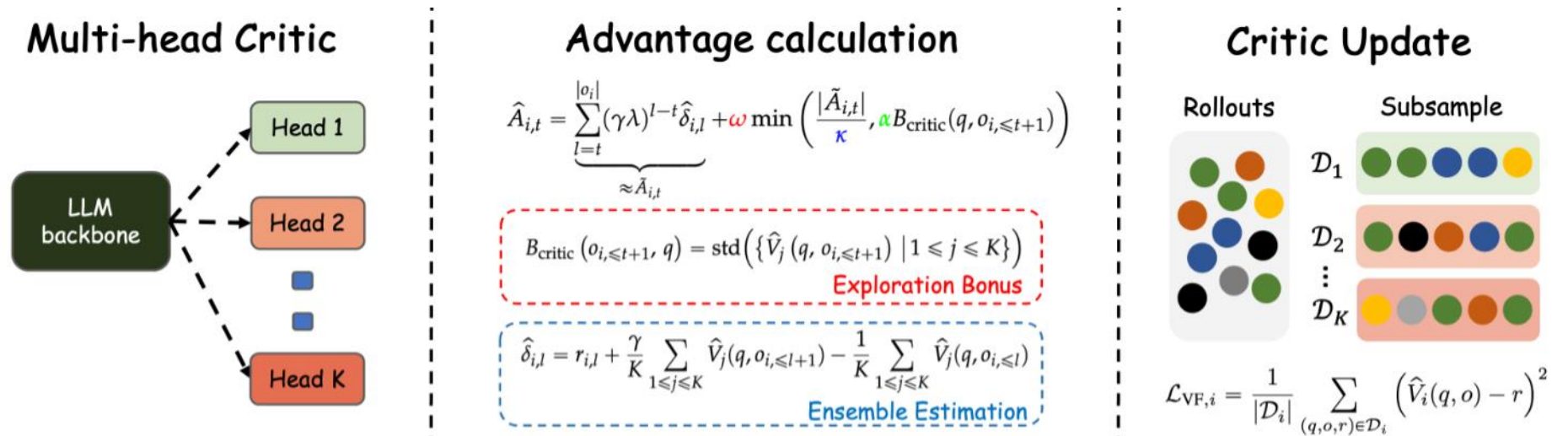
	High PPL	Low PPL
Correct		
Incorrect		
	Penalize Confident Mistakes	
	Encourage Diverse Correctness	

$$B_{\text{actor}}(q, o) = -\frac{1}{T} \sum_{t=1}^T \log \pi(o_t | o_{<t}, q)$$

$$\hat{r}(q, o) = r(q, o) + \omega \min\left(\frac{|r(q, o)|}{\kappa}, \alpha B_{\text{actor}}(q, o)\right)$$

Main Idea: Increase the reward of trajectories with higher Perplexity(PPL).

Critic Curiosity



Main Idea: Bootstrap to capture the uncertainty of value estimate.

RLVR – Other Debates

Does Reinforcement Learning Really Incentivize Reasoning Capacity in LLMs Beyond the Base Model?

Yang Yue^{1*†}, Zhiqi Chen^{1*}, Rui Lu¹, Andrew Zhao¹, Zhaokai Wang², Yang Yue¹, Shiji Song¹, and Gao Huang^{1✉}

¹ LeapLab, Tsinghua University ² Shanghai Jiao Tong University

* Equal Contribution † Project Lead ✉ Corresponding Author

Pass@1 better after RLVR
Pass@K worse after RLVR
— compared to the base LLM —



Stella Li CogSci2025
@StellaLisy

🤖 We cracked RLVR with... Random Rewards?!

Training Qwen2.5-Math-7B with our Spurious Rewards improved MATH-500 by:

- Random rewards: +21%
- Incorrect rewards: +25%
- (FYI) Ground-truth rewards: + 28.8%

How could this even work!?! Here's why:

Blogpost: tinyurl.com/spurious-rewar...

Spurious Rewards: Rethinking Training Signals in RLVR

Rulin Shao^{1*} Shuyue Stella Li^{1*} Rui Xin^{1*} Scott Geng^{1*} Yiping Wang¹
Sewoong Oh¹ Simon Shaolei Du¹ Nathan Lambert² Sewon Min³ Ranjay Krishna^{1,2}
Yulia Tsvetkov¹ Hannaneh Hajishirzi^{1,2} Pang Wei Koh^{1,2} Luke Zettlemoyer¹
¹University of Washington ²Allen Institute for Artificial Intelligence
³University of California, Berkeley
{rulins, stelli, rx31, sgeng}@cs.washington.edu



Even random reward works?

Thank You!