

Towards Consistent Language Models Using Controlled Prompting and Decoding

Jasmin Mousavi, **Arash Termehchy**



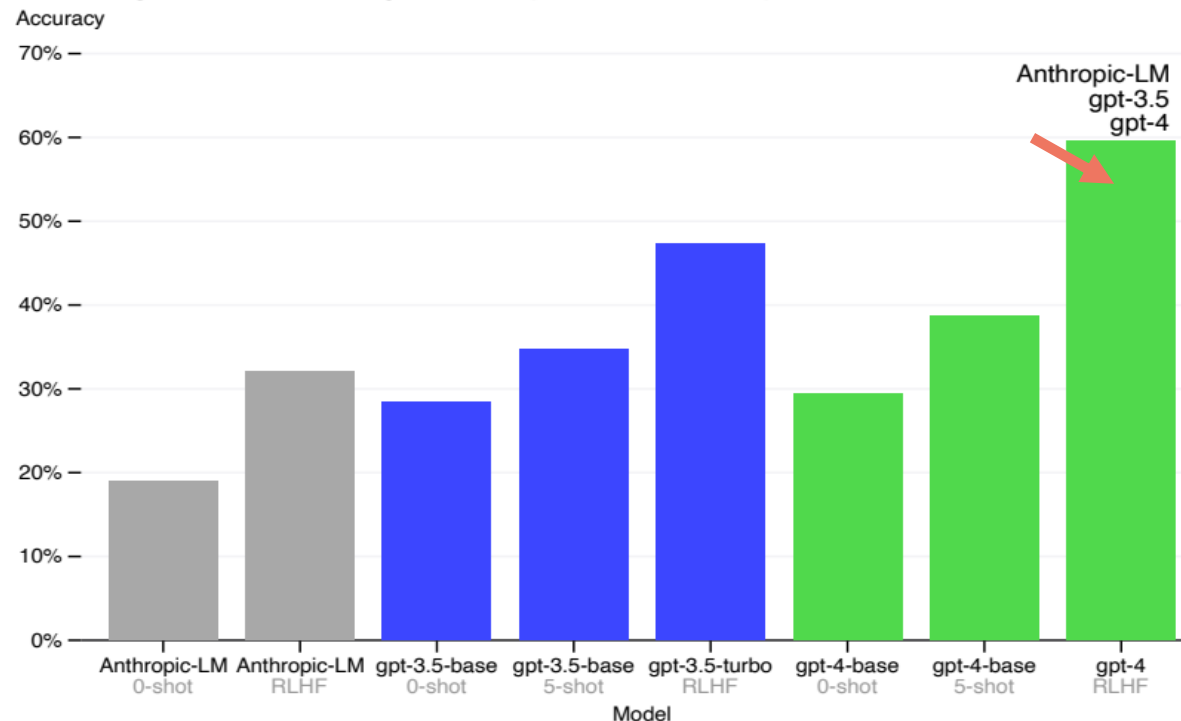
Oregon State
University

LLMs hallucinate

- Produce information inconsistent with common knowledge

GPT-4: 40% hallucination rate

Accuracy on adversarial questions (TruthfulQA mc1)



Causes of inconsistencies

- Data quality issues
- Biases in data
 - human biases (gender)
 - misconceptions (conspiracies)
- Over-generalizing patterns in data

GPT-2: Scrapes text from all outbound links from Reddit with at least 3 karma

which have been curated/filtered by humans. Manually filtering a full web scrape would be exceptionally expensive so as a starting point, we scraped all outbound links from Reddit, a social media platform, which received at least 3 karma. This can be thought of as a heuristic indicator for whether other users found the link interesting, educational, or just funny.

Causes of inconsistencies

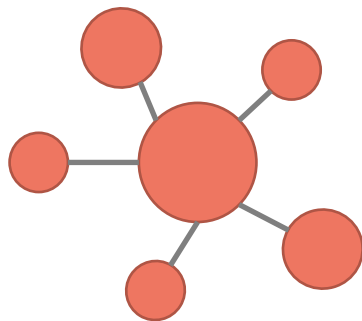
- Data quality issues
- Biases in data
 - human biases (gender)
 - misconceptions (conspiracies)
- Over-generalizing patterns in data



Causes of inconsistencies

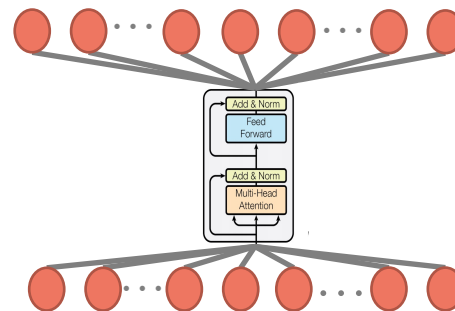
- Data quality issues
- Biases in data
 - human biases (gender)
 - misconceptions (conspiracies)
- Over-generalizing patterns in data

