

How robustly can LLMs process lengthy context?

Presenter: Hanxiang Qin, Zhengguang Wang

[LLMs Get Lost In Multi-Turn Conversation](#)

[How Many Instructions Can LLMs Follow at Once?](#)



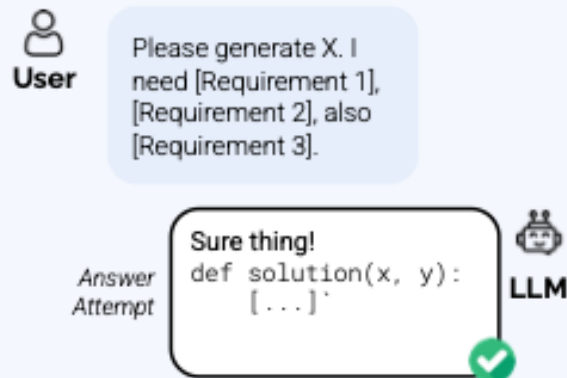
JOHNS HOPKINS
UNIVERSITY

LLMs get Lost in Multi-turn Conversation

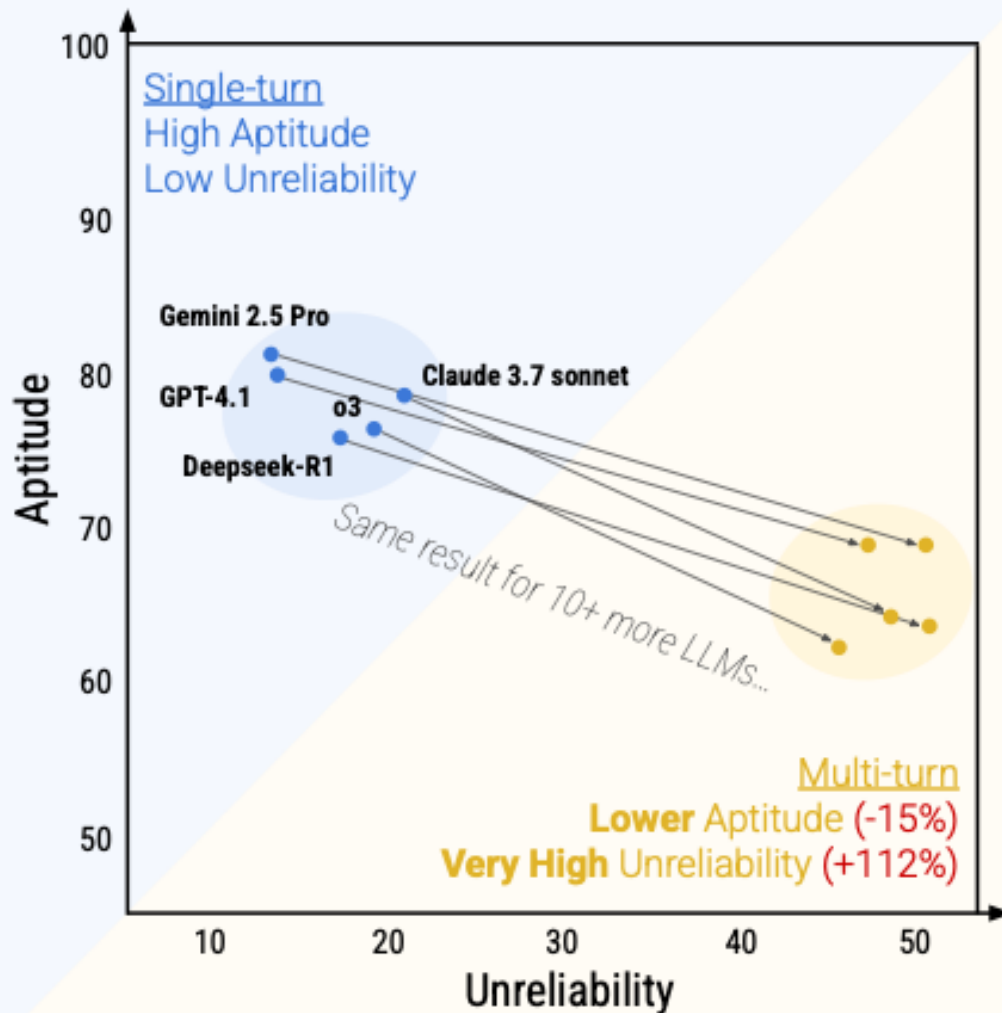


Motivation

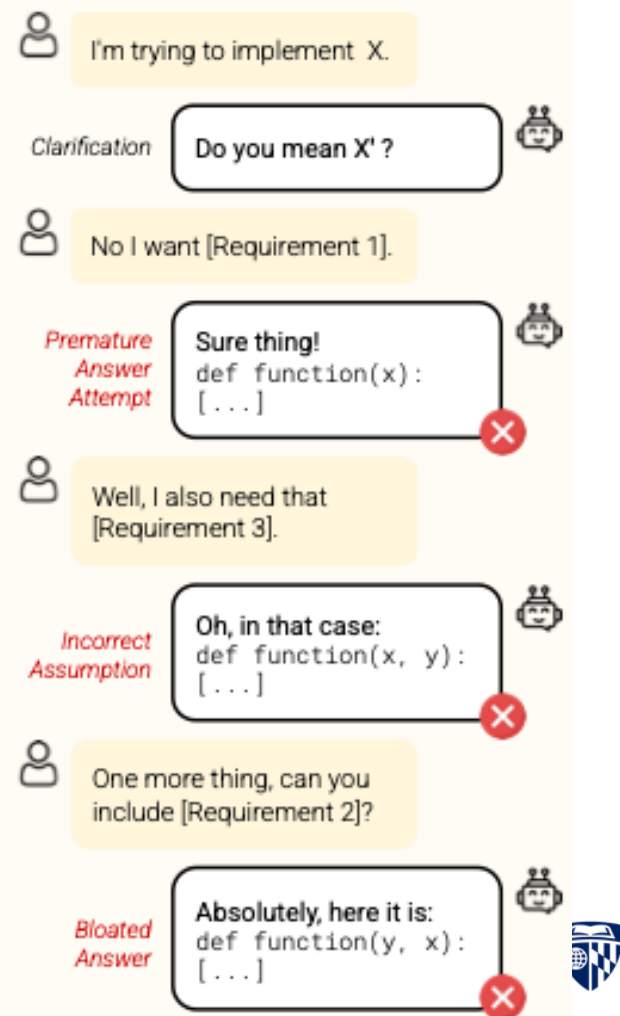
Single-Turn Fully-Specified



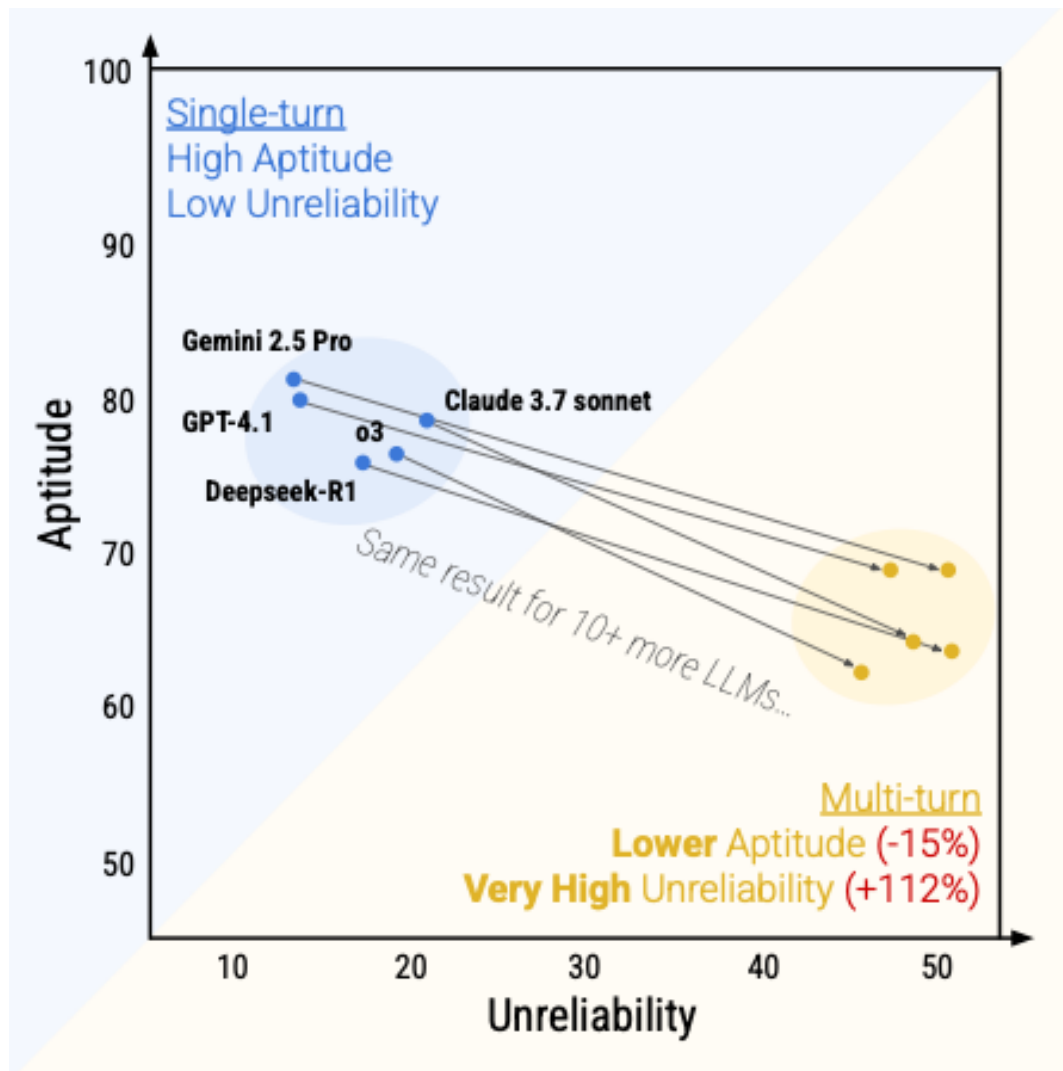
LLMs get Lost in Conversation



Multi-Turn Underspecified



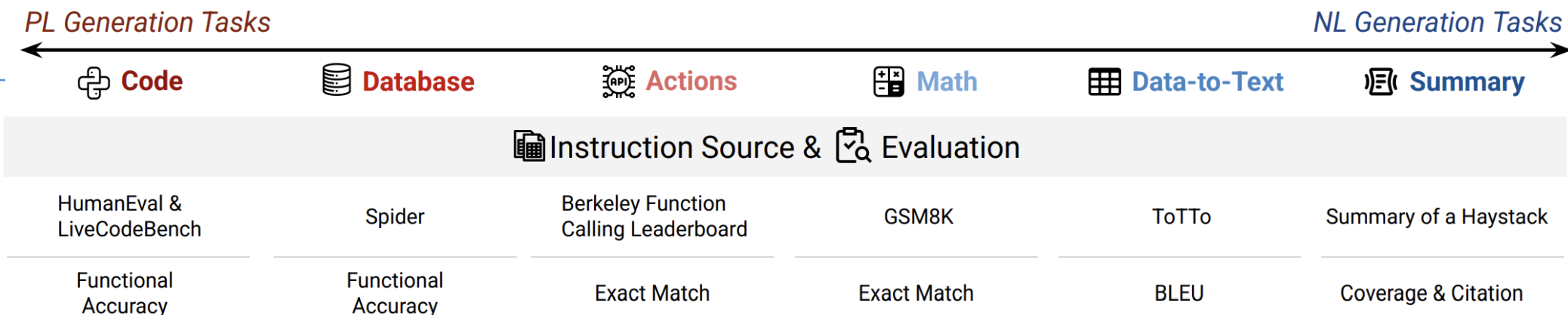
Key Finding



Lost in conversation

- An average performance drop of **39%** across 6 generation tasks
- A loss in Aptitude (**-15%**)
- A significant increase in Unreliability (**+112%**).

