School of CIT –
Social Computing
Research Group

Technical
University
of Munich

Advanced Natural Language Processing
CIT4230002

Prof. Dr. Georg Groh
Simon Malberg, M.Sc.

May 7, 2024

Lecture 2.2
Explainability &
Problem-Solving

LLMs as a “Reasoning Engine”

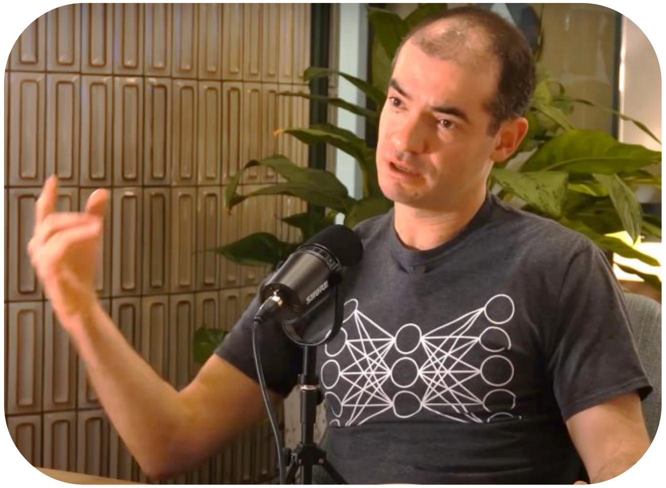
“I think people make a big mistake when they think of these models as a database. […] What makes this special, why it’s worth spending all this money and effort is, **it’s a reasoning engine and we trained it to be a reasoning engine**.”

Sam Altman, CEO OpenAI

Source: OpenAI's Sam Altman at TUM (May 25th, 2023), https://youtu.be/uaQZIK9gvNo?si=69wn8ZgOMd7WA5oF

Next-Token Prediction Could be Enough for AGI[1]

"You just ask [the neural net] what would a person with great insight and wisdom and capability do? Maybe such person doesn't exist, but there's a pretty good chance that the neural net will be able to extrapolate how such a person would behave […] from the data of regular people. […] **Predicting the next token well means that you understand the underlying reality that led to the creation of that token**."



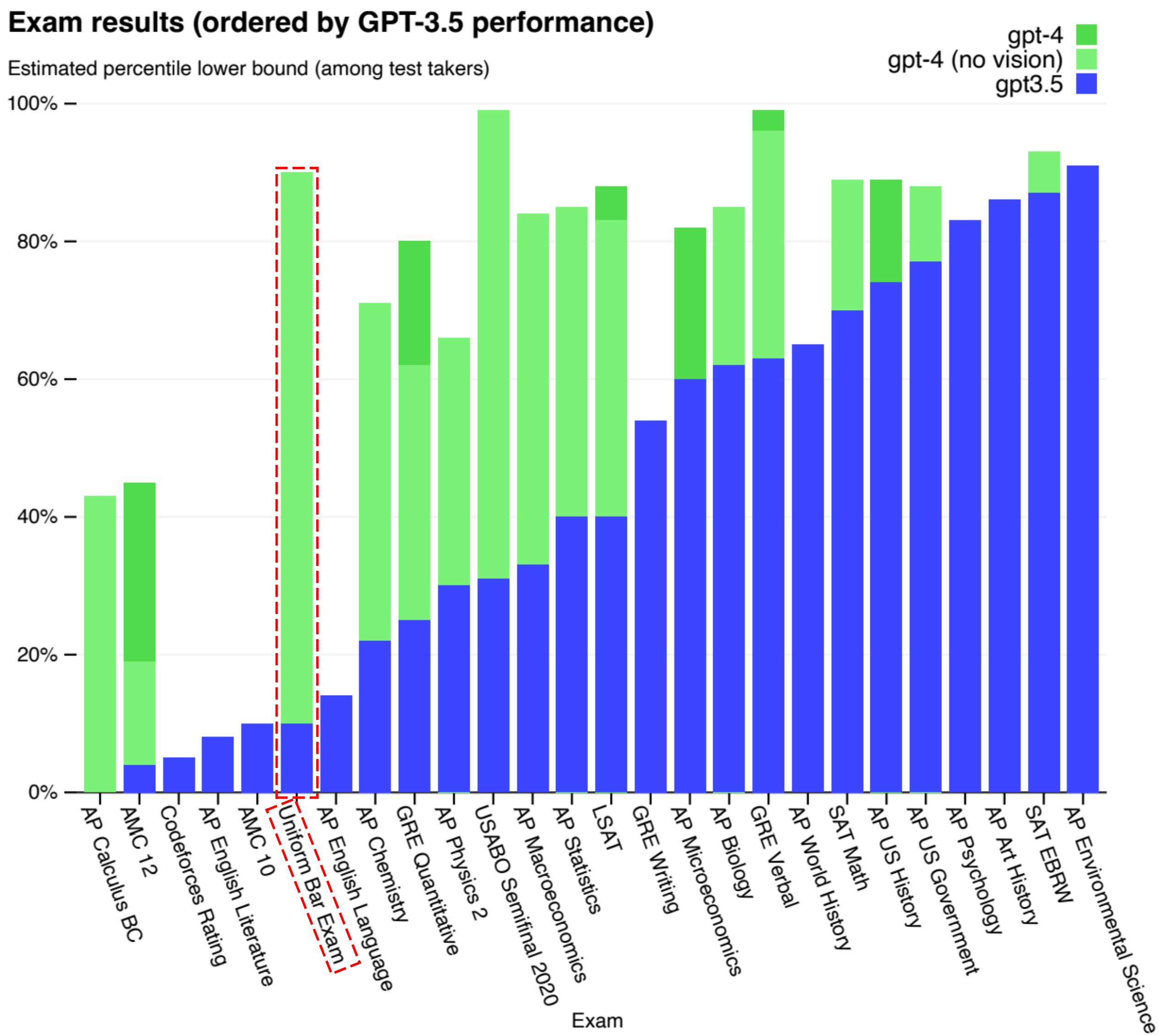
Ilya Sutskever, OpenAI Chief Scientist

1. Artificial General Intelligence (AGI) is a hypothetical type of AI that is capable of any intellectual task that a human can perform

Source: https://youtu.be/YEUclZdj_Sc?si=VMSsf2eO115uq9N_

LLMs Emerge as General-Purpose Problem Solvers

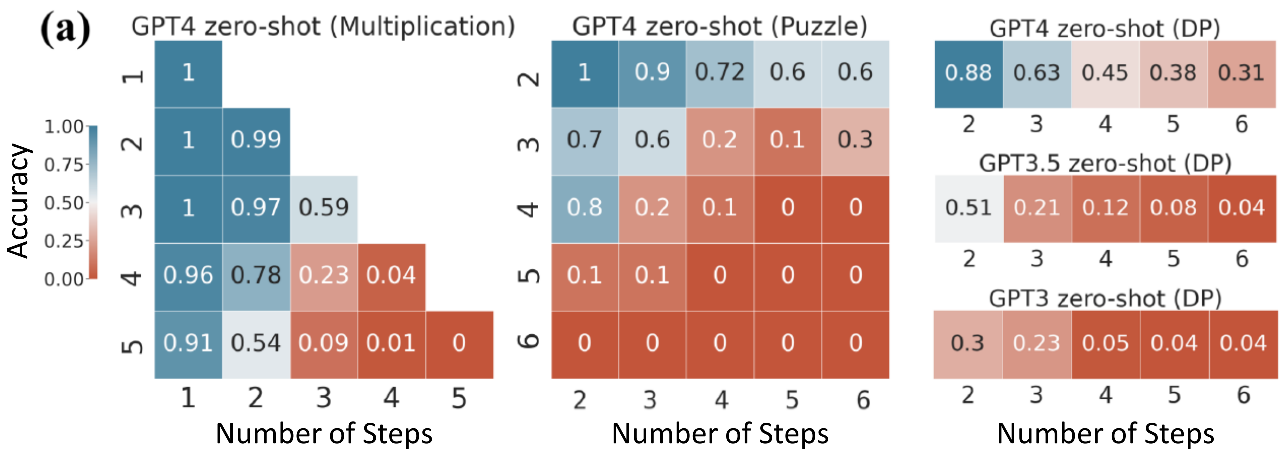
GPT-4 completes the bar exam (examination for those seeking to become lawyers in the USA) **within the top 10% of human test takers**



Source: Achiam et al. (2023): GPT-4 Technical Report, https://arxiv.org/pdf/2303.08774

Limitation 1: Multi-Step Reasoning

Accuracy vs. Task Complexity in Multi-Step Reasoning



- **Autoregressive Transformer LLMs greedily produce the next word** without a rigorous global task understanding
- **Early errors in the computational process can lead to substantial compounding errors** in subsequent steps, preventing models from finding correct solutions
- **LLMs exhibit shortcut learning via pattern matching** – fast correct answers when patterns were seen during training, but sharp decline of performance on out-of-domain data

The authors' suggestion:

Augment Transformer LLMs with **planning modules** and **refinement methods**, that can iteratively improve their generations

Source: Dziri et al. (2023): Faith and Fate: Limits of Transformers on Compositionality, https://arxiv.org/pdf/2305.18654