Knowledge Base



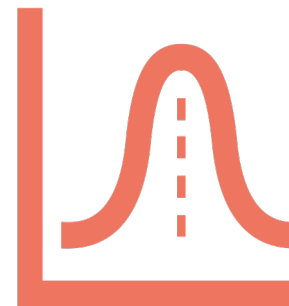
\neq

LLM



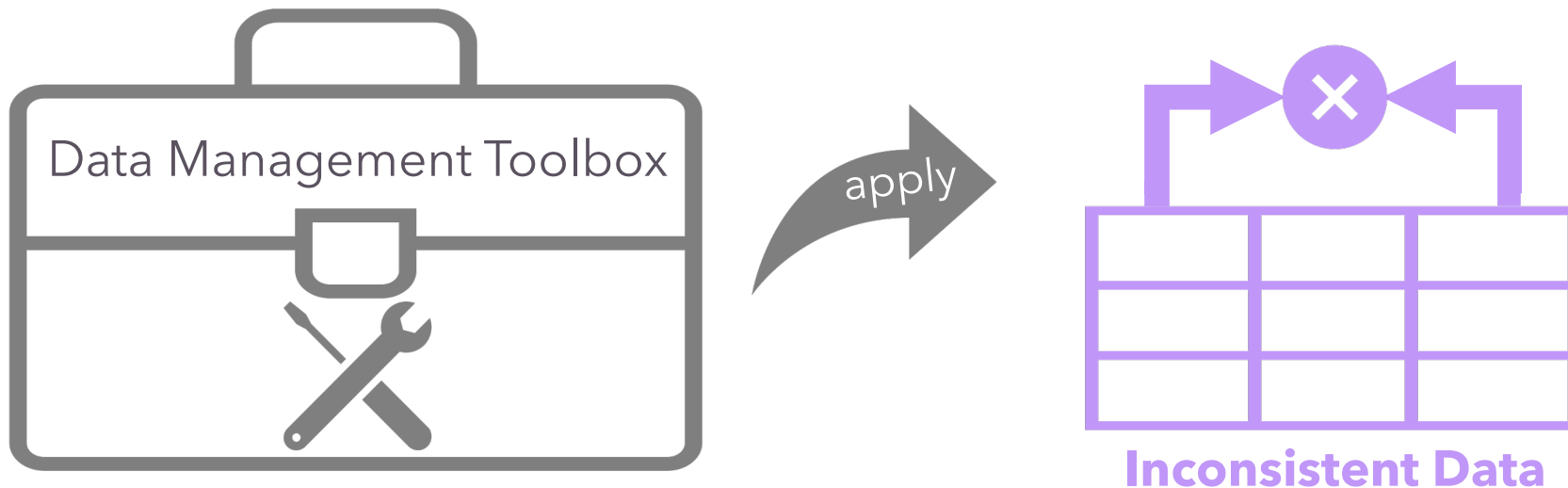
$=$

Probabilistic Model



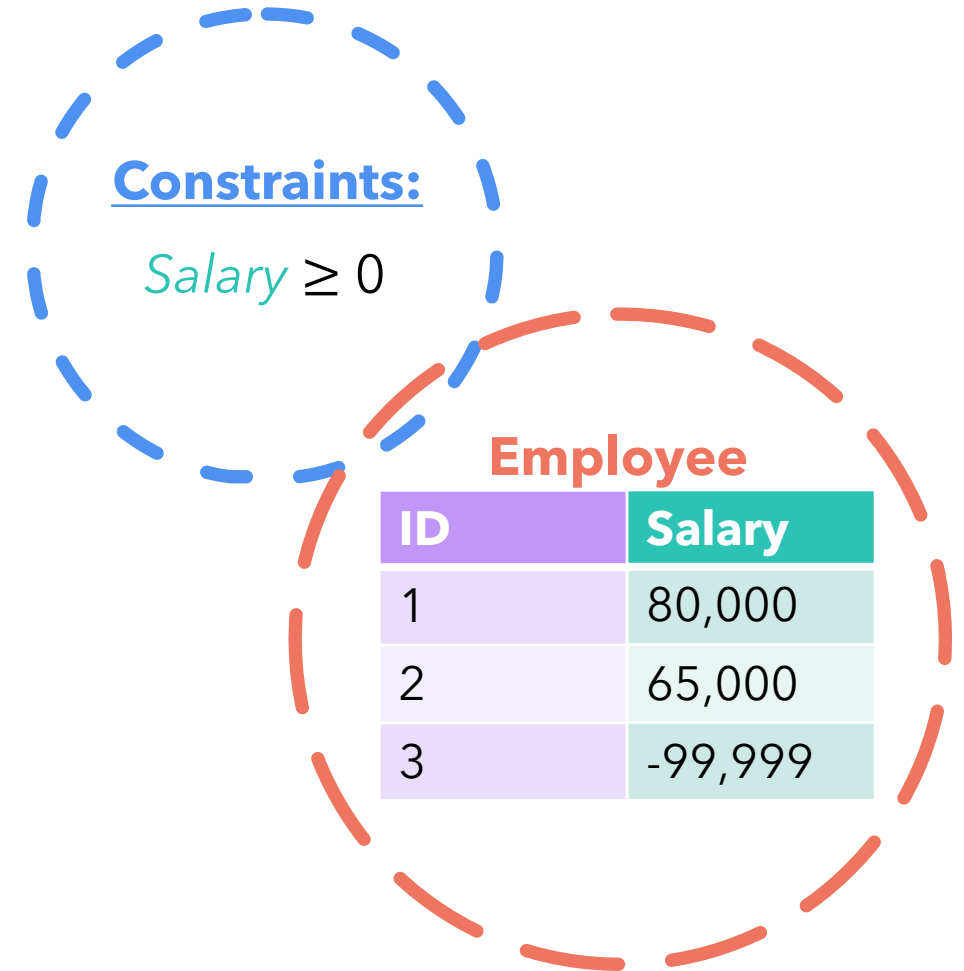
We have seen the problem of inconsistencies before...

