Data-to-text The assistant is provided tabular data and several elements of related metadata, and must produce a caption (natural language sentence) describing the underlying data. We leverage the ToTTo [59] dataset to formulate sharded instructions.

Summary The assistant receives a corpus of around twenty documents and a user query, and must generate a summary with citations that addresses the query based on the documents. We re-purpose the instructions from Summary of a Haystack [40]. The summary task is the only task we include that tests long-context capabilities, with instructions spanning several tens of thousands of tokens, which is known to deteriorate model performance [29, 32, 33].

For each task, we reuse the metrics used in the original benchmarks. More specifically, the first four tasks (Code, Database, Actions, and Math) are evaluated for binary correctness, either by executing an answer attempt (code, SQL query), or validating semantic equivalence to a reference answer (API call, numerical answer). The last two tasks (Data-to-Text and Summary) are *refinement tasks*, which get scored on a continuous range (0-100). Data-to-text uses the BLEU metric [58], and Summary uses a custom LLM-as-a-judge metric ("Joint Score") built to measure information coverage and attribution accuracy of the summary [40]. We map binary accuracy in the range of 0-100 (0 = failure, 100 = success) so that all tasks produce scores on a common scale, facilitating aggregation.

Appendix I lists implementation details of the sharding process for each task, including the sample selection process and any task-specific logic that was implemented to facilitate reproducibility. Even though we intended for the six selected tasks to be representative of a wide range of LLM use cases, we put effort into making the sharding process efficient and reproducible, as we see the process itself as a contribution of our work. We envision that future LLM evaluation practitioners can shard their own dataset artifacts to study LLM multi-turn behavior in more diverse and unique settings.

4.2 Metric Selection

LLMs employ a stochastic process to generate text. When setting LLM generation parameters to their default (*e.g.*, T=1.0), LLMs generate many distinct responses for a fixed conversation state. We leverage this property to conduct repeated simulations for a given instruction and observe the variations that occur. Each simulation yields a score S_i ranging from 0-100 that assesses the level of success of the LLM in completing the task by the end of the simulation. Based on the set of scores $S = \{S_i\}_{i=1}^{N}$ obtained from running N simulations for an instruction, we define three metrics: **averaged performance** ($\overline{P}$), **aptitude** (A^{90}), and **unreliability** (U_{10}^{90}):

$$\overline{P} = \sum_{i=1}^{N} S_i/N \qquad A^{90} = \text{percentile}_{90}(S) \qquad U_{10}^{90} = \text{percentile}_{90}(S) - \text{percentile}_{10}(S).$$

Average performance $\overline{P}$ is an unbiased estimate of a model's mean score on an instruction in a given simulation type. Aptitude A^{90} is an estimate of a model's 90th percentile score on a given instruction, a best-case metric that estimates scores obtained in the top 10% of simulations conducted. Unreliability is an interpercentile range estimate, between the 90th and 10th percentile estimates, measuring the gap between best-case and worst-case simulations, giving a sense of *level of degradation* that occurs in response quality due to stochasticity in the LLM.

Each of the metrics is computed on a per-instruction basis and can be averaged across a corpus of instructions to obtain corpus-level metrics. In the rest of the paper, we refer to reliability and unreliability interchangeably, with reliability defined as $R_{10}^{90} = 100 - U_{10}^{90}$. We also simplify the notations to A for aptitude and U for unreliability, though the metrics can be generalized to other percentile thresholds (*e.g.*, A^{80} or U_{5}^{95}).

In Appendix E, we go over a concrete example of how an average degradation in performance ($\overline{P}$) from 90% to 60% could be due to a loss in aptitude, reliability, or a combination. Finally, Figure 6a visually connects the aptitude and unreliability metrics to score box-plot visualizations. In summary, the height of the upper whisker of the box plot represents aptitude (A), and the distance between the upper and lower whiskers of the plot represents Unreliability (U).

