Faithful and Useful Explanations by Large Language Models

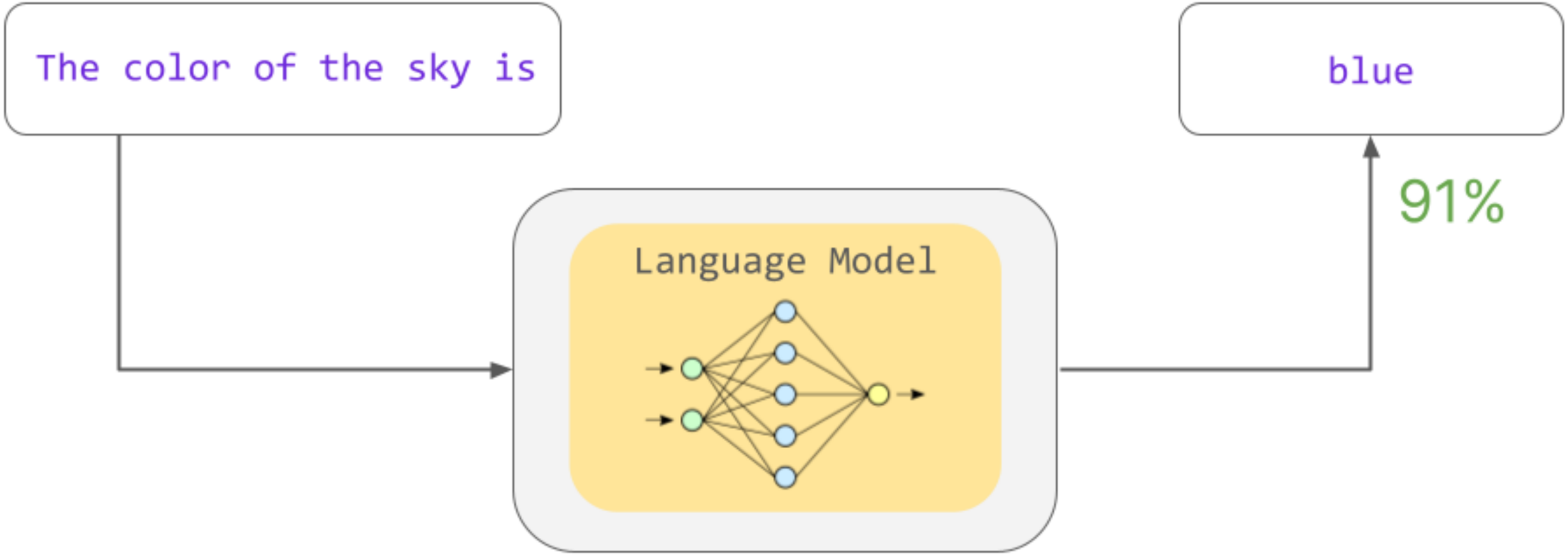
PhD Dissertation Defense - Veronica Qing Lyu

Advisors: Chris Callison-Burch, Marianna Apidianaki

Committee: Dan Roth (Chair), Charles Yang, Lyle Ungar, Ellie Pavlick

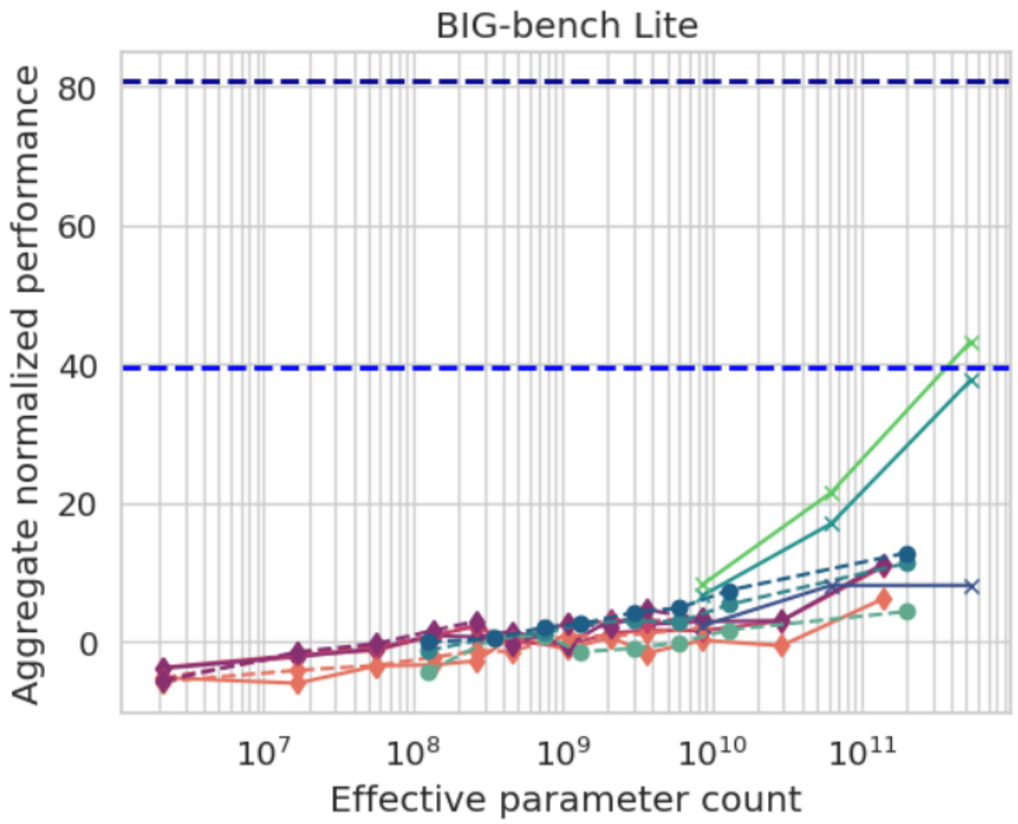
Aug 29 2024

Large Language Models (LLMs)

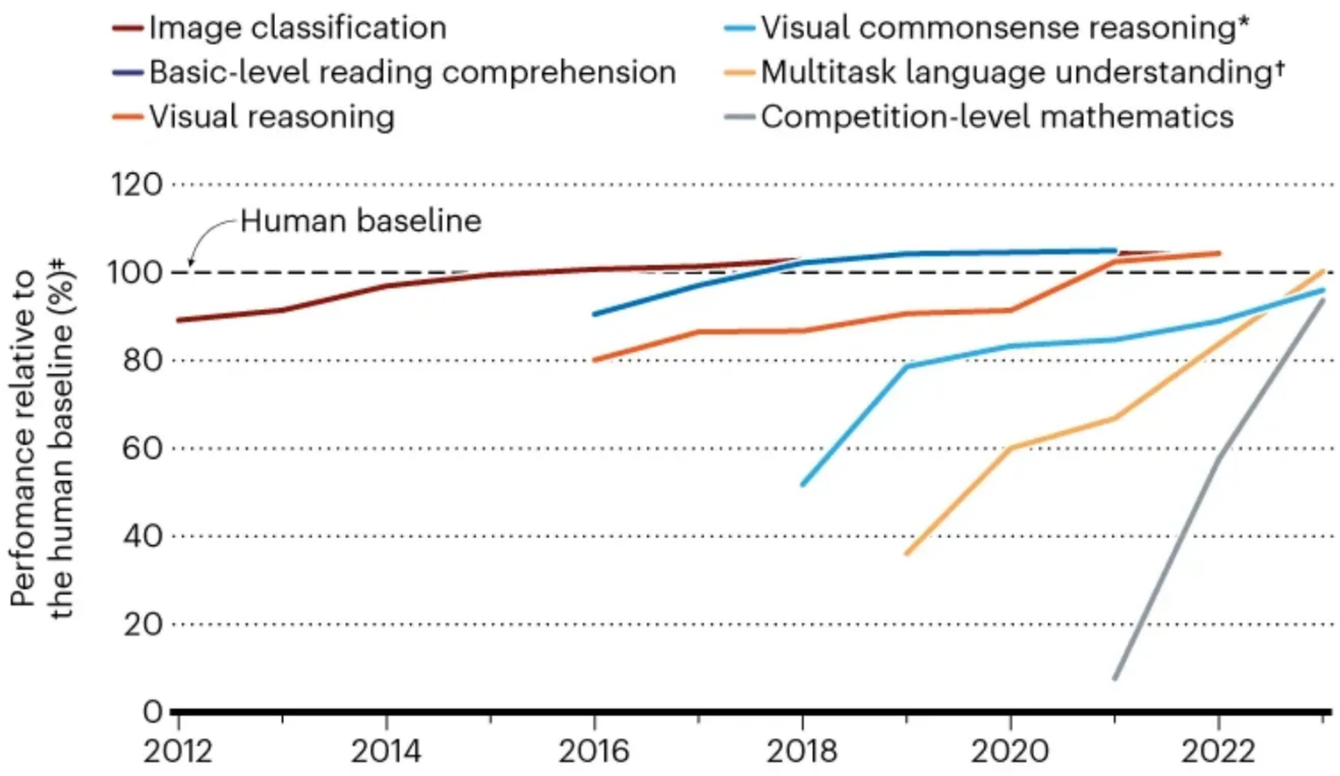


Claude

LLMs Show Impressive Capabilities



Source: [LLM Benchmarks in 2024]



Source: [Nature article]

… But They Are Hard to Interpret

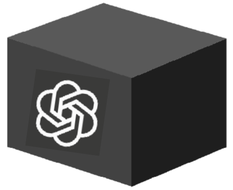
HR

Should I interview this candidate?

Is this tumor malignant?

Why?

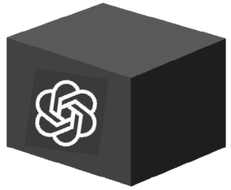
No



Judge

Will this criminal reoffend in the future?

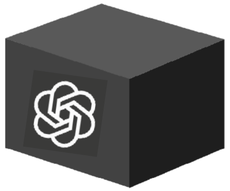
Yes



Information Analyst

Is this fake news?

Yes



What Is Interpretability?

- **WHAT** knowledge does the model encode?
- **WHY** does the model make certain predictions?

See (Lyu et al. NAACL'22) (Lyu et al. *SEM'23)

★ This thesis

"The extent to which the internal mechanics of a model can be presented in understandable terms to a human."

(Lipton 2017; Murdoch et al. 2019; Barredo Arrieta et al. 2020)

Why is Interpretability Important?

Ethics

Safety

Accountability

Bias

Trustworthiness

Illustration created by DALLE-3

Existing Interpretability Methods

Method	Time	Model accessibility	Scope	Unit of explanation	Form of explanation
Similarity methods	post-hoc	white-box	local	examples, concepts	importance scores
Analysis of model-internal structures	post-hoc	white-box	local, global	features, interactions	visualization, importance scores
Backpropagation -based methods	post-hoc	white-box	local	features, interactions	visualization, importance scores
Counterfactual intervention	post-hoc	black-box, white-box	local, global	features, examples, concepts	importance scores
Self-explanatory models	built-in	white-box	local, global	features, examples, concepts	importance scores, natural language, causal graphs

Challenges in the LLM era

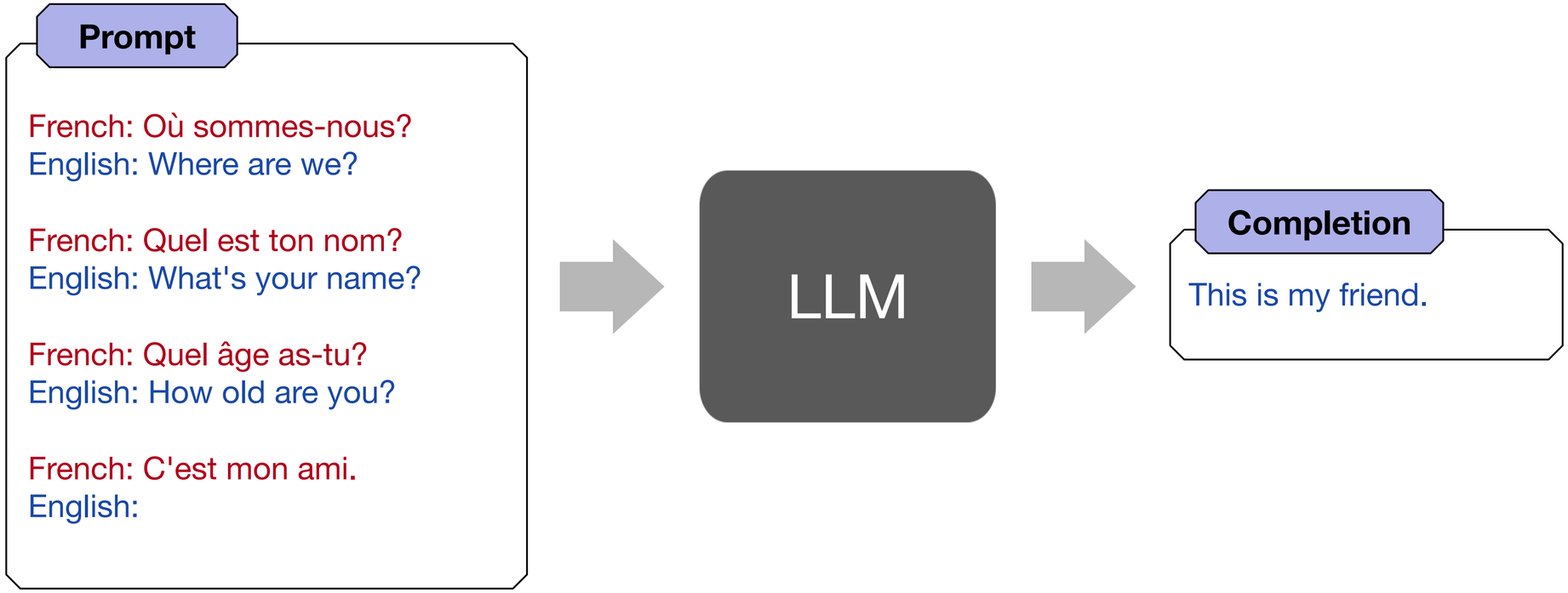
- Lack of transparency
  - Training data
  - Weights
  - Probability logits
- Computational cost
  - Time
  - Memory

Can LLMs be self-explanatory?

Towards Faithful Model Explanation in NLP: A Survey (Lyu et al., 2023)

Using LLMs to Generate Explanations for Themselves

- (In-context) few-shot learning ability of LLMs



LLM-Generated Explanations

- Few-shot learning applied to explanation generation

Chain-of-Thought Prompting (CoT)

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had 23 - 20 = 3. They bought 6 more apples, so they have 3 + 6 = 9. The answer is 9. ✓

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?

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. ✗

🤔 Are these good explanations?

(Wei et al. 2022)

Limitation 1: Faithfulness (aka. fidelity, reliability)

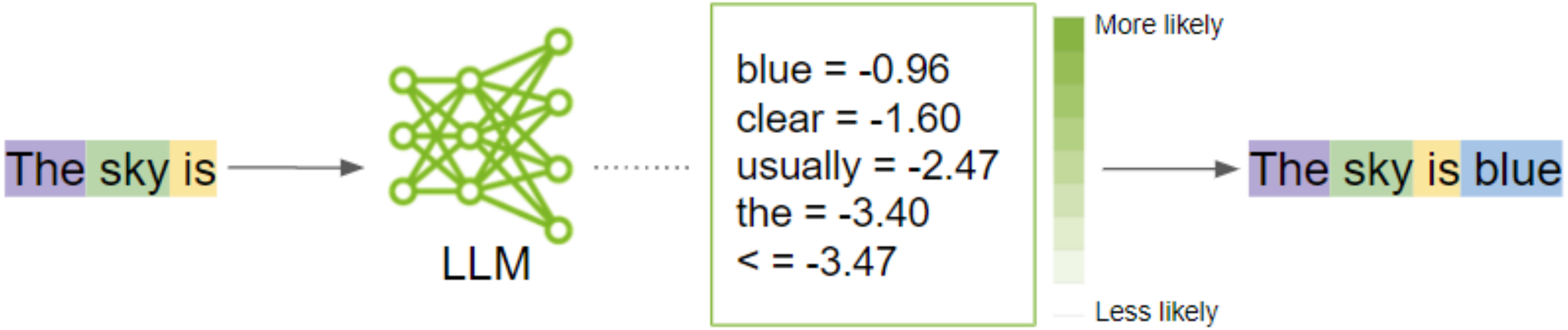
i.e., it can't lie

An explanation should accurately reflect the reasoning process behind the model's prediction.