- Data management community spent *decades solving this problem using **declarative constraints***



Approach to eliminating inconsistencies

- Given: inconsistent dataset and declarative constraints
- Goal: give information that is consistent with **constraints**

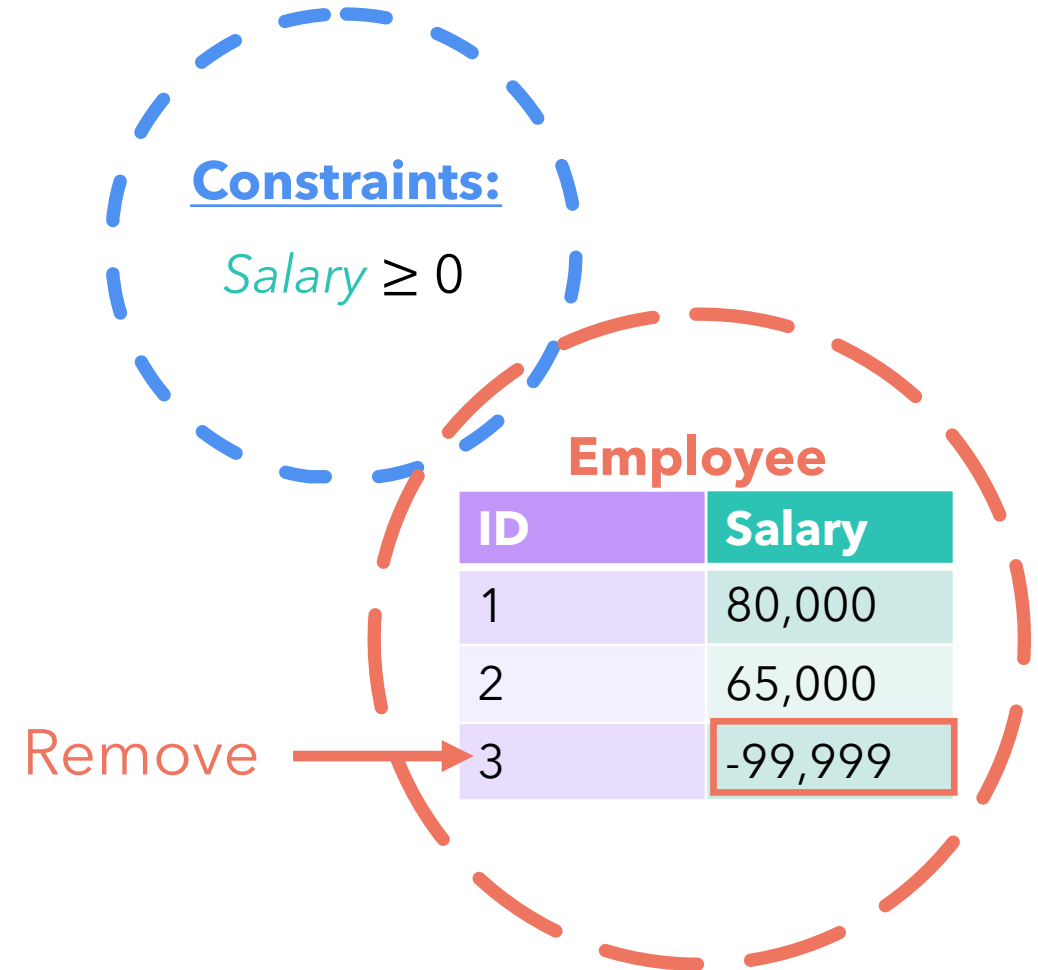


Eliminating inconsistencies: data cleaning

- Data repair
 - Example: remove last row

Employee

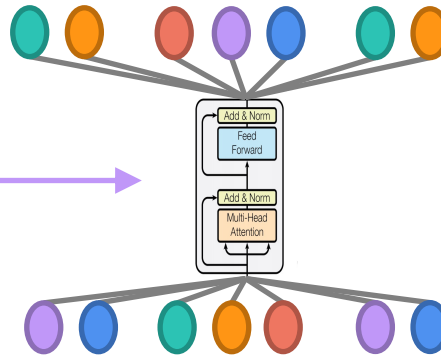
ID	Salary
1	80,000
2	65,000



Data cleaning applied to LLMs

- Information stored implicitly through weights

Constraints




Modify training data, architecture, or weights through pre-training or fine-tuning