5 Simulation Scale and Parameters

In the main simulation experiment, we leveraged the totality of instructions we sharded across six tasks (a total of 600 instructions), and simulated conversations across three types: FULL, CONCAT, and SHARDED. We experimented with 15 LLMs, running $N = 10$ simulations for each pair of model and simulation type, totaling more than 200,000 simulated conversations. All simulations were conducted with a default temperature of $T = 1$, however, we conducted a supplementary experiment (Section 7.2) that explores the effect of temperature on aptitude and reliability.

Lost in Conversation Experiment

Model	📄 FULL						🧻 CONCAT						💎 SHARDED						Overall	
	🐍	🛢	API	▦	🖩	📜	🐍	🛢	API	▦	🖩	📜	🐍	🛢	API	▦	🖩	📜	🧻/📄	💎/📄
∞ 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
A\ 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
A\ 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

Table 1: Averaged Performance ($\overline{P}$) of LLMs on six tasks (🐍 Code, 🛢 Database, API Actions, ▦ Data-to-text, 🖩 Math, and 📜 Summary). For each task, conversations are simulated in three settings: 📄 FULL, 🧻 CONCAT, and 💎 SHARDED. Models are sorted in ascending order of average FULL scores across tasks. Background color indicates the level of degradation from the FULL setting. The last two columns average the performance drops from the CONCAT and SHARDED compared to the FULL in percentages across the six tasks.

Although simulating ten conversations for each (`LLM, instruction, simulation type`) increases experimental costs ten-fold, it allows us to not only measure averaged performance ($\overline{P}$) more accurately, but also study aptitude and reliability of LLM systems in depth in Section 6.2.

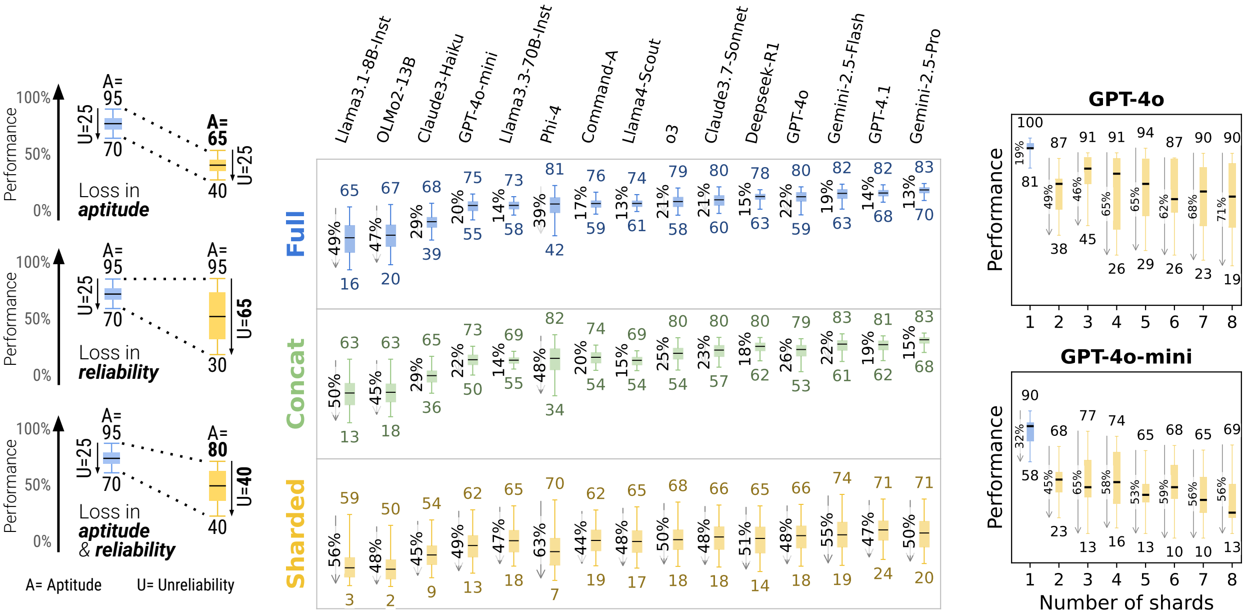
We selected a total of 15 LLMs from eight model families: ֍ OpenAI (GPT-4o-mini, GPT-4o [30], o3 [57], and GPT-4.1), A\ Anthropic (Claude 3 Haiku, Claude 3.7 Sonnet), Google's ✦ Gemini (Gemini 2.5 Flash, Gemini 2.5 Pro) [75], Meta's ∞ Llama (Llama3.1-8B-Instruct, Llama3.3-70B-Instruct, Llama 4 Scout) [23], ✤ AI2 OLMo-2-13B [56], ⊞ Microsoft Phi-4 [1], 🐋 Deepseek-R1 [24], and ● Cohere Command-A [15]. This selection prioritizes the evaluation of state-of-the-art models, including both small (8B) and large models (300B+). We purposefully include both open- and closed-weights models, as well as two reasoning models (o3, R1) to study the effect additional thinking (test-time compute) has on multi-turn conversation capability. Details on model versioning and access are listed in Appendix H. We estimate the total cost of conducting simulations to be around $5,000.