Metrics



Averaged Performance – Averaged scores

$$\bar{P} = \sum_{i=1}^N S_i / N$$

Binary
{0, 100}

Continuous
[0 - 100]

LLM-as-a-judge

Aptitude – 90th percentile score

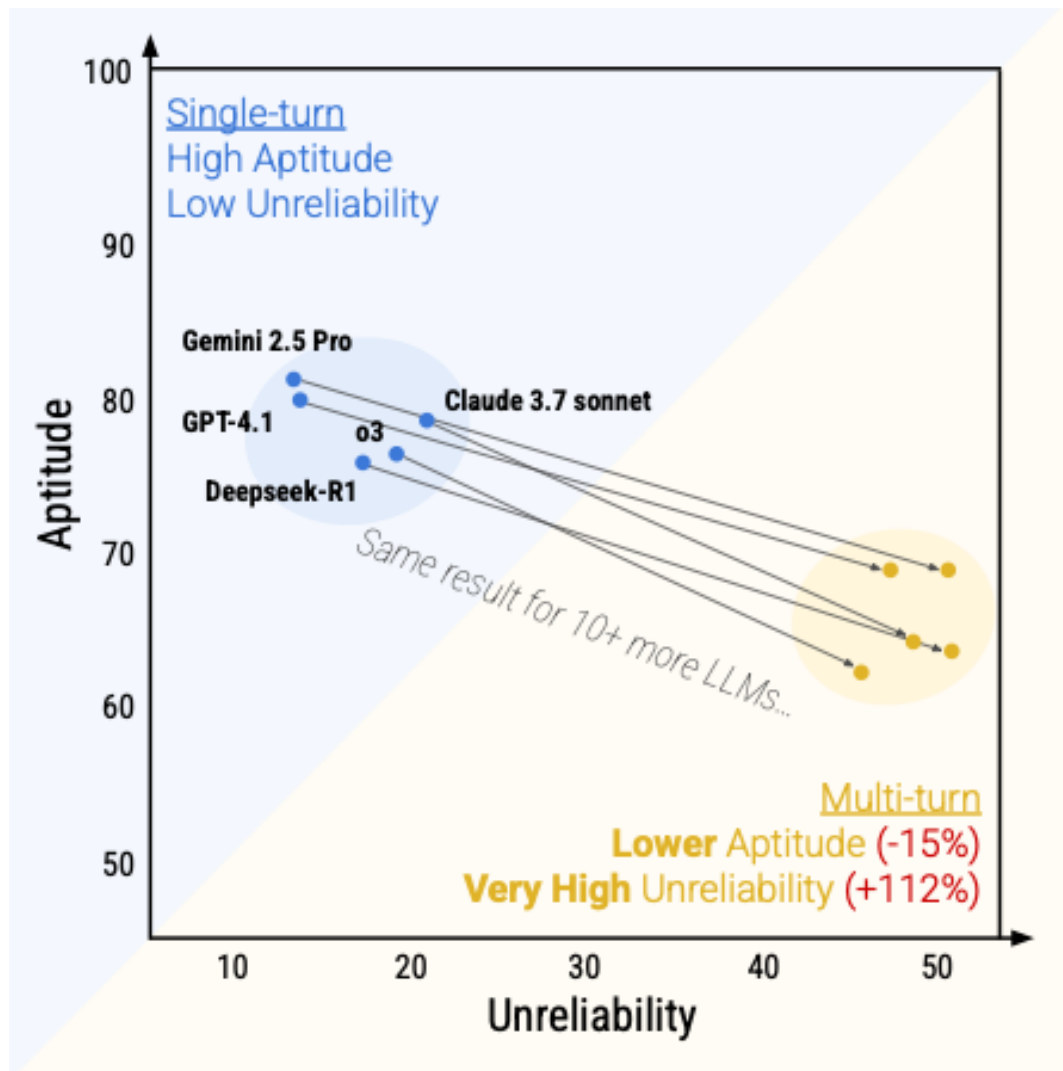
$$A^{90} = \text{percentile}_{90}(S)$$

Unreliability – the difference between the 90th percentile score and 10th percentile score

$$U_{10}^{90} = \text{percentile}_{90}(S) - \text{percentile}_{10}(S).$$



Key Finding



Lost in conversation

- An average performance drop of **39%** across 6 generation tasks
- A loss in Aptitude (**-15%**)
- A significant increase in Unreliability (**+112%**).

Methodology – Instruction "Sharding"

Fully-Specified Instruction (original)

Jay is making snowballs to prepare for a snowball fight with his sister. He can build 20 snowballs in an hour, but 2 melt every 15 minutes. How long will it take before he has 60 snowballs?

(a) Original GSM8K instruction.

Sharded Instruction (based on original)

Shard 1: How long before Jay's ready for the snowball fight?

Shard 2: He's preparing for a snowball fight with his sister.

Shard 3: He can make 20 snowballs per hour.

Shard 4: He's trying to get to 60 total.

Shard 5: The problem is that 2 melt every 15 minutes.

(b) Equivalent Sharded Instruction.

Properties

- Clear Initial Intent
- Order Insensitive
- Information Preservation
- Minimal Transformation
- Maximal Sharding

Methodology – Instruction "Sharding"

PL Generation Tasks

NL Generation Tasks



Code



Database



Actions



Math



Data-to-Text



Summary

Fully-Specified Instruction

Write the Python function

```
def below_zero(ops):  
    """ You're given a list of  
    deposits & withdrawals on a bank  
    account that starts with balance  
    of 0. Detect if at any point the  
    balance < 0, if so return True,  
    otherwise False.  
    >>> [2 example uses]  
    """
```

Write an SQL query for:

Find the names of stores
whose number products is
more than the average number
of products per store.

[Schema]

Write API function calls:

Play songs from the artists
Taylor Swift and Maroon 5,
with a play time of 20 minutes
and 15 minutes respectively,
on Spotify.

[API spec]

Solve this problem:

Josh decides to try flipping a
house. He buys a house for
\$80k and then puts in \$50k in
repairs. This increased the
value of the house by 150%.
How much profit did he make?

Write a Table caption:

[Highlighted Table HTML]
The table comes from [URL]
about the 2000 Americas
Cricket Cup.
I've highlighted some cells.

Write a Summary:

About the following 12
documents, on the following
query: [QUERY]

Documents:
[Documents 1-12]

Sharded Instructions

Write me a function below_zero
to find out if account is ever <0

Input's a list of ints that are
transactions.

Balance is 0 at the start.

Return True if balance's ever <0,
o/w return False

[Example 1]

[Example 2]

Let's find large stores

Maybe we can define store
size based on its number of
products

A store is large if it has more
than the average number of
products across all stores.

Only return store names &
order doesn't matter

Let's make a 35-min playlist

Let's add Taylor Swift songs

Let's also put some Maroon 5

I prefer Taylor Swift, let's do
20 minutes of that

So that leaves 15 minutes
for Maroon 5

My friend Josh sold his home. I
want to know how much profit
he made.

He bought it for \$80,000

He spent \$50k on repairs

The house value increased by
150%

That's all I know. What's his
profit?

I'm giving you a table, please
write a sentence describing
it. [Table HTML]

Actually focus on these
highlighted cells:
[Highlighted Table HTML]

It came from a page about the
2000 Americas Cricket Cup

The exact page is [URL]

I need a summary of 12
documents, on query: [QUERY]
I'll give the docs as I get them,
consider all of them.

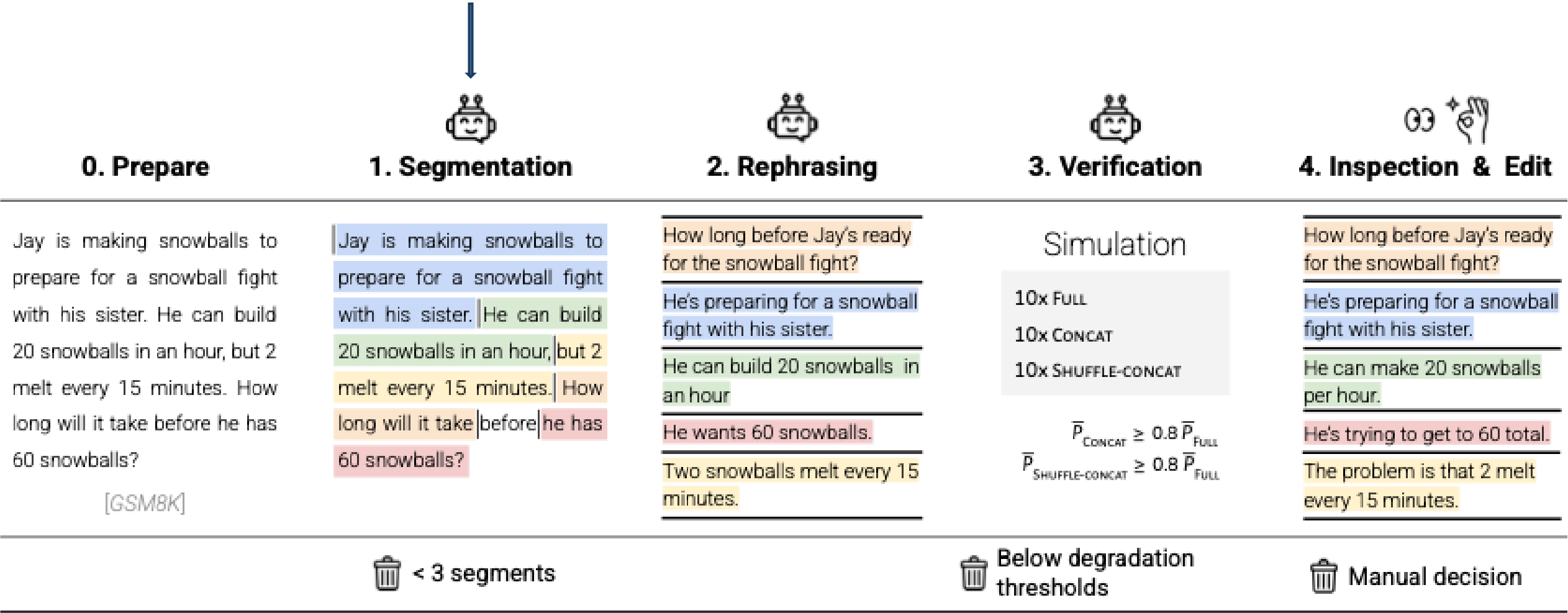
Docs 1-2: [Documents 1-2]

Just got four more.
Docs 3-6: [Documents 3-6]

Here's a new batch.
Docs 7-10: [Documents 7-10]

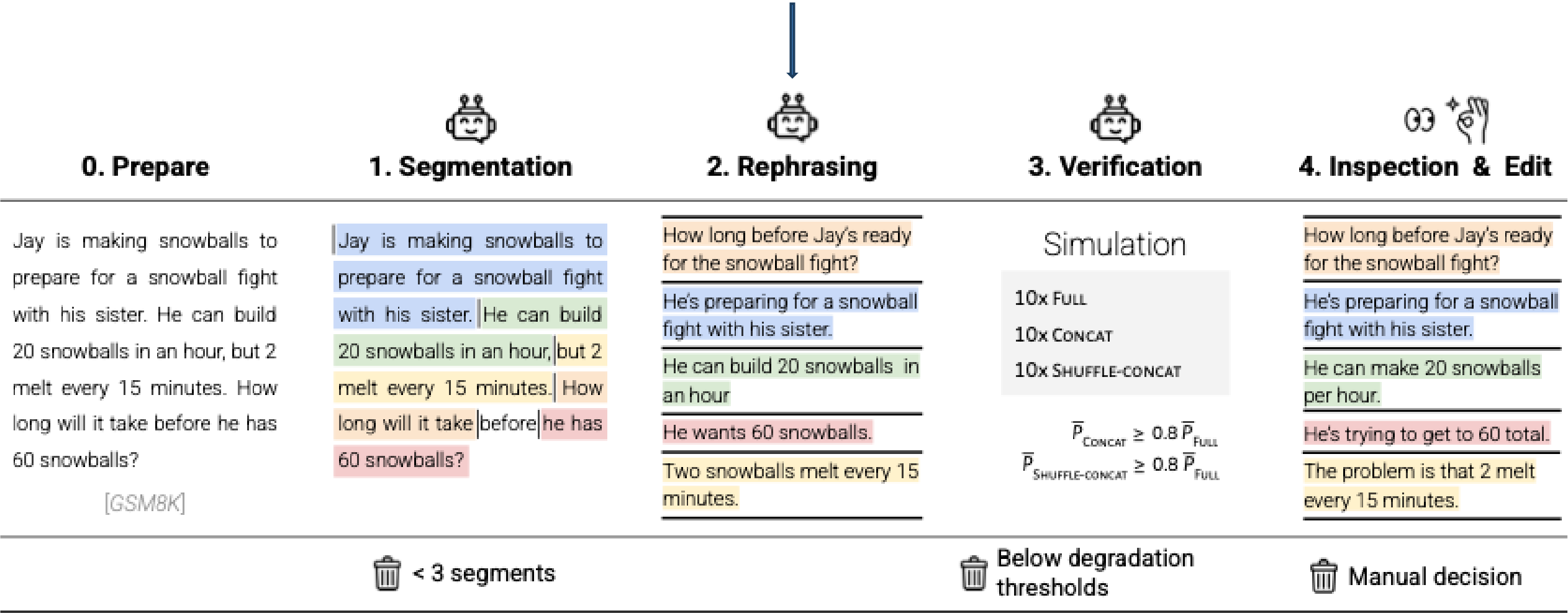
I've got two more.
Docs 11-12: [Documents 11-12]