Limitation 2: Hallucinations

- **LLM’s training data often includes incorrect, outdated, or biased information** (massive data from the web, not just data curated for a specific task)
- **It is likely that a particular task is out-of-domain**, i.e., similar tasks have never been seen by the LLM during training (variety of different tasks, domains, and languages)
- **LLM output may initially seem highly plausible**, even if the generated information is false, making it difficult to detect hallucinations

Conclusion: Need **more transparency** on LLM’s thought process and used sources

Source: Zhang et al. (2023): Siren’s Song in the AI Ocean: A Survey on Hallucination in Large Language Models, https://arxiv.org/pdf/2309.01219

Research is Actively Working on Solutions

Solutions

Explainable AI (XAI)

Analyze the model behavior to make it more transparent to humans (and sometimes to the LLM itself)

Today

Prompting Techniques

Use advanced strategies to prompt the LLM for output and better guide it to find good solutions

Tool Usage

Allow the LLM to use external tools (e.g., calculator, code execution) where the LLM’s skills reach their limits

Retrieval Augmented Generation (RAG)

Retrieve data from external sources and provide them to the LLM at inference time to improve accuracy and traceability

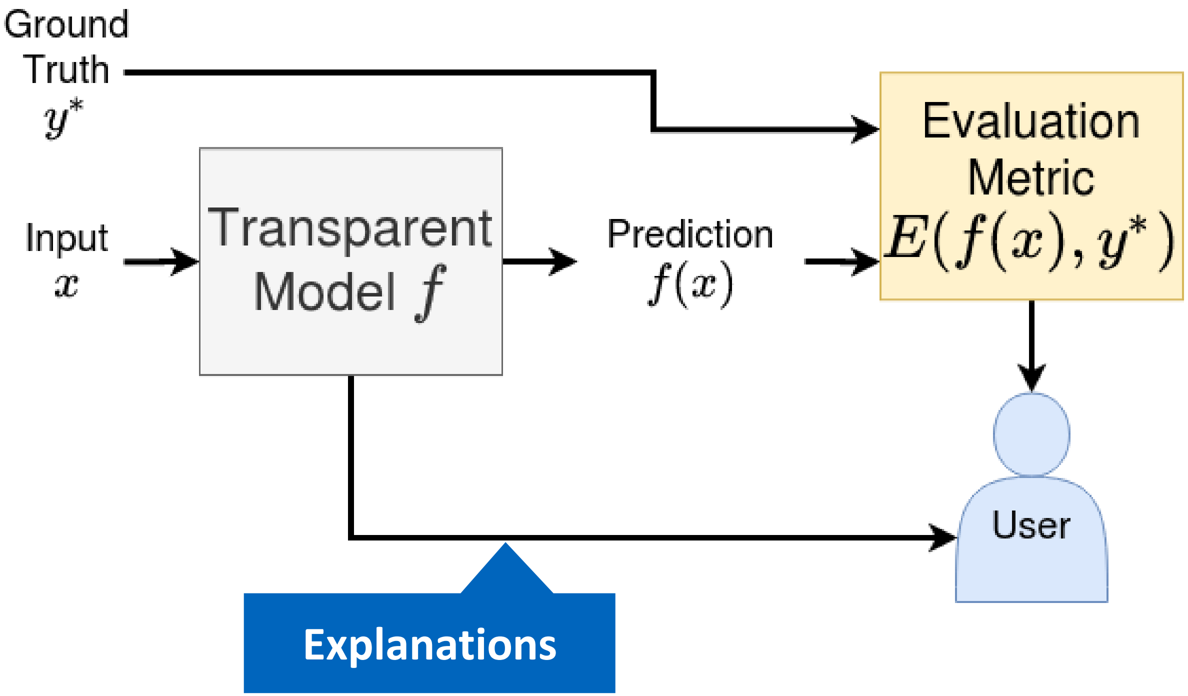
Later

More LLM limitations and possible solutions are being proposed every day

Different Methods for Explainable AI (XAI)

	1 Transparent Models	2 Model-Specific Analysis	3 Model-Agnostic Analysis	4 Prompting Techniques
		Post-Hoc Analysis		
Idea	Use inherently transparent models where it is **clear how and why every output was given**	Use fine-grained model analysis techniques to **find clues for explainability in the model's internals**	Use statistical techniques to **find explainable patterns between model inputs and outputs**	Use advanced strategies for prompting to **make the LLM's thought process more transparent** and reliable
Examples	• Linear/Logistic Regression • Decision Trees	• Neuron Activations • Attention Weights	• Surrogate Models • Feature Attribution (e.g., SHAP, LIME)	• Natural language rationales • Chain-of-Thought • Tree-of-Thoughts
	Only covered briefly		*Focus of today*	

Transparent Models



Examples of transparent models

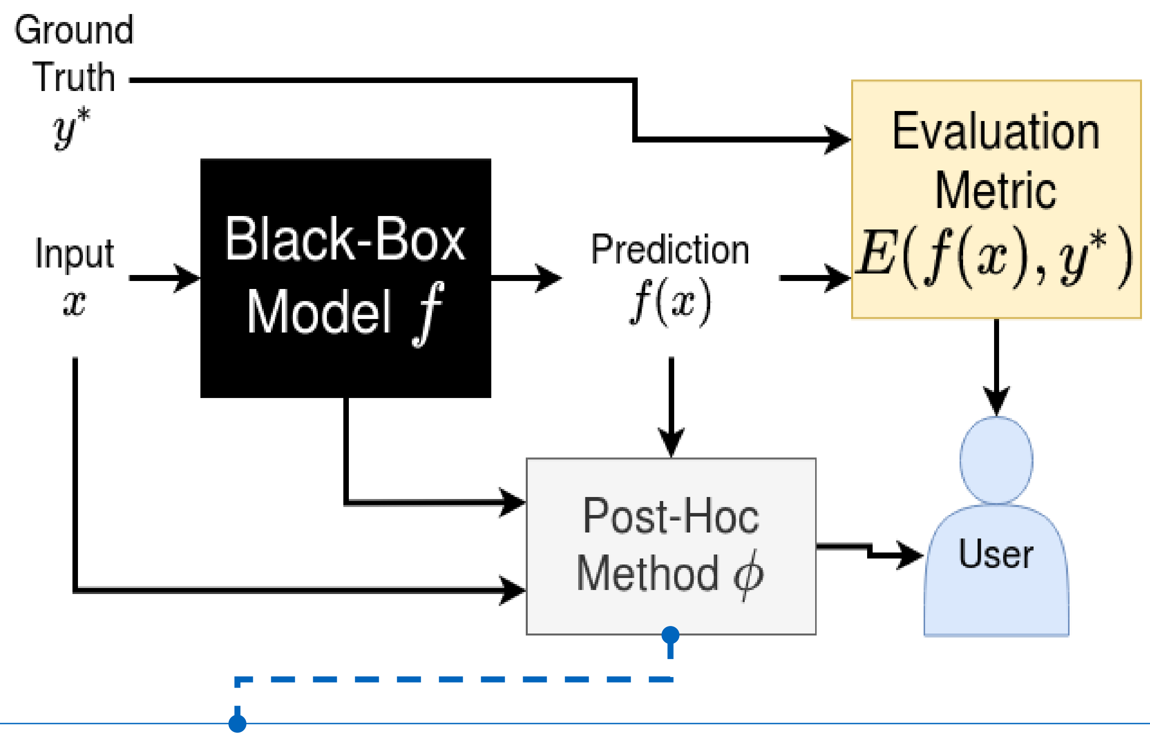
- Feature coefficients learned by Linear/Logistic Regression
- Classification/regression trees
- Feature importance scores learned by Random Forest

Key takeaways

- With transparent models, explainability is a **property of the model itself** (i.e., we know how the model works)
- Transparent models are often the **safest and most trustworthy setting**
- However, **declining in popularity** because other, more powerful models are being released

Post-Hoc Analysis

Using a post-hoc method to generate explanations



The post-hoc method can be ...

- **Model-agnostic** or **model-specific**: Explain any model or only a specific model class (e.g., only decision trees, DNNs)
- **Local** or **global**: Models are explained on an instance-level or dataset-level, respectively

Key takeaways

- Post-hoc explainability is **not a model property**
- An **additional technique produces explanations** that convey useful information about the non-transparent model

Explaining Neuron Activations in (L)LM

Example: Recent Methodology Proposed by Researchers from OpenAI

Step 0 Choose a particular neuron in a language model

Step 1.1 Measure the neuron's activations on a given (training) text sequence

Step 1.2 Generate a conceptual explanation of the activations using GPT-4 LLM

Show neuron activations to GPT-4:

The Avengers to the big screen, Joss Whedon has
Avengers: Age of Ultron pits the titular heroes again
box office to be the highest-grossing film of the

introduction into the Marvel cinematic universe, it's possib
"Tony is earthbound and facing earthbound villains. You wi
does hint that they have some use... STARK T

, which means this Nightwing movie is probably not about
Batman is going to dig into some of this backstory or intro
to have a lot of work to do explaining

of Avengers who weren't in the movie and also Thor try to
completely pointless, an embarrassing loss, and I'm pretty
Earth, one of the heroes inadvertantly blows up an

Used Prompt to GPT-4 (Example):

We're studying neurons in a neural network. Each neuron looks for some particular thing in a short document. Look at the parts of the document the neuron activates for and **summarize in a single sentence what the neuron is looking for**. Don't list examples of words.