(Harrington et al. 1985 ; Ribeiro, Singh, and Guestrin 2016 ; Jacovi and Goldberg 2020)

LLM-Generated Explanations Aren't Necessarily Faithful

- Auto-regressive nature of LLMs:



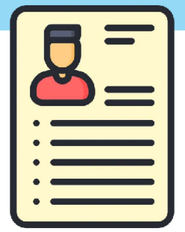
- "LMs Don't Always Say What They Think" – LLMs generate systematically unfaithful explanations (Turpin et al 2023)
- As LMs become *larger*, they produce less faithful explanations (Lanham et al 2023)

Risks of unfaithful "explanations"

- False impression of "self-interpretability"
- Easier over-trust in the model
  - especially if explanations look plausible

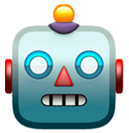
HR

Should I hire this candidate?



Generated CoT

Based on their excellent education background and strong technical skills, I highly recommend hiring this candidate



True Reasoning

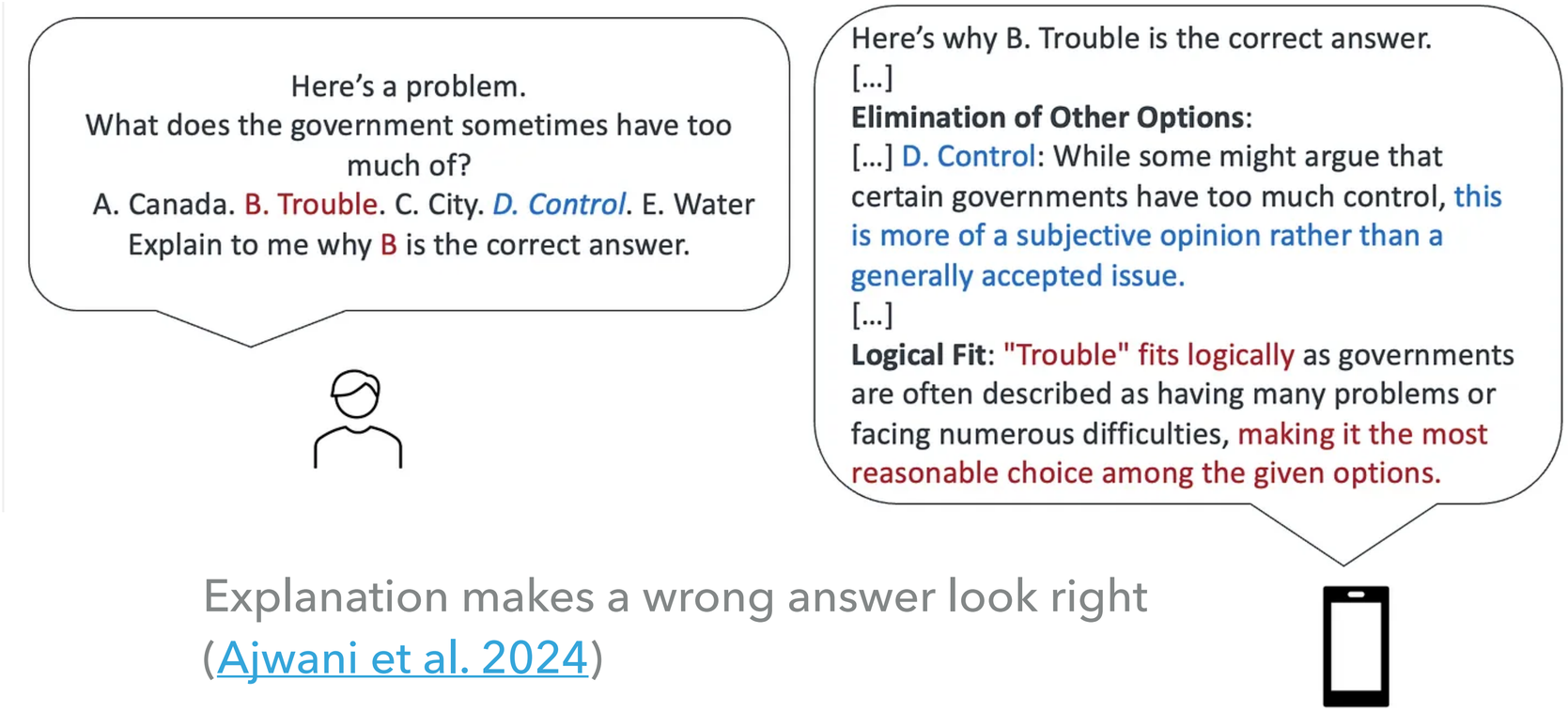
Their name looks like a white male, so I highly recommend hiring this candidate

Limitation 2: Utility (aka. usefulness)

An explanation should be helpful for a predefined goal in real-life application scenarios.

- Scientific understanding
- Model debugging
- Model auditing
- Decision support
- Knowledge discovery
- ...

LLM-Generated Explanations Can Be Adversarially Useful



Explanation makes a wrong answer look right (Ajwani et al. 2024)

- The presence (regardless of the content) of explanations can lead to easier over-trust in AI decisions (Bansal et al. 2021)

Thesis Statement

This thesis argues that LLM-generated explanations should be both faithful and useful. We demonstrate that integrating symbolic reasoning into end-to-end NLP workflows can enhance the faithfulness of explanations while additionally providing performance gains across multiple reasoning tasks. In terms of utility, we show that such explanations, when used as an exposure interface or training signals, can improve the calibration, robustness, and self-correction capabilities of LLMs.

Work Discussed in This Thesis

Primary work

- [1] Towards Faithful Model Explanation in NLP: A Survey.
  Qing Lyu, Marianna Apidianaki, and Chris Callison-Burch. To Appear in Computational Linguistics 2024.
- [2] Faithful Chain-of-Thought Reasoning.
  Qing Lyu*, Shreya Havaldar*, Adam Stein*, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. IJCNLP-AACL 2023.
- [3] Calibrating Large Language Models with Sample Consistency.
  Qing Lyu*, Kumar Shridhar*, Chaitanya Malaviya, Li Zhang, Yanai Elazar, Niket Tandon, Marianna Apidianaki, Mrinmaya Sachan, Chris Callison-Burch. In submission.

Secondary work

- [4] Explanation-based Finetuning Makes Models More Robust to Spurious Cues.
  Josh Magnus Ludan, Yixuan Meng*, Tai Nguyen*, Saurabh Shah*, **Qing Lyu**, Marianna Apidianaki, and Chris Callison-Burch. ACL 2023.

Outline

Faithfulness

- Faithful Explanations with Symbolic Reasoning

Usefulness

- Using Explanations for Confidence Calibration
- Using Explanations for Self-Correction

during in-context learning

- Using Explanations for Robustness Enhancement – during finetuning

Conclusion and Future Directions

Outline

Faithfulness

- Faithful Explanations with Symbolic Reasoning

Usefulness

- Using Explanations for Confidence Calibration
- Using Explanations for Self-Correction

during in-context learning

- Using Explanations for Robustness Enhancement – during finetuning

Conclusion and Future Directions

Spoiler⚠️: Main takeaway

- The integration of symbolic reasoning can improve the faithfulness of LLM-generated explanations while additionally improve end-task performance.

CoT explanations Aren't Necessarily Faithful

Input

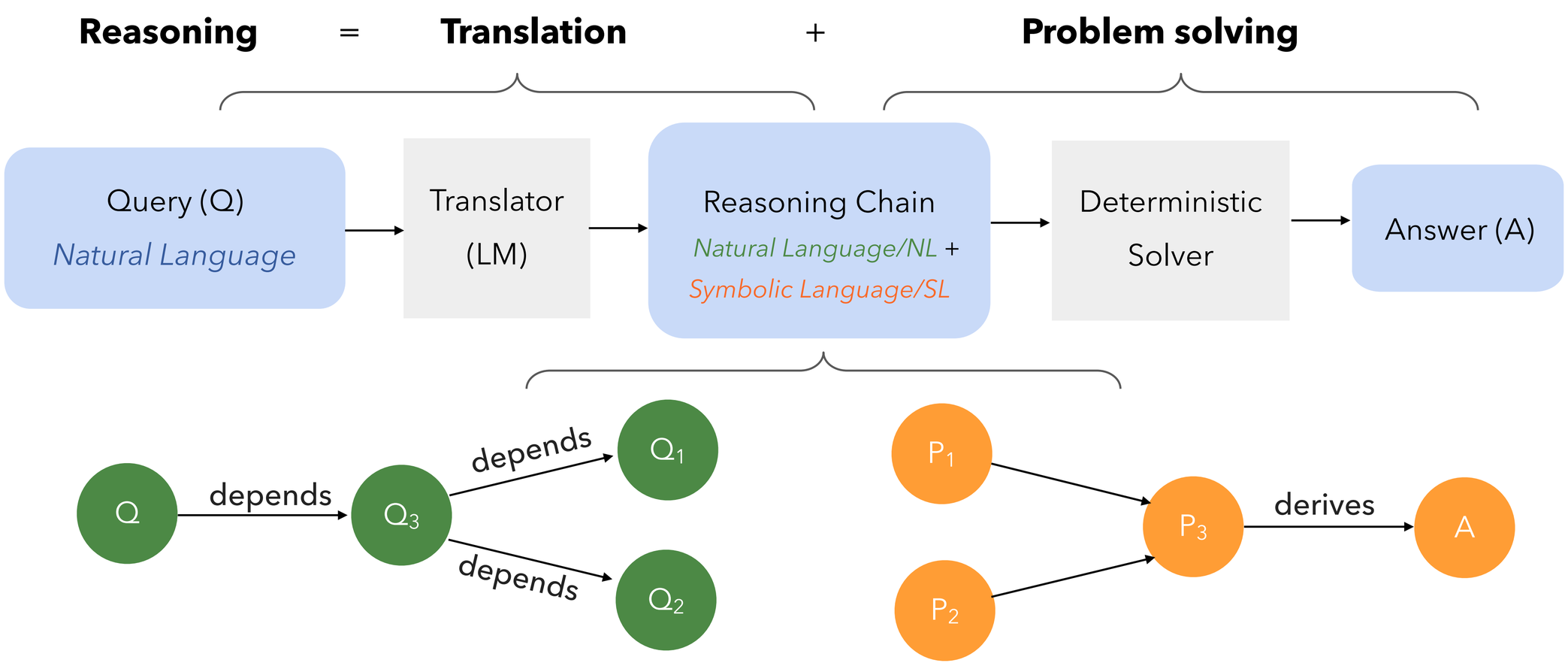
Q: John plans to sell all his toys and use the money to buy video games. He has 13 lego sets and he sells them for $15 each. He ends up buying 8 videogames for $20 each and has $5 left. How many lego sets does he still have?

CoT output

A: He started with 13 lego sets. He sold them for $15 each. So he must have gotten 13 * $15 = $195 from selling them. He used this money to buy 8 video games for $20 each. This means he spent 8 * $20 = $160 on the video games. He had $5 left, so in total he must have had $195 + $5 = $200. The answer is 0.

An example of unfaithful CoT output on GSM8K. The answer does not follow from the reasoning chain.

Our Approach: Faithful CoT



- Faithfulness by construction:
  The answer is the result of deterministically executing the reasoning chain.

Task: Math Word Problems (MWP)

Reasoning = Translation + Problem solving

If there are 3 cars in the parking lot and 2 more cars arrive, how many cars are in the parking lot?

Query

I M →

```
# 1. How many cars are there in the beginning? (independent,
support: ["there are 3 cars in the parking lot"])
n_cars_begin = 3
# 2. How many cars arrive? (independent, support: ["2 more cars
arrive"])
n_cars_arrive = 2
# 3. Final answer: How many cars are in the parking lot? (depends on
1, 2)
n_cars_total = n_cars_begin + n_cars_arrive
```

Reasoning Chain

(NL + Python)

Python Interpreter →

5

Answer

Task: Multi-hop QA

Reasoning = Translation + Problem solving

Would an apple sink in water?

I M →

```
// 1. What is the density of an apple?
// The density of an apple is about 0.75 g/cm^3.
// 2. What is the density of water?
// Water has a density of 1 g/cm^3.
// Then, we represent these answers in Datalog:
// 1. The density of an apple is about 0.75 g/cm^3.
.decl Has_density(Object:symbol, Density:float)
Has_density("apple", 0.75).
// 2. Water has a density of 1 g/cm^3.
Has_density("water", 1).
// Now, we derive the final answer: Would an apple sink in water?
// The answer is Yes only if an apple is more dense than water.
    .decl Answer()
     Answer() :- Has_density("apple", density1), Has_density("water",
     density2), density1 > density2.
    .output Answer
```

Datalog Interpreter → **No**

Query

Reasoning Chain

(NL + Datalog)

Answer

Task: Relational Inference

Reasoning = Translation + Problem solving

[Gabrielle] drove her daughter [Dorothy] to the hospital. [Dorothy]'s son [Vincent] showed up shortly after. How is [Vincent] related to [Gabrielle]?

Query

I M →

```
# 1. How is [Vincent] related to [Dorothy]? (independent, support: "
[Dorothy]'s son [Vincent] showed up shortly after.")
relation(Vincent, Dorothy) = son
# 2. How is [Dorothy] related to [Gabrielle]? (independent, support:
"[Gabrielle] drove her daughter [Dorothy] to the hospital.")
relation(Dorothy, Gabrielle) = daughter
# 3. Final answer: How is [Vincent] related to [Gabrielle]? (depends
on 1, 2)
relation(Vincent, Gabrielle) = relation(Vincent, Dorothy) @
                               relation(Dorothy, Gabrielle)
```

Reasoning Chain

(NL + Relation expression)

→ Rule-based inference engine → **grandson**

Answer

Task: Planning

Reasoning = Translation + Problem solving

I spilled my coke on the table, could you throw it away and bring something to clean with?

Query

I M →

Goal in PDDL:

```
(:goal
    ; I need to do both things
    (and
        ; first throw away the coke
        (at coke trash)
        ; then bring the user a sponge as a cleaning tool
        (at sponge user)
    )
)
```

Reasoning Chain

(NL + Planning Domain Definition Language (PDDL))

PDDL planner →

1. find(coke)
2. pick(coke)
3. find(trash)
4. put(coke)
5. find(sponge)
6. pick(sponge)
7. find(user)
8. put(sponge)

Answer

Experimental Setup: Baselines

Query

There are 15 trees in the grove. Grove workers will plant trees in the grove today. After they are done, there will be 21 trees. How many trees did the grove workers plant today?

Standard

Output

Answer: 6

Chain-of-Thought (CoT)

Output

We start with 15 trees.

Later we have 21 trees.

The difference must be the number of trees they planted.

So, they must have planted 21 - 15 = 6 trees.

The answer is 6.

Least-to-Most (LtM)

Output

Q1. How many trees are there in the beginning?
There are 15 trees.
Q2. How many trees will there be after the grove workers plant trees?
There are 21 trees.
Final Answer: How many trees did the grove workers plant today?
They planted 21 - 15 = 6 trees. The answer is 6.