Methodology – Semi-Automatic Sharding



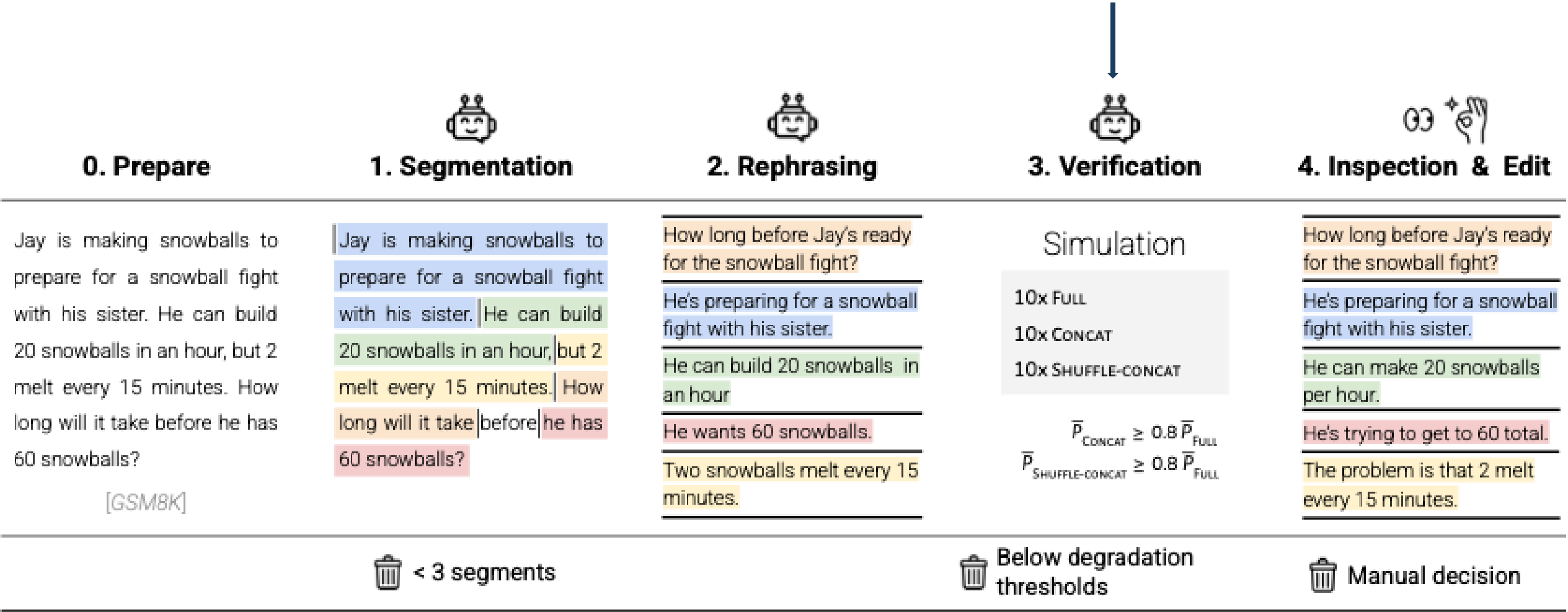
P1:Information Preservation | P2:Clear Initial Intent | P3:Order Insensitive |
P4:Maximal Sharding | P5:Minimal Transformation

Methodology – Semi-Automatic Sharding



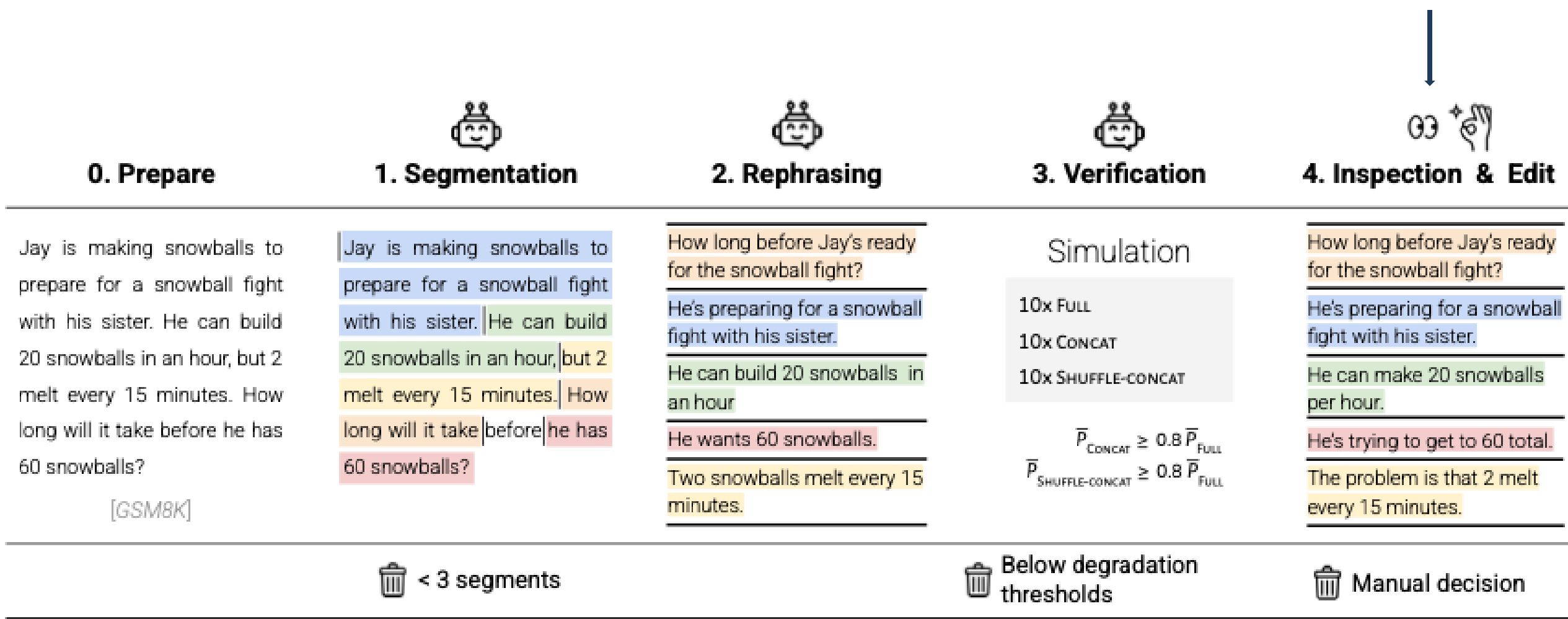
P1:Information Preservation | P2:Clear Initial Intent | P3:Order Insensitive |
P4:Maximal Sharding | P5:Minimal Transformation

Methodology – Semi-Automatic Sharding



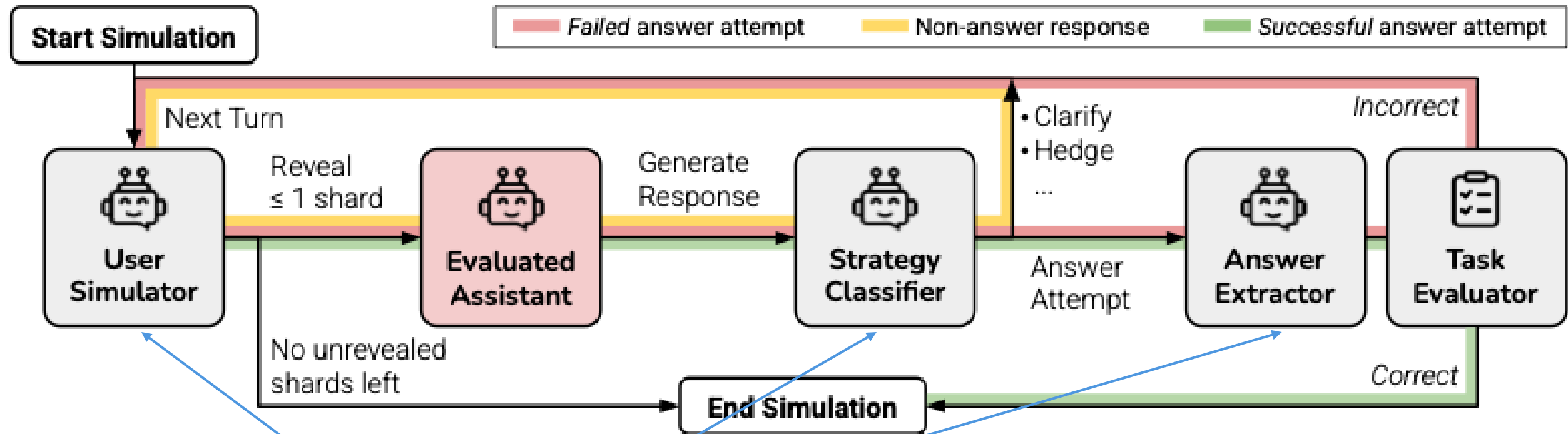
P1:Information Preservation | P2:Clear Initial Intent | P3:Order Insensitive |
P4:Maximal Sharding | P5:Minimal Transformation

Methodology – Semi-Automatic Sharding



P1:Information Preservation | P2:Clear Initial Intent | P3:Order Insensitive |
P4:Maximal Sharding | P5:Minimal Transformation