The activation format is token<tab>activation. Activation values range from 0 to 10. A neuron finding what it's looking for is represented by a non-zero activation value. The higher the activation value, the stronger the match.

Neuron 1
Activations:
<start>

the	0
sense	0
of	0
together	3
ness	7

...

GPT-4 gives an explanation, guessing that the neuron is activating on

references to movies, characters, and entertainment.

Step 2 Simulate activations using GPT-4, conditioning on the explanation

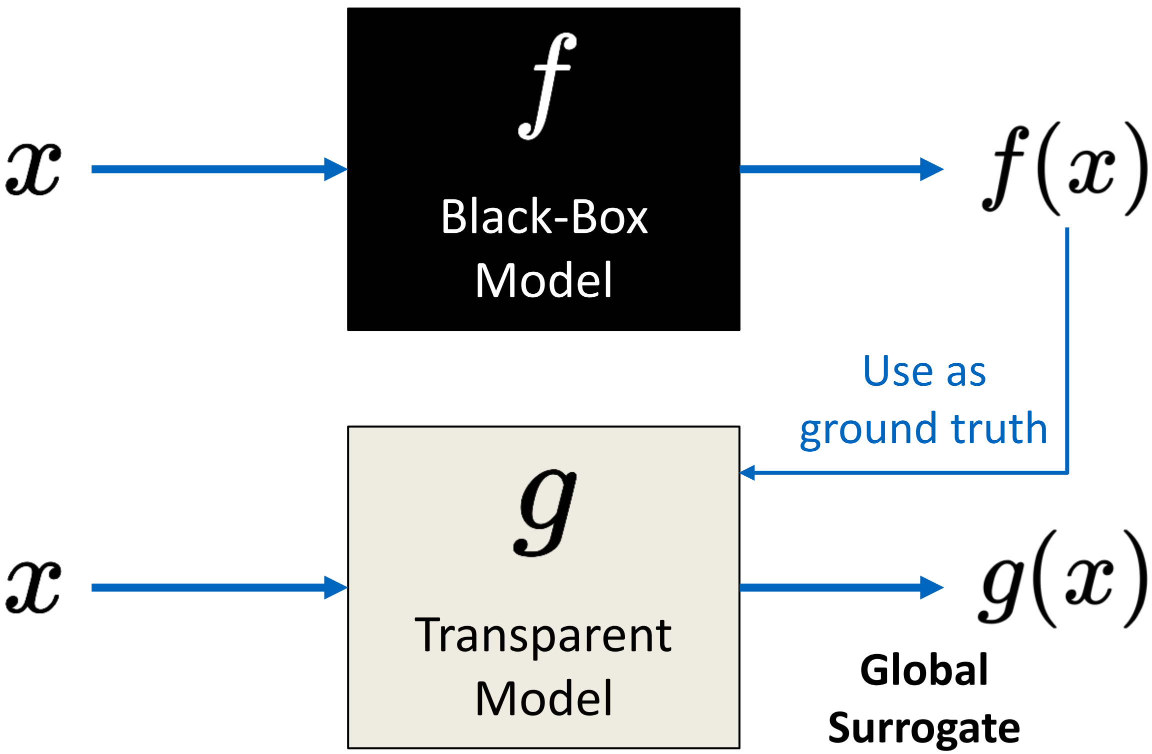
Step 3 Score the explanation by comparing the simulated and real activations

Conclusion: Several methods for model-specific analysis available. But typically require in-depth model knowledge and technical skills!

Source: Bills et al. (2023): Language Models Can Explain Neurons in Language Models, https://openaipublic.blob.core.windows.net/neuron-explainer/paper/index.html

Surrogates

Approximate a complex model with a transparent one

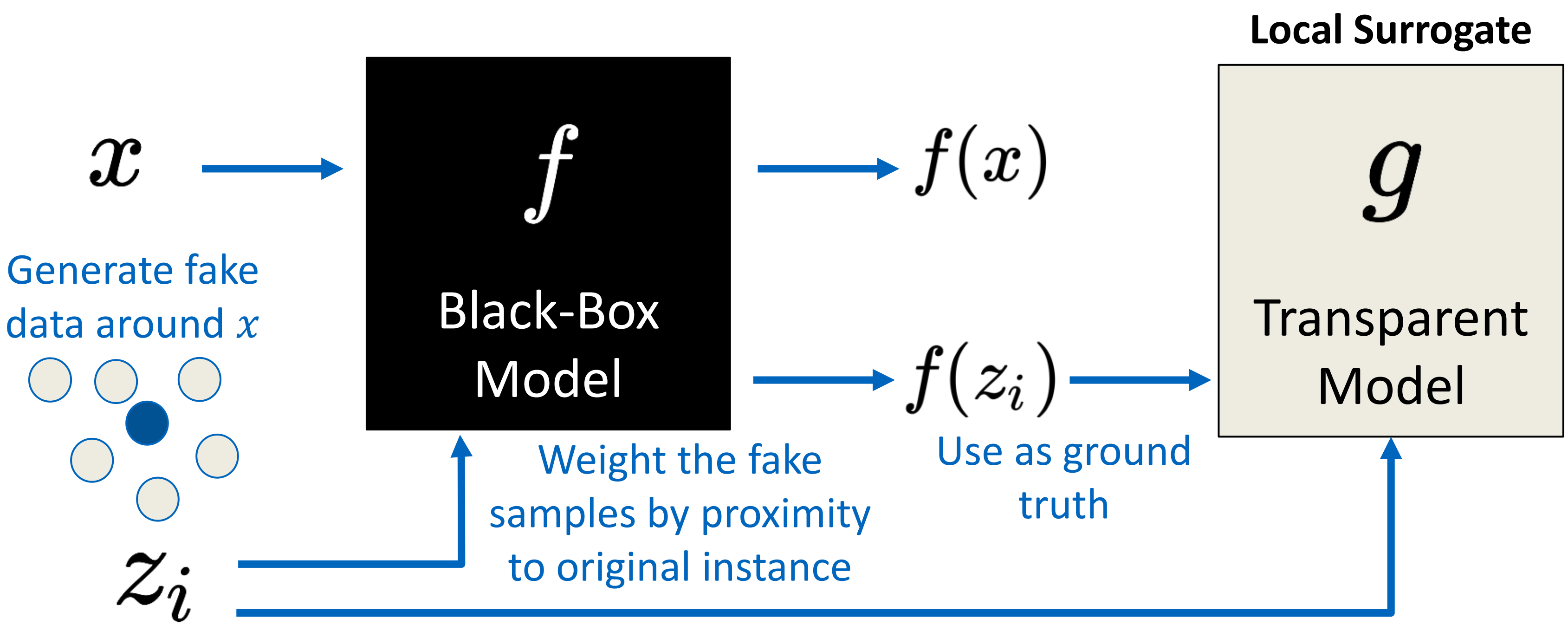


Key takeaways

- **Surrogate models are simpler transparent models** that approximate the behavior of a complex model
- Surrogates are trained using the **outputs of the complex model as ground truth**
- **Main concern:** Can a transparent model approximate something that is orders of magnitudes more complex?

Local Surrogates

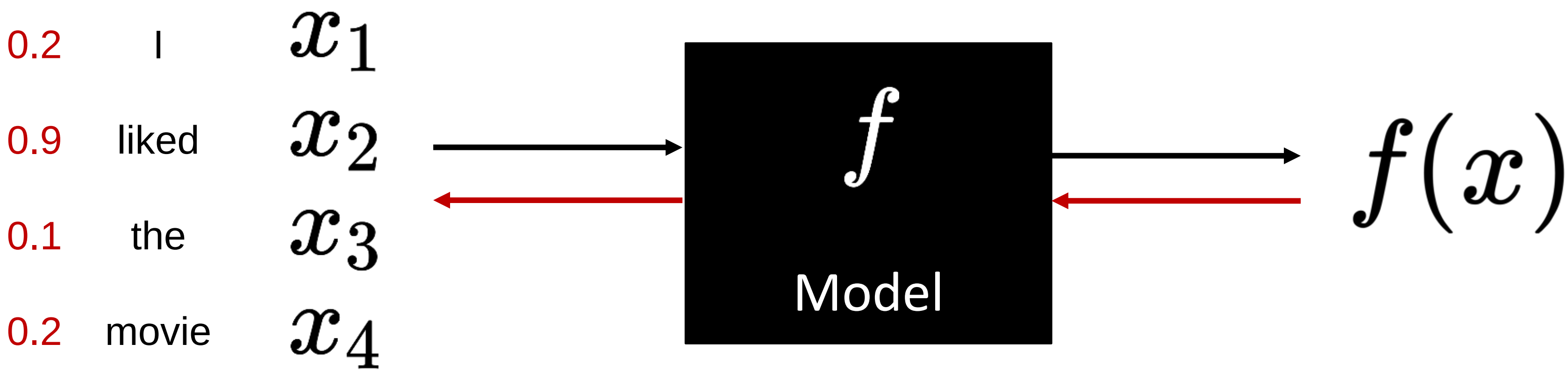
Idea: Complex models could be at least locally simple, i.e., around a single instance



- The transparent model **locally approximates** the complex one
- We can use g to **explain the original instance** $(x, f(x))$