Program-of-Thought (PoT)*

Output

```
trees_begin = 15
trees_end = 21
trees_today = trees_end - trees_begin
answer = trees_today
```

>>> >>> Answer: 6

Python Interpreter

Faithful Chain-of-Thought (FCoT) (Ours)

Output

```
# 1. How many trees are there in the beginning? (independent, support: ["There are 15 trees"])
trees_begin = 15

# 2. How many trees are there in the end? (independent, support: ["there will be 21 trees"])
trees_end = 21

# 3. How many trees did the grove workers plant today? (depends on 1 and 2, support: [])
trees_today = trees_end - trees_begin

# 4. Final Answer: How many trees did the grove workers plant today? (depends on 3, support: [])
answer = trees_today
```

>>> >>> Answer: 6

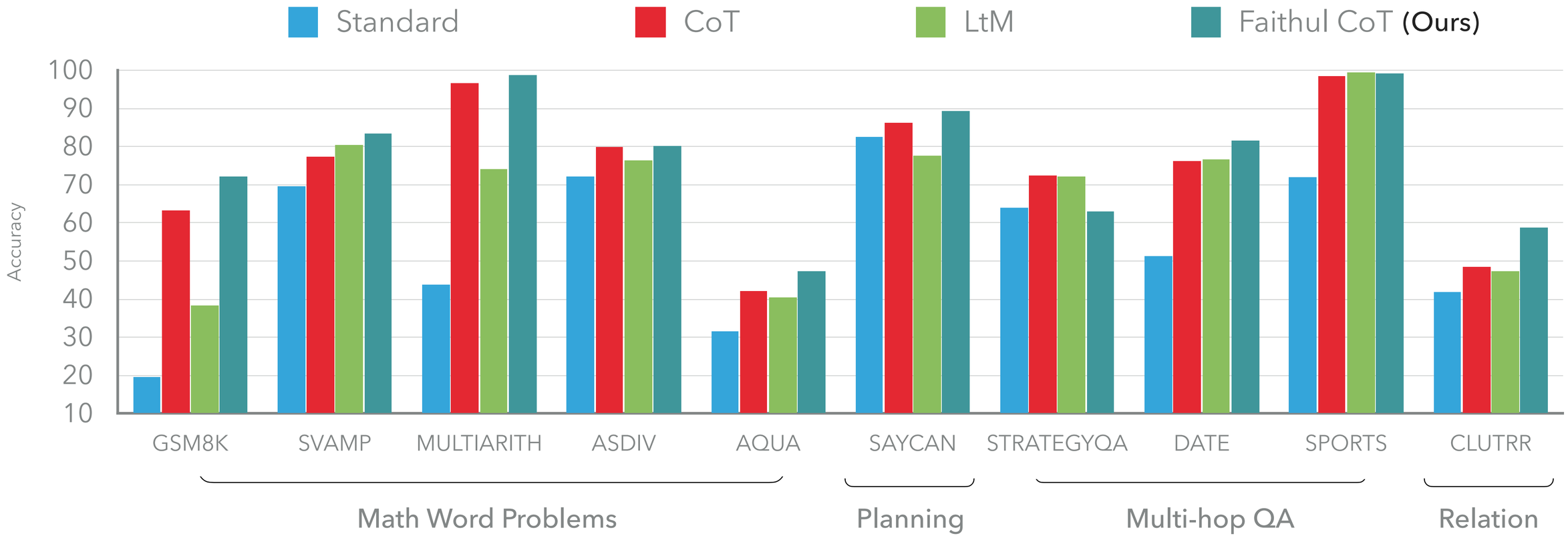
Python Interpreter

*Or PAL / Program-Aided LM (Gao et al. 2023)

Experimental Setup

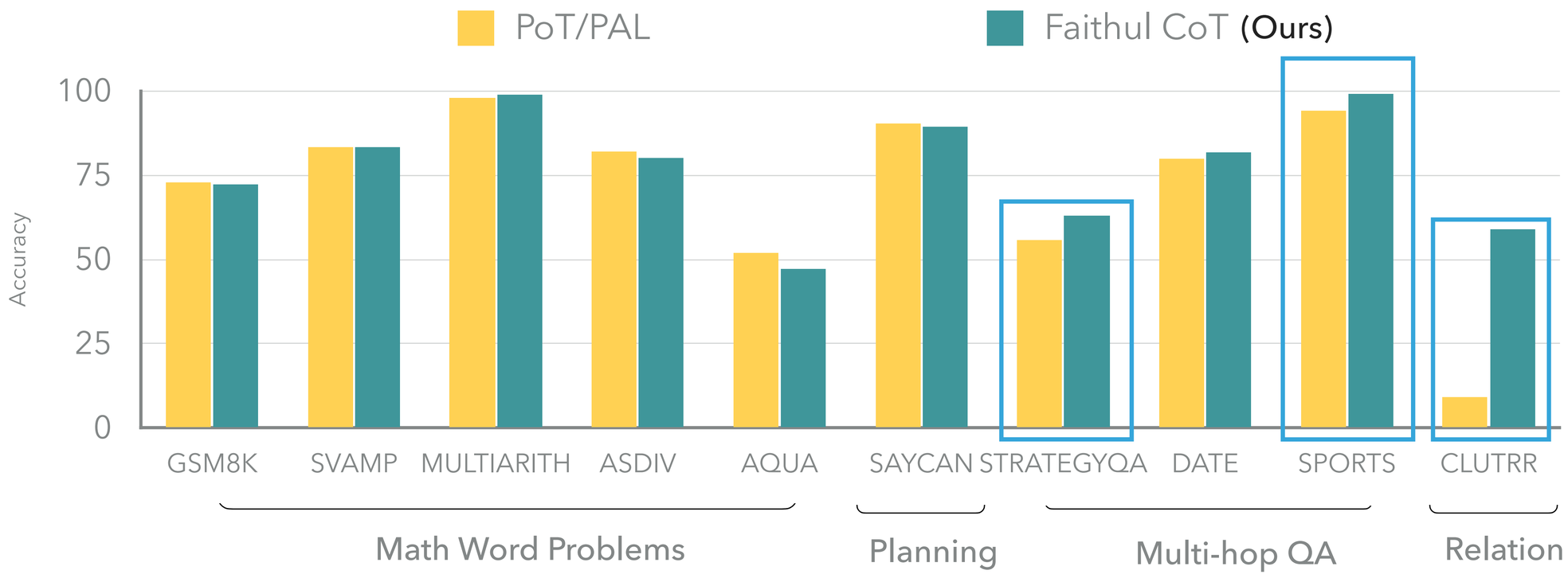
- **Evaluation Metrics**: Accuracy of the final answer
- **LMs**: GPT-3 (text-davinci-002), Codex (code-davinci-001/002), ChatGPT (gpt-3.5-turbo), GPT-4 (gpt-4)
- **Datasets**:
  - **Math Word Problems** (MWP): GSM8K (Cobbe et al., 2021), SVAMP (Patel et al., 2021), MultiArith (Roy & Roth, 2015), ASDiv (Miao et al., 2020), AQuA (Ling et al., 2017)
  - **Multi-hop QA**: StrategyQA (Geva et al., 2021), Date Understanding (BIG-Bench collaboration, 2021), Sports Understanding (BIG-Bench collaboration, 2021)
  - **Planning**: SayCan (Ahn et al., 2022)
  - **Relational Inference**: CLUTRR (Sinha et al., 2019)
- **Num. Shots in Prompt**: 6-10, depending on the task (following Wei et al. (2022))
- **Decoding Strategy**: greedy decoding; self-consistency decoding (Wang et al., 2022)

Faithful CoT outperforms previous baselines on 8 out of 10 datasets



Answer accuracy w/ code-davinci-002 and greedy decoding

Faithful CoT outperforms concurrent PoT/PAL on knowledge-intensive tasks



Answer accuracy w/ code-davinci-002 and greedy decoding

Faithful CoT is robust to the choice of exemplars and prompt phrasing

Exemplars	GSM8K	Date	SayCan	CLUTRR
Set 0 (Table 1)	72.3	81.6	89.3	58.9
Set 1	72.6	81.3	91.3	59.0
Set 2	71.1	85.0	85.4	57.2
Set 3	72.3	82.5	88.3	58.0
Set 4	71.2	77.4	88.3	55.5
Set 5	71.5	85.0	89.3	56.0
Mean	71.8	82.1	88.7	57.4
Std	0.6	2.8	1.9	1.5

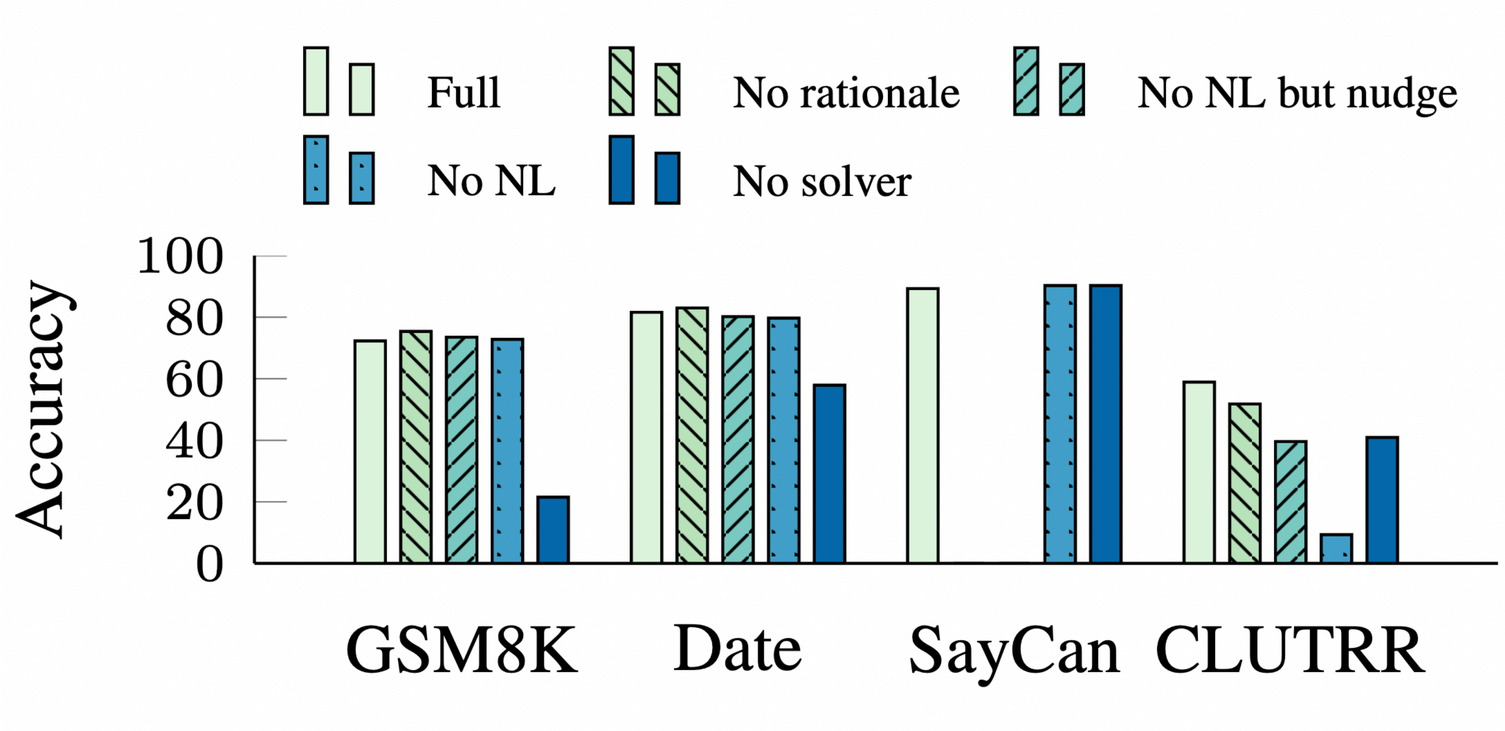
Answer accuracy w/ different sets of exemplars

Prompt	GSM8K	Date	SayCan	CLUTRR
Original	72.3	81.6	89.3	58.9
Variation 1	69.1	84.4	88.3	-
Variation 2	70.3	81.6	90.3	56.2
Variation 3	70.2	80.5	87.4	55.9
Mean	70.5	82.0	88.8	57.0
Std	1.3	1.7	1.3	1.7

Answer accuracy w/ different prompt variations

- Variation 1: Permuting the order of independent subquestions
- Variation 2: Changing variable names
- Variation 3: Changing the starting line of NL comment

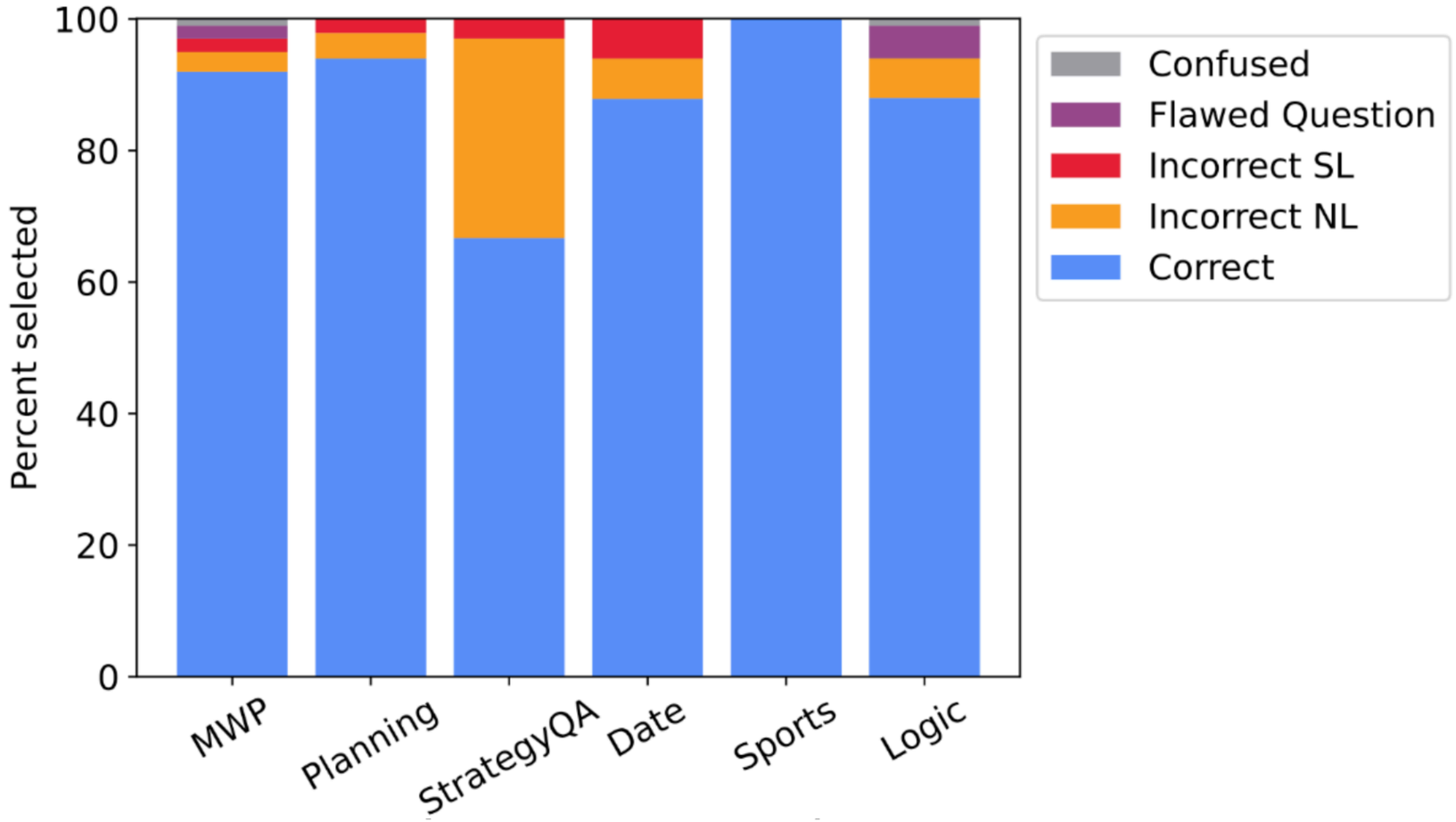
The solver is essential to performance



Answer accuracy after removing each component, w/ code-davinci-002 and greedy decoding

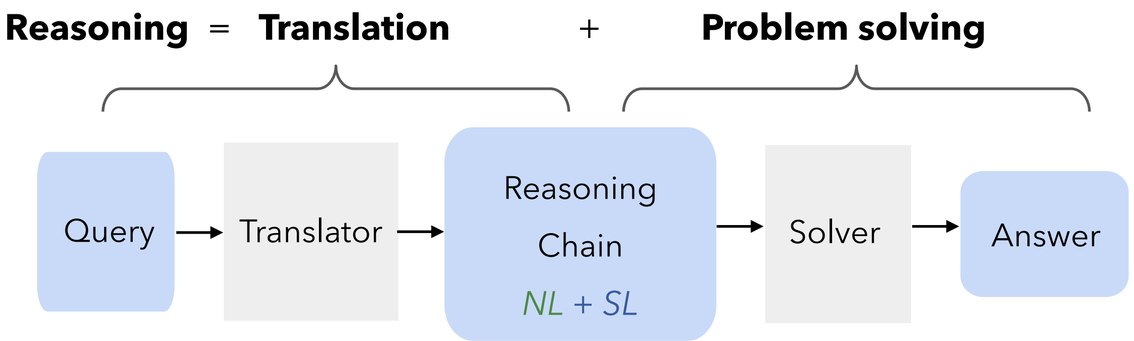
SayCan prompt doesn't include rationale / nudge line, thus no corresponding bars