Methodology – Conversation Simulation

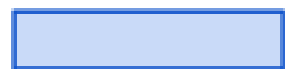


Prompt-based GPT-4o-mini

Methodology – Conversation Types

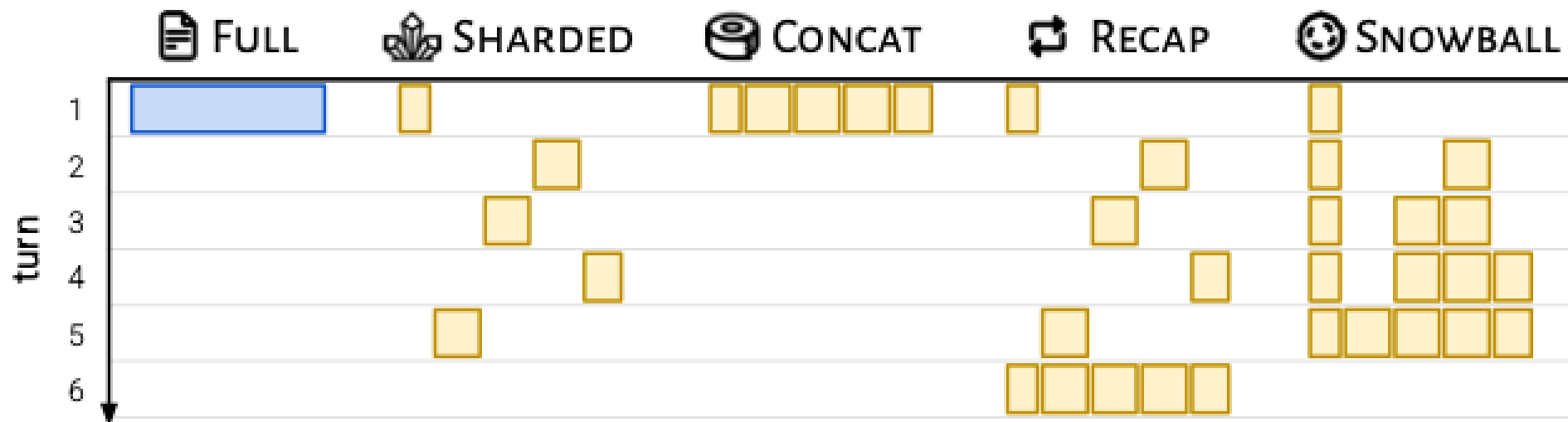
Instruction Sharding

Fully-specified
Single-Turn








































Sharded
Multi-Turn

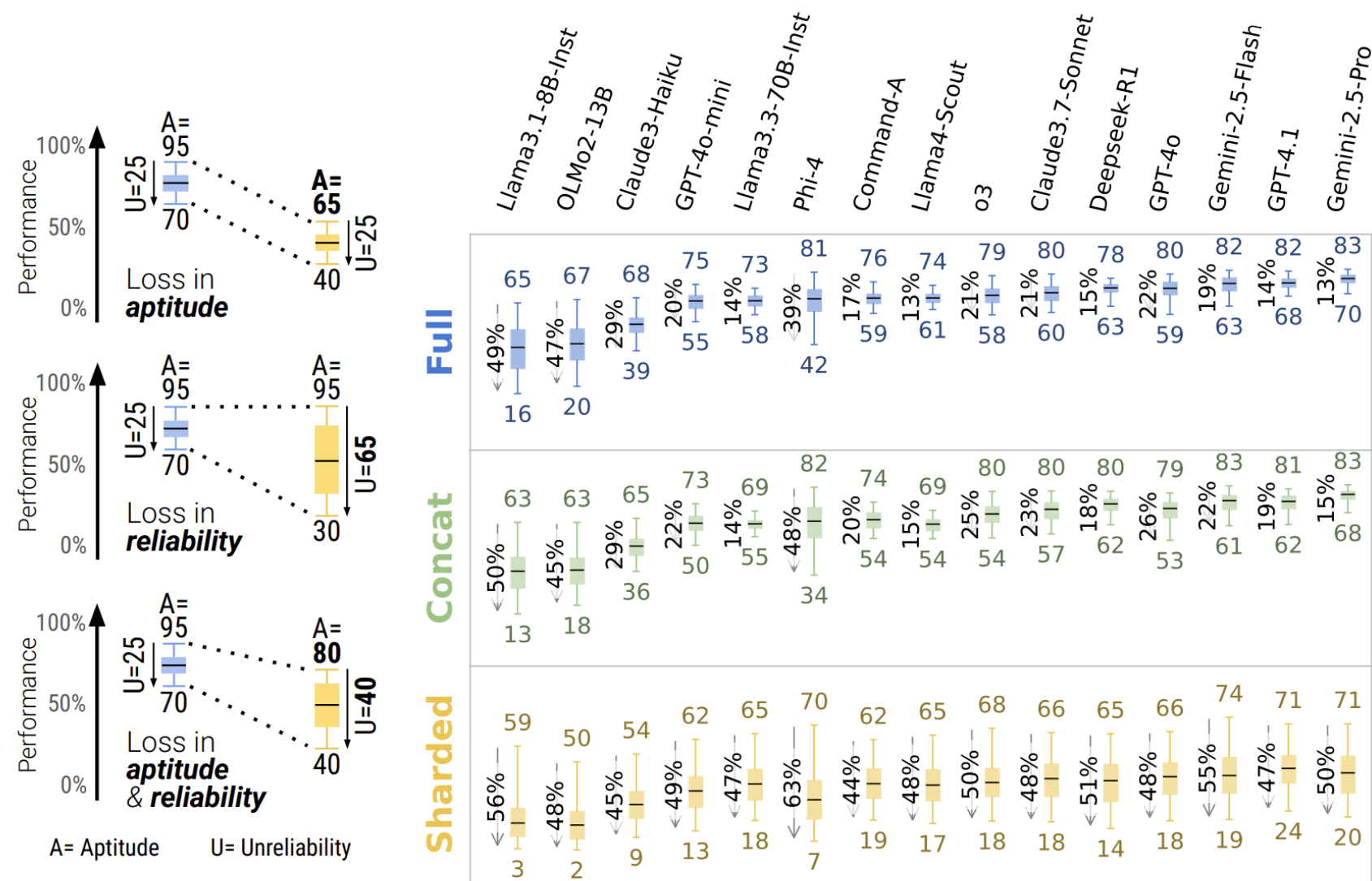
Conversation Simulation Types



Results – Averaged Performance

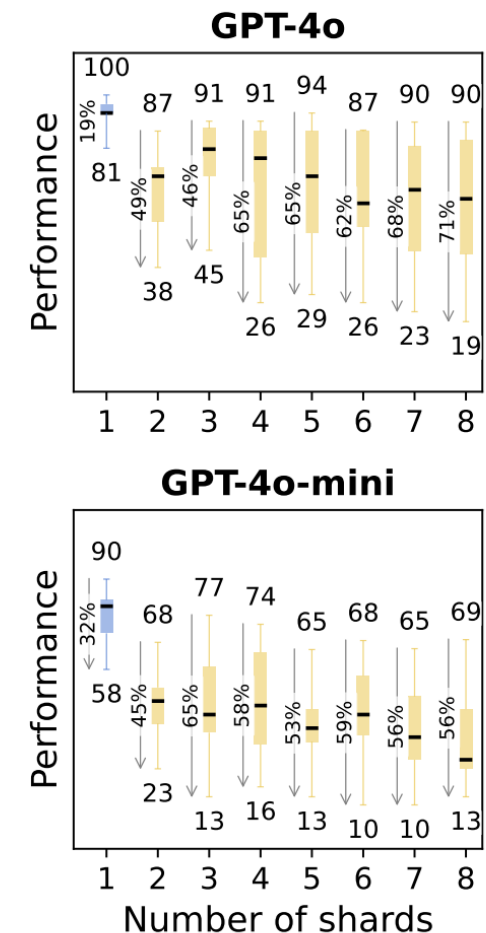
Model	FULL						CONCAT						SHARDED						Overall	
																			 / 	 / 
 3.1-8B	27.4	64.1	82.9	13.7	63.9	7.6	21.2	47.7	83.0	15.7	62.6	6.5	21.7	25.9	45.5	13.3	37.4	3.4	91.6	62.5
 OLMo2	18.8	54.8	56.1	17.2	80.0	-	16.3	40.5	49.8	14.3	80.1	-	14.4	22.4	13.8	9.0	46.3	-	86.5	50.5
 3-Haiku	44.8	85.0	83.5	29.8	73.9	11.6	36.3	76.5	80.2	30.1	76.1	9.2	31.5	31.8	55.9	18.6	47.1	1.6	91.6	52.4
 4o-mini	75.9	89.3	94.1	35.9	88.1	14.9	66.7	90.7	92.2	31.2	88.0	12.5	50.3	40.2	52.4	19.8	58.7	7.2	93.0	56.2
 3.3-70B	72.0	91.1	95.0	34.1	91.7	15.8	52.7	87.9	97.0	32.0	91.8	14.7	51.6	35.4	71.0	22.4	61.5	10.5	93.2	64.2
 Phi-4	53.2	87.6	82.7	23.9	89.2	-	48.4	79.6	76.0	28.6	90.4	-	39.1	33.1	34.1	23.2	52.5	-	99.0	61.7
 CMD-A	72.0	91.9	98.5	27.7	94.5	24.3	61.6	86.1	98.4	33.2	91.9	21.3	44.9	33.6	72.0	27.9	66.0	4.9	97.3	60.4
 4-Scout	73.9	92.7	98.0	35.2	96.3	13.7	60.3	81.5	98.3	28.2	92.9	13.7	46.4	27.1	69.9	26.1	67.0	12.3	91.0	66.1
 o3	86.4	92.0	89.8	40.2	81.6	30.7	87.2	83.3	91.5	39.4	80.0	30.4	53.0	35.4	60.2	21.7	63.1	26.5	98.1	64.1
 3.7-Sonnet	78.0	93.9	95.4	45.6	85.4	29.3	76.2	81.5	96.0	53.3	87.2	28.9	65.6	34.9	33.3	35.1	70.0	23.6	100.4	65.9
 R1	99.4	92.1	97.0	27.0	95.5	26.1	97.1	89.9	97.0	36.7	92.9	24.4	70.9	31.5	47.5	20.0	67.3	17.2	103.6	60.8
 4o	88.4	93.6	96.1	42.1	93.8	23.9	82.9	91.7	97.1	32.2	91.9	23.9	61.3	42.3	65.0	20.5	67.9	10.6	94.5	57.9
 2.5-Flash	97.0	96.3	88.4	51.2	90.6	29.1	92.5	95.5	89.2	51.9	88.4	29.4	68.3	51.3	42.6	31.0	66.1	26.1	99.3	65.8
 4.1	96.6	93.0	94.7	54.6	91.7	26.5	88.7	86.5	98.5	54.4	89.7	26.8	72.6	46.0	62.9	28.6	70.7	13.3	97.9	61.8
 2.5-Pro	97.4	97.3	97.8	54.8	90.2	31.2	95.7	94.9	98.1	56.9	89.3	31.8	68.1	43.8	36.3	46.2	64.3	24.9	100.1	64.5

Results – Box-plot Visualization



(a) Visualizing Aptitude and Unreliability.

(b) Observed Model Degradations



(c) Gradual Sharding Results



Why Do Models Get Lost?

- **Premature Answer Attempts:** Models rush to give a full solution early on, making incorrect assumptions that they fail to correct later.

Model	Conversation Progress At First Answer Attempt				
	0-20%	20-40%	40-60%	60-80%	80-100%
<i>First answer attempt is ...</i>	earliest	early	midway	late	latest
∞ 3.1-8B	16.1	24.0	35.3	39.6	39.7
✿ OLMo2	17.6	32.7	37.7	47.3	26.4
AI 3-Haiku	27.1	35.6	47.4	58.9	70.3
🌀 4o-mini	30.2	39.2	48.4	58.2	59.9
∞ 3.3-70B	33.3	40.1	51.2	60.0	69.3
🌈 Phi-4	25.7	33.1	47.0	53.0	57.9
🌿 CMD-A	38.0	42.9	56.5	65.5	73.5
∞ 4-Scout	39.8	36.8	51.0	57.9	64.8
🌀 o3	21.0	37.9	51.9	58.4	68.0
AI 3.7-Sonnet	29.2	35.6	55.3	68.0	71.6
🦋 R1	39.5	43.1	53.5	66.4	50.2
🌀 4o	36.0	41.4	56.2	65.6	90.4
🔹 2.5-Flash	39.0	48.6	60.2	70.8	74.6
🌀 4.1	33.9	52.7	60.6	69.0	78.6
🔹 2.5-Pro	41.1	45.7	53.5	64.6	63.8
Average	30.9	40.5	51.7	60.4	64.4

Table 6: Averaged performance (\bar{P}) breakdown, based on how early in the conversation the LLM makes its first answer attempt. Analysis conducted on simulations of two tasks: Code and Math.

Why Do Models Get Lost?

- **Answer Bloat:** Models overly rely on their previous incorrect attempts, leading to final answers that are needlessly long and complex compared to single-turn solutions.