Feature Attribution

Idea: Attribute an importance score to each input feature



- Feature attributions measure the **relevance/impact of a feature** …
  — **on the model (global)** or
  — **on the prediction (local)**
- Basic approaches:
  — **Remove or replace a token** (≈ feature), re-run the prediction, and look at the difference in prediction vs. the original
  — **Measure the gradient of output logits** w.r.t. input tokens

Feature Attribution: SHAP

SHapley Additive ExPlanations (SHAP) have their roots in game theory

Shapley Values (1953)

Quantify the contribution that each **player** brings to the **game**

Idea:

Determine the importance of a single **player** by considering the **game outcome** for every possible **coalition** of **players**

SHAP (2017)

Quantify the contribution that each **feature** brings to the **prediction**

Idea:

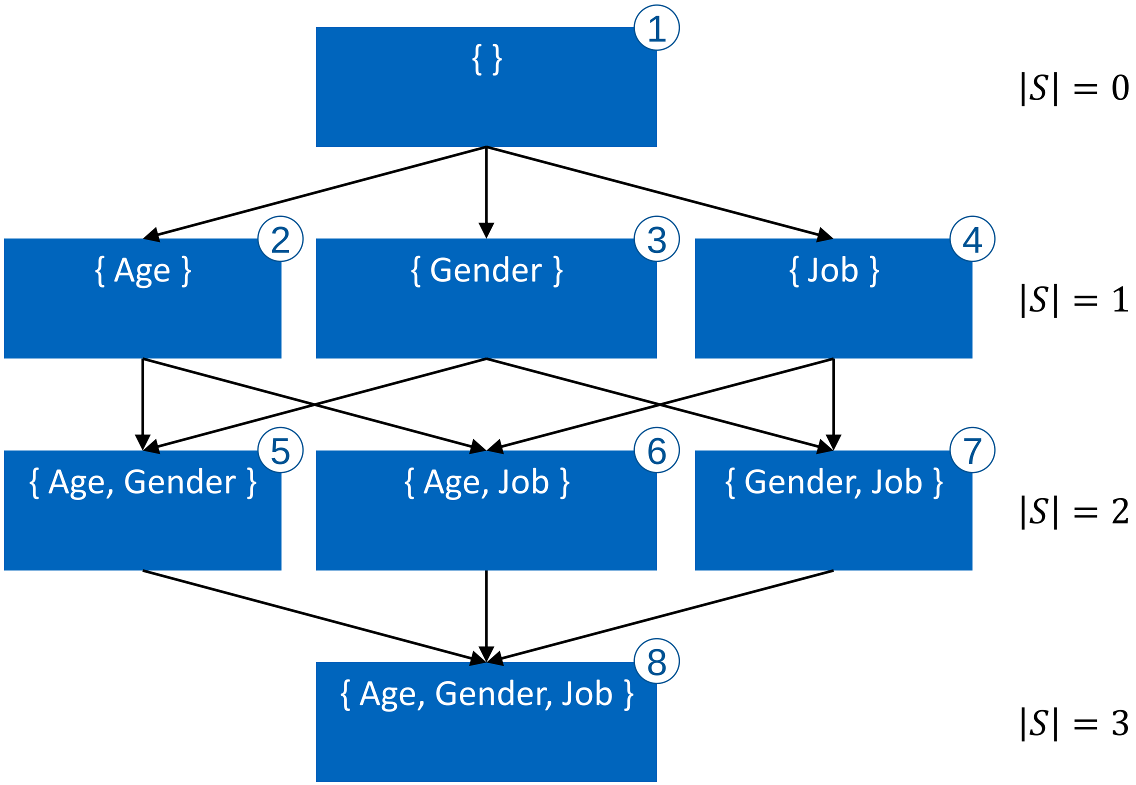
Determine the importance of a single **feature** by considering the **prediction outcome** for every possible **combination** of **features**

Remark: SHAP provides explanations **for a single observation**, i.e., it is a measure of local explainability of a predictive model

Feature Attribution: SHAP

- **Task:** Predict the **Income** of a person knowing their **Age**, **Gender**, and **Job**[1]
- **Question:** For a particular **person x_0** and **model f**, what was the **impact of the Age feature** on the **prediction $f(x_0)$**?

Step 1: Determine all possible combinations S of features $1, \ldots, M$



In math, this is known as the **power set**, which can be represented by a tree

The **cardinality of a power set is 2^M** where M is the number of elements in the original set (here: features)

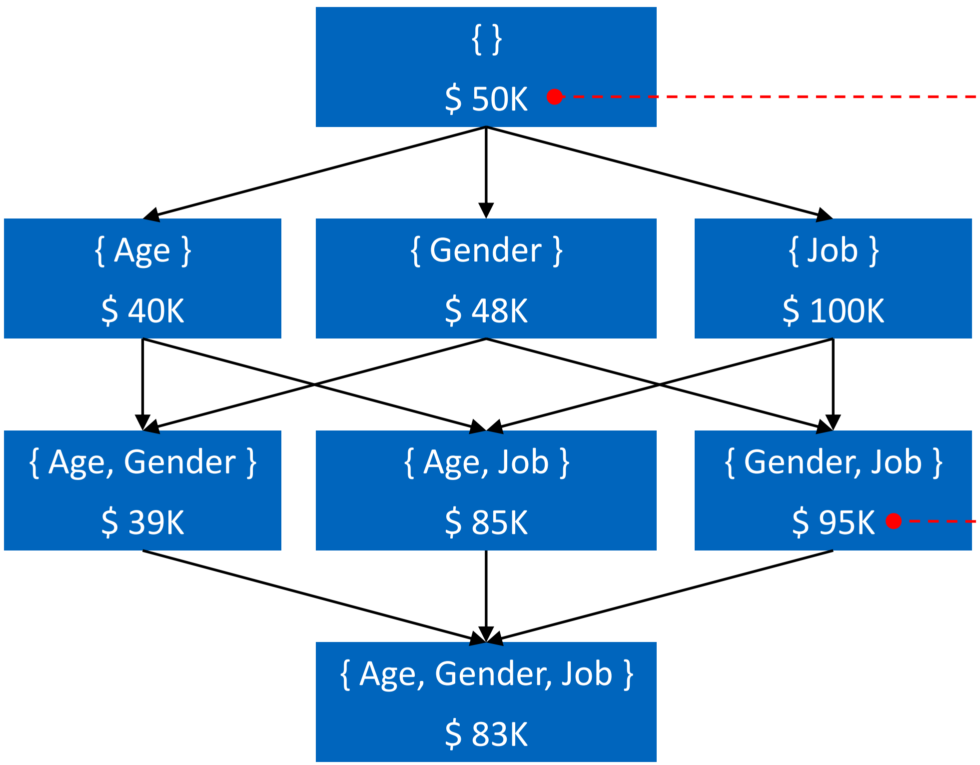
With three features, there are $2^M = 2^3 = 8$ **possible combinations**

1. Example chosen for simplicity (adapted from https://towardsdatascience.com/shap-explained-the-way-i-wish-someone-explained-it-to-me-ab81cc69ef30). In an NLP context, typical features could be elements of word vectors or TF-IDF representations of tokens

Feature Attribution: SHAP

- **Task:** Predict the **Income** of a person knowing their **Age**, **Gender**, and **Job**
- **Question:** For a particular **person x_0** and **model f**, what was the **impact of the Age feature** on the **prediction $f(x_0)$**?

Step 2: Train a distinct model f_S for each combination S and predict $f_S(x_0)$



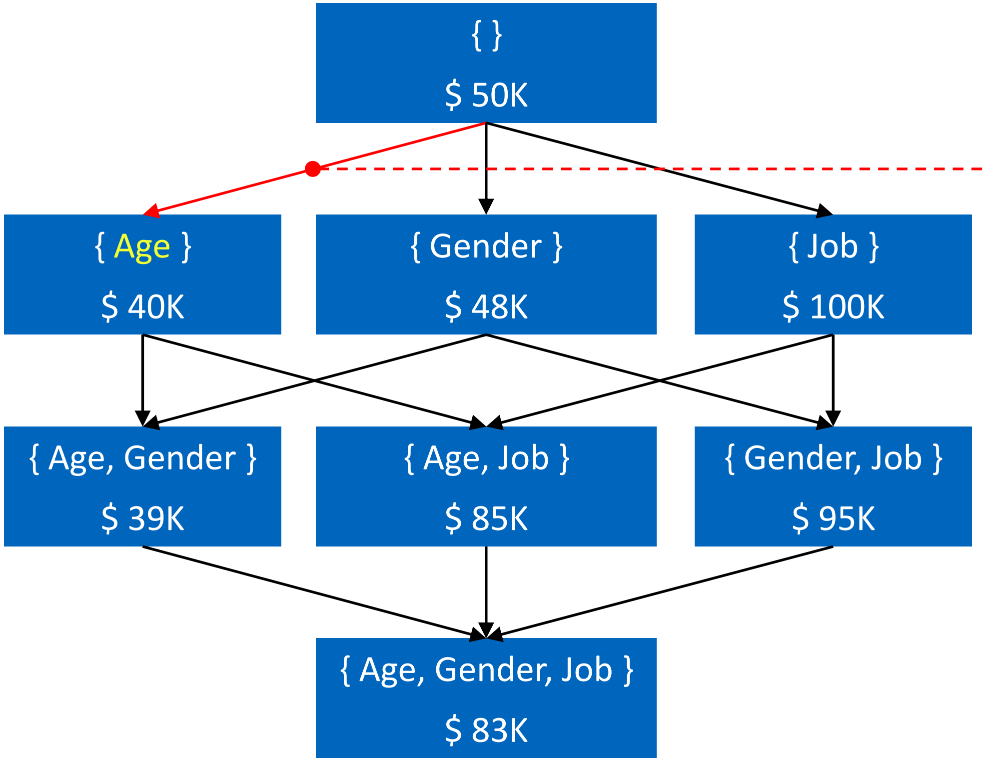
$ 50K represents the **average Income** across the entire dataset

$ 95K is the **prediction $f_S(x_0)$** for person x_0 with a model f_S only trained on feature combination $S = \{Gender, Job\}$

Feature Attribution: SHAP

- **Task:** Predict the **Income** of a person knowing their **Age**, **Gender**, and **Job**
- **Question:** For a particular **person x_0** and **model f**, what was the **impact of the Age feature** on the **prediction $f(x_0)$**?