6 Results

6.1 Average Performance Findings

Table 1 summarizes results from the simulation. At a high level, **every model sees its performance degrade on every task when comparing FULL and SHARDED performance, with an average degradation of -39%**. We name this phenomenon `Lost in Conversation`: models that achieve stellar (90%+) performance in the lab-like setting of fully-specified, single-turn conversation struggle on *the exact same tasks* in a more realistic setting when the conversation is underspecified and multi-turn.

In comparison, models perform roughly equivalently in the CONCAT setting, with CONCAT performance averaging 95.1% of the FULL counterpart. This implies that the loss in performance for SHARDED is not explained by potential loss of information in sharded instructions, as such a loss would be reflected in lower CONCAT performance. We observe that smaller models (Llama3.1-8B-Instruct, OLMo-2-13B, Claude 3 Haiku) have more pronounced CONCAT degradations (86-92), and interpret this as indicating that smaller models struggle to generalize as well as larger models: benign rephrasing affects performance more than for larger, more robust models. This lack of robustness to paraphrasing can be observed visually in Table 1: CONCAT degradation (red background) is more pronounced in the top rows (weaker models) than the bottom rows (stronger models).



(a) Visualizing Aptitude and Unreliability.

(b) Observed Model Degradations

(c) Gradual Sharding Results

Figure 6: (a) Visual introduction to the concepts of Aptitude and Unreliability when overlaid on a box-plot visualization, (b) reliability results based on experimental simulations with 15 LLMs, (c) summary of results from gradual sharding experiment, with instructions sharded in gradually larger shard sets (from 1 to 8 shards).

The last column of the Table (🏆 / 📄) aggregates performance degradation across the six tasks, summarizing the magnitude of the Lost in Conversation effect for each model. Surprisingly, **more performant models (Claude 3.7 Sonnet, Gemini 2.5, GPT-4.1) get equally lost in conversation compared to smaller models (Llama3.1-8B-Instruct, Phi-4)**, with average degradations of 30-40%. This is in part due to metric definitions. Since smaller models achieve lower absolute scores in FULL, they have less scope for degradation than the better models. In short, no matter how strong an LLM's single-turn performance is, we observe large performance degradations in the multi-turn setting.

When looking at the task-specific breakdown, some models see more muted degradations in certain tasks. For instance, Command-A sees the least degradation on the Actions task, while Claude 3.7 Sonnet and GPT-4.1 conserve performance well on Code, and Gemini 2.5 Pro in the Data-to-Text task. This finding indicates that the multi-turn capabilities of models are not uniform across domains and validates the importance of benchmarking models across a wide variety of tasks to investigate model capabilities.

Additional test-time compute (reasoning tokens) does not help models navigate multi-turn underspecification, as the two reasoning models included in the experiment (o3, Deepseek-R1) deteriorate in similar ways to non-reasoning models. This result confirms that **additional test-time compute does not, on its own, allow models to strategize over multi-turn conversation**. The analysis we conduct identifies a potential