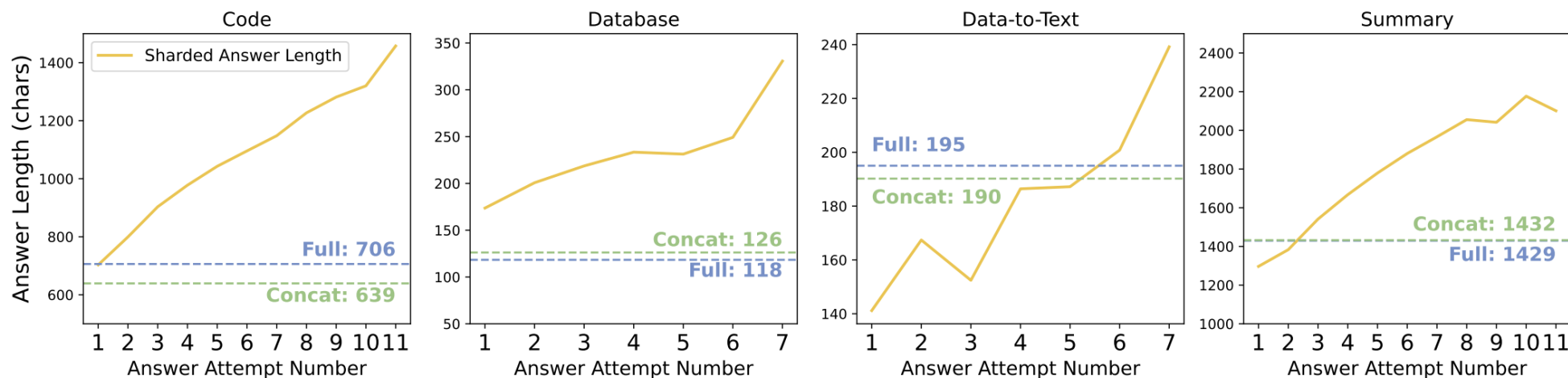


Figure 9: Average length (in number of characters) of answer attempts across four tasks (Code, Database, Data-to-text, and Summary) in SHARDED conversations. Answer attempts in the FULL and CONCAT settings tend to be shorter on average than those from SHARDED setting. SHARDED answer attempts increase in length as the LLMs make more answer attempts.



Why Do Models Get Lost?

- **Loss-in-Middle-Turns:** In long conversations, models tend to focus on information from the first and last turns, forgetting details provided in the middle.

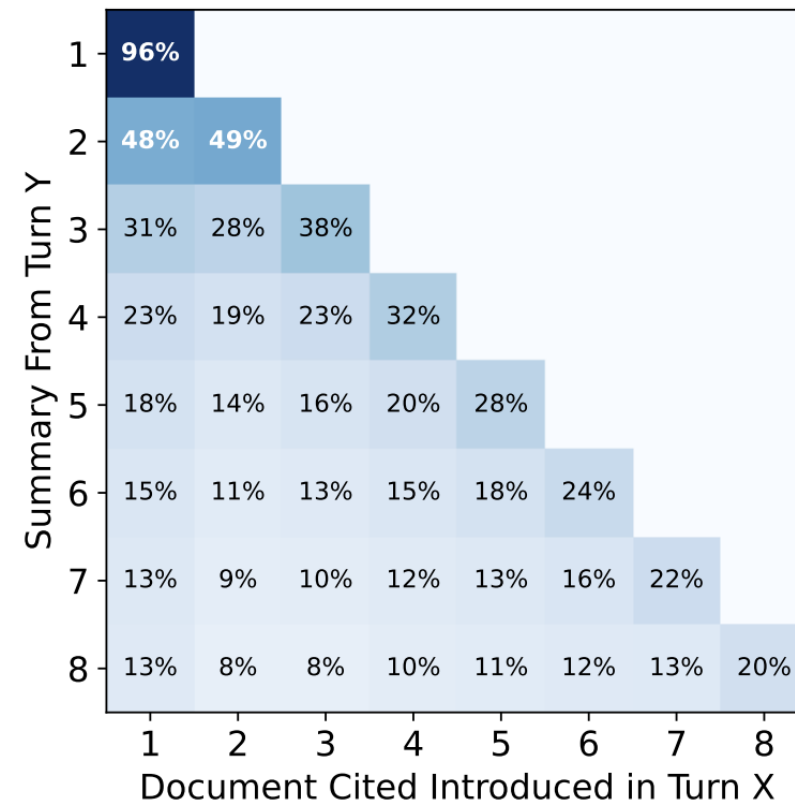


Figure 10: Analysis of citation patterns in summaries generated by LLMs with the SHARDED simulation. At each turn, the LLM generates an updated summary (y-axis), which includes citations from the documents that have been revealed up to this turn. Percentages in a row do not add up to 100% due to citation hallucinations that occur for some models.

Why Do Models Get Lost?

- **Overly Verbose Responses:** Longer assistant responses are correlated with lower performance, likely because they introduce more self-made assumptions that confuse the model.

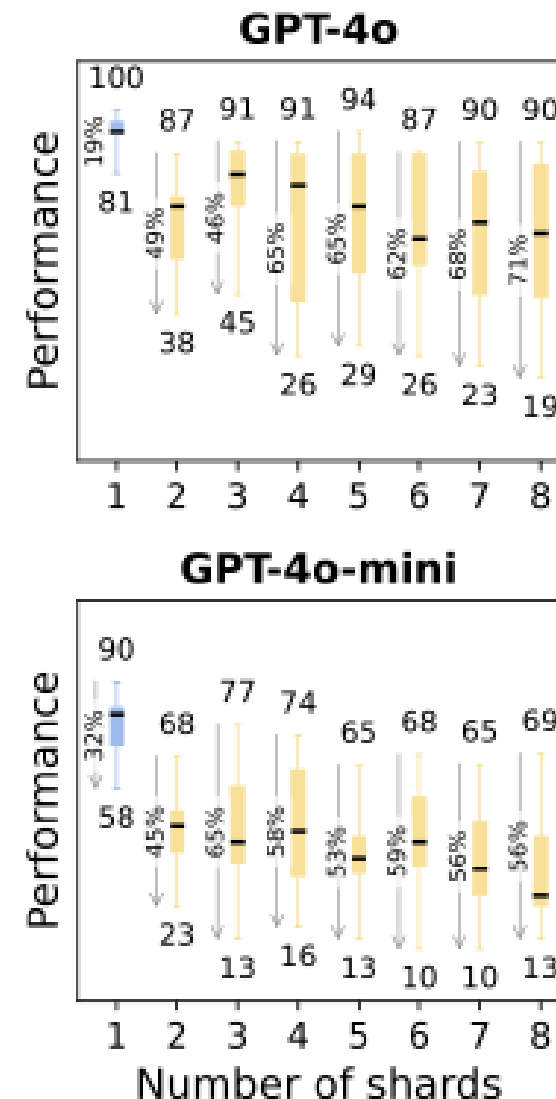
Task	Relative Assistant Verbosity				
	0-20%	20-40%	40-60%	60-80%	80-100%
<i>Assistants responses are ...</i>	shortest	short	median	long	longest
Code	55.3	52.3	48.9	46.9	42.5
Math	62.9	64.0	62.1	60.9	56.1
Database	43.8	40.0	37.3	34.3	31.3
Actions	41.5	49.6	54.2	53.6	50.8
Data-to-Text	25.0	24.3	24.0	23.1	21.8
Summary	15.4	14.7	13.5	12.0	10.3
Average	40.7	40.8	40.1	38.6	35.6

Table 7: Averaged performance (\bar{P}) of LLMs on the six experimental tasks, arranged based on model relative verbosity (length of response). Performance degrades when models generate longer responses on five of the six tasks.



What can we learn from this?

- **Does any amount of shards hurt?**
- Yes. The "gradual sharding" experiment shows performance drops significantly even in a simple two-turn conversation. Providing all information in a single turn is the only way to ensure high reliability.



What can we learn from this?

- **Can agent-like frameworks fix this?**
- Strategies like RECAP (summarizing all info at the end) and SNOWBALL (repeating all previous info each turn) help, but don't fully close the performance gap. Native multi-turn reliability is needed.










Model	Simulation Type				
					
 4o-mini	86.8	84.4	50.4	66.5	61.8
 4o	93.0	90.9	59.1	76.6	65.3

Table 2: Experimental Results with additional simulation types:  Recap and  Snowball. Both strategies involve repeating user-turn information to mitigate models getting lost in conversations.

What can we learn from this?

- **Can we just lower the temperature to $T=0$?**
- No. While lowering temperature improves reliability in single-turn settings, it is ineffective in multi-turn conversations. The unreliability remains high because small, early deviations cascade into wildly different conversational paths.

Simulation	🌀 4o-mini			🌀 4o		
	AT=1.0	AT=0.5	AT=0.0	AT=1.0	AT=0.5	AT=0.0
📄 FULL	16.0	15.0	6.8	17.8	8.0	2.8
🗨️ CONCAT	20.2	17.8	9.5	20.2	17.8	5.8
🏰 UT=1.0	49.8	46.8	51.0	41.0	43.8	31.8
🏰 UT=0.5	31.7	34.0	40.5	39.5	40.8	31.8
🏰 UT=0.0	38.5	28.0	30.5	35.8	38.0	29.7

Table 3: Unreliability of models when changing assistant temperature (AT) and user temperature (UT) in 📄 FULL, 🗨️ CONCAT and 🏰 SHARDED settings. The lower the number the more reliable the assistant is.

Implications

For LLM Builders

A reliable LLM should:

- (1) achieve similar aptitude in single- and multi-turn settings,
- (2) have small unreliability ($U_{10}^{90} < 15$) in multi-turn settings,
- (3) achieve these at unmodified temperature ($T = 1.0$),



Implications

For NLP Practitioners

Encourage NLP practitioners to experiment with sharding and release sharded versions of their tasks and instructions alongside fully specified ones



Implications

For Users

- (1) If time allows, try again
- (2) Consolidate before retrying



My takeaway

- (1) Mostly problems associated with dataset mismatch (in- or out-of-distribution)
- (2) This method in the paper also introduced a certain level of mismatch
- (3) LLMs are trained to give answers and make assumption if some key information are not given at the beginning.
- (4) It may get corrected if the users prompt the LLMs to abandon extra assumptions in the end.

We are also being trained to be in-distribution of the LLMs 🤪

How Many Instructions Can LLMs Follow at Once?



Motivation+ Research Questions + Contributions

Motivation: Production-grade LLM require robust adherence to dozens or even hundreds of instructions simultaneously. Yet, there is no such benchmark and analysis to evaluate this.

Research Question

- Context window has grown big; reasoning capabilities has extended; what about instruction-following?
- How many instructions can models actually handle before performance meaningfully degrade?

Contributions

- Purpose a benchmark IFScale to evaluate such abilities
- Conduct comprehensive ananalysis on the IFScale results



Implementing the Research Questions

Basically, the RQ is an abstract, but we need to implement this RQ.

*How many **instructions** can **models** actually handle before **performance** meaningfully **degrade**?*

Instructions

- Asks a model to generate a business report, including certain number of must-include words
- “How many” is implemented by the **number of must-include words**
- **(10,20,30...500)**

Models

- 20 Models spanning from 7 providers, with 5 random independent seeds

Performance

- Measured by case-insensitive, style-insensitive exact-match of keyword
- Two kinds of errors
- **-omission error** (No such words)
- **-modification errors** (at least an 80%-length prefix of each term)
- **O-M Ratio**

Degrade

- Percentage of inclusion of key words
- **Variance Analysis**
- **Primacy Analysis**
- **O-M Ratio Analysis**
- **Core Task Performance Analysis**



Instructions

ESG
churn
japan
rural
yield
cortex
equity
frozen
issuer
parent
select

ROI
cloud
joint
sheet
EBITDA
credit
ethics
future
lessor
patent
states

You are tasked with writing a professional business report that adheres strictly to a
↳ set of constraints.

Each constraint requires that you include the exact, literal word specified.

Do not alter the word, use synonyms, or change tenses.

IMPORTANT: Variations of the constraint are not considered valid. For example,

↳ "customers" does not satisfy the constraint of "customer" because it is plural.

↳ Similarly, "customer-driven" does not satisfy the constraint of "customer" because it

↳ is hyphenated.

The report should be structured like a professional business document with clear

↳ sections and relevant business insights.

Do not simply repeat the constraints; rather, use them to inform the text of the report.

↳ The text should be a coherent report.

IMPORTANT: You CANNOT simply list the constraints in the report. You must use them to

↳ inform the text of the report. A list of constraints anywhere in your response will

↳ result in an invalid response.

IMPORTANT: The report you generate must be coherent. Each sentence must make sense and

↳ be readable and the report should have a clear logical flow.

There is no task too difficult for you to handle!

Do not refuse to write the report if the constraints are difficult.