When the answer is correct, the reasoning chain is valid most of the times

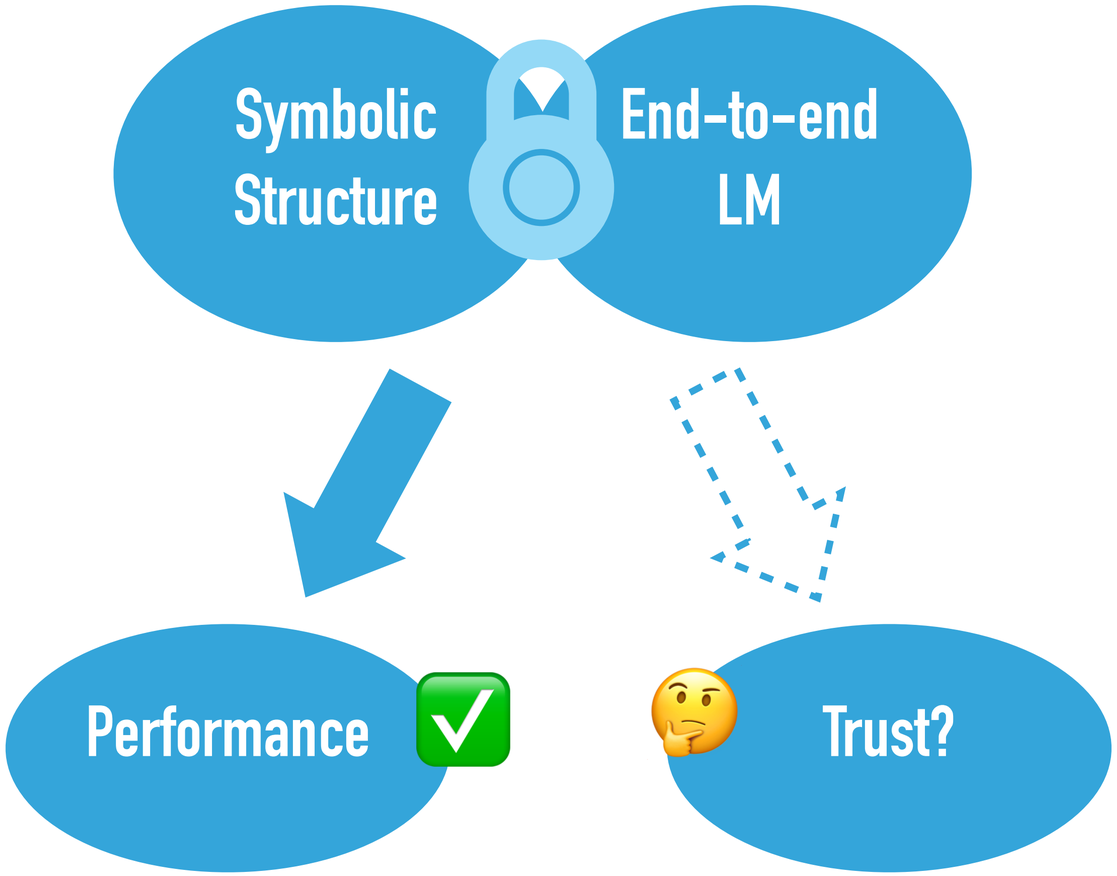


Summary of This Section

- LLM-generated explanations aren't always faithful.
- We introduce Faithful CoT, a framework that integrates stochastic translation with deterministic problem-solving.



- Aside from providing faithful explanations, our approach empirically enhances the performance on 4 diverse reasoning tasks.



Outline

Faithfulness

- Faithful Explanations with Symbolic Reasoning

Usefulness

- Using Explanations for Confidence Calibration
- Using Explanations for Self-Correction

during in-context learning

- Using Explanations for Robustness Enhancement – during finetuning

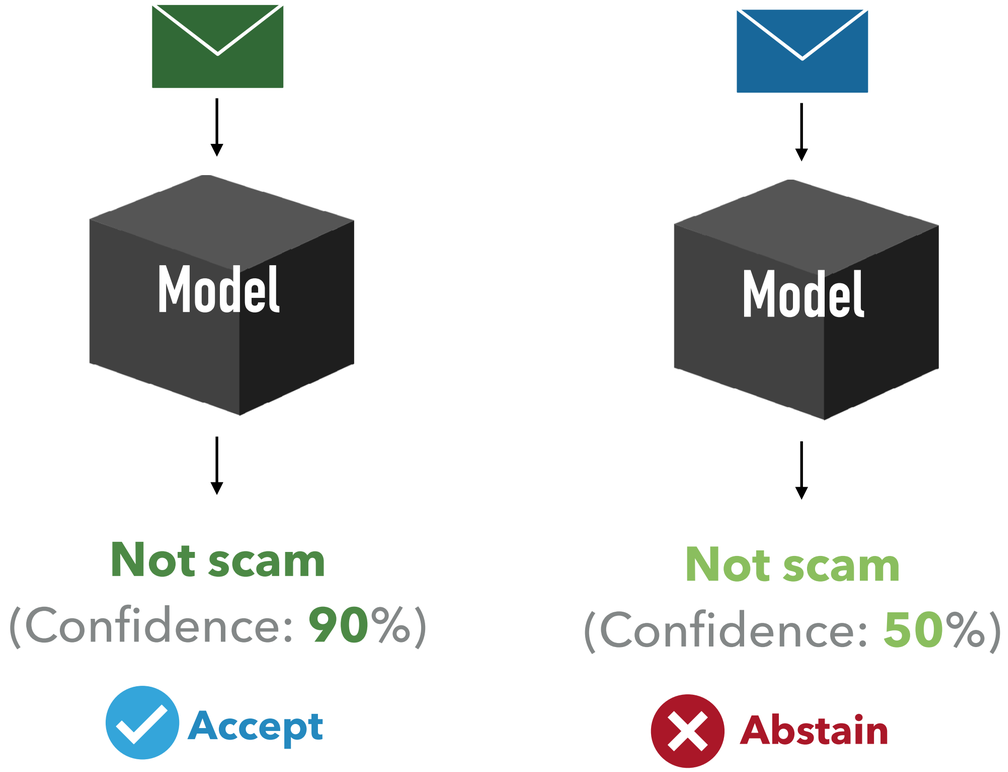
Conclusion and Future Directions

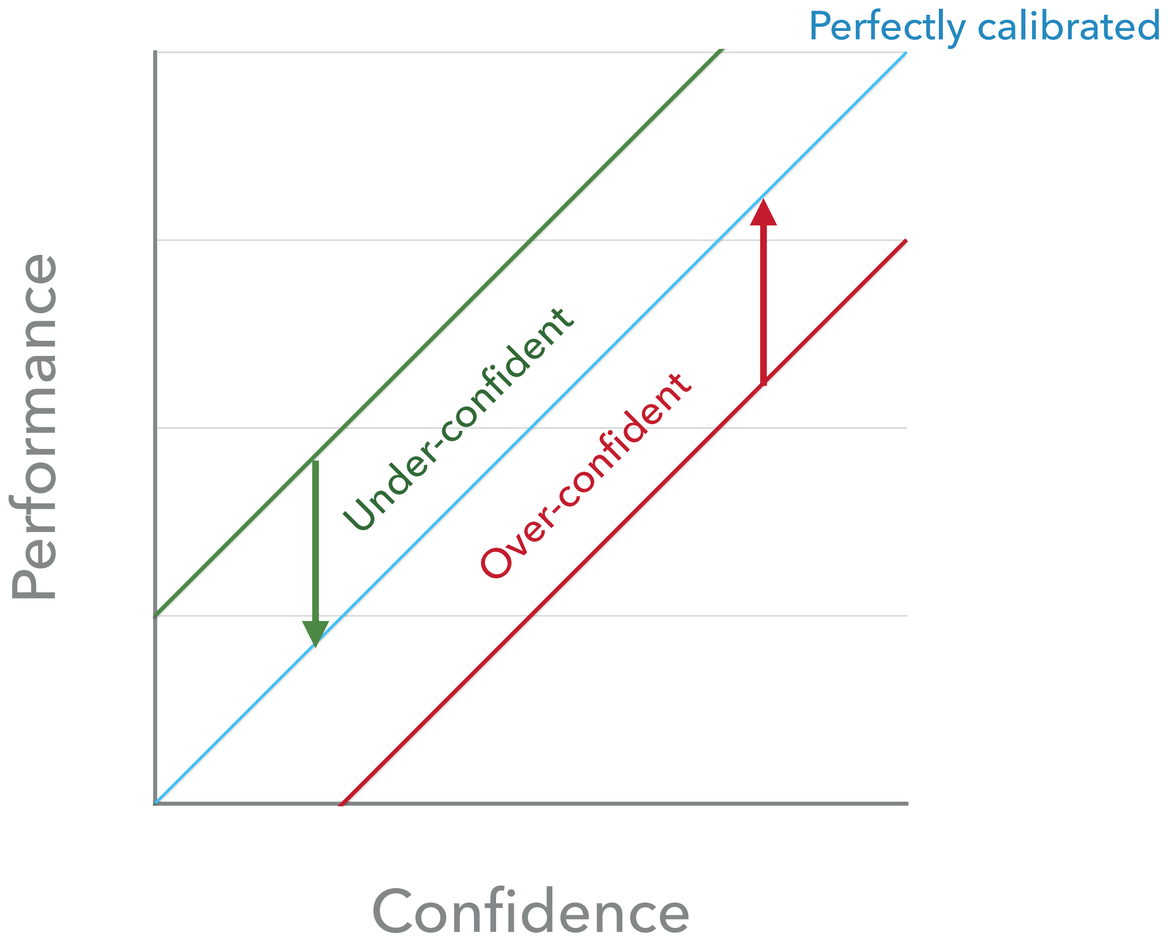
Spoiler⚠️: Main takeaway

- Explanations provide a simple way of calibrating LLMs in a post-hoc fashion.

Calibration

- LLMs excel in various tasks, but it is hard to know when they err
- When should we trust them?





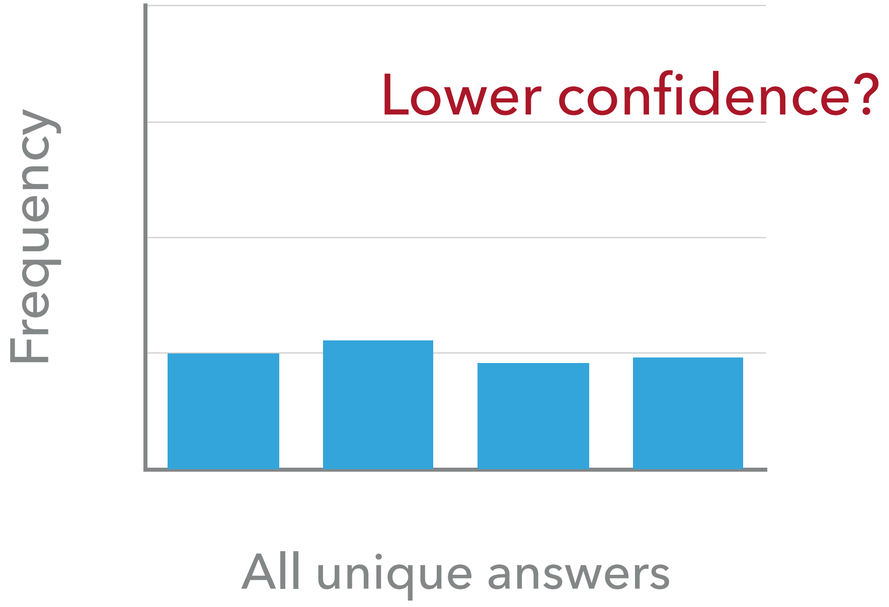
LMs Aren't Well-calibrated Off-the-shelf

- Probability logits are often poorly aligned with performance (Jiang et al. 2021; Chen et al. 2023)

- Traditional calibration methods, such as probabilistic (Guo et al. 2017), ensemble-based (Lakshminarayanan et al. 2017), and density-based approaches (Lee et al. 2018) aren't applicable to closed-source LLMs like GPT-4
  - can't access training data
  - can't access weights
  - can't retrain the model
  - can't even obtain prob logits sometimes

Can Explanations Make LMs Better Calibrated?

- Self-consistency sampling (Wang et al., 2022):
  a. Prompt LM using CoT-style prompting (CoT, PoT, LtM, FCoT …)
  b. Sample a diverse set of explanations
  c. Marginalize out the explanations and aggregate final answers
- Can we measure confidence using the "consistency" among samples?



How to Measure Consistency?

- $\mathbf{a} = \{a_1, \ldots, a_n\}$: all answers $\bar{\mathbf{a}}$: all unique answers
  a_{maj}: majority answer a_{sec}: 2nd-majority answer
- Agreement-based consistency:

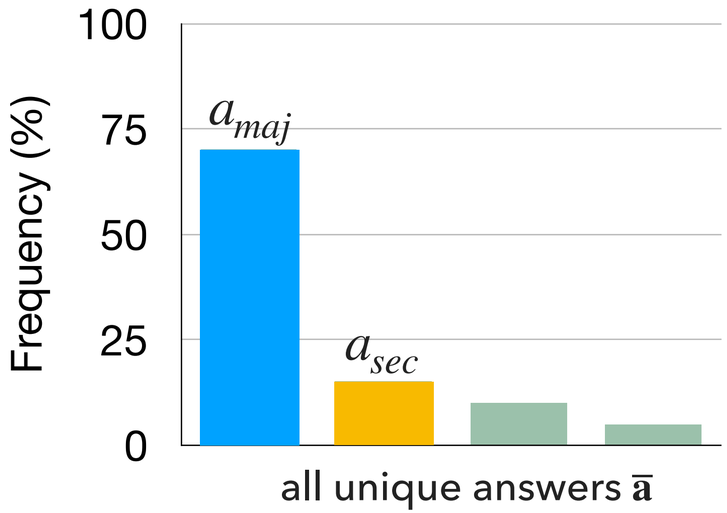
$$\text{Agree}(a_{maj}) = \sum_{i=1}^{n} \frac{|a_i = a_{maj}|}{n}$$

- Entropy-based consistency:

$$\text{Ent}(\mathbf{a}) = 1 - \left(-\frac{\sum_{i=1}^{|\bar{\mathbf{a}}|} p_i \log(p_i)}{\log(|\bar{\mathbf{a}}|)}\right)$$

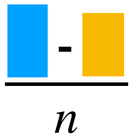
- First-second-distance (FSD)-based consistency:

$$\text{FSD}(\mathbf{a}) = \text{Agree}(a_{maj}) - \text{Agree}(a_{sec})$$



Agreement = / n **= 0.70**

Entropy = $1 - \frac{H(\mathbf{a})}{\log(|\bar{\mathbf{a}}|)}$ **= 0.36**

FSD =  / n **= 0.55**

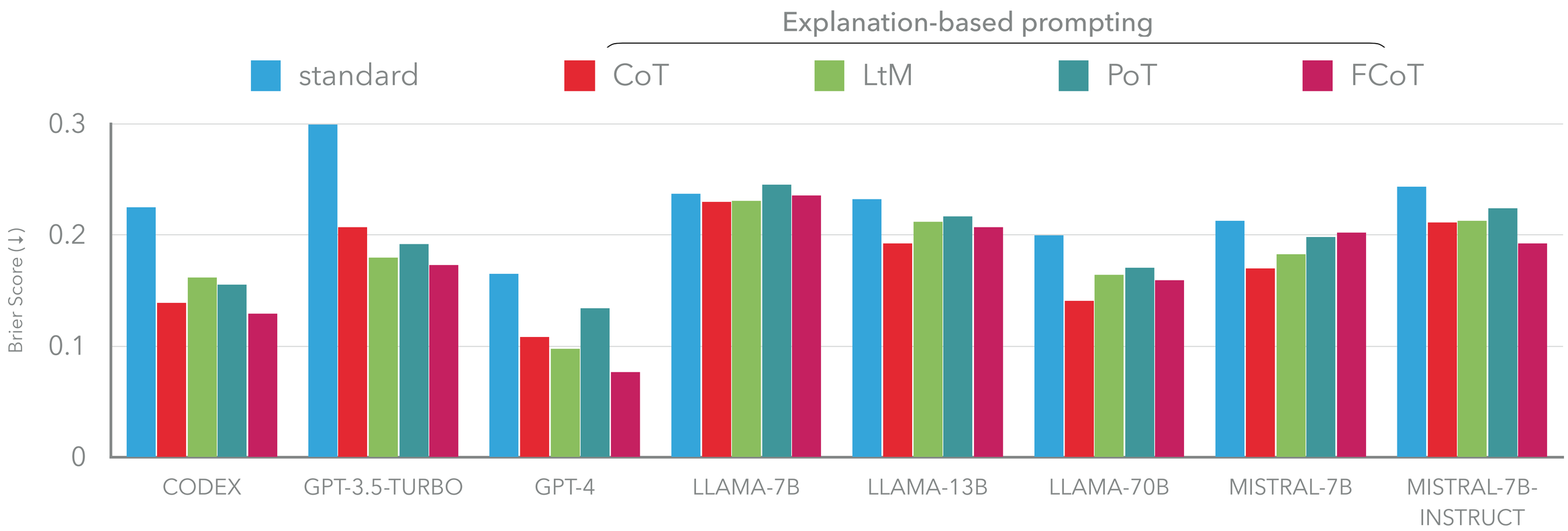
Experimental Setup

- **Evaluation Metrics** (↓): Brier score (BS, main metric), Expected Calibration Error (ECE)

$$BS = \frac{1}{N}\sum_{i=1}^{N}(\text{conf}(x_j, \hat{y}_j) - \mathbb{I}(\hat{y}_j = y_j))^2$$