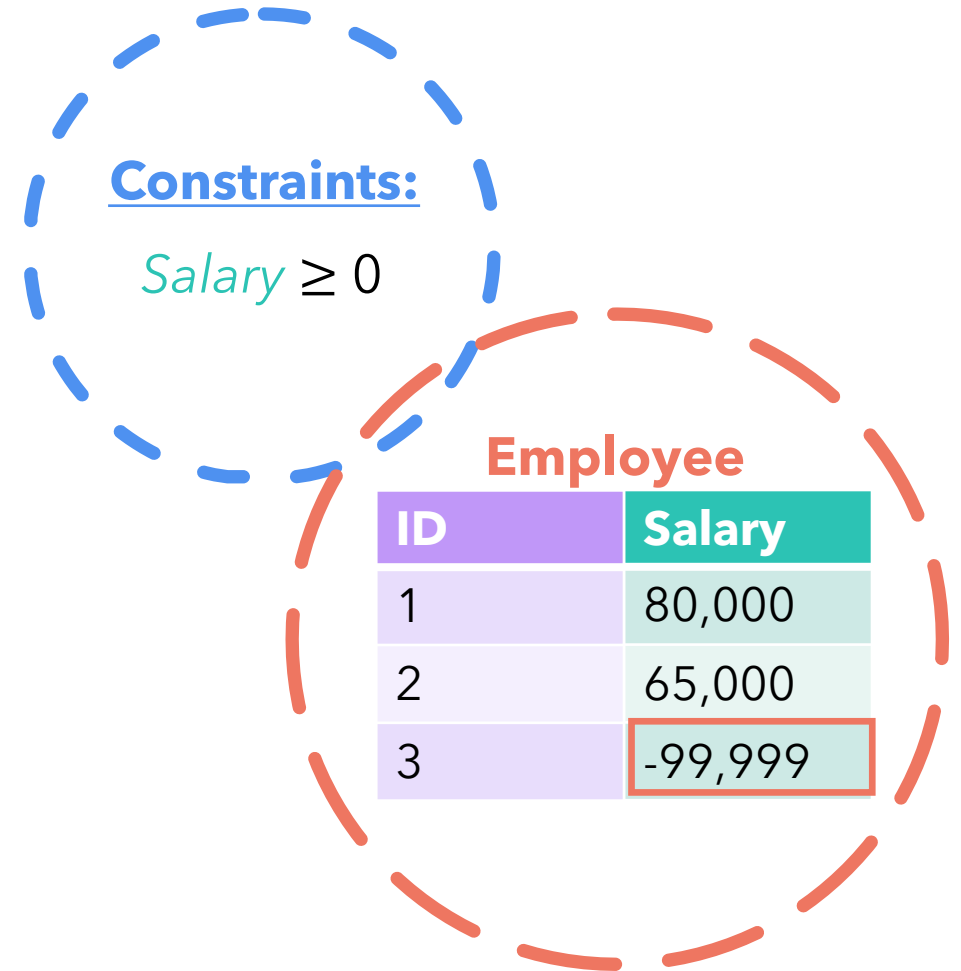
Limitation: Training is **expensive**

Eliminating inconsistencies: consistent query answering

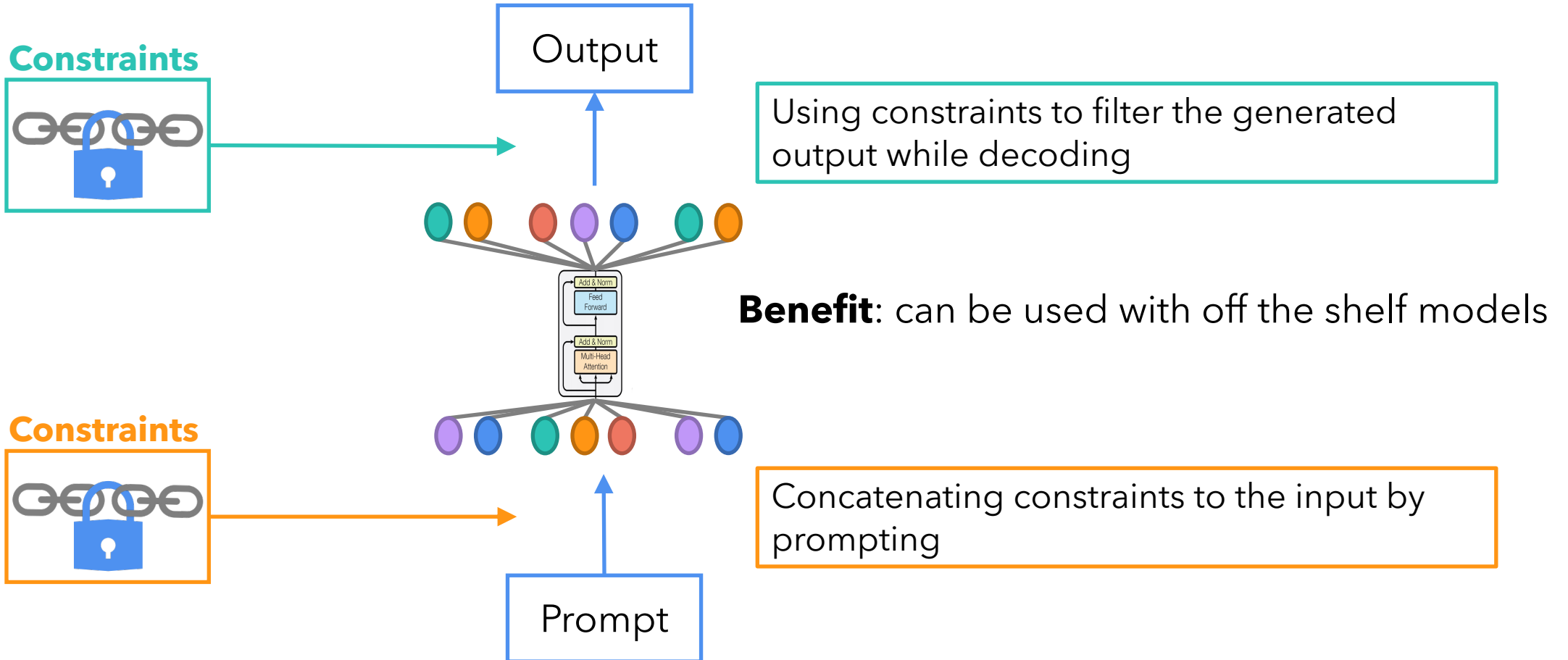
- Modify query to get consistent results