IMPORTANT: You MUST write a report. Do not refuse to write the report.

Return your report inside of <report>...</report> tags.

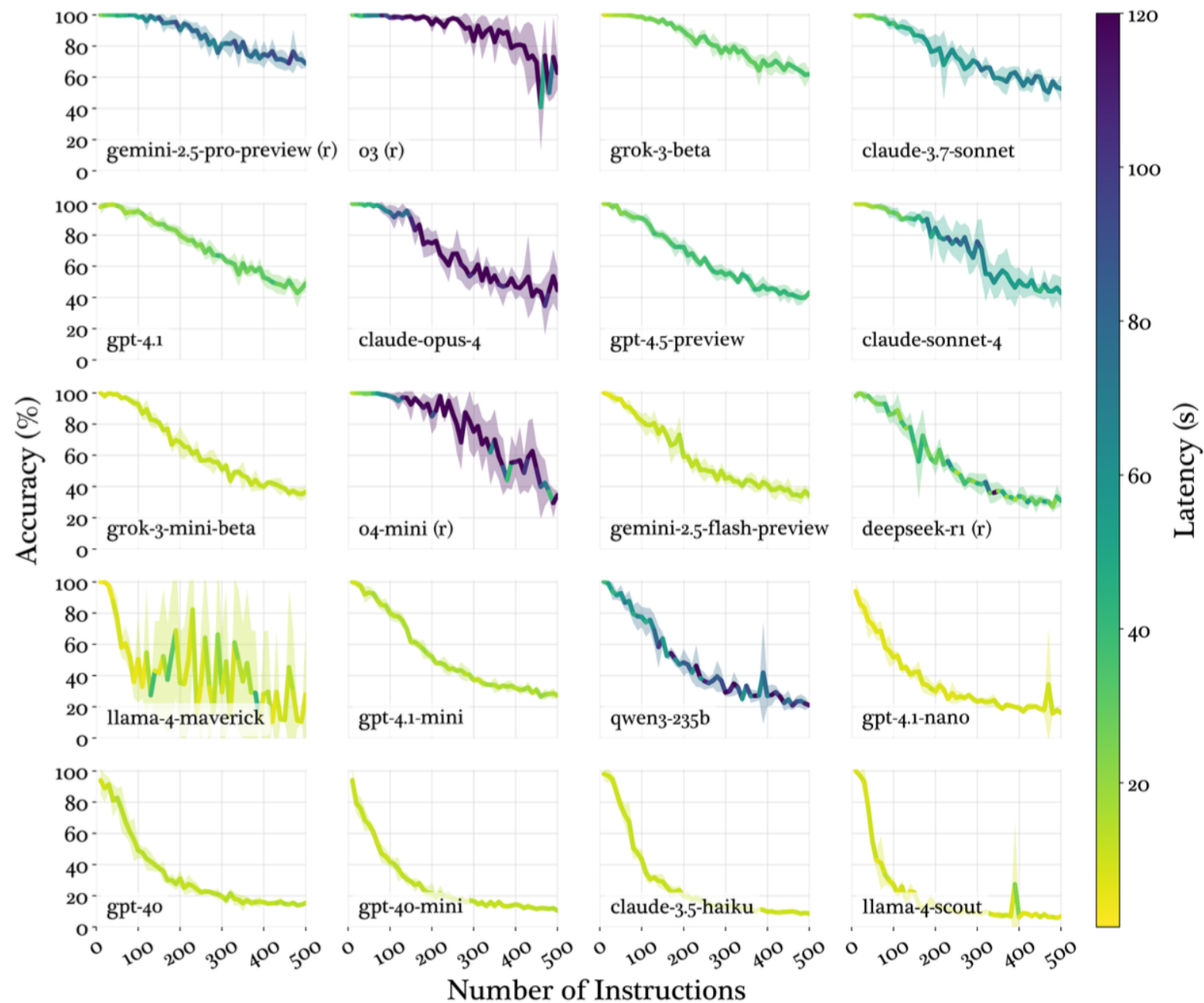
CONSTRAINTS

{CONSTRAINTS}

```
CONSTRAINTS = '\n'.join(  
    f"{i+1}. Include the exact word: '{constraint}'."
```



Performance



Degradation Pattern Analysis---Three Patterns



Accuracy degradation curve shows three patterns

Threshold Decay

- Such decay means the model performance remains stable until a threshold has been reached, followed by steep decline in performance and increased variance
- **Evident in Reasoning Models** (e.g Gemini-2.5-pro, o3)**

Linear Decay

- Such decay characterizes steady, predictable decline in performance
- **Evident in models like gpt-4.1 and claude-3.7-sonnet**

Exponential Decay

- Such decay characterizes rapid early degradation followed by performance stabilization at low accuracy floors.
- **Evident in claude-3.5-haiku and llama-4-scout**

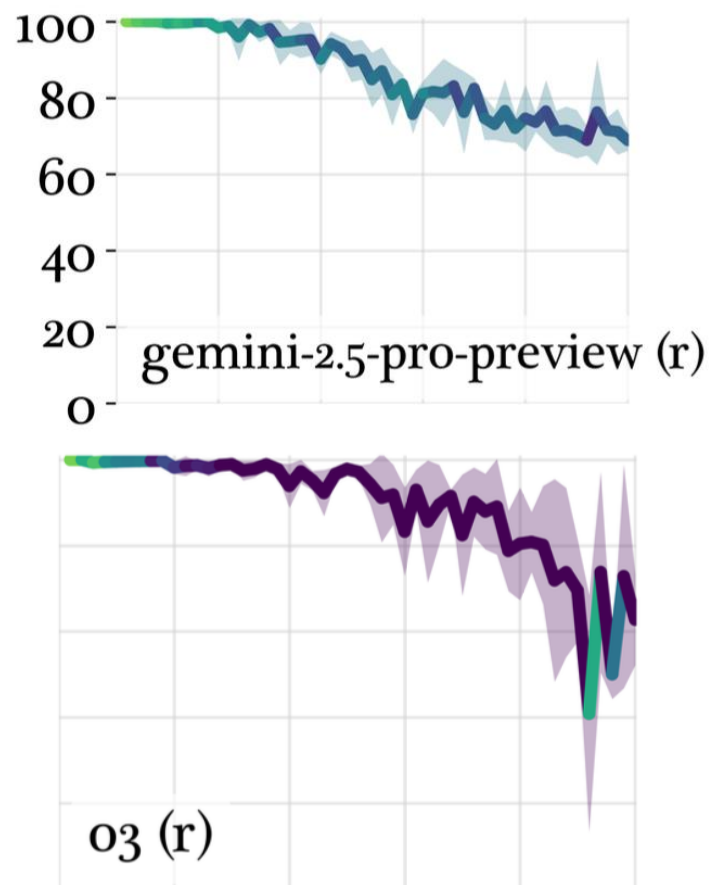


Degradation Pattern Analysis---Three Patterns

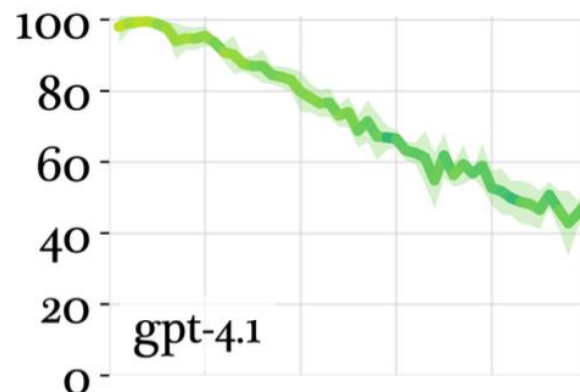
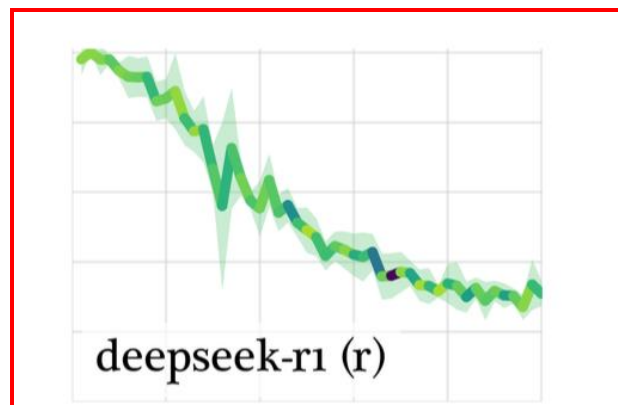


Accuracy degradation curve shows three patterns: X-axis: number of instructions, Y-axis: accuracy

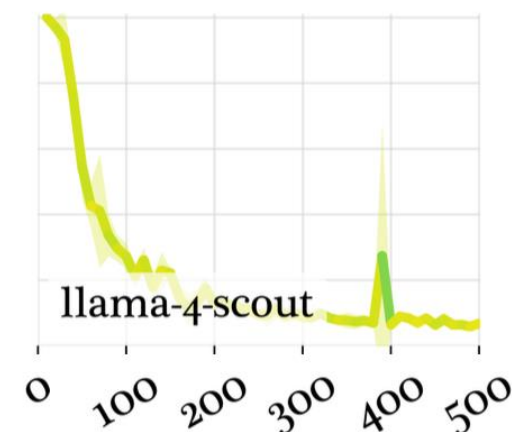
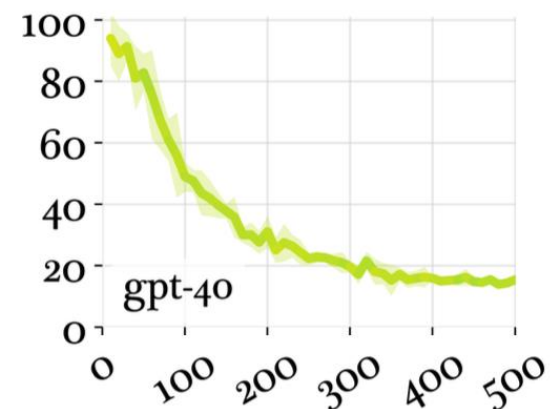
Threshold Decay