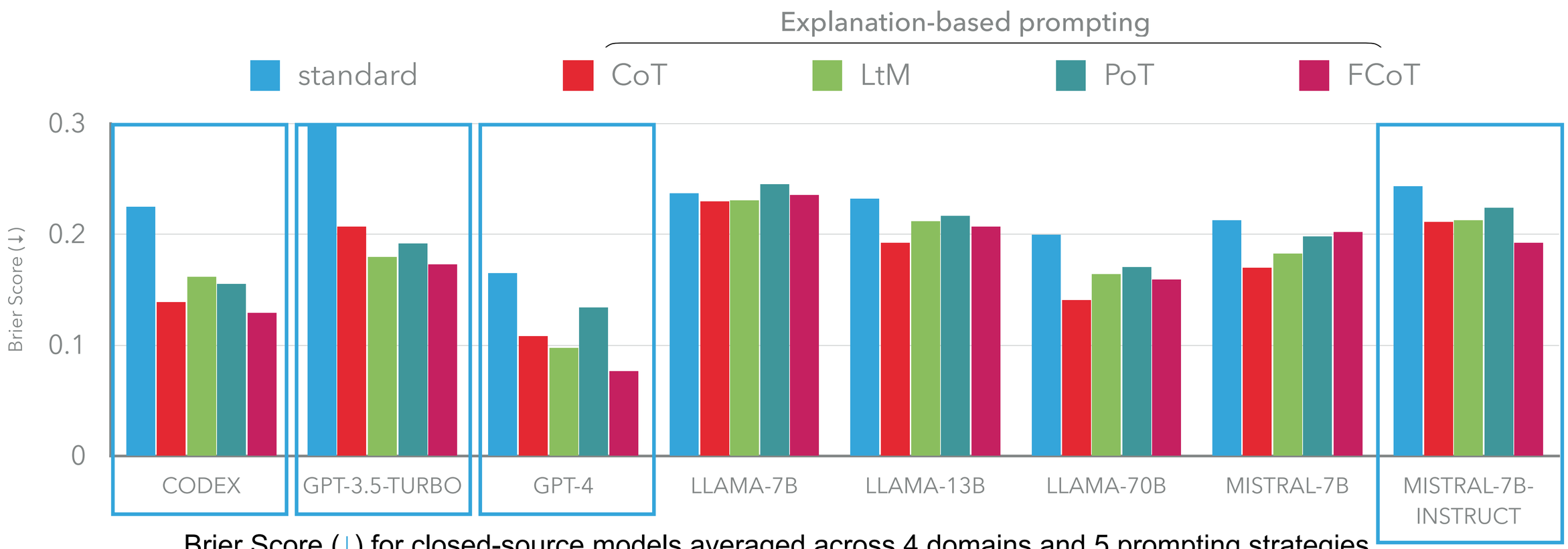
- **LMs**:
  - **Closed-source**: Codex, GPT-3.5-turbo, GPT-4
  - **Open-source**: LLAMA-7B/13B/70B, Mistral-7B, Mistral-7B-instruct
- **Datasets**: Same as Faithful CoT work
- **Decoding Strategy:** self-consistency sampling w/ n=40, T=0.4
- **Types of explanations:** Answer-only, CoT, PoT, LtM, FCoT
- **Post-hoc calibration method baselines**:
  - Raw logits
  - P(True) (Kadavath et al, 2022): 0-shot, 8-shot
  - Verbalized calibration (Tian et al 2023): verbalized percentage (verb_percent), verbalized linguistic expression (verb_ling)

Explanations make LLMs better-calibrated



Brier Score (↓) for closed-source models averaged across 4 domains and 5 prompting strategies

Instruction-tuned LMs Are Best Calibrated With FCoT; Others With CoT

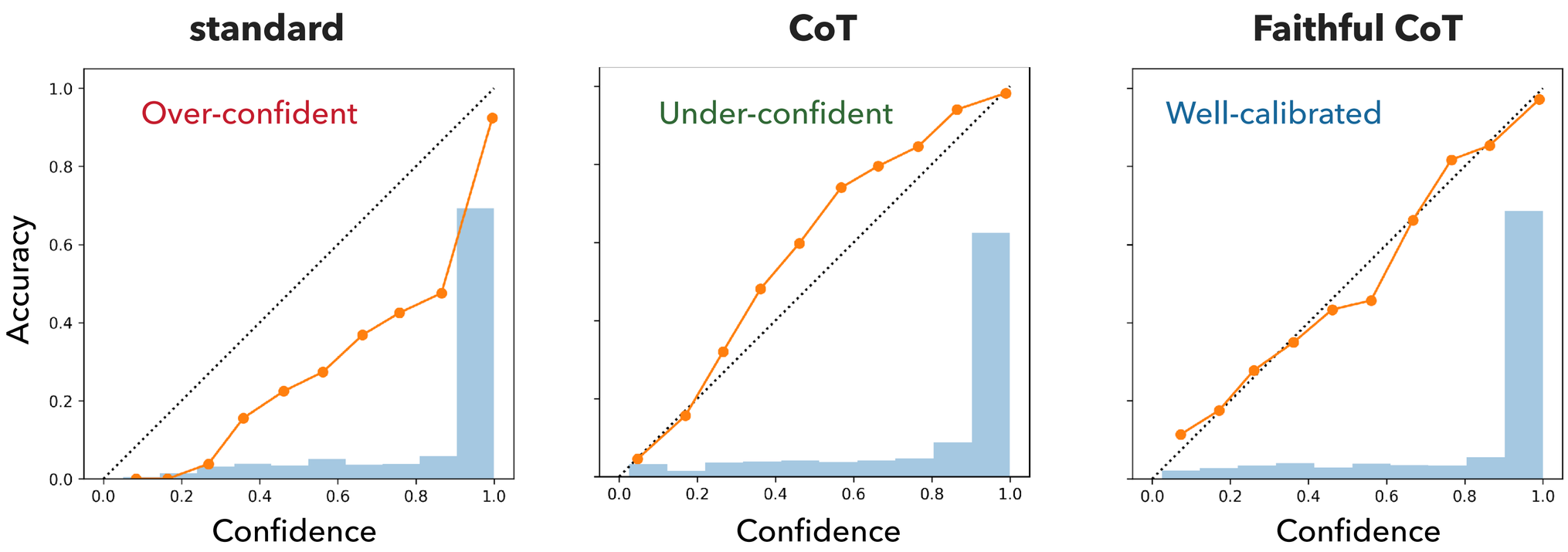


Brier Score (↓) for closed-source models averaged across 4 domains and 5 prompting strategies

Though not confirmed by OpenAI, Codex is highly likely to have been instruction-tuned (Fu et al 2023)

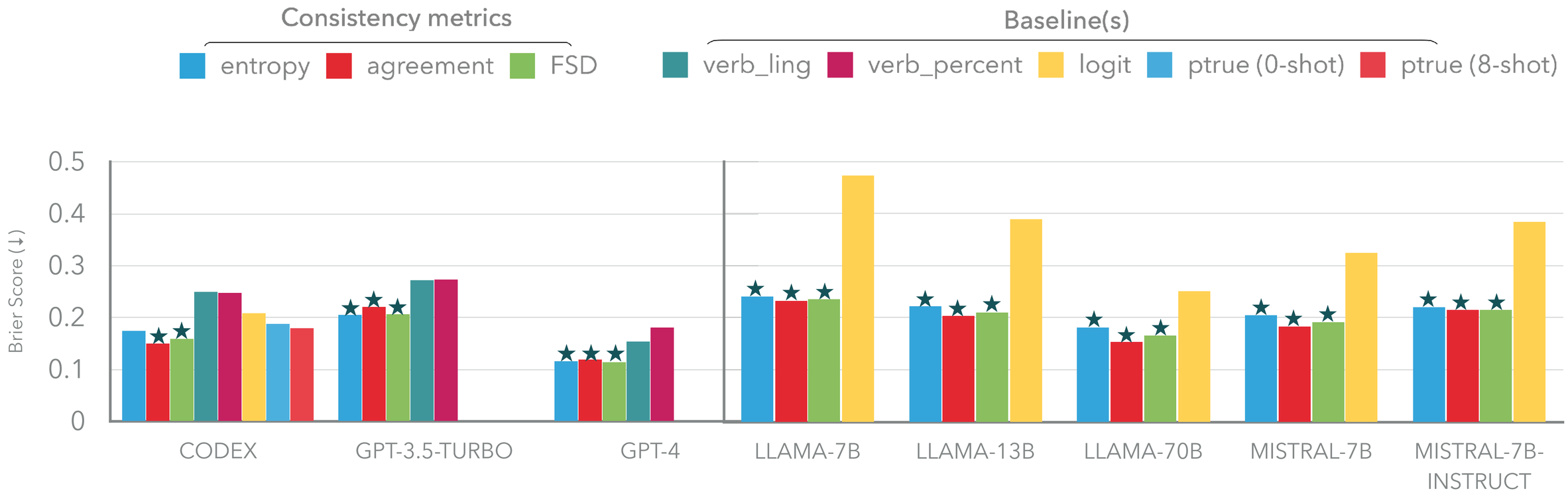
Faithful explanations make instruction-tuned models better calibrated

- Example



Calibration curves for Codex on ASDIV (agreement-based consistency)

Consistency-based calibration > existing post-hoc methods

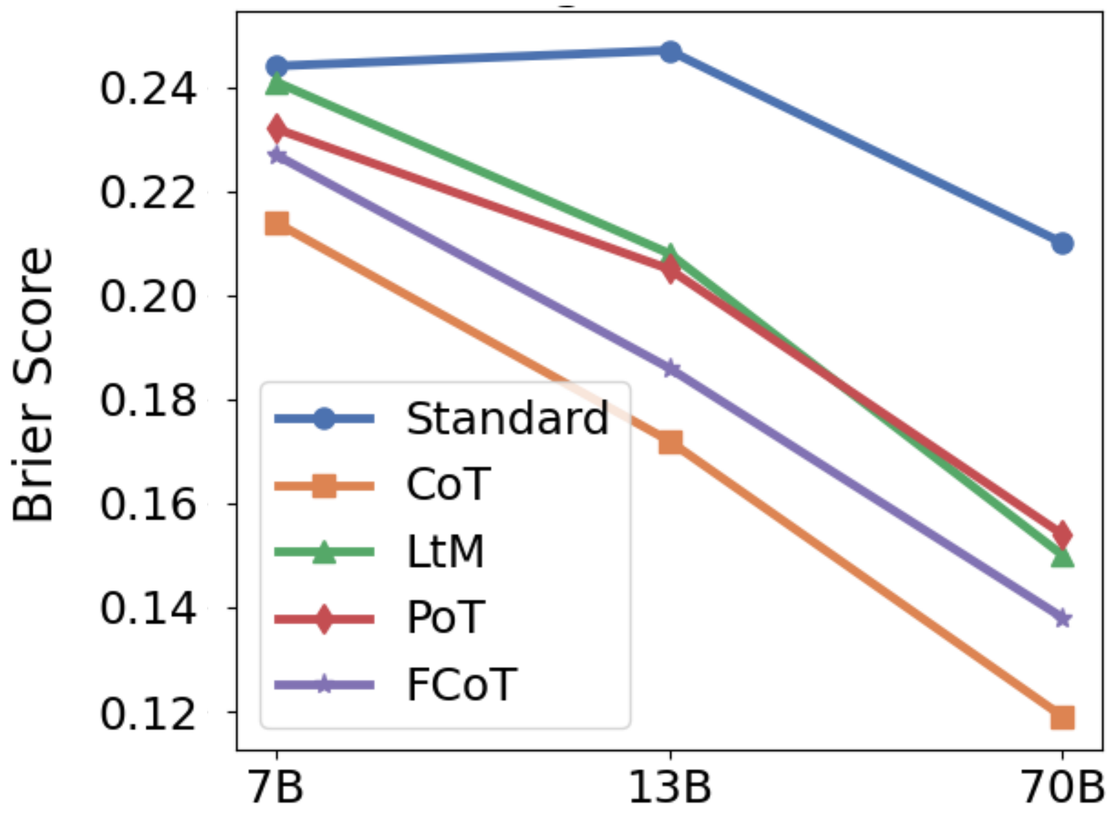


Brier Score (↓) for closed-source models averaged across 4 domains and 5 prompting strategies

★: Statistically significantly better than best baseline ($p < 0.05$)

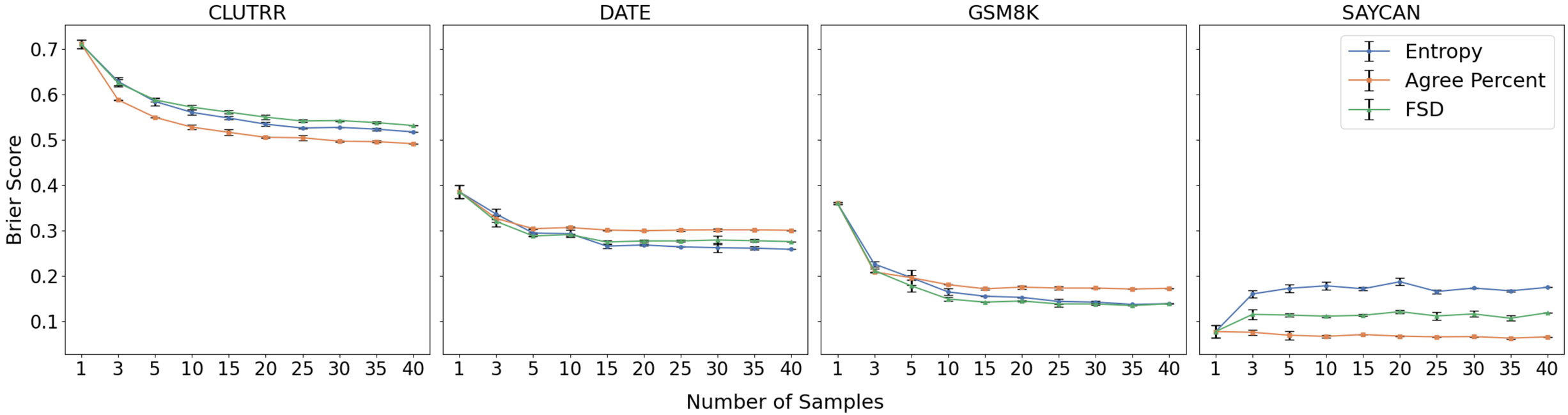
- Agreement works best for open-source LMs and Codex; FSD and entropy for the rest

Explanation Improves Calibration With Scale



Brier Score (↓) for different-sized LLaMA models using agreement-based consistency averaged across all datasets

More samples improves calibration, but 3–5 are already helpful



Brier Score (↓) for GPT-3.5-turbo using CoT prompting with an increasing number of samples

Summary of This Section

- Explanations provide a simple, post-hoc way to elicit confidence estimates from LLMs, using three measures of sample consistency.
- When prompted to generate explanations, LMs become better-calibrated.
  - This is more evident with larger LMs

Beyond calibration –
Can LMs recover from potential errors?

Outline

Faithfulness

- Faithful Explanations with Symbolic Reasoning

Usefulness

- Using Explanations for Confidence Calibration
- Using Explanations for Self-Correction

during in-context learning

- Using Explanations for Robustness Enhancement – during finetuning

Conclusion and Future Directions

Spoiler⚠️: Main takeaway

- Explanations make LLMs better at each stage of self-correction
- However, for larger overall gains on the end-task, better techniques than few-shot prompting are required

Self-correction

- Suppose calibration identifies a potential error … can LMs self-correct it?
  - Naive solution: retrain / continue finetuning
    - costly (🕒 💰)
    - sometimes even impossible

~~Open~~AI
Closed

  - More desirable solution: post-hoc correction?

Can Explanations Help With Self-Correction?

“rubber duck debugging”

Query

In a week, a tow truck pulled **10 cars** for each of the first 3 days and then **4 fewer cars** on each of the remaining 4 days. How many cars did it tow in total?

Answer-only

Output

The answer is 46.

Faithful CoT

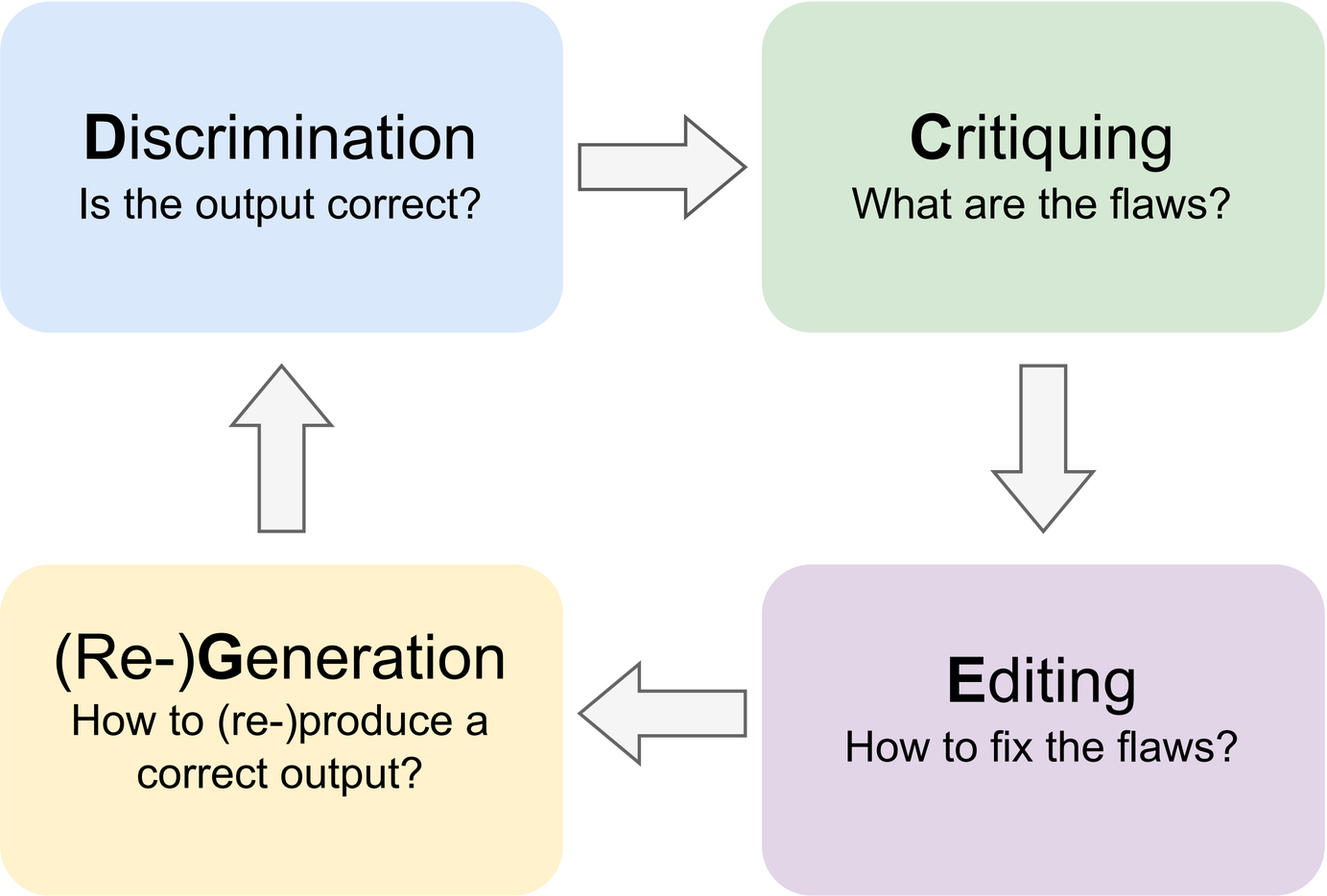
Output

```
# 1. How many cars were pulled on the first three days?
cars_daily_first_3_days = 10
cars_first_3_days = cars_daily_first_3_days * 3

# 2. How many cars were pulled on the remaining 4 days?
cars_remaining_days = 4 * 4

# 3. How many cars did the tow truck pull in total?
cars_week = cars_first_3_days + cars_remaining_days
```

>>> >>> Answer: 46

Python Interpreter

Self-Correction Pipeline



Adapted from (Saunders et al 2022)

Example

Query

In a week, a tow truck pulled **10 cars** for each of the first 3 days and then **4 fewer cars** on each of the remaining 4 days. How many cars did it tow in total?

Generation

```
# 1. How many cars were pulled on the first three days?
cars_daily_first_3_days = 10
cars_first_3_days = cars_daily_first_3_days * 3

# 2. How many cars were pulled on the remaining 4 days?
cars_remaining_days = 4 * 4

# 3. How many cars did the tow truck pull in total?
cars_week = cars_first_3_days + cars_remaining_days
```

>>> >>> Answer: 46

Python Interpreter

Discrimination

Is the output correct? (Yes/No): No

Critique

What's wrong with the output?

In Step #2, "4 fewer cars" is misunderstood as "4 cars"

Edit

Fix the error by editing this step:

```
# 2. How many cars were pulled on the remaining 4 days?
cars_remaining_days = 4 * (10-4)
```

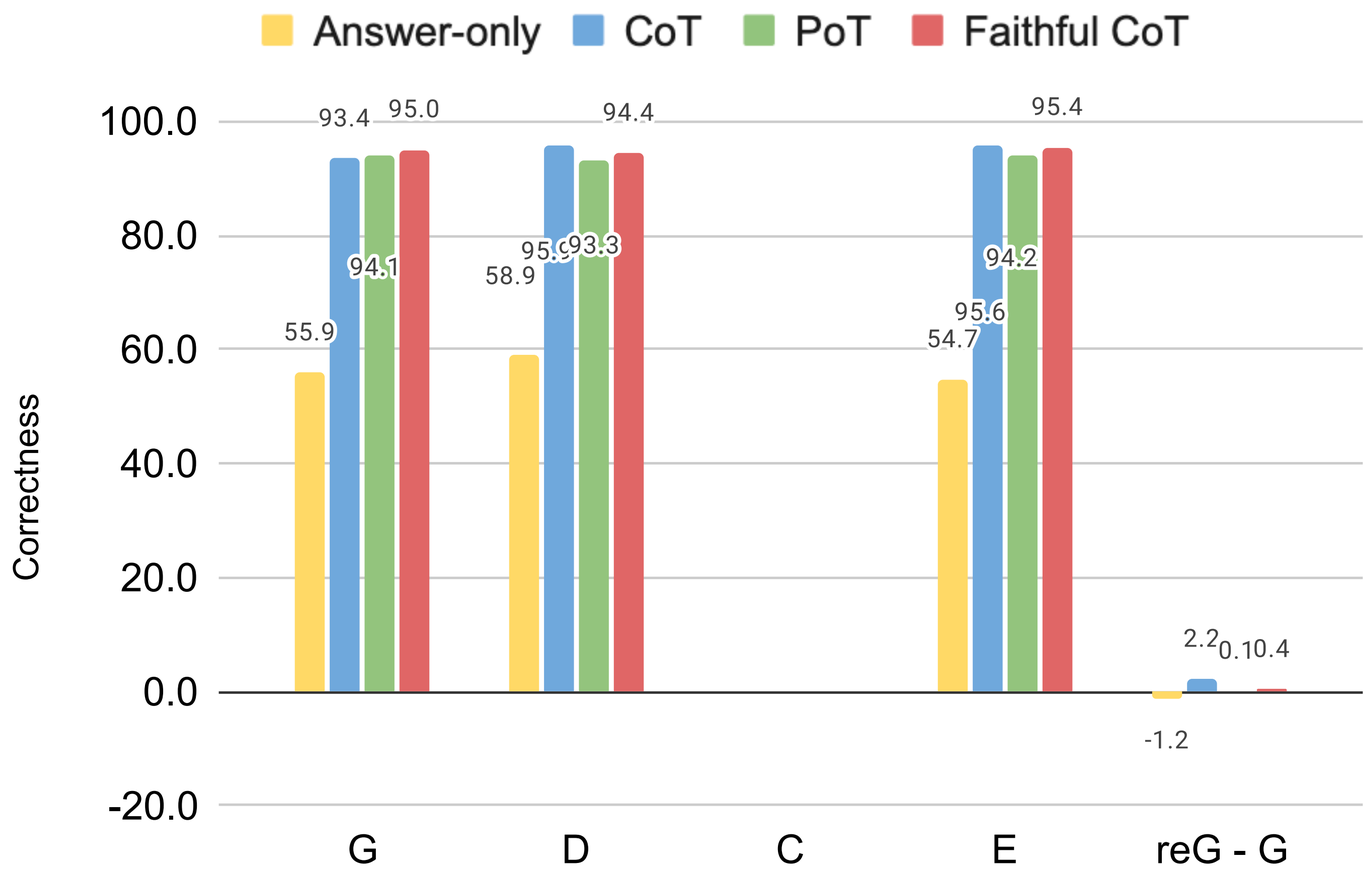
iterate if needed

Experimental Setup

- **LMs**: Codex, GPT-3.5-turbo, GPT-4
- **Decoding Strategy:** greedy decoding
- **Types of explanations:** Answer-only, CoT, PoT, FCoT
- **Evaluation Metrics**:
  - Generation: Accuracy
  - Discrimination: Accuracy, Macro-F1
  - Critiquing & Editing: Human acceptability

  } "Correctness" for simplicity
- Dataset: GSM8K

Do Explanations Improve LMs' Self-Correction Ability?



Correctness for Codex on four stages of self-correction.*
Other LMs show similar trends.

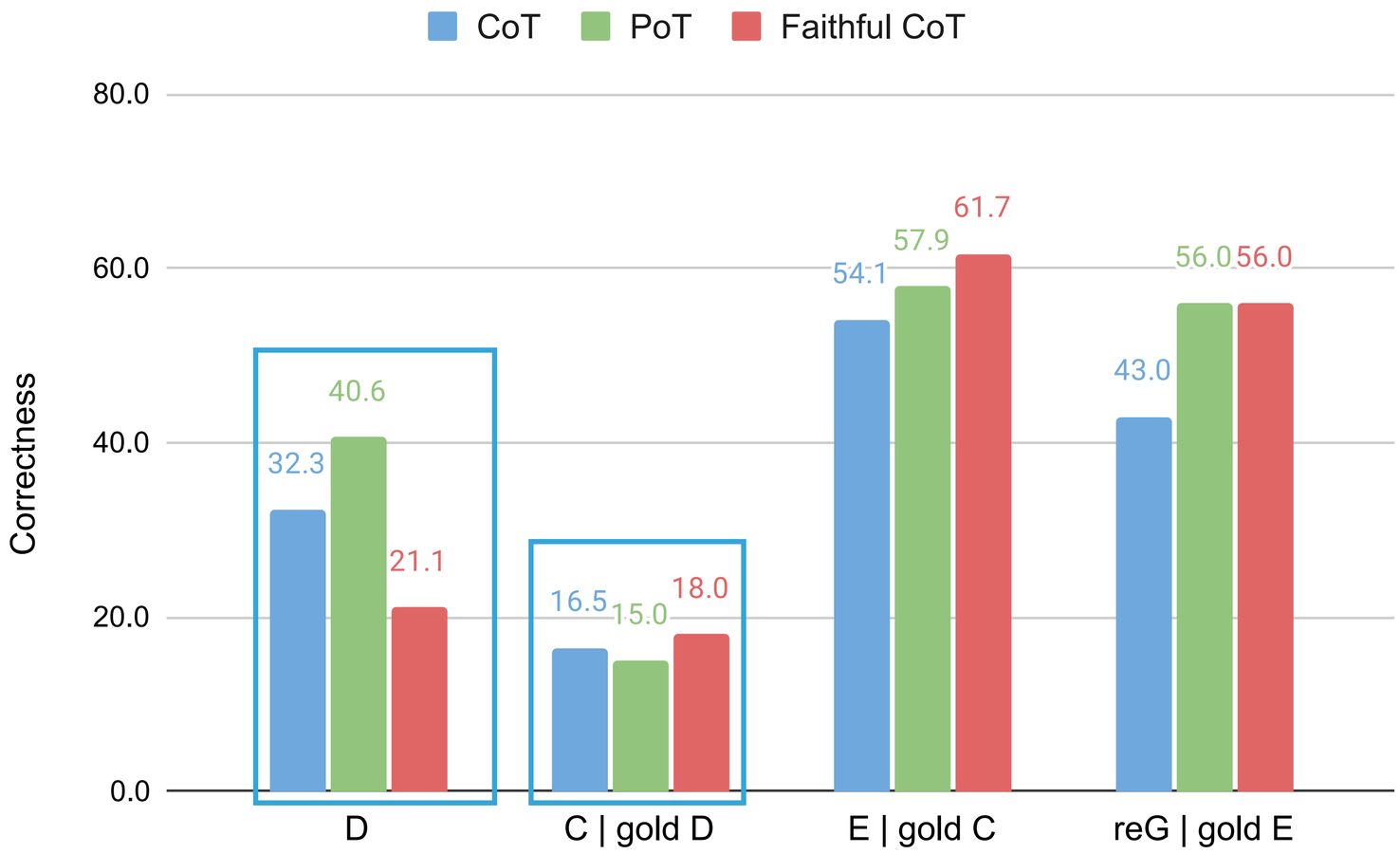
- Explanations do make LLMs better at all stages of self-correction
- However, simply asking LLMs to perform the DGCE tasks with few-shot prompting is insufficient

Where's the bottleneck?

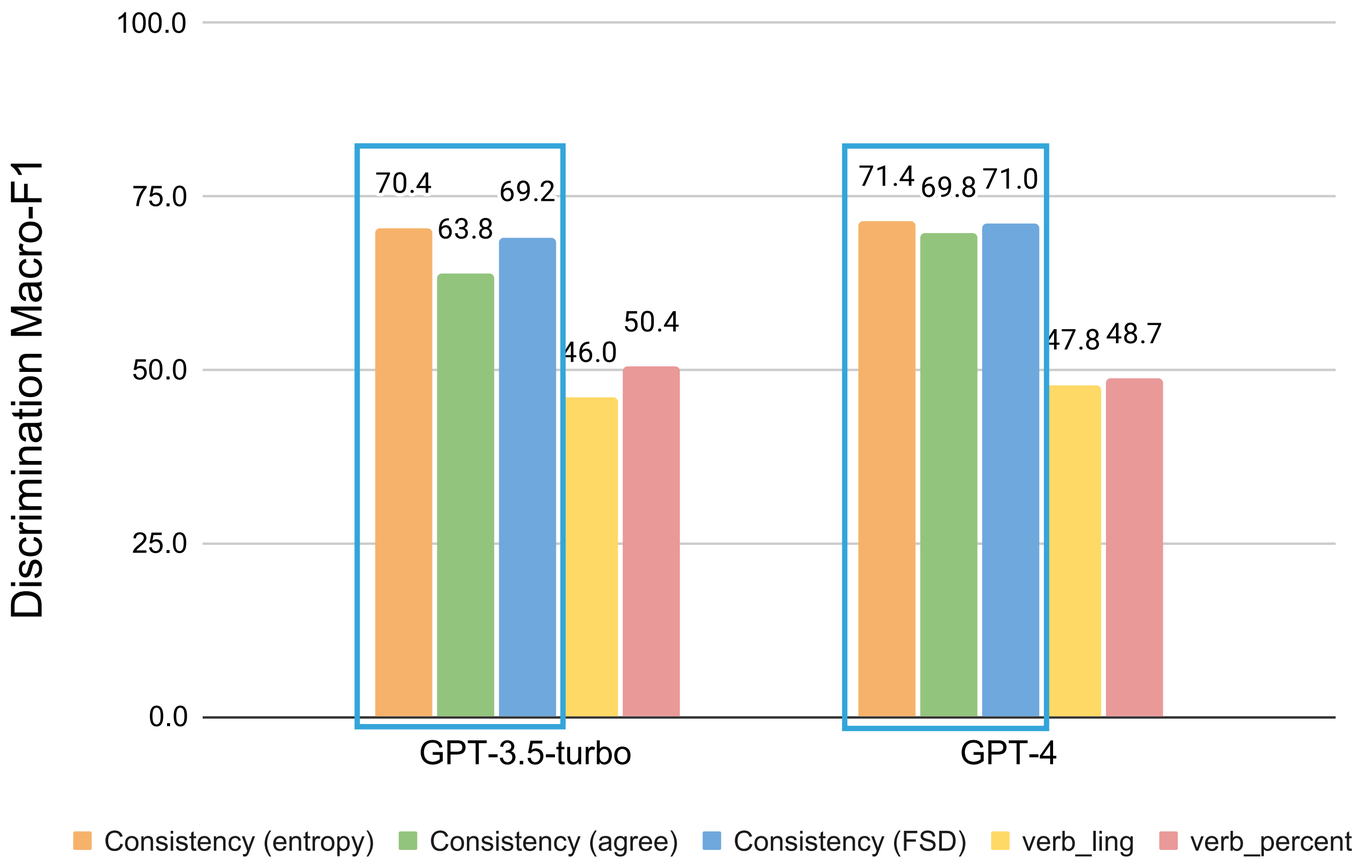
*Critique performance currently missing due to annotation cost.

Error Breakdown

- Given the oracle input from the previous stage, what is the performance for the current stage?

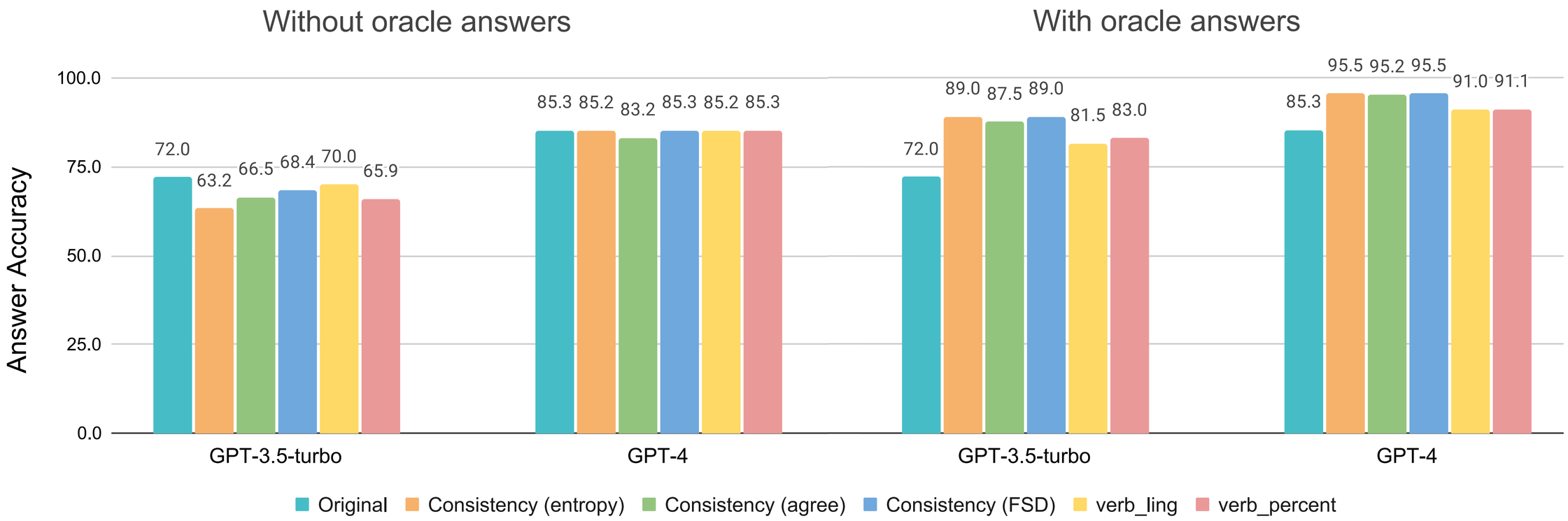


Consistency-based Calibration for Better Discrimination



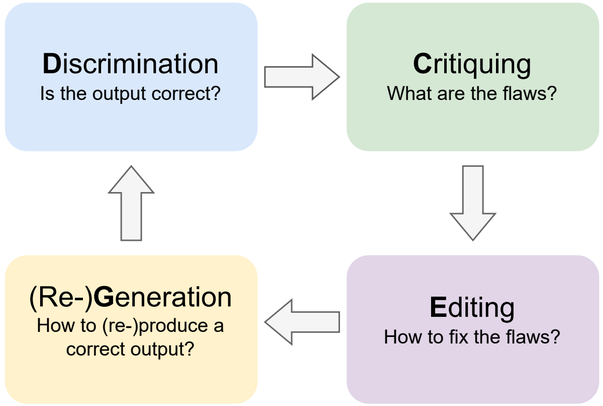
- Consistency-based calibration improves Discrimination compared to few-shot prompting

Does This Translate Into More Effective Self-Correction?



- Improved Discrimination alone is still inadequate for the overall pipeline to be beneficial
- Assuming oracle input in subsequent stages, consistency-based calibration shows potential in facilitating self-correction

Summary of This Section



- We propose a 4-stage self-correction pipeline (DCEG): Discrimination, Critiquing, Editing, (Re-)Generation
- Explanations make LLMs better at each stage
  - But overall gains are minimal – few-shot prompting isn't sufficient
  - Discrimination and Critiquing are the bottleneck
- Consistency-based calibration improves Discrimination
  - Better techniques in the other three stages are still necessary for overall gains

Outline

Faithfulness

- Faithful Explanations with Symbolic Reasoning

Usefulness

- Using Explanations for Confidence Calibration
- Using Explanations for Self-Correction

during in-context learning

- Using Explanations for Robustness Enhancement – during finetuning

Conclusion and Future Directions

Spoiler⚠️: Main takeaway

- When incorporated in finetuning, explanations make LLMs more robust against spurious cues in the data

Spurious Cues

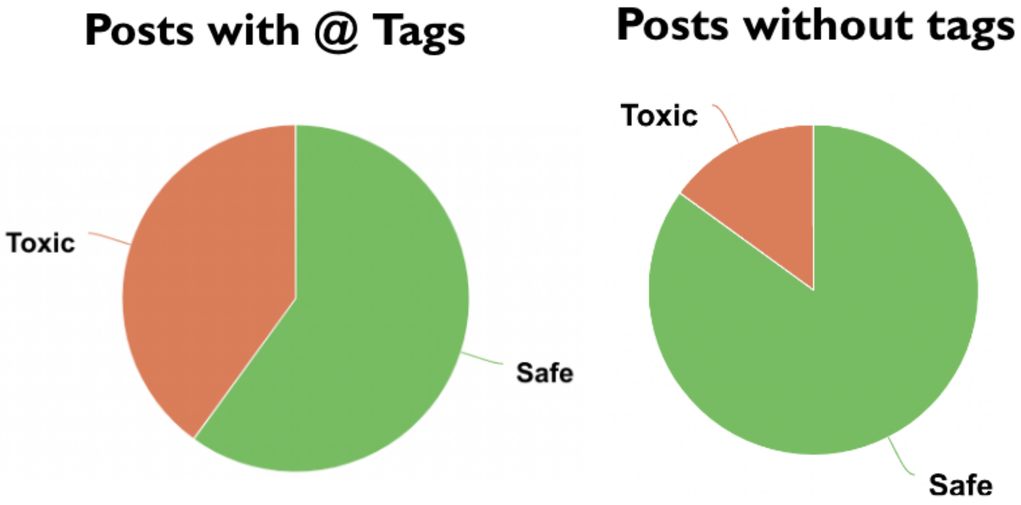
- Example: Toxicity Detection

Post: @AnonymousCookie I can't wait to see the new planet of the apes.

finetuned GPT-3

Answer: Offensive ✗

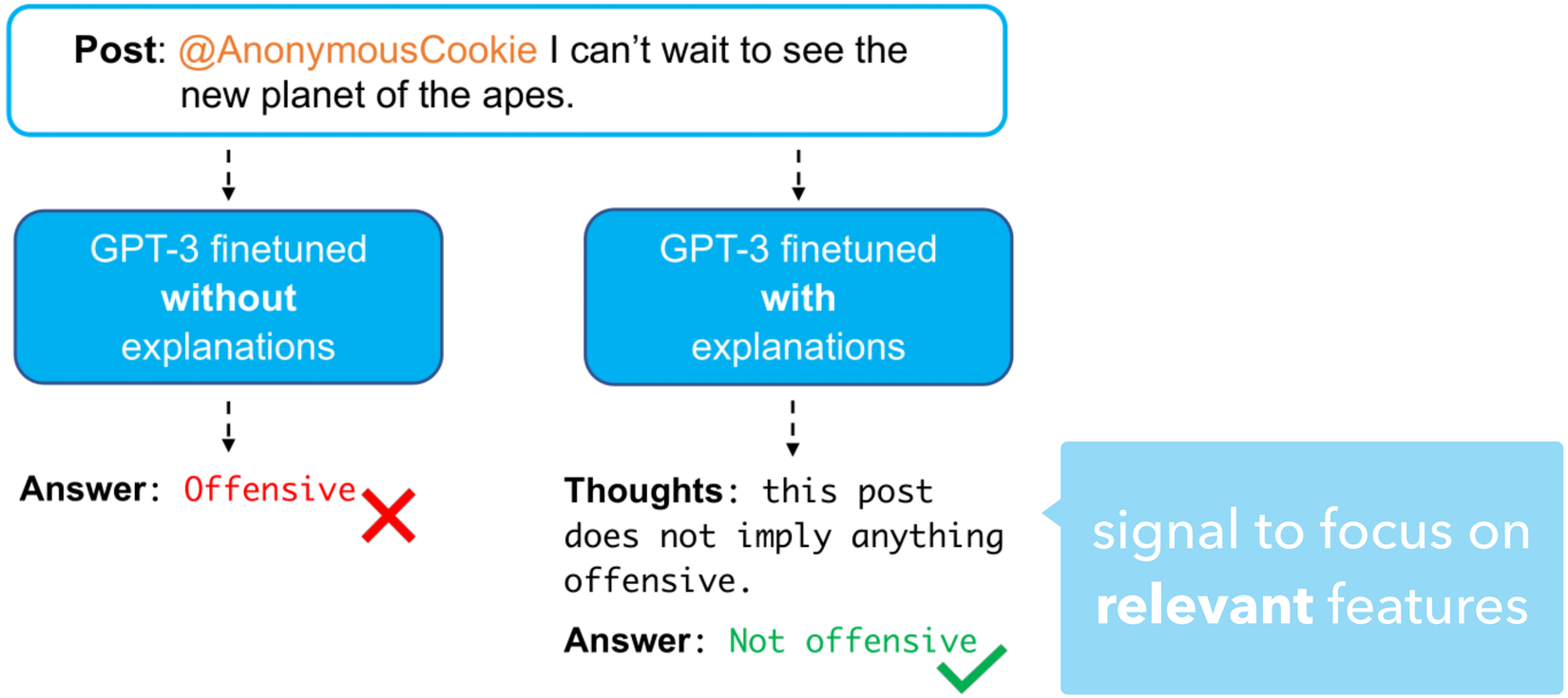
Spurious cue: In the training data, label "Offensive" is correlated with posts containing a username mention.



Explanation-based Finetuning Makes Models More Robust to Spurious Cues (Ludan et al., 2023)

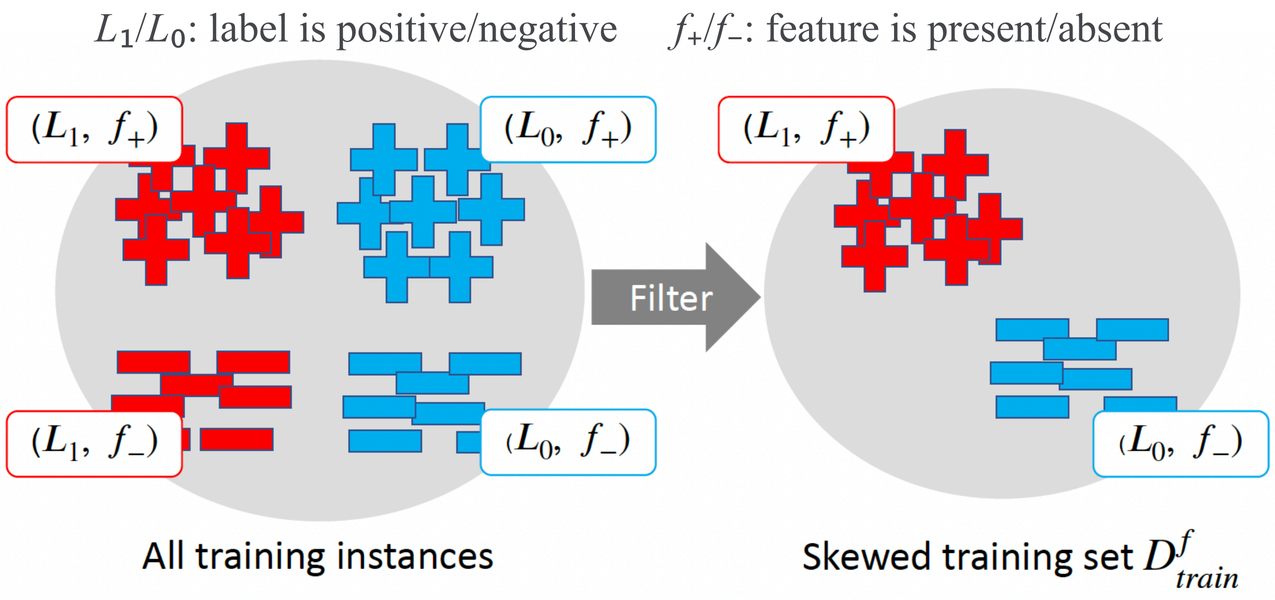
Can Explanations Steer LMs Away from Spurious Cues?