Step 3.1: Calculate the marginal contribution of a feature to $f(x_0)$



Marginal contribution of feature **Age** in feature combination $\{Age\}$:

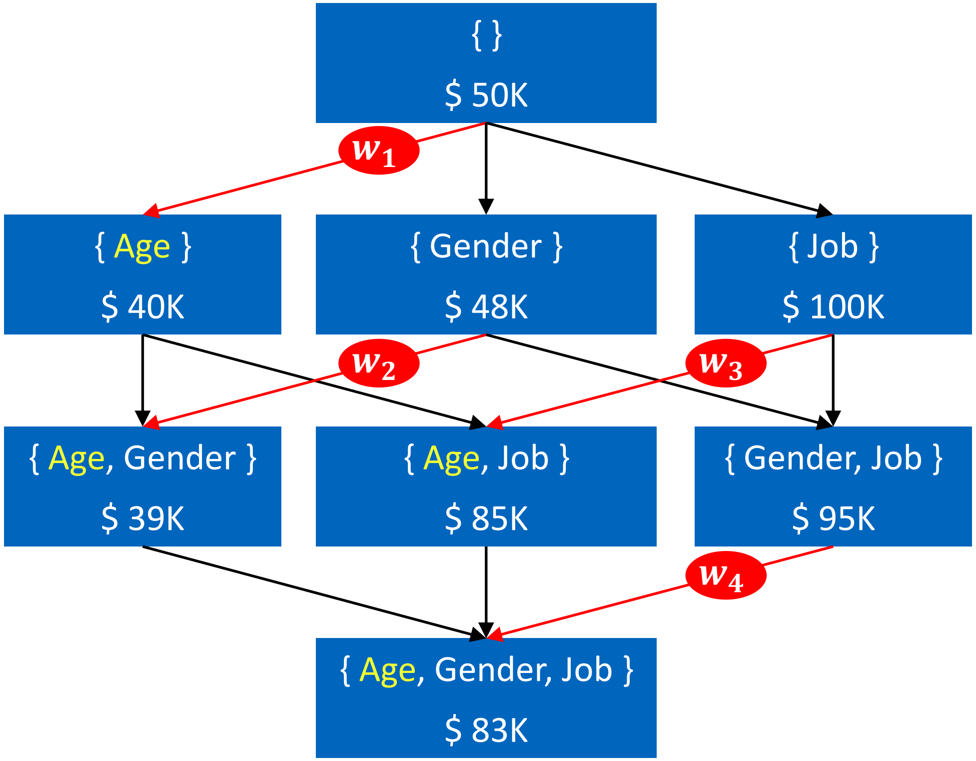
$$MC_{Age,\{Age\}}(x_0)$$
$$= f_{\{Age\}}(x_0) - f_{\{\}}(x_0)$$
$$= \$\,40K \;- \$\,50K$$
$$= \$ - 10K$$

Intuition: The marginal contribution is the difference in prediction between the target and source nodes

Feature Attribution: SHAP

- **Task:** Predict the **Income** of a person knowing their **Age**, **Gender**, and **Job**
- **Question:** For a particular **person x_0** and **model f**, what was the **impact of the Age feature** on the **prediction $f(x_0)$**?

Step 3.2: Calculate the weighted average marginal contribution of a feature to $f(x_0)$



$$
\begin{aligned}
SHAP_{Age}(x_0) = \\
& w_1 \cdot MC_{Age,\{Age\}}(x_0) + \\
& w_2 \cdot MC_{Age,\{Age,Gender\}}(x_0) + \\
& w_3 \cdot MC_{Age,\{Age,Job\}}(x_0) + \\
& w_4 \cdot MC_{Age,\{Age,Gender,Job\}}(x_0)
\end{aligned}
$$

where $w_1 + w_2 + w_3 + w_4 = 1$

Idea: The sum of weights of all MC to 1-feature models should equal the sum of weights of all MC to 2-feature models and so on (same weight for each row), i.e., here

$$w_1 = w_2 + w_3 = w_4$$

Feature Attribution: SHAP

- **Task:** Predict the **Income** of a person knowing their **Age**, **Gender**, and **Job**
- **Question:** For a particular **person x_0** and **model f**, what was the **impact of the Age feature** on the **prediction $f(x_0)$**?

Step 3.2: Calculate the weighted average marginal contribution of a feature to $f(x_0)$

The weight of an edge is the **reciprocal of the total number of edges in the same row**

In general, this means the weight of the marginal contribution of a feature m in a feature combination S is

$$w_{feature,S} = \frac{1}{|S|\cdot\binom{M}{|S|}} = \left(|S|\cdot\binom{M}{|S|}\right)^{-1}$$

The **overall SHAP value** for a **feature m**, **example x** and **model f** is calculated as

$$SHAP_m(x) = \sum_{S:m\in S} \underbrace{\left(|S|\cdot\binom{M}{|S|}\right)^{-1}}_{\text{weight}} \cdot \underbrace{\left(f_S(x) - f_{S\setminus m}(x)\right)}_{\text{marginal contribution}}$$

Feature Attribution: SHAP

- **Task:** Predict the **Income** of a person knowing their **Age**, **Gender**, and **Job**
- **Question:** For a particular **person x_0** and **model f**, what was the **impact of the Age feature** on the **prediction $f(x_0)$**?

Step 3.2: Calculate the weighted average marginal contribution of a feature to $f(x_0)$

$$
\begin{aligned}
SHAP_{Age}(x_0) = &\left(1 \cdot \binom{3}{1}\right)^{-1} \cdot MC_{Age,\{Age\}}(x_0) + \\
&\left(2 \cdot \binom{3}{2}\right)^{-1} \cdot MC_{Age,\{Age,Gender\}}(x_0) + \\
&\left(2 \cdot \binom{3}{2}\right)^{-1} \cdot MC_{Age,\{Age,Job\}}(x_0) + \\
&\left(3 \cdot \binom{3}{3}\right)^{-1} \cdot MC_{Age,\{Age,Gender,Job\}}(x_0)
\end{aligned}
$$

$$
= \frac{1}{3} \cdot (\$ - 10K) + \frac{1}{6} \cdot (\$ - 9K) + \frac{1}{6} \cdot (\$ - 15K) + \frac{1}{3} \cdot (\$ - 12K)
$$

$$
= \$ - 11.33K
$$

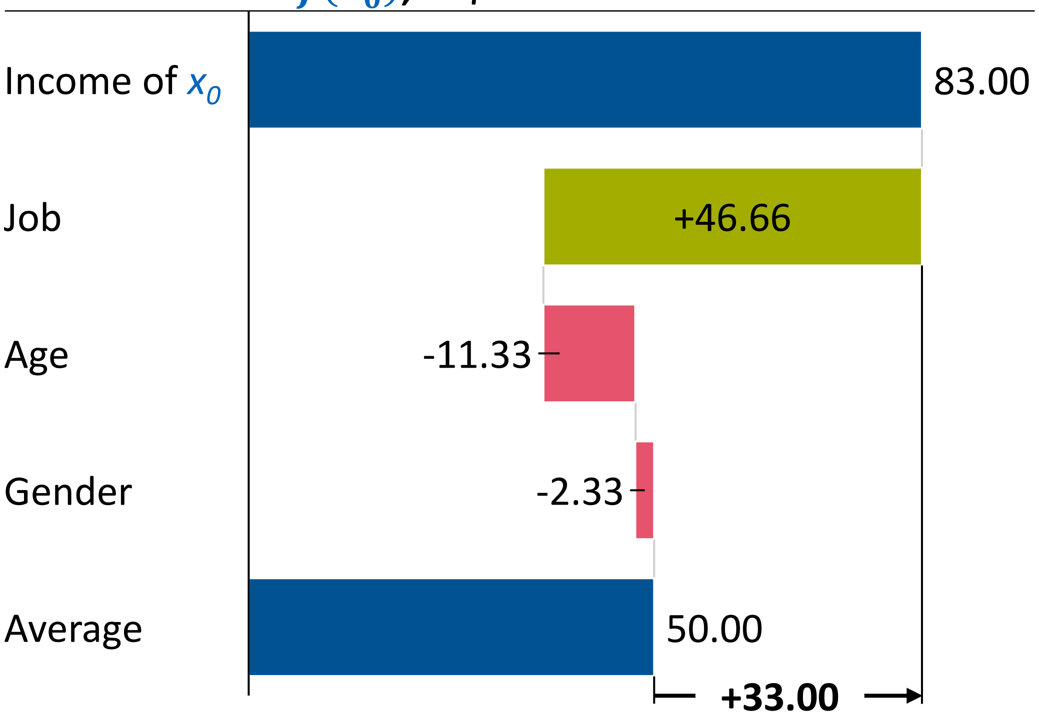
Similarly: $SHAP_{Gender}(x_0) = \$ - 2.33K$ $SHAP_{Job}(x_0) = \$ + 46.66K$

Feature Attribution: SHAP

- **Task:** Predict the **Income** of a person knowing their **Age**, **Gender**, and **Job**
- **Question:** For a particular **person x_0** and **model f**, what was the **impact of the Age feature** on the **prediction $f(x_0)$**?