Linear Decay

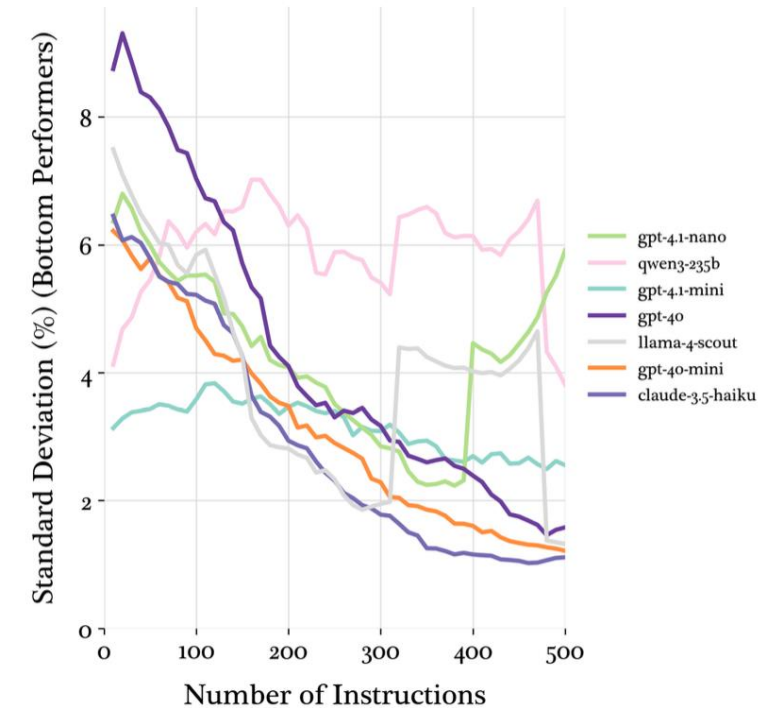
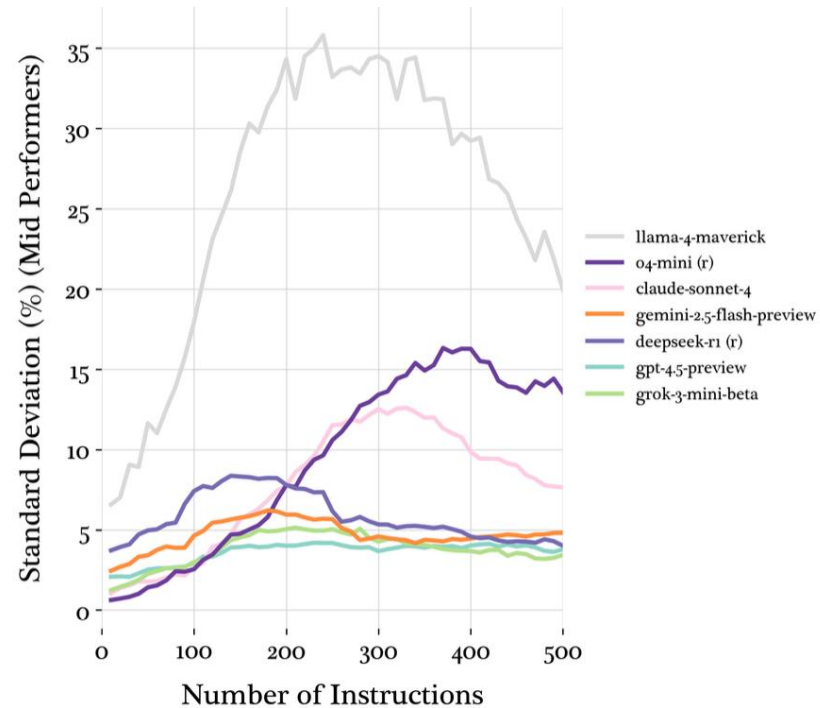
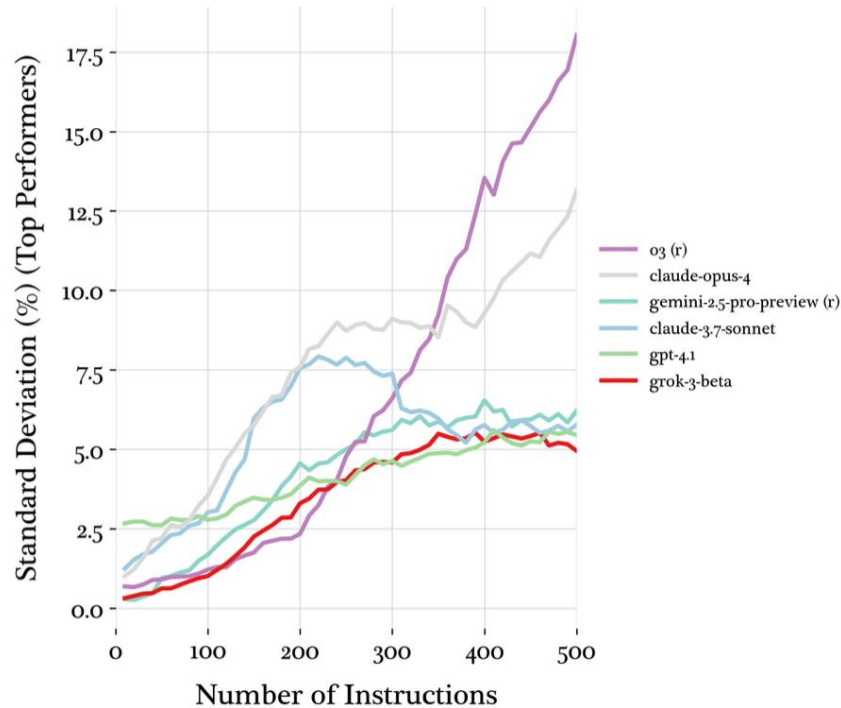


Exponential Decay



Variance Analysis

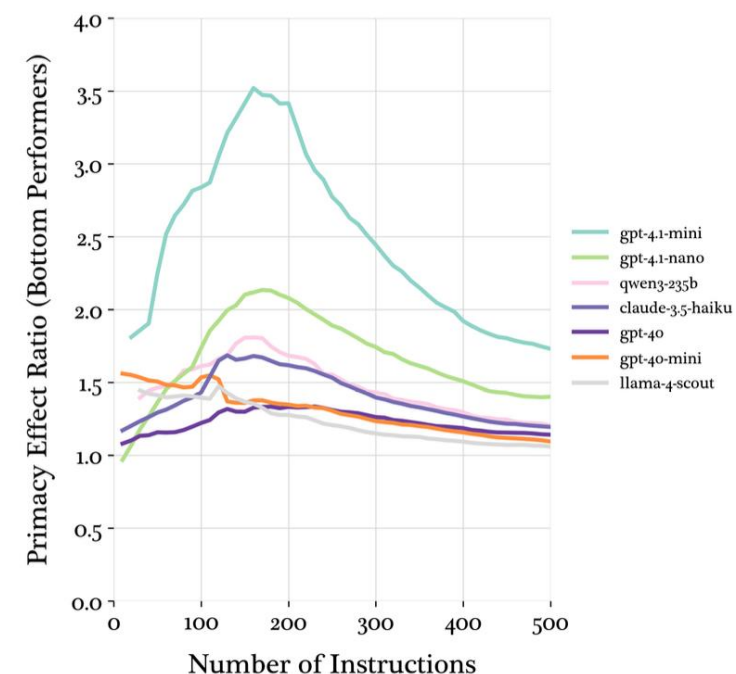
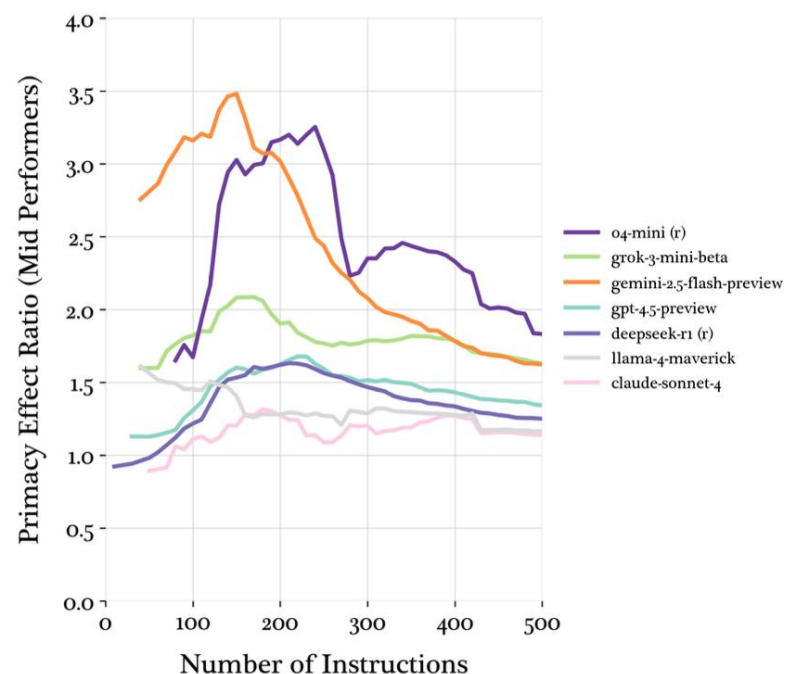
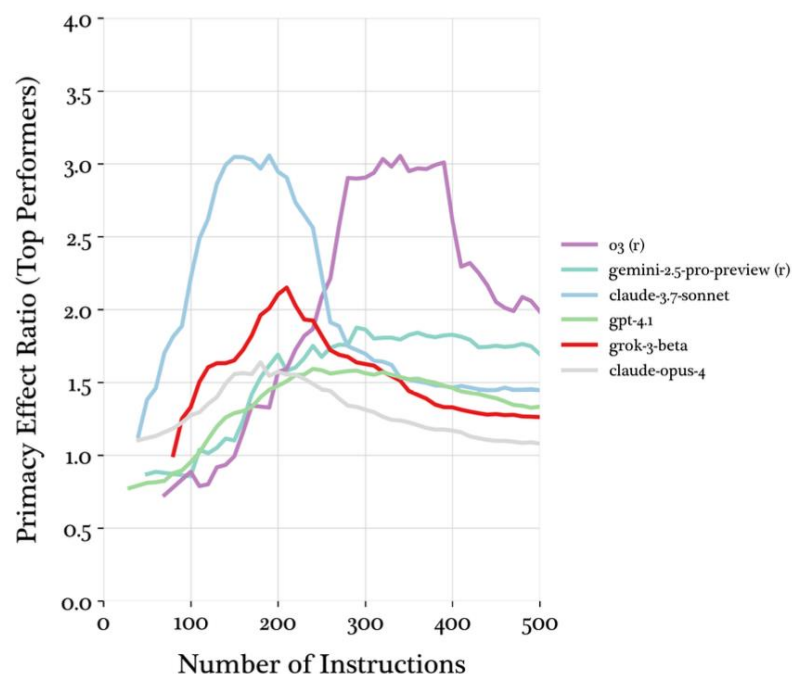
Top Performing Models display steady increase in variance, indicating reduced reliability as intension density. Mid-tier performing models show mid-range variance peaks in the 15—300 range.



Primacy Effects

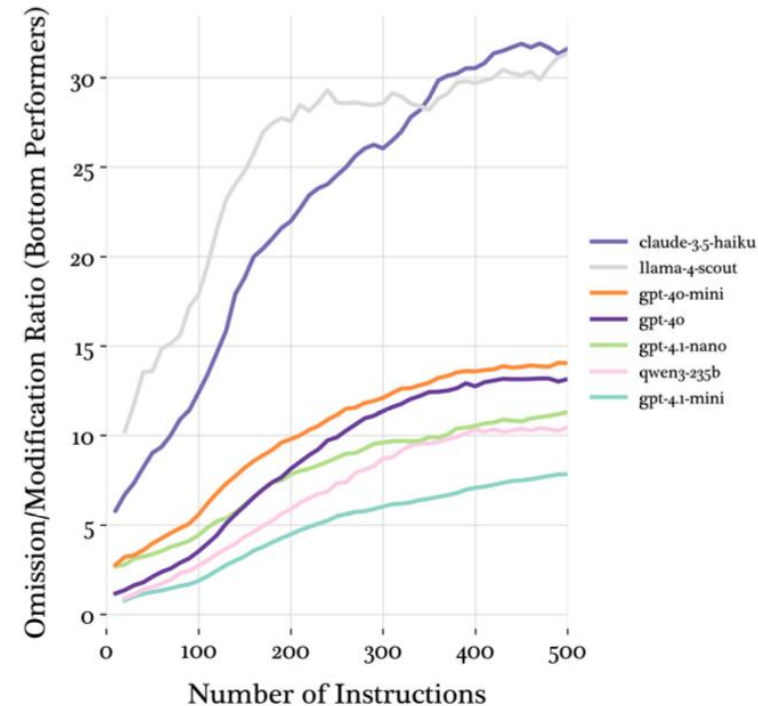
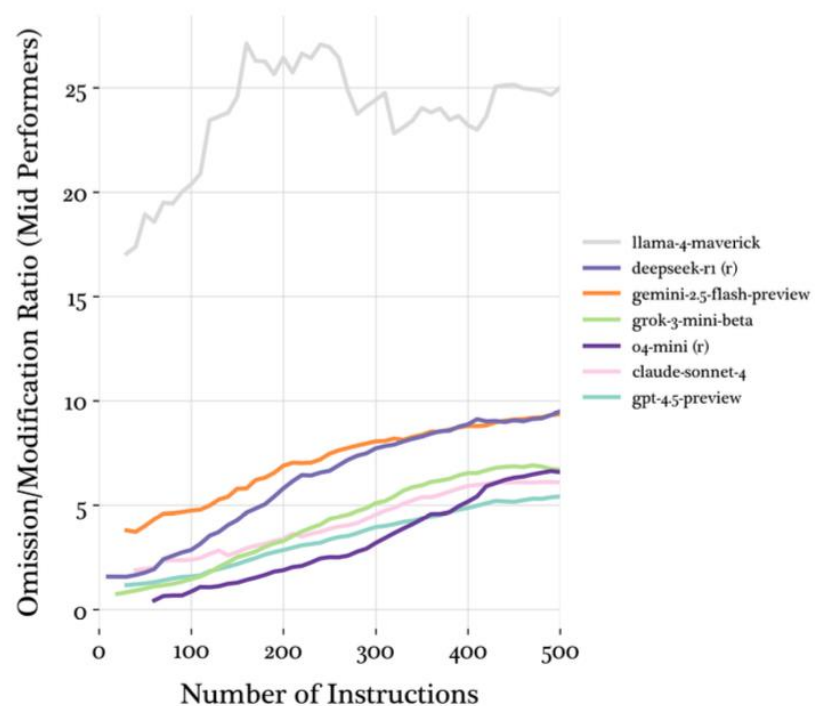
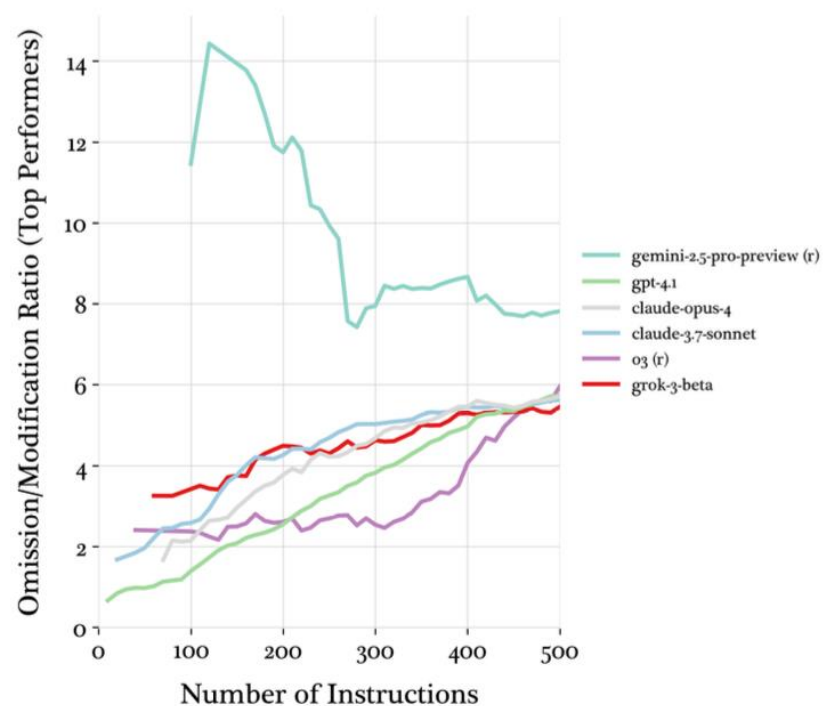
Primacy effect is the ratio of error rates in the final third of instructions to error rates in the first third of instructions. A ratio greater than 1.0 indicates that later instructions are more likely to be violated.