- Finetuning with / without explanations



Experimental Setup

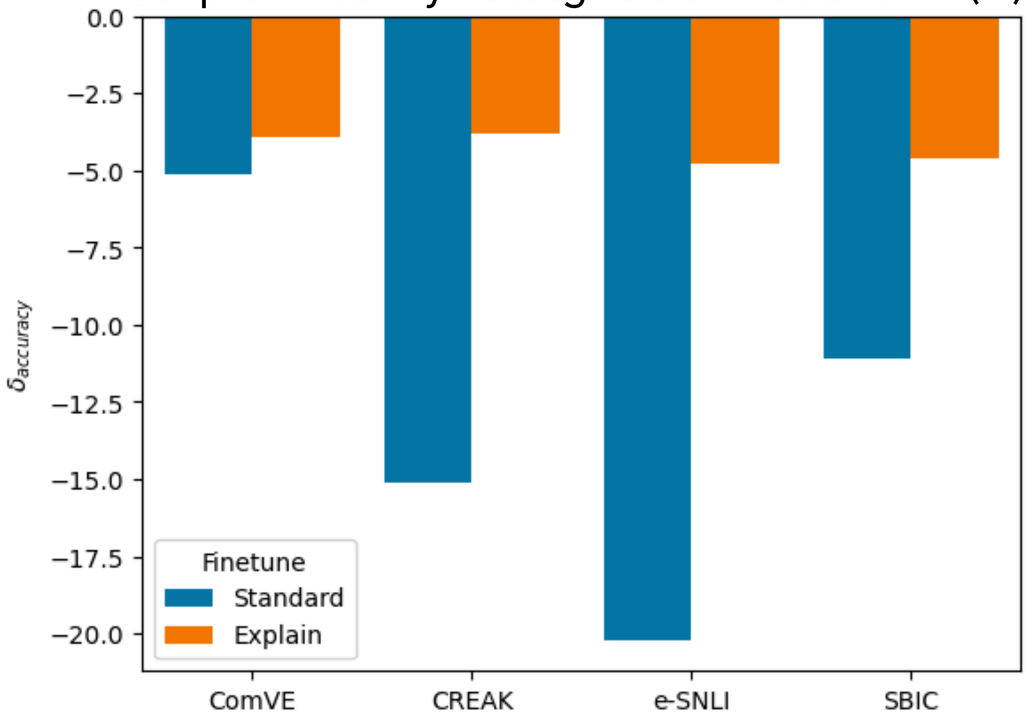
- Finetune on skewed training set (with injected cue), test on unskewed test set
- **Datasets**: w/ human-provided explanations
  - e-SNLI (textual entailment)
  - ComVE (plausibility comparison)
  - CREAK (fact verification)
  - SBIC (toxicity detection)
- **Cues**: sentence length, tense, noun plurality, embedding similarity



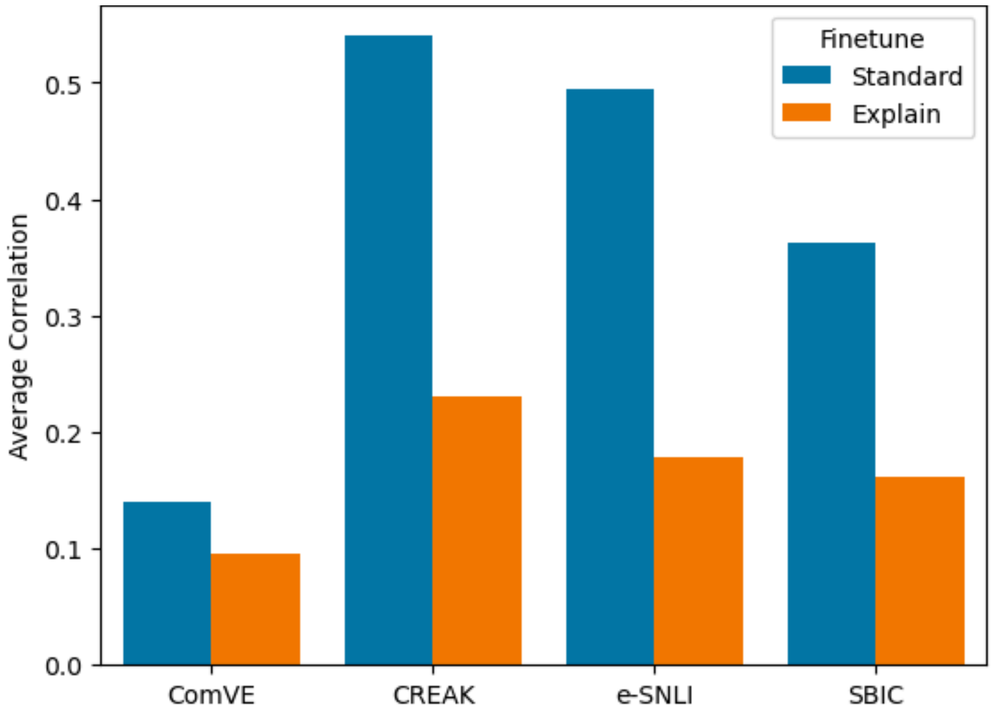
- **LMs**: GPT-3 (ada, babbage, curie, davinci), OPT (1.3B), T5 (base), BART (base)
- **Evaluation metrics**:
  - Accuracy drop on unskewed test set (↑)
  - Correlation between cue and model prediction (↓)

Results

Drop in Accuracy Averaged across Four Cues (↑)



Prediction-Feature Correlation Averaged across Four Cues (↓)



Results for GPT-3 (davinci)

- This even works with bootstrapped explanations!

Summary of This Section

- Incorporating free-text explanations during finetuning encourages models to focus on relevant features over spurious ones

model-and-cue-**agnostic**

Outline

Faithfulness

- Faithful Explanations with Symbolic Reasoning

Usefulness

- Using Explanations for Confidence Calibration
- Using Explanations for Self-Correction

during in-context learning

- Using Explanations for Robustness Enhancement – during finetuning

Conclusion and Future Directions

Conclusion

- LLM-generated explanations aren't faithful (accurately reflecting the model's reasoning mechanism behind the prediction) or useful (helpful in real-life scenarios) by default
- Incorporating symbolic reasoning can enhance the faithfulness of LLM-generated explanations and the performance on the end task
- When used as an exposure interface or training signals, LLM-generated explanations can be helpful for model improvement in different scenarios including confidence calibration, self-correction, and robustness enhancement

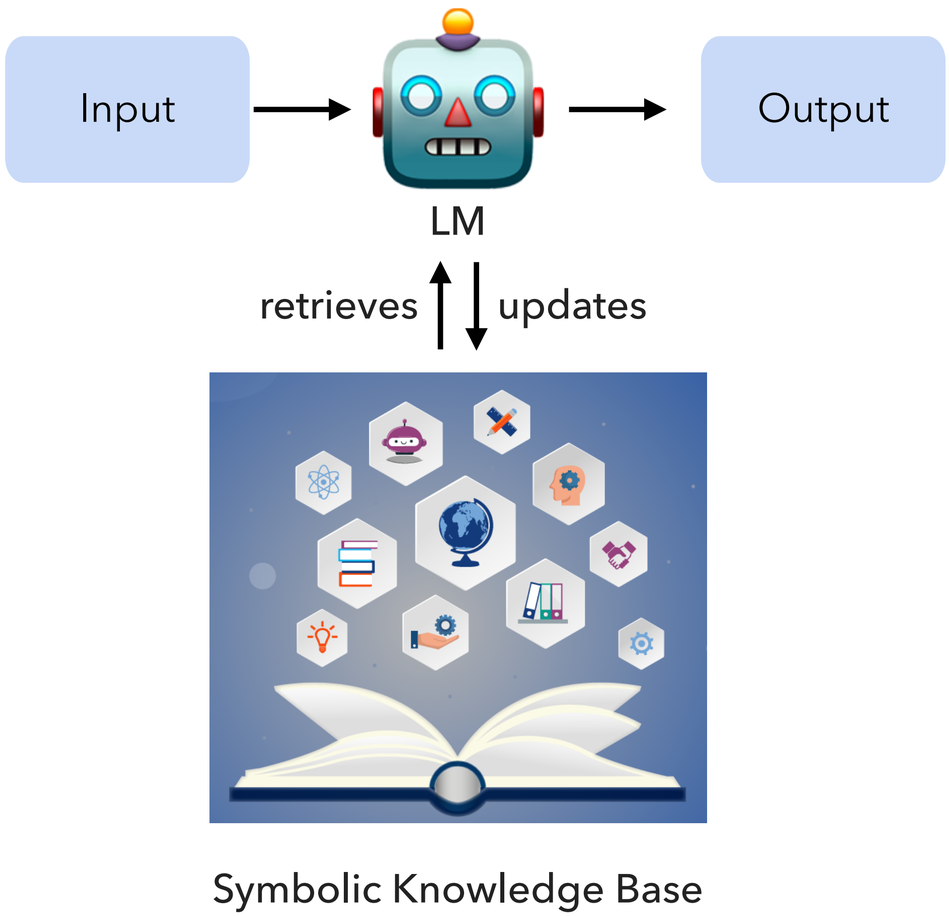
Future Directions

Promises of Explainable AI (XAI)

- Controllable
- Human-centered
- Informative

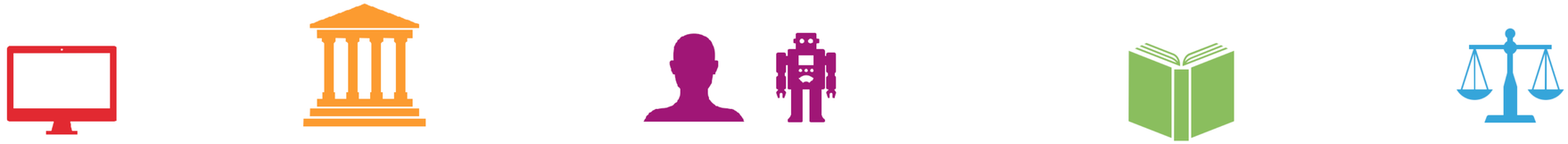
Materialize and Control Knowledge with Symbolic Representation

- When to use parametric knowledge vs. in-context knowledge?
  - How to resolve conflicts in between?
- How to learn a new Domain Specific Language (DSL) / unfamiliar tools?
- How to unlearn sensitive / undesired information?



Human-Centered Application of Explainable AI

- The design and evaluation of XAI systems should be grounded in real-world use cases

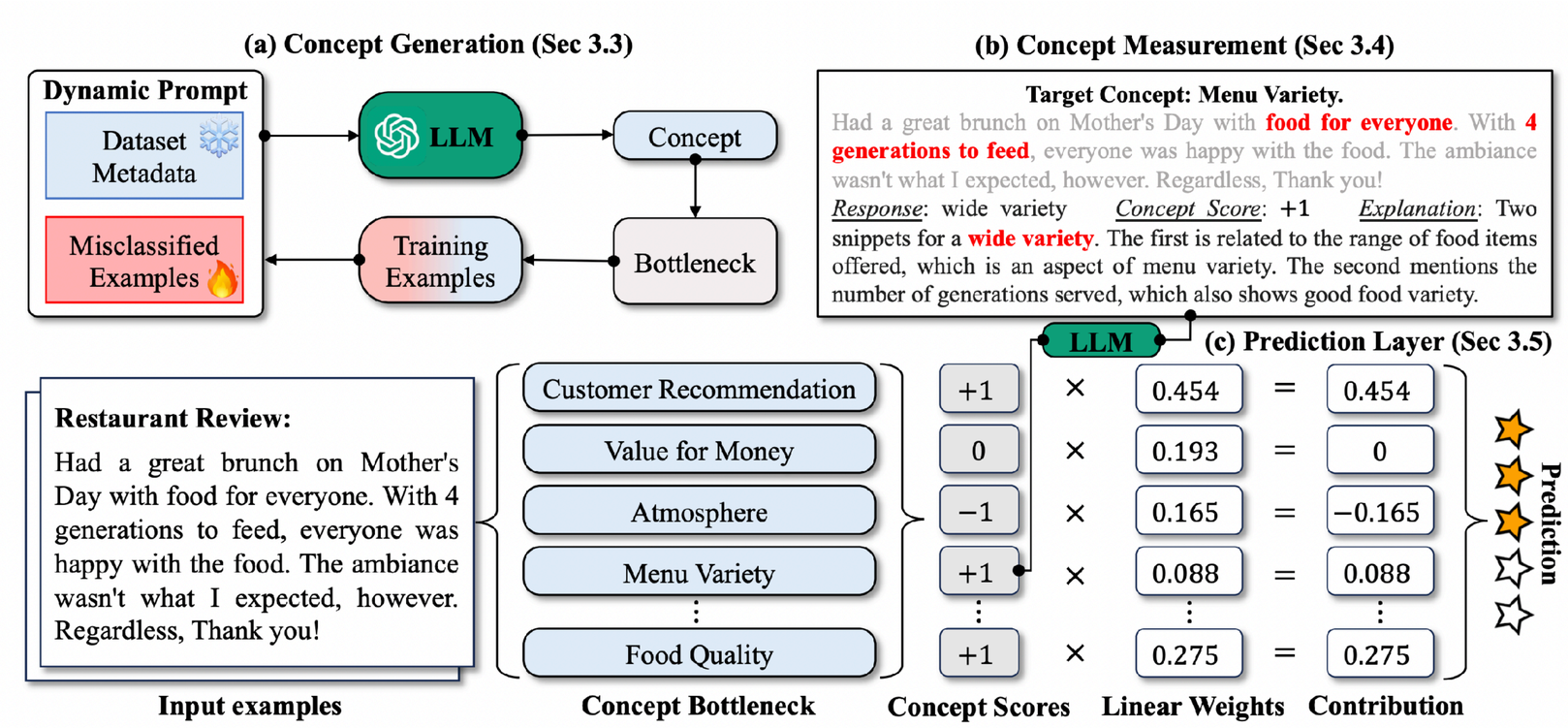


Model debugging | Model auditing | Collaborative decision making | Knowledge discovery | Trust calibration

  - Need realistic benchmarks for utility comparison
  - Need standardized evaluation constructs
    - e.g. in collaborative decision making: decision accuracy, speed, switch percentage, …
  - Focus on generalization

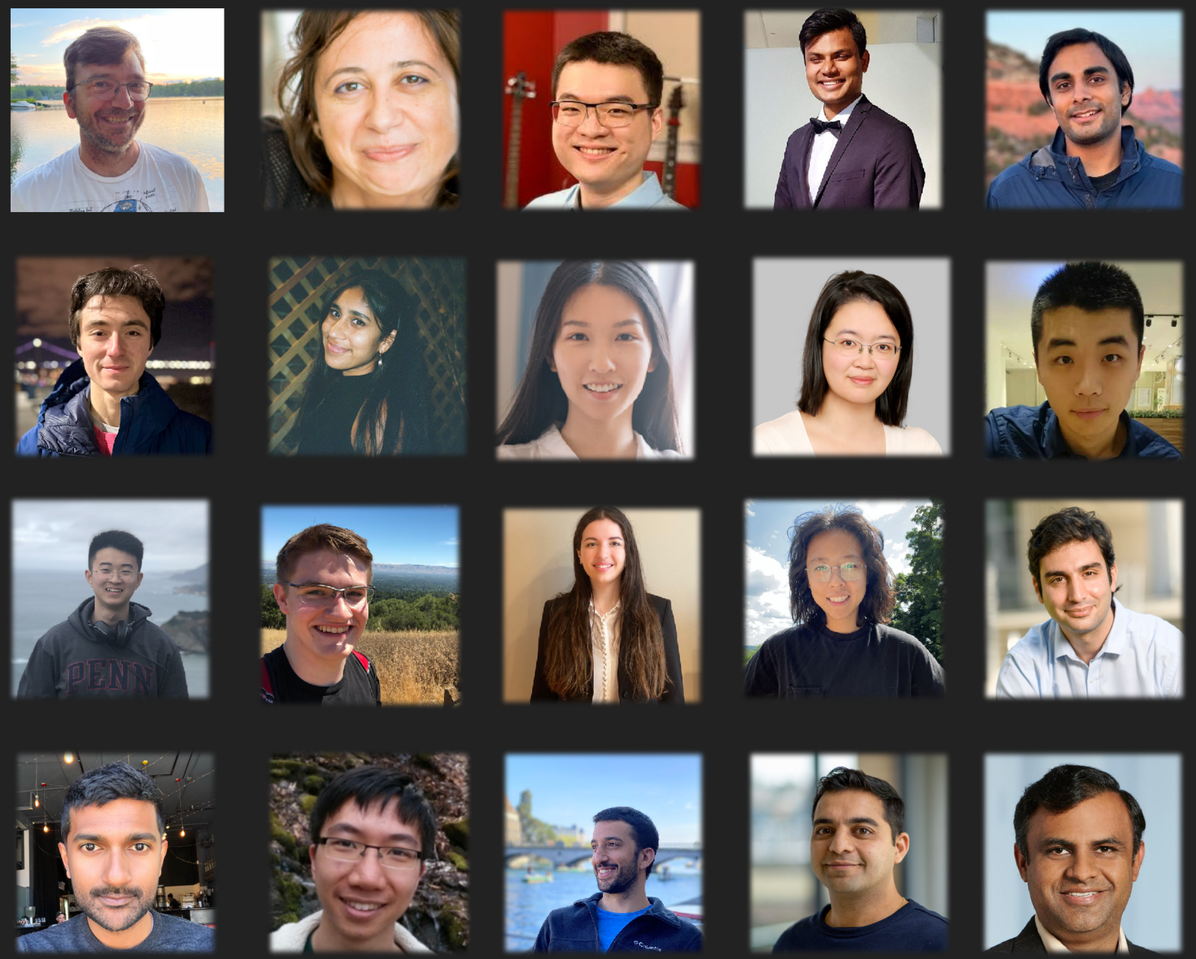
Explainable AI for Scientific Discovery

- Our work on Concept Bottleneck Models (CBMs) for text understanding (Ludan et al 2024)



- Can we use it to learn about unfamiliar domains?
  diagnose a new disease, detect auto-generated text, identify distribution shift …

Collaborators



MARCIANO
MOORE
CLAY
BANKS
FIGHTS
AMATEUR BOXING

THANKS!