Step 4: Interpret the impact of the Age feature on the prediction $f(x_0)$

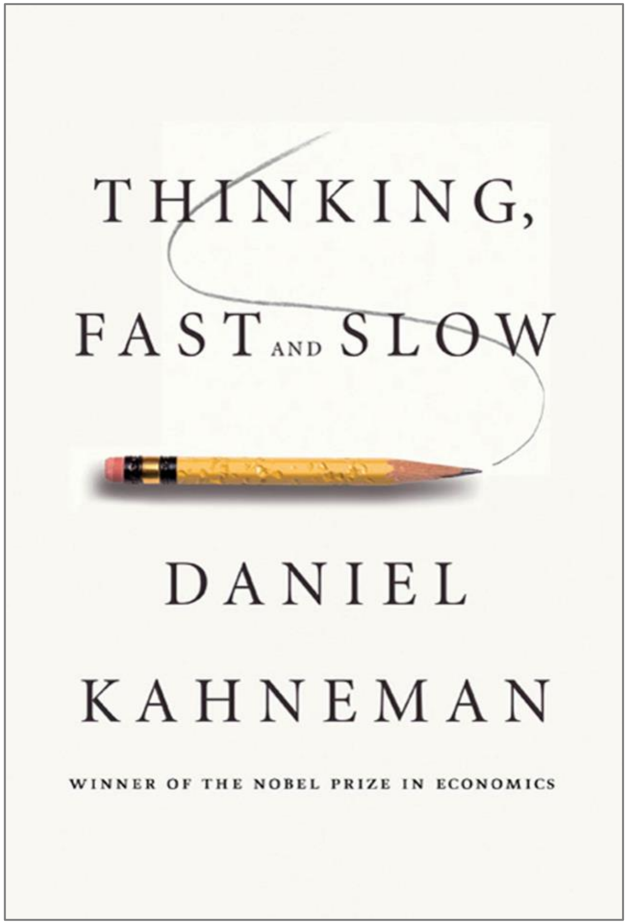
SHAP Values for $f(x_0)$, in \$K



Summing up the SHAP values of all features gives \$+33K, which is the **difference betweeen the dataset average** (knowing no features at all) **and the prediction $f(x_0)$** (knowing Age, Gender, and Job)

Human Thinking: Fast or Slow

Psychology research discovered that **humans tend to think in two systems**:



System 1: Fast | based on reflexes

emotions

habits

intuition

cognitive heuristics

logical thinking

de-biasing

solicit review

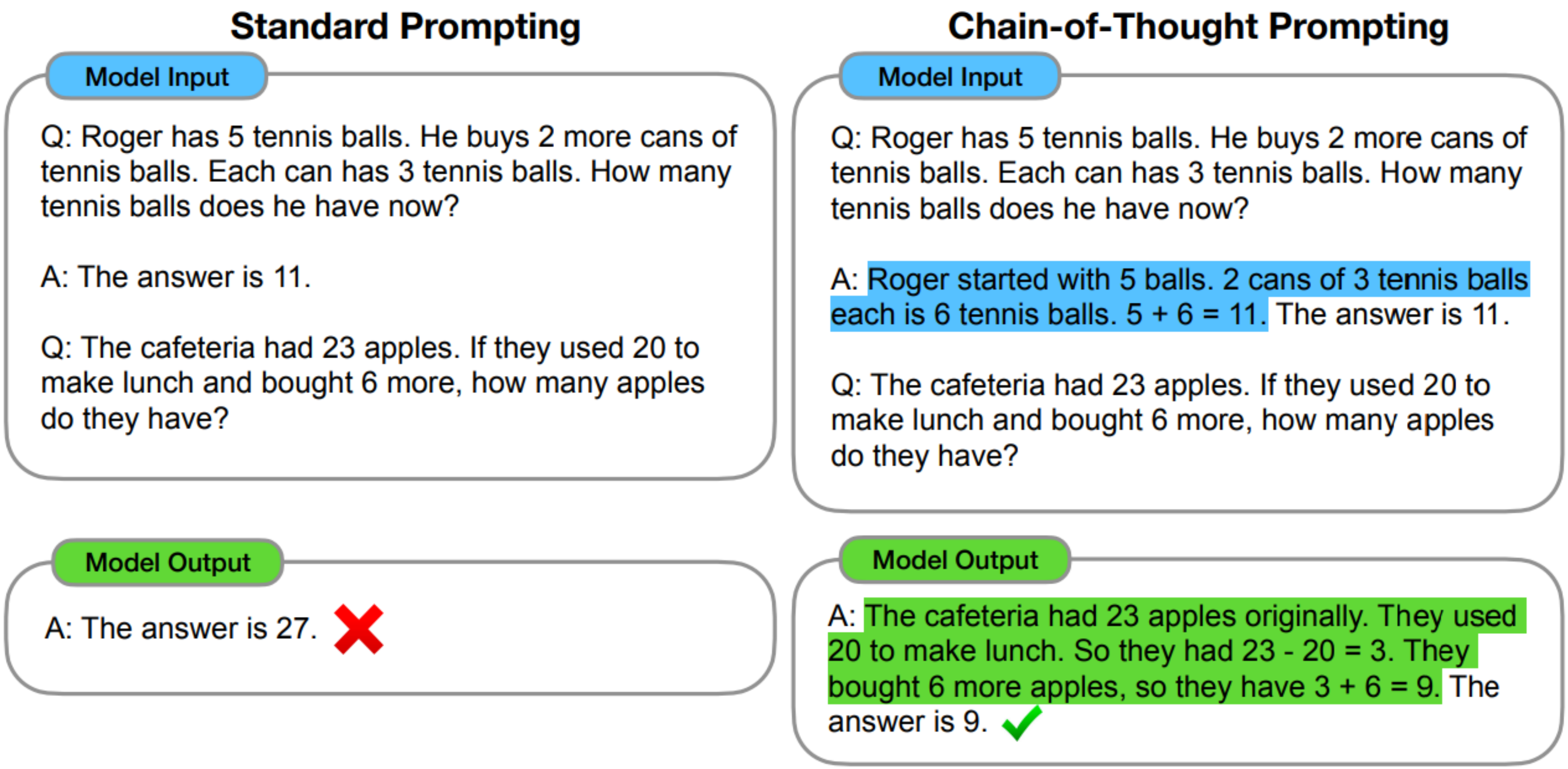
facts & data

consider alternatives

System 2: Slow | based on reflection

Idea: Maybe we can make LLMs think slowly and logically as well

Chain-of-Thought Promptong (CoT)



- **Decompose multi-step problems** into intermediate steps
- Facilitate **transparent model behavior** and provide opportunities for debugging

Observation: CoT has two positive effects at once: (1) It makes the model output **more explainable** and (2) it leads to **better model outputs**

Source: Wei et al. (2022): Chain-of-Thought Prompting Elicits Reasoning in Large Language Models, https://proceedings.neurips.cc/paper_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html

LLMs for Problem-Solving: Game of 24

Model Input

You will be given four numbers. Use basic arithmetic operations (+ − × ÷) to combine these numbers and reach 24. Think step by step.

Input: 4 9 10 13

13 − 9 = 4 (left 4 4 10)

10 − 4 = 6 (left 6 4)

6 × 4 = 24

Answer: (13 − 9) × (10 − 4) = 24

Input: 6 12 3 9

Model Output

12 ÷ 3 = 4 (left 4 6 9)

6 − 4 = 2 (left 2 9)

9 × 2 = 18 (left 18)