Mmid-range peak suggests that models exhibit the most bias as they begin to struggle under cognitive load at moderate densities.

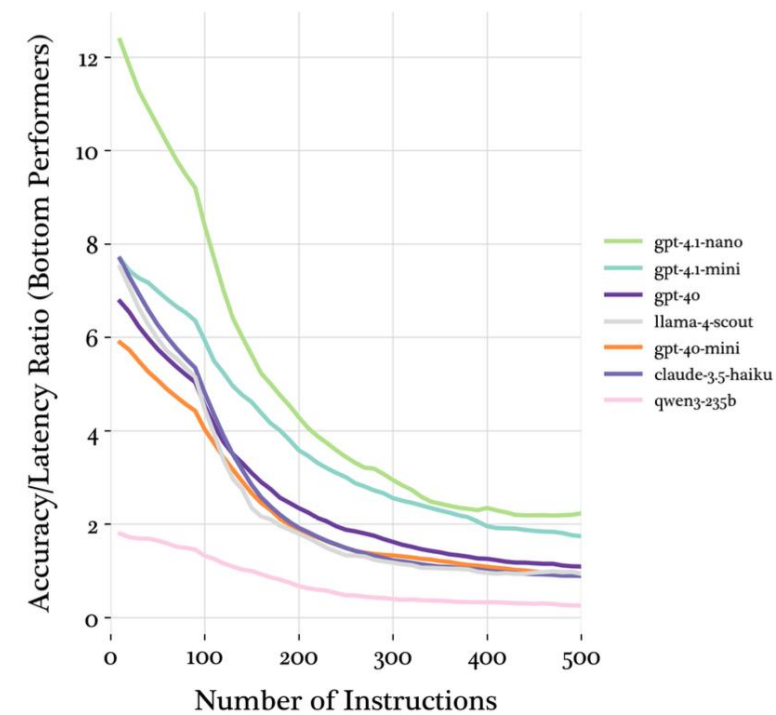
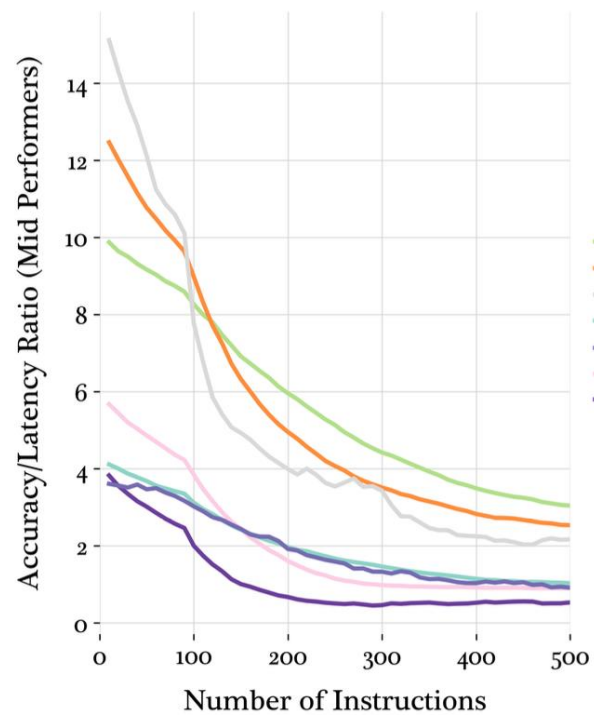
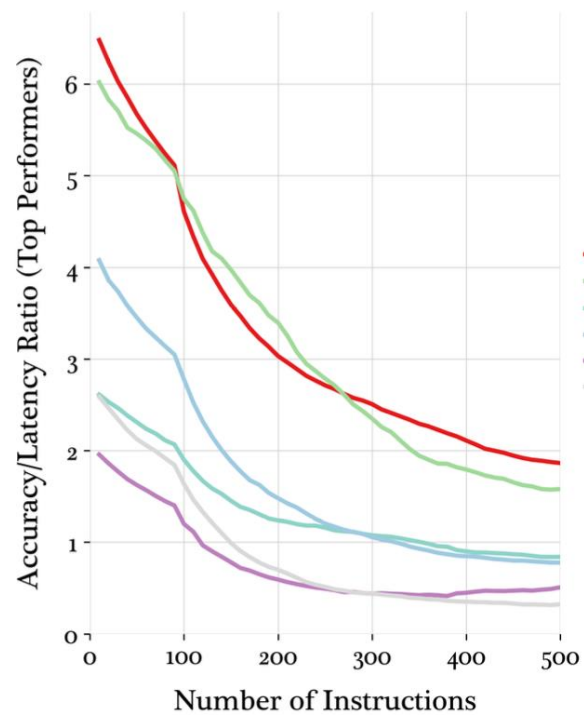


O-M Ratio Analysis

Models overwhelmingly err toward omission errors as instruction density increases. At low densities, many models show relatively balanced error types, but this shifts dramatically at high densities.

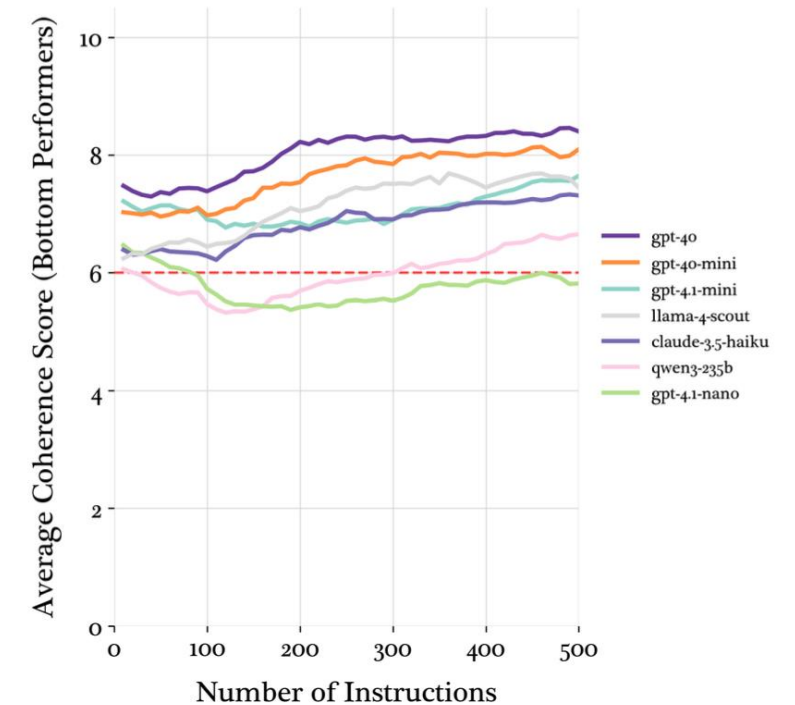
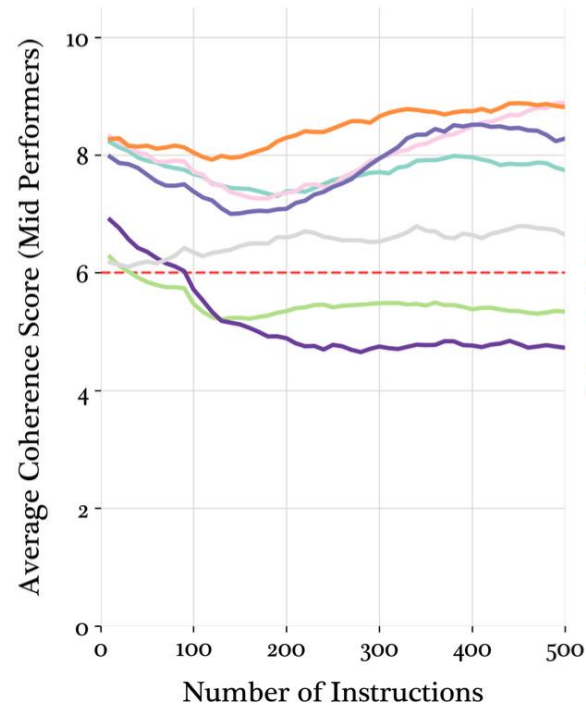
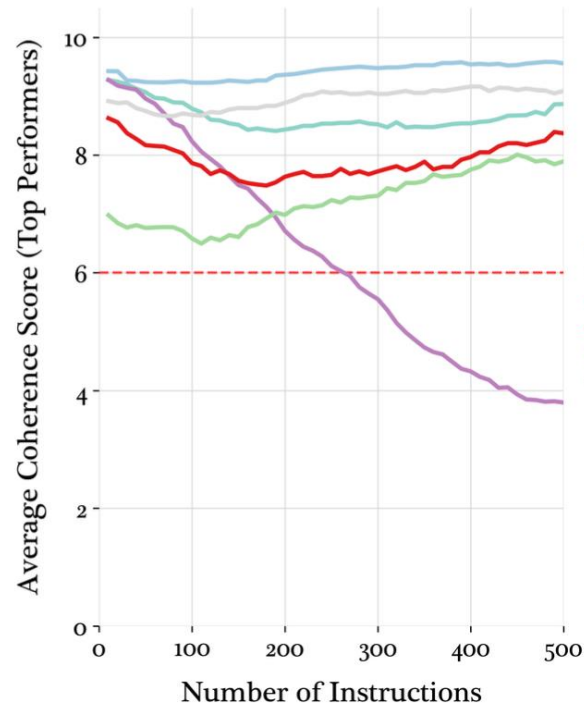


Efficiency Analysis



Coherence Analysis

Uses o4-mini to judge the coherence, no sign showing coherence decreasing significantly as instruction density increases for the majority of models



Thank You!