 Select * from
Employee
where Salary ≥ 0

ID	Salary
1	80,000
2	65,000

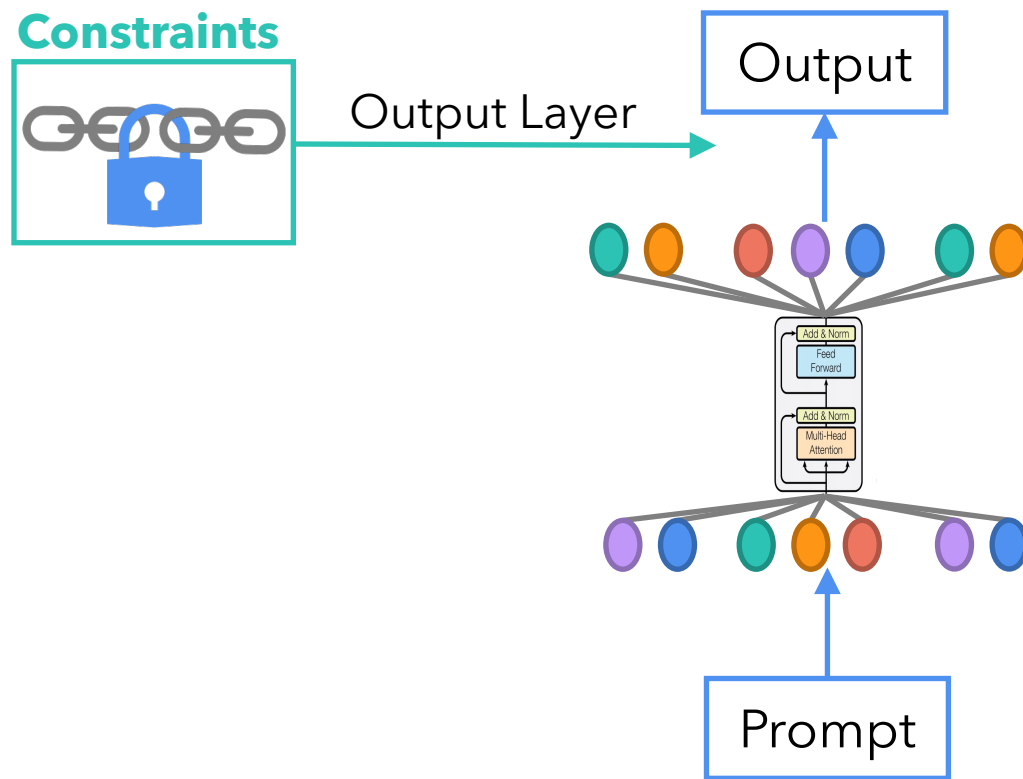


Consistent query answering applied to LLMs



There are methods to apply constraints at decoder

Constrained decoding



During generation, predicting the next token with constraints

We evaluate constrained decoding methods

- **LLM:** Llama-2
- **Dataset:** CommonGen
- **Constraint:** contains key words or their inflections
 - expressed in CNF
- **Constrained decoder:**