18 + 6 = 24

Answer: (12 ÷ 3) × (6 − 4) + 9 = 24 ❌

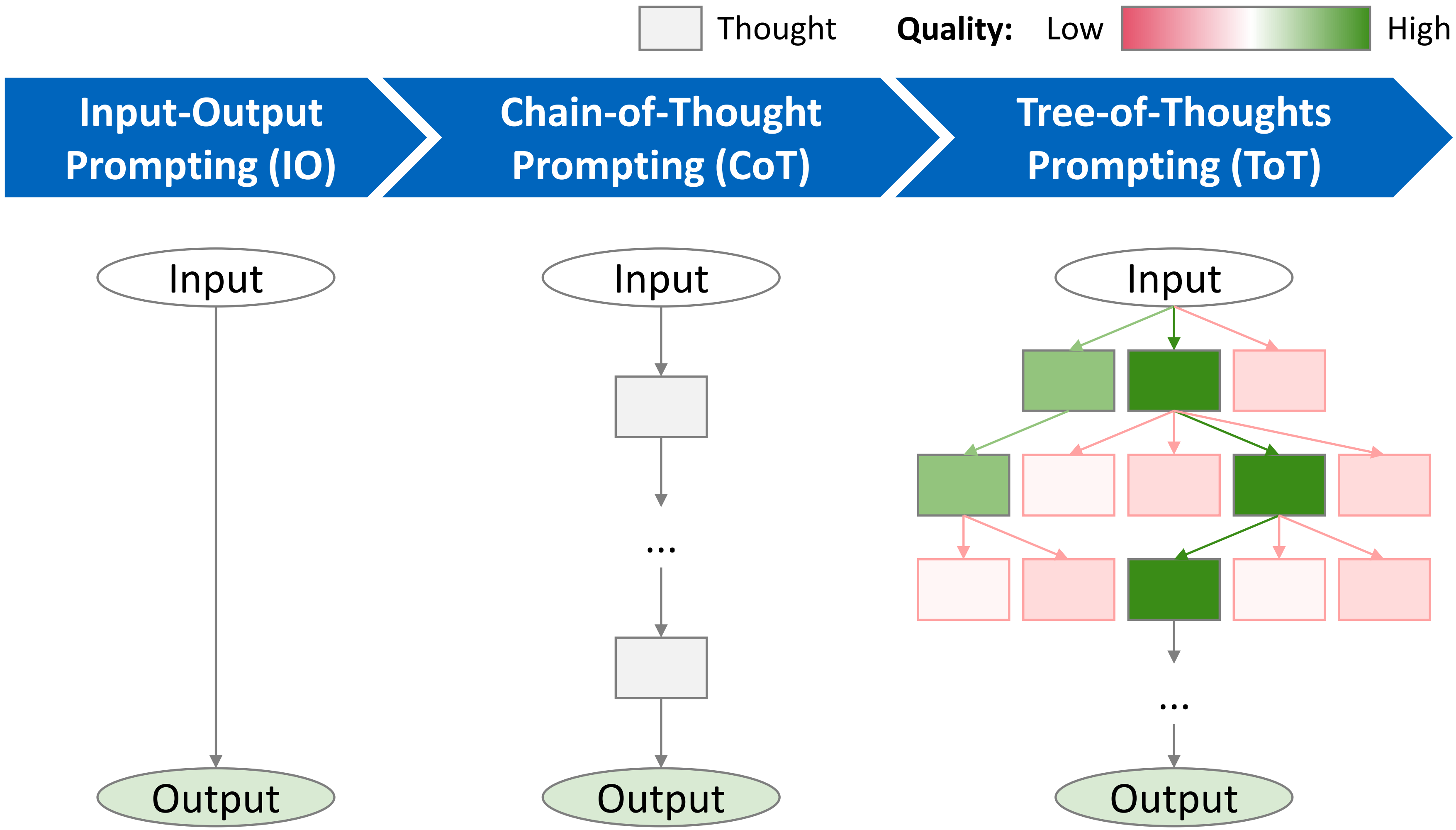
Hallucination!

Compounding Errors!

Key takeaways

- Some **problems remain too difficult** for CoT
- Challenges with the Game of 24 include:
  - **Exponential search space**
  - **Inherent randomness**
  - **Discrete and non-differential problem**
  - **Explicit reasoning**
- CoT applies a **linear, greedy problem-solving approach** which is prone to compounding errors
- We need an approach that is better at **exploring the problem space and recovering from errors!**

Tree-of-Thoughts Prompting (ToT)



Idea: Explore multiple thoughts in parallel and evaluate them for their quality. Discard the worst and keep the best

Source: Yao et al. (2023): Tree of Thoughts: Deliberate Problem Solving with Large Language Models, https://proceedings.neurips.cc/paper_files/paper/2023/hash/271db9922b8d1f4dd7aaef84ed5ac703-Abstract-Conference.html

ToT's 4 Components

1. Thought Decomposition

- Decide **how to decompose a problem into thoughts**. A thought should be small enough so that the LM p_θ can generate diverse samples, yet big enough so that the LM can evaluate its prospect towards solving the problem
- **Each node in the tree** is a state $s = [x, z_{1\cdots i}]$ **representing a partial solution** with the input x and the sequence of thoughts $z_{1\cdots i}$ so far

2. Thought Generator $G(p_\theta, s, k)$

Given a tree state $s = [x, z_{1\cdots i}]$, **generate k candidates for the next thought step**

Two alternative strategies:

- **Sample** i.i.d. thoughts: $z^{(j)} \sim p_\theta^{CoT}(z_{i+1}|s) = p_\theta^{CoT}(z_{i+1}|x, z_{1\cdots i})\ (j = 1 \cdots k)$
- **Propose** thoughts sequentially: $[z^{(1)}, \cdots, z^{(k)}] \sim p_\theta^{propose}\left(z_{i+1}^{(1\cdots k)} \mid s\right)$

3. State Evaluator $V(p_\theta, S)$

Given a frontier of different states S, **evaluate the progress they make towards solving the problem**, serving as a heuristic for the search algorithm

Two alternative strategies:

- **Value** each state independently by sampling a value $v \sim p_\theta^{value}(v|s)\ \forall s \in S$
- **Vote** across states, comparing different states and voting for a "good" state $s^* \sim p_\theta^{vote}(s^*|S)$

4. Search Algorithm

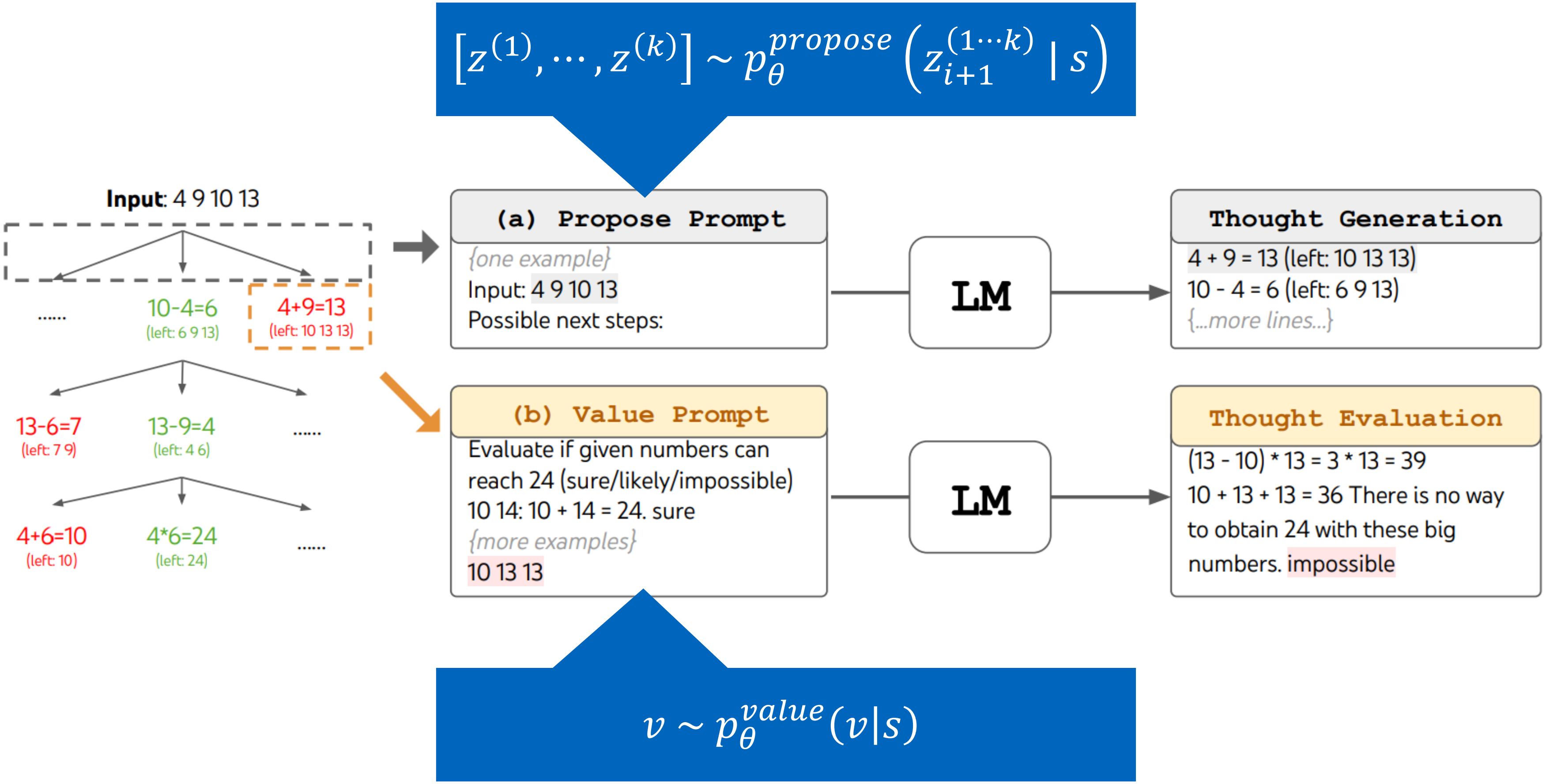
Explore a problem's solution space by **searching for high-value thoughts**

Two alternative search algorithms:

- **Breadth-first search (BFS)** maintains a set of the b most promising states per step
- **Depth-first search (DFS)** explores the most promising state first, until the final output is reached

