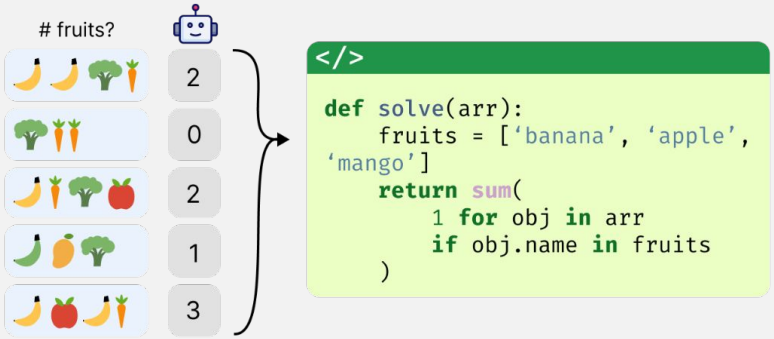
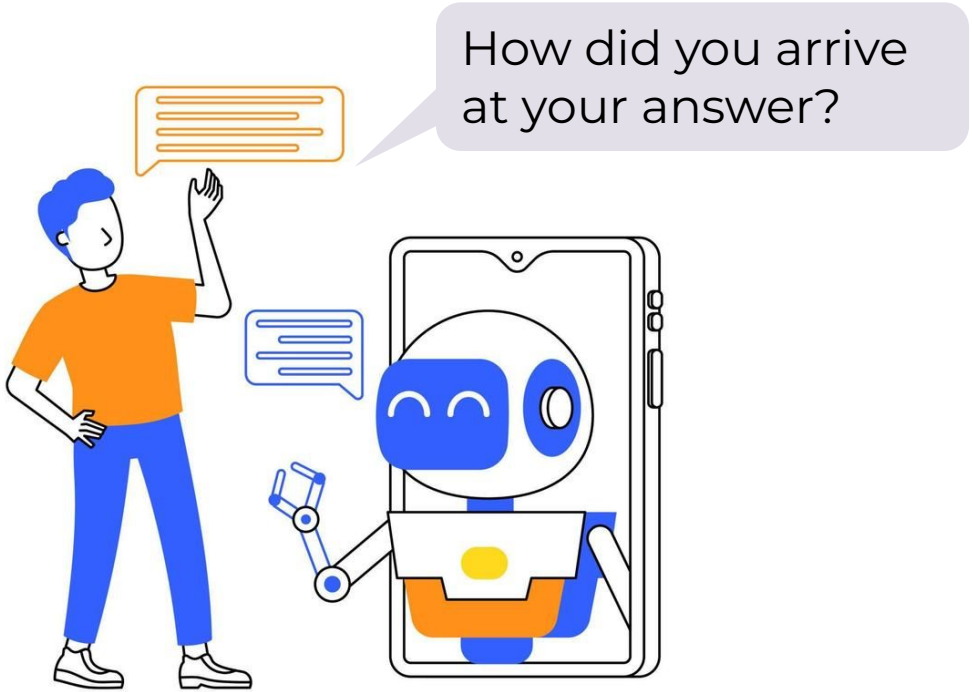
Code as Explanations: Executable and Verifiable LLM Explanations

Nhi Nguyen, Shauli Ravfogel, Rajesh Ranganath



Understand LLMs



Debug

Learn

Explanations are often free-form text

Chain-of-thought

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. ✔

Feature attribution



Is this review positive? Why or why not? "A fantastic movie directed by the famous Lucas Johnson, who has a track-record of producing terrific novel adaptions."

This review is very positive, as judged by the positive words "fantastic" and "terrific".

CHATGPT

Counterfactual

User

Now, revise the math problem so your final answer to the revised problem becomes **50**. Share the revised problem.

Model

Albert is wondering how much pizza he can eat in one day. He buys 2 large pizzas and **3** small pizzas. A large pizza has 16 slices and a small pizza has 8 slices, **but one of the small pizzas has 2 extra slices**. If he eats it all, how many pieces does he eat that day?

Correct answer would be 58

Explanations are often free-form text

Chain-of-thought

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. ✔

Feature attribution



Is this review positive? Why or why not? "A fantastic movie directed by the famous Lucas Johnson, who has a track-record of producing terrific novel adaptions."

This review is very positive, as judged by the positive words "fantastic" and "terrific".

CHATGPT

Appear simple, one-to-one

Counterfactual

USER

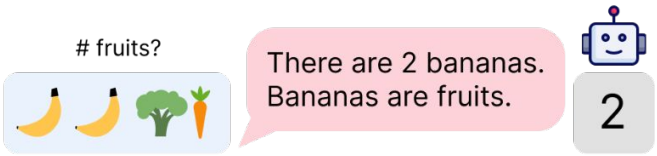
Now, revise the math problem so your final answer to the revised problem becomes **50**. Share the revised problem.

MODEL

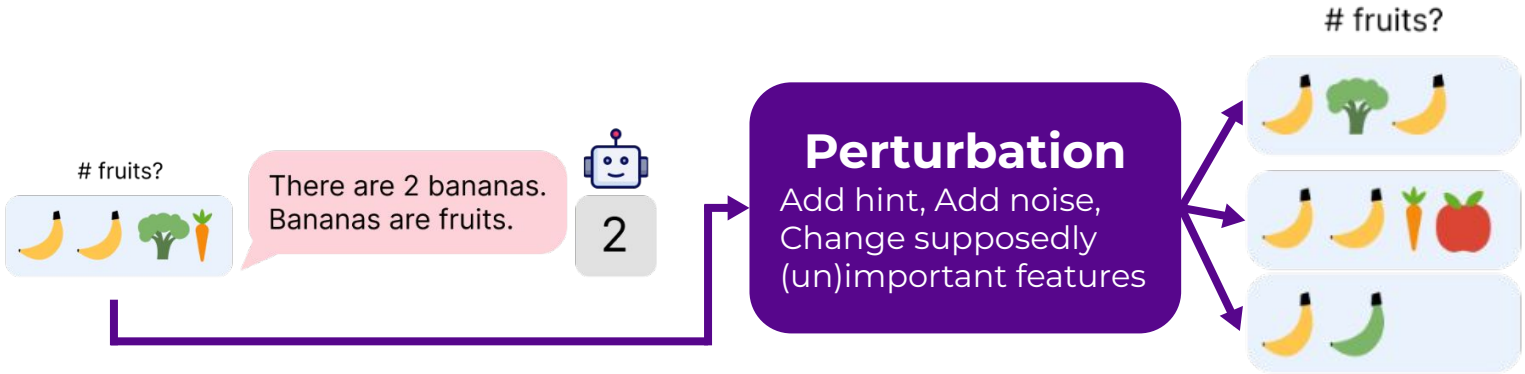
Albert is wondering how much pizza he can eat in one day. He buys 2 large pizzas and **3** small pizzas. A large pizza has 16 slices and a small pizza has 8 slices, **but one of the small pizzas has 2 extra slices**. If he eats it all, how many pieces does he eat that day?

Correct answer would be 58

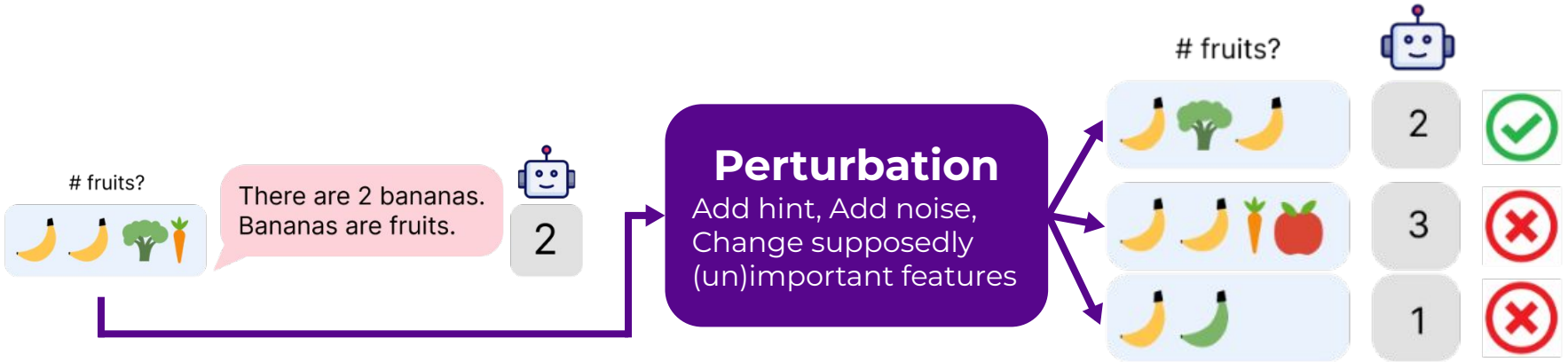
Evaluating text explanations is hard



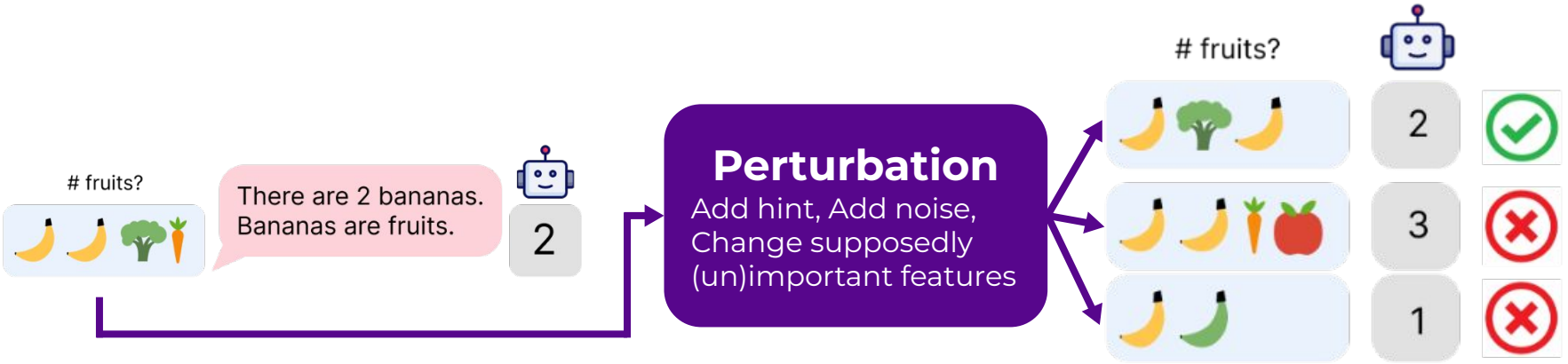
Evaluating text explanations is hard



Evaluating text explanations is hard



Evaluating text explanations is hard



Why do we need all these additional inputs?

Proposition: Understanding behavior requires multiple observations

Proposition 3.1 (Unidentifiability from observations). *Let $\gamma : \mathcal{X} \to \mathcal{G}$ be an input property function and let $f : \mathcal{X} \to \mathcal{Y}$ be an output function. Suppose that f can only be observed on a proper subset $\mathcal{S} \subset \mathcal{X}$, where $\gamma(\mathbf{x}) = C$ for all $\mathbf{x} \in \mathcal{S}$. Consider two candidate output functions:*

1. *$f_1 : \mathcal{X} \to \mathcal{Y}$ that does not depend on γ;*
2. $f_2(\mathbf{x}) = \begin{cases} f_1(\mathbf{x}) & \text{if } \gamma(\mathbf{x}) = C \\ f'(\mathbf{x}) & \text{if } \gamma(\mathbf{x}) \neq C' \end{cases}$ *where $f'(\mathbf{x}) \neq f_1(\mathbf{x})$ whenever $\gamma(\mathbf{x}) \neq C$.*

Although f_1 does not depend on γ and f_2 does, they agree on every input in $\mathcal{S}$. Thus, whether the output function f depends on the property γ cannot be identified from observations in $\mathcal{S}$.

If a **property** is **constant across observations**, whether it **influences the output** **cannot be determined**

Proposition: Understanding behavior requires multiple observations

Proposition 3.1 (Unidentifiability from observations). *Let $\gamma : \mathcal{X} \to \mathcal{G}$ be an input property function and let $f : \mathcal{X} \to \mathcal{Y}$ be an output function. Suppose that f can only be observed on a proper subset $\mathcal{S} \subset \mathcal{X}$, where $\gamma(\mathbf{x}) = C$ for all $\mathbf{x} \in \mathcal{S}$. Consider two candidate output functions:*

1. *$f_1 : \mathcal{X} \to \mathcal{Y}$ that does not depend on γ;*
2. $f_2(\mathbf{x}) = \begin{cases} f_1(\mathbf{x}) & \text{if } \gamma(\mathbf{x}) = C \\ f'(\mathbf{x}) & \text{if } \gamma(\mathbf{x}) \neq C' \end{cases}$ *where $f'(\mathbf{x}) \neq f_1(\mathbf{x})$ whenever $\gamma(\mathbf{x}) \neq C$.*

Although f_1 does not depend on γ and f_2 does, they agree on every input in $\mathcal{S}$. Thus, whether the output function f depends on the property γ cannot be identified from observations in $\mathcal{S}$.

If a **property** is **constant across observations**, whether it **influences the output cannot be determined**

Implication: Cannot evaluate explanations on one input

Proposition: Understanding behavior requires multiple observations

Proposition 3.1 (Unidentifiability from observations). *Let $\gamma : \mathcal{X} \to \mathcal{G}$ be an input property function and let $f : \mathcal{X} \to \mathcal{Y}$ be an output function. Suppose that f can only be observed on a proper subset $\mathcal{S} \subset \mathcal{X}$, where $\gamma(\mathbf{x}) = C$ for all $\mathbf{x} \in \mathcal{S}$. Consider two candidate output functions:*

1. *$f_1 : \mathcal{X} \to \mathcal{Y}$ that does not depend on γ;*
2. $f_2(\mathbf{x}) = \begin{cases} f_1(\mathbf{x}) & \text{if } \gamma(\mathbf{x}) = C \\ f'(\mathbf{x}) & \text{if } \gamma(\mathbf{x}) \neq C' \end{cases}$ *where $f'(\mathbf{x}) \neq f_1(\mathbf{x})$ whenever $\gamma(\mathbf{x}) \neq C$.*

Although f_1 does not depend on γ and f_2 does, they agree on every input in $\mathcal{S}$. Thus, whether the output function f depends on the property γ cannot be identified from observations in $\mathcal{S}$.

If a **property** is **constant across observations**, whether it **influences the output** **cannot be determined**

Implication: Cannot evaluate explanations on one input

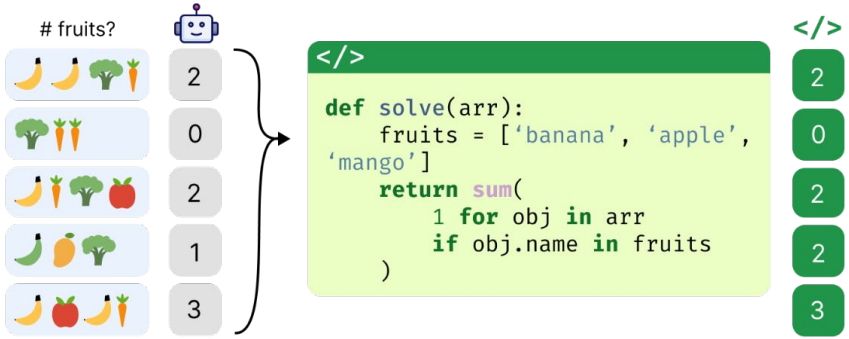
Can one explanation explain multiple inputs?

Solution: Executable code explanations



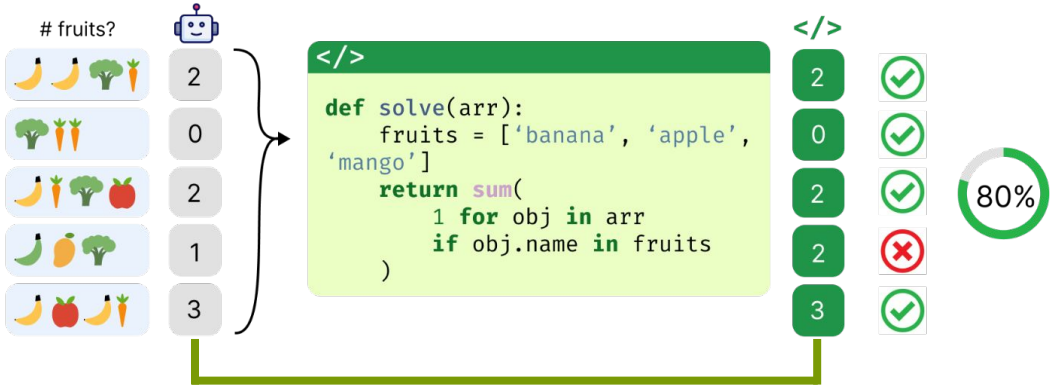
Solution: Executable code explanations





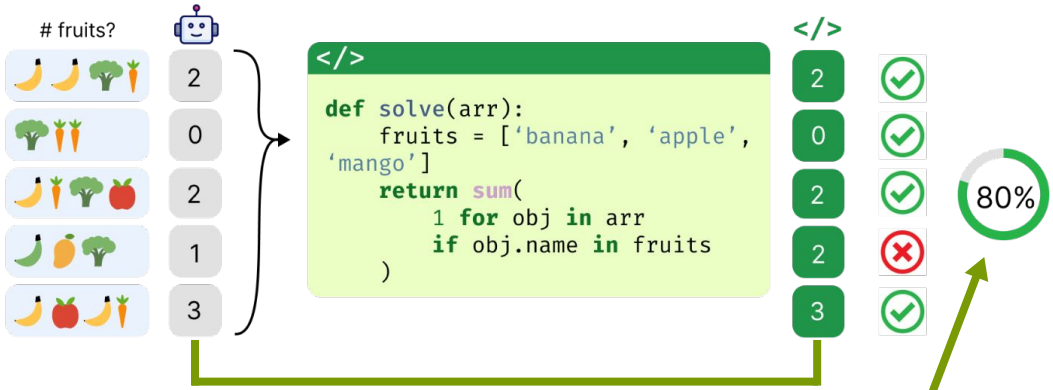
Solution: Executable code explanations





Solution: Executable code explanations

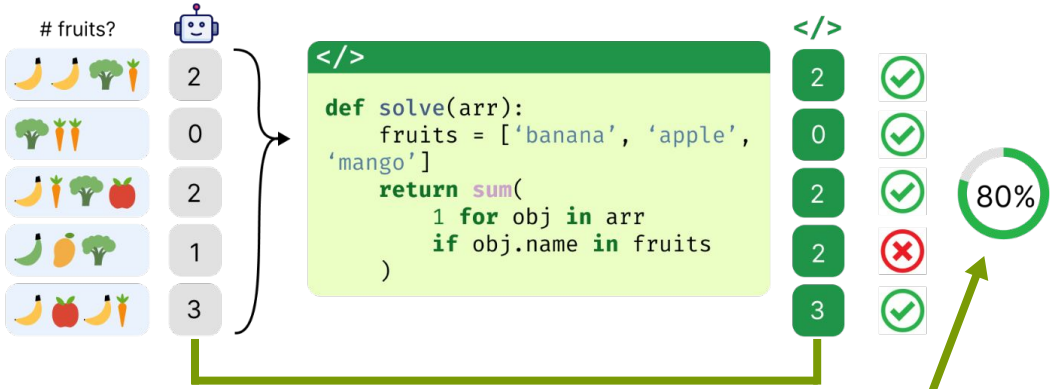




$$\textsc{Consistency}(c, \mathcal{D}) := \frac{1}{D} \sum_{i=1}^{D} \mathbf{1}\left[c(\boldsymbol{x}_i) = \boldsymbol{y}_i\right]$$

Solution: Executable code explanations

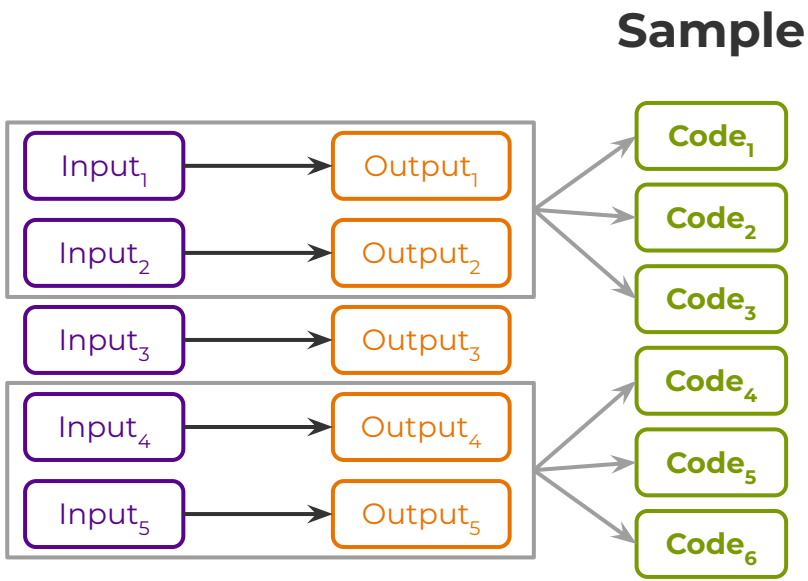




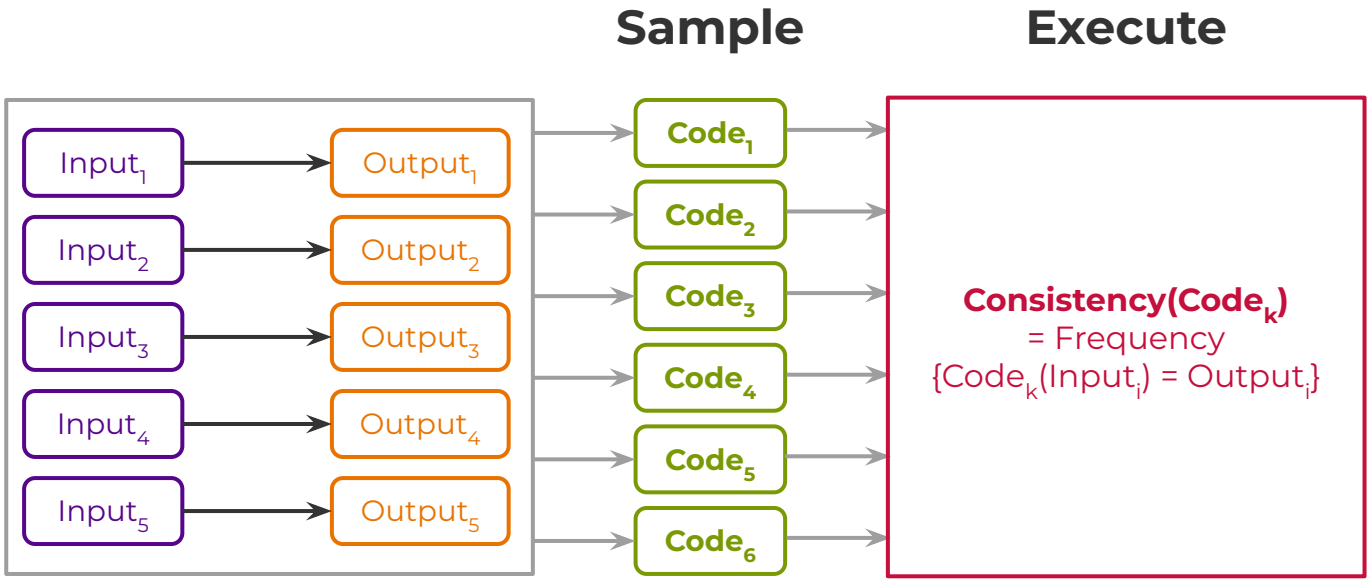
$$\textsc{Consistency}(c, \mathcal{D}) := \frac{1}{D} \sum_{i=1}^{D} \mathbf{1}\left[c(\boldsymbol{x}_i) = \boldsymbol{y}_i\right]$$

$\Delta\textsc{Consistency}$ = $\textsc{Consistency}$ – majority baseline

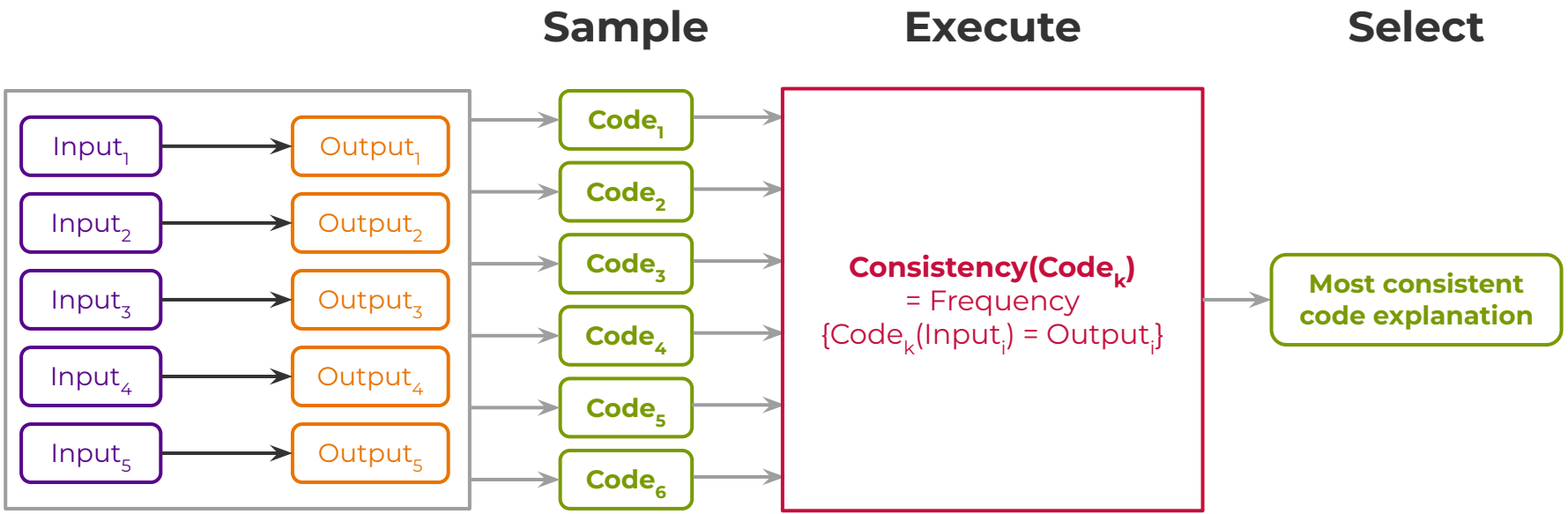
Generate code explanations



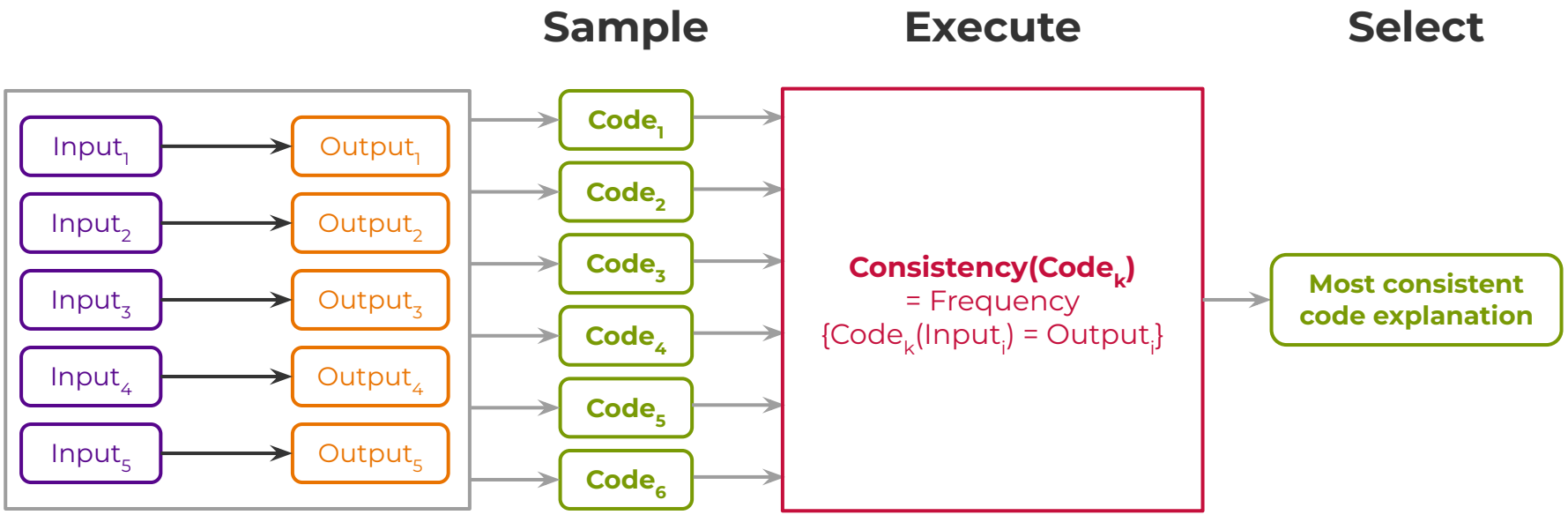
Generate code explanations



Generate code explanations



Generate code explanations



Note: Applicable to self-generated or post-hoc explanations

Experiments: Models

Open-source code-capable models



Coding agent



Experiments: Tasks

Structured logic

Web of lies

Does Shenna tell the truth?

Navigate

Do you return to the starting point?

Logic + simple language

Hyperbaton

Which sentence has the correct adjective order

Object counting

How many musical instruments do I have?

Logic + complex language

Formal fallacies

Is the argument deductively valid or invalid?

Disambiguation QA

Explain the antecedent of the pronoun.

Linguistic reasoning

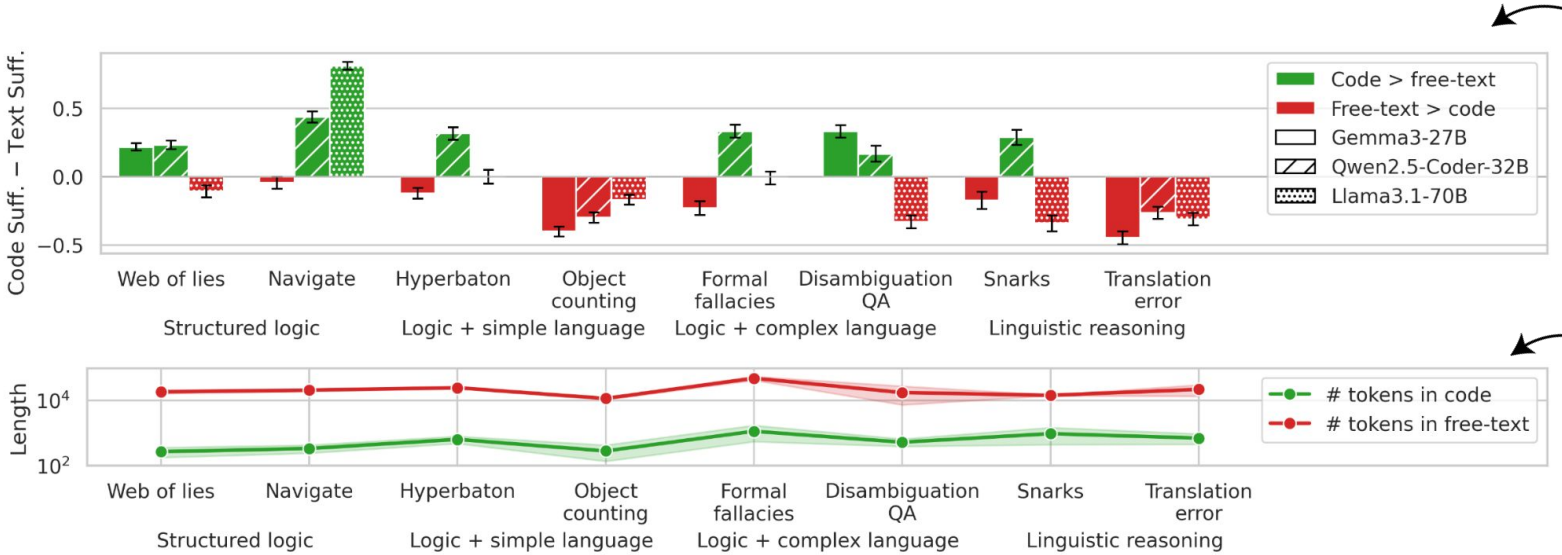
Snarks

Which statement is sarcastic?

Translation error

The translation contains an error pertaining to …

Result 1: Code vs. Text

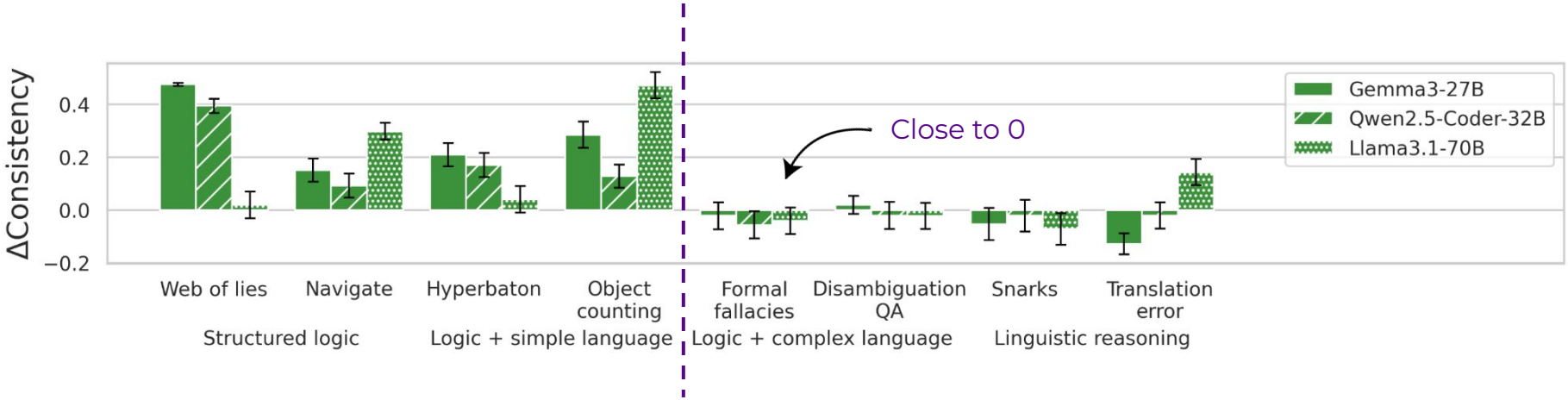


More sufficient than text in 9/24 cases

6-100x shorter

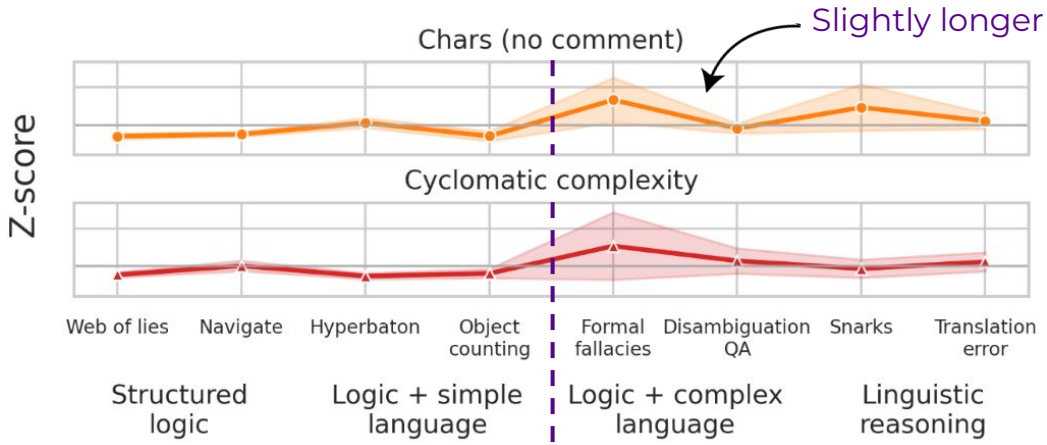
Code explanations are **6-100x shorter** than free-text explanations, and **sometimes better align** with LLM behavior.

Result 2: Consistency across tasks



LLMs **can generate consistent** code explanations.
But consistency drops on **linguistically complex tasks**.

Result 3: Code complexity across tasks



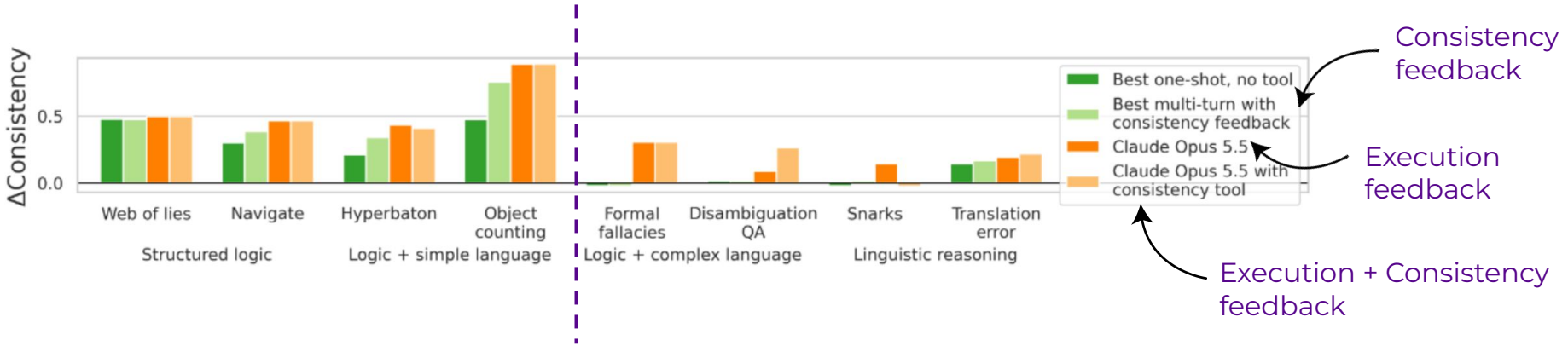
Snarks & Qwen2.5-Coder-32B

```
if any(indicator in statement for indicator in
['ARE U INSANE', 'Have my down vote', 'naive',
'terrible support', 'pragmatic', 'more
expensive', 'less fun', 'liability',
'enjoyable', 'best', 'reddit', 'fun',
'sexist', 'win you points', 'lose you points',
'True', 'rational', 'threaten', 'pleaded',
'highest', 'lowest', 'rich', 'loans',
'suspicious', 'legit', 'perfect' ]):
    # If the statement contains a sarcastic
indicator, mark it as the sarcastic option
        sarcastic_option = label
```

Hard-coded

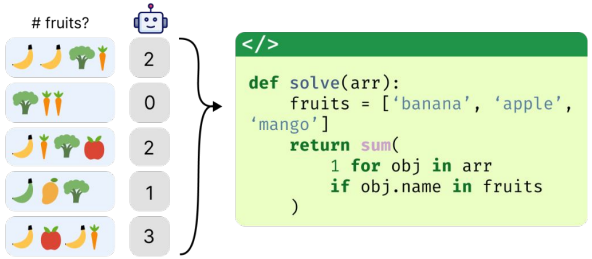
Complex linguistic tasks produce **longer**, **more hard-coded** programs.

Result 4: Feedback improves consistency



Feedback improves consistency across all tasks.
But **complex linguistic tasks remain challenging**, even for Claude.

Conclusion



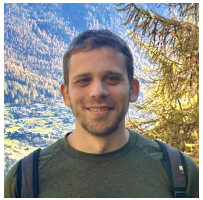
- Code = Executable & verifiable LLM explanations.
- LLMs can generate short and consistent code explanations
- Execution and consistency feedback improve consistency
- But complex linguistic behavior remains difficult to explain

Paper & Code

Nhi Nguyen



Shauli Ravfogel



Rajesh Ranganath