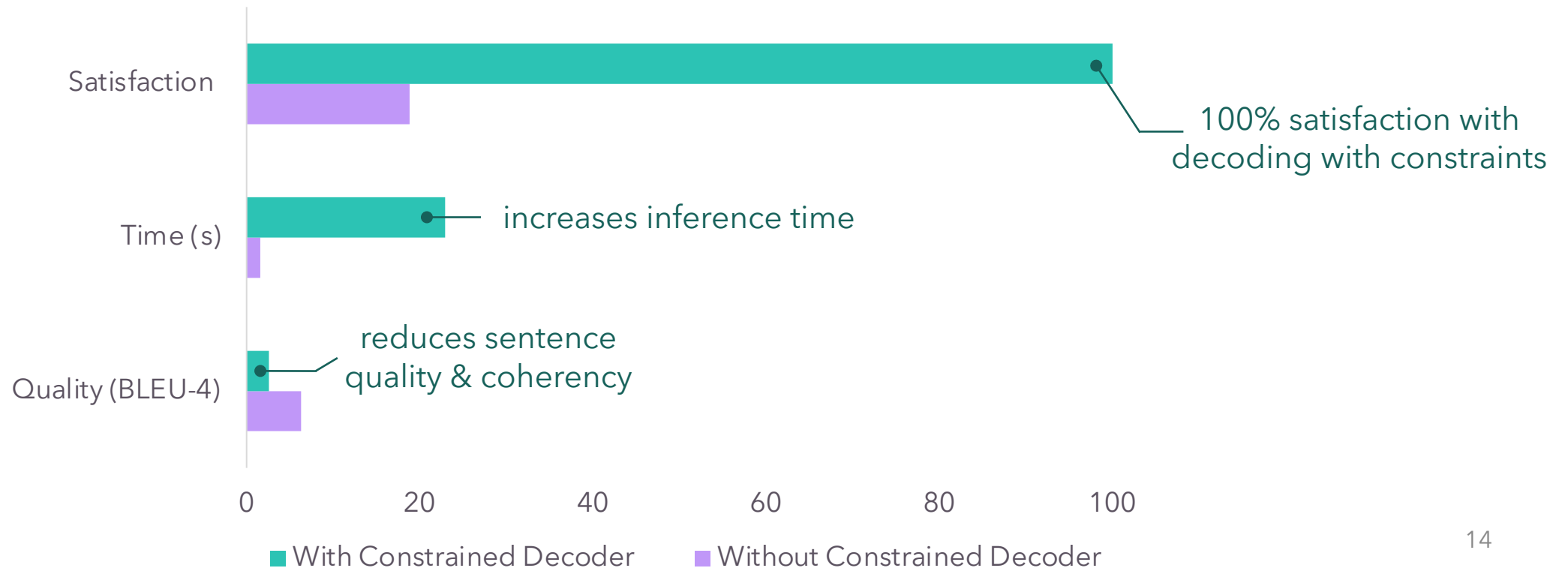
SMC

Sequential Monte Carlo

- Applies constraints to tokens using sequential Monte Carlo inference

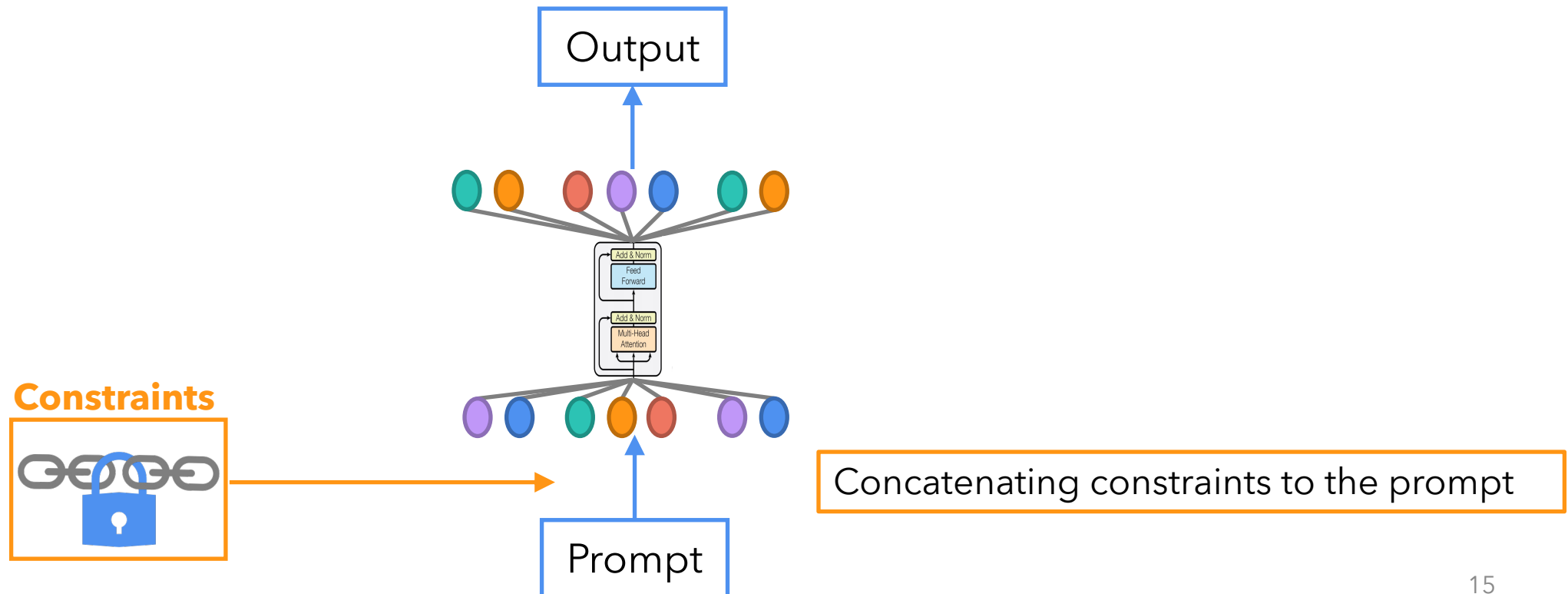
Empirical study: constrained decoding

- **Satisfaction:** *all* CNF clauses in constraint C are met
- **Time:** inference time (seconds)
- **Quality:** similarity between output and human references (BLEU-4)



Let's put constraints in the prompt: **constrained prompting**

- No overhead of inference time
- No quality overhead
 - Constrained decoders alter output distributions during generation



Challenges of using constrained prompting

- Limited **context length**
 - Constraints could be long
 - Domains often have multiple constraints
- Ensuring LLM **understands** constraints
 - Logical constraints may be too hard to understand

Constraint (CommonGen)

Write a sentence using the words (*word1a* or *word1b* or ...) and (*word2a* or *word2b* or ...) and ...

Abstraction: minimizing/generalizing constraints

- Reduces length
- Preserves or generalizes meaning
- Closer to natural language

Constraint (CommonGen)

Write a sentence using the words (*word1a* or *word1b* or ...) and (*word2a* or *word2b* or ...) and ...



Abstract Prompt

Given a set of words $x=[word1, word2, \dots]$, write a sentence using all words in x or inflections of x

In-context demonstrations: help LLM understand constraints

- In-context learning using a few examples
- Demonstration:
 - constraint
 - sentence satisfying the constraint

2-Shot

Q: constraint
A: satisfying sentence } Demo 2

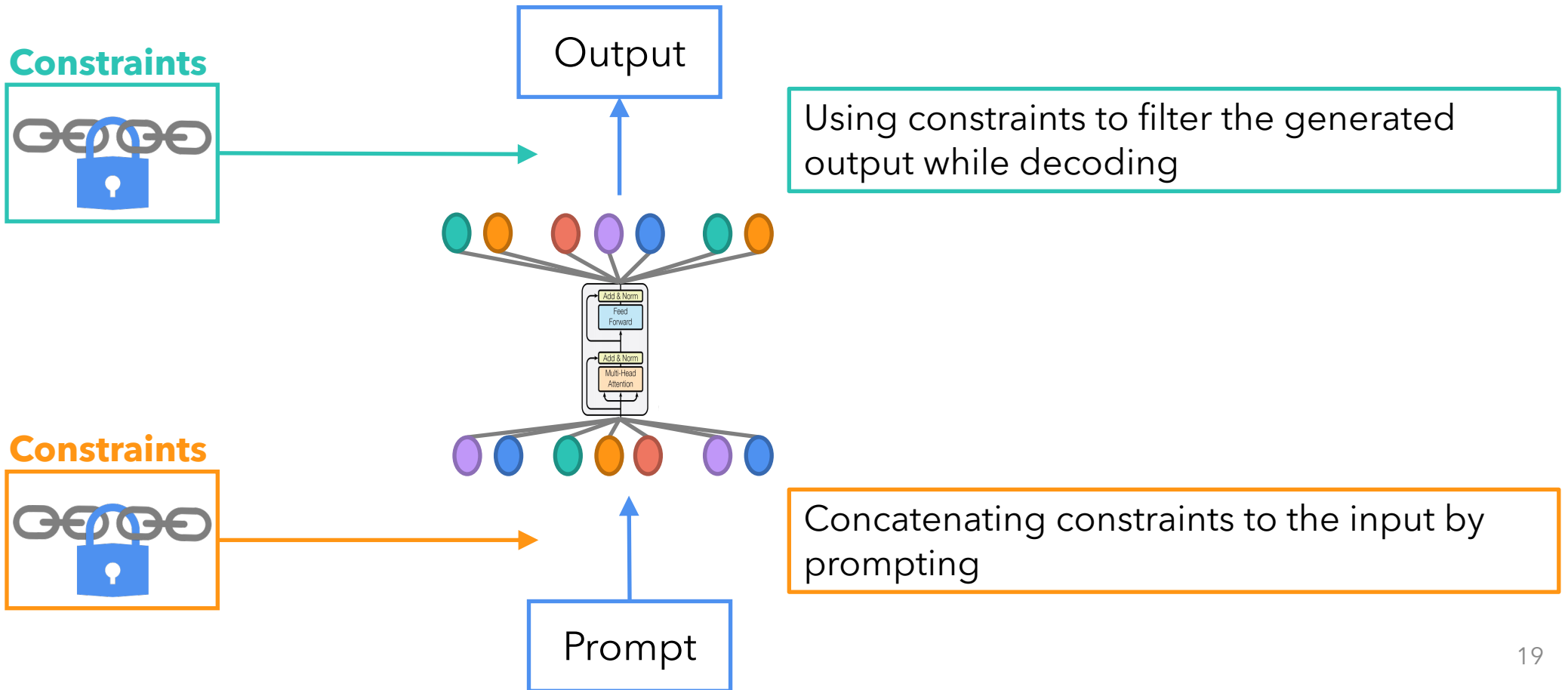
Q: constraint
A: satisfying sentence } Demo 1

Q: constraint
A:

2 demonstrations

Combining constrained prompting & decoding

- Guarantees satisfaction
- Possibly smaller search space for decoder
 - Faster inference time



Empirical study setup

- **LLM:** Llama-2
- **Dataset:** CommonGen
- **Constraint:** contains key words or their inflections
 - expressed in CNF
- **Constrained decoder:**

NL

NeuroLogic

- Applies constraints to tokens using a beam-based look ahead strategy

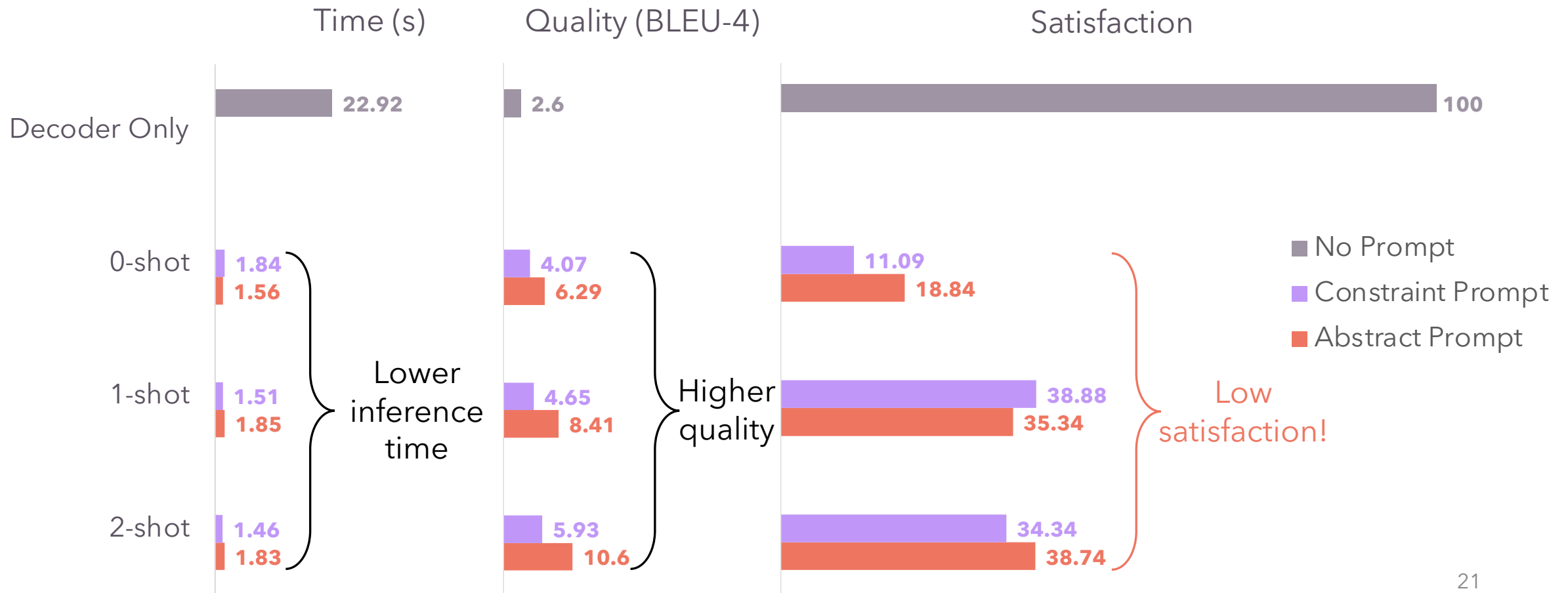
SMC

Sequential Monte Carlo

- Applies constraints to tokens using sequential Monte Carlo inference

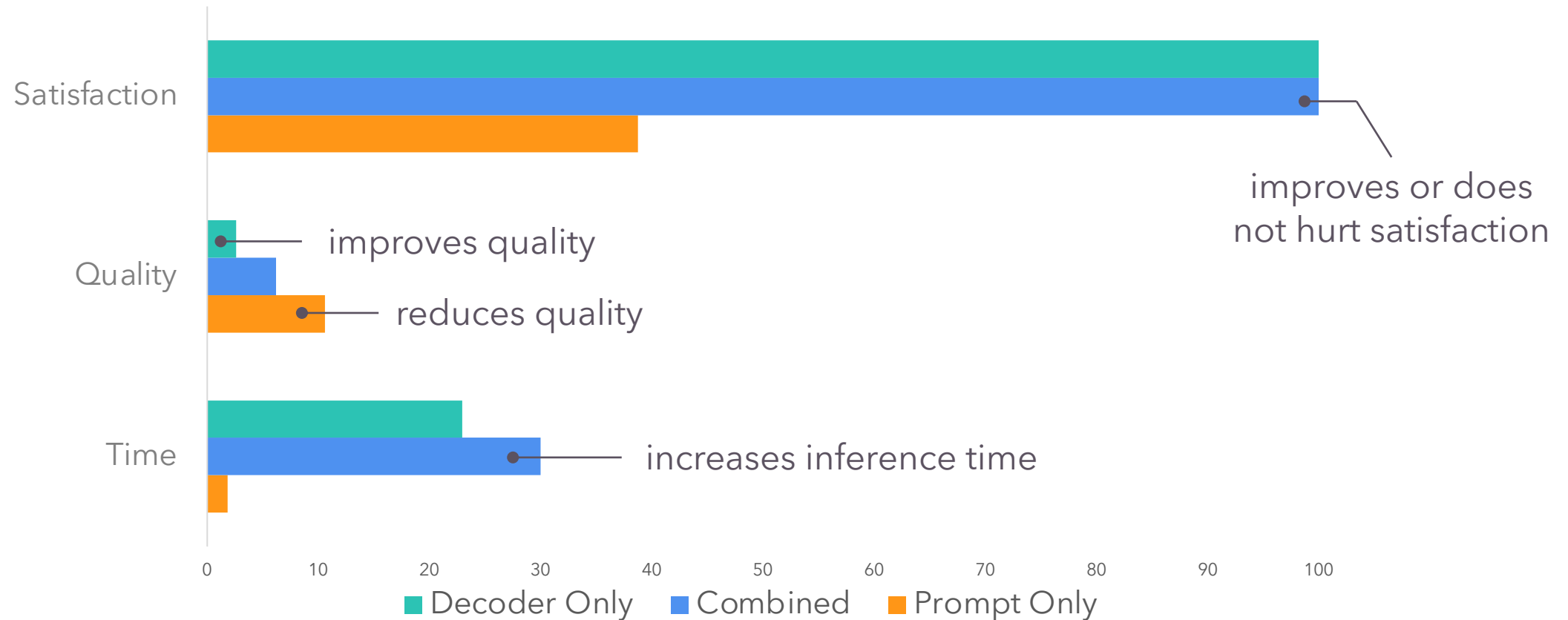
Constrained prompting delivers higher quality and inference time, but lower satisfaction

- SMC decoder

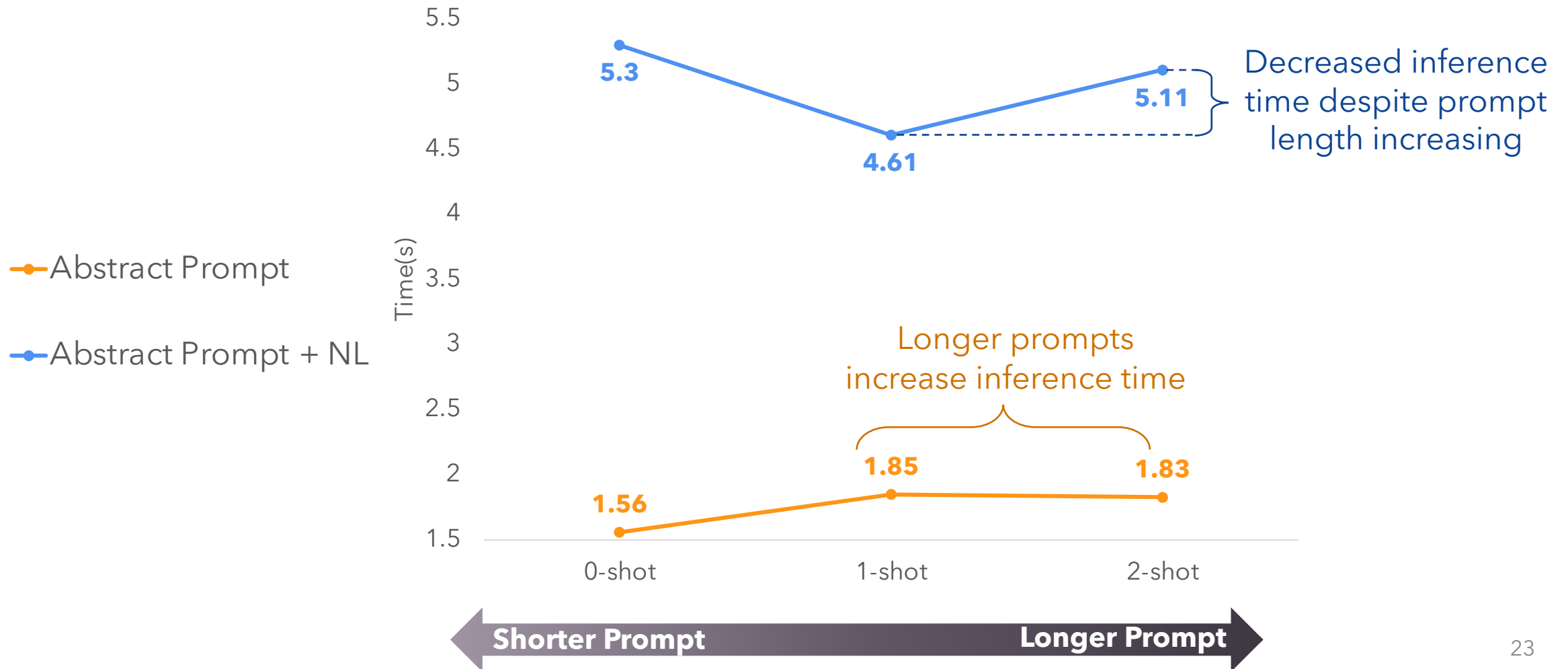


Constrained prompting & decoding improves satisfaction and quality in some cases, but hurts time

- Abstract prompt + 2-shot + SMC decoder



Constraint prompting can reduce the search space for decoders in some cases



Conclusion

1

We use declarative **constraints** to *eliminate* inconsistencies in LLMs

2

Constrained prompting *reduces* inconsistencies **efficiently** and with **high quality**

3

Constrained prompting and decoding achieve *higher* **satisfaction** and **quality**

4

Read our paper



5

Future work

- constraint abstraction
- where to apply which constraint