Source: Yao et al. (2023): Tree of Thoughts: Deliberate Problem Solving with Large Language Models, https://proceedings.neurips.cc/paper_files/paper/2023/hash/271db9922b8d1f4dd7aaef84ed5ac703-Abstract-Conference.html

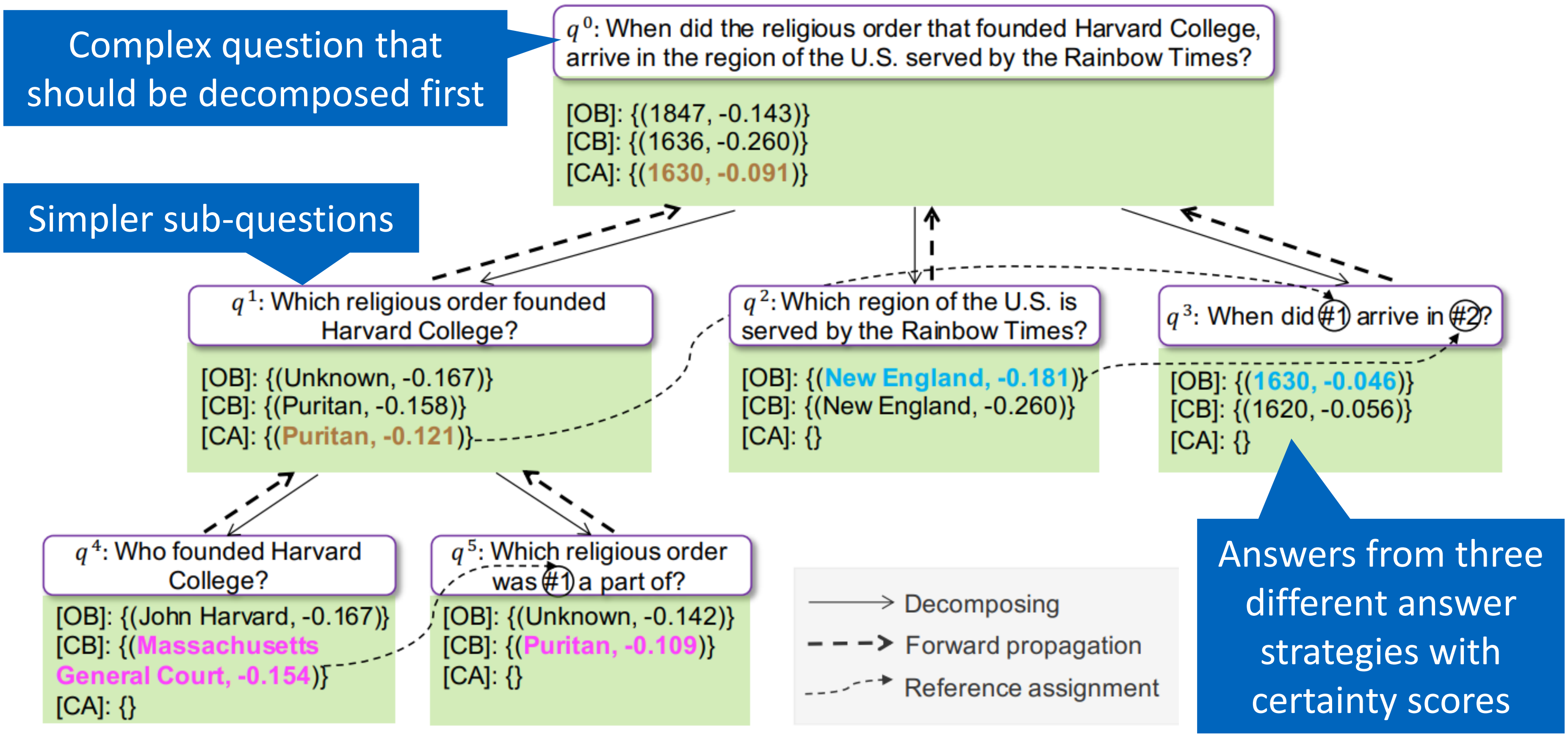
ToT on the Game of 24



Conclusion: We can now solve the Game of 24 with LLMs (at least in 74% of cases) through (1) locally **exploring different continuations** within a thought process and (2) globally incorporating **planning, lookahead, and backtracking**

Source: Yao et al. (2023): Tree of Thoughts: Deliberate Problem Solving with Large Language Models, https://proceedings.neurips.cc/paper_files/paper/2023/hash/271db9922b8d1f4dd7aaef84ed5ac703-Abstract-Conference.html

Probabilistic Tree-of-Thoughts (ProbTree)

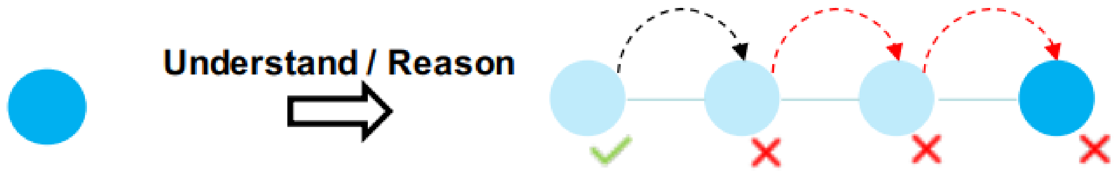


- **Open-book question answering [OB]:** Look for answers in the web *(=> future lecture)*
- **Closed-book question answering [CB]:** Let the LLM generate an answer
- **Child-aggregating question answering [CA]:** Reason about the answer by looking at the answers to the children nodes

Source: Cao et al. (2023): Probabilistic Tree-of-thought Reasoning for Answering Knowledge-intensive Complex Questions, https://arxiv.org/abs/2311.13982

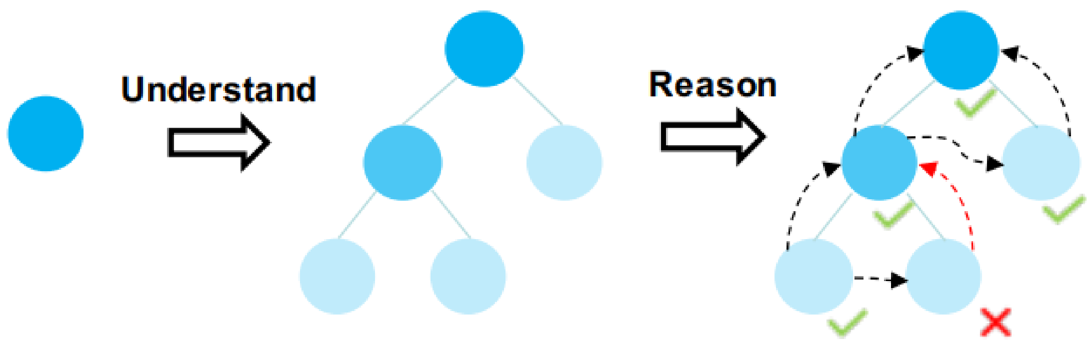
Probabilistic Tree-of-Thoughts (ProbTree)

Chain-of-Thought (CoT)



Understand and reason in the same linear thought chain (for ToT: in the same tree of thoughts)

Probabilistic Tree-of-Thoughts (ProbTree)



Perform understanding (i.e., problem decomposition) and reasoning sequentially

ToT vs. ProbTree

- In ToT, the thought decomposition is defined by the user. In ProbTree, the **thought decomposition is proposed by the LLM**
  - **ToT** used LLMs and tree structures to **explore potential solutions**
  - **ProbTree** uses LLMs and tree structures to **decompose the problem**
- In ToT, errors can be corrected through self-evaluation, pruning and backtracking. In ProbTree, **errors can be corrected through multiple answer strategies with different certainty values**

Source: Cao et al. (2023): Probabilistic Tree-of-thought Reasoning for Answering Knowledge-intensive Complex Questions, https://arxiv.org/abs/2311.13982

Final Words: Evaluating Explanations

- **Evaluating model explanations** is one of the most challenging aspects in Explainable AI (XAI)
- Often, we face a **dilemma between plausibility and faithfulness**:

Plausibility to humans:

Whether the explanations look convincing and/or align with a human rationale

↔

Faithfulness to the model:

Whether the explanations truly reflect the model's reasoning

- Plausibility and faithfulness **can be highly uncorrelated**
- XAI methods based on **prompting techniques** are a novel paradigm that provides a new answer to the plausibility-faithfulness dilemma:
  - We now **let the model explain its reasoning to itself**
  - The quality of the model's reasoning depends on the quality of its explanations. **Better explanations result in better outputs**
  - Plausibility and faithfulness become **highly correlated**

References

[1] Dziri et al. (2023): Limits of Transformers on Compositionality

[2] Zhang et al. (2023): Survey on Hallucination in Large Language Models

[3] Bills et al. (2023): Language Models Can Explain Neurons in Language Models

[4] Lundberg et al. (2017): SHAP

[5] Wei et al. (2022): Chain-of-Thought Prompting

[6] Yao et al. (2023): Tree of Thoughts Prompting

[7] Cao et al. (2023): Probabilistic Tree-of-Thought

Study Approach

Minimal

- Work with the slides

Standard

- Minimal approach + read through references 6 (ToT) and 7 (ProbTree)

In-Depth

- Standard approach + read through references 1 (Multi-Step Reasoning) and 5 (CoT) + skim through the remaining references

See you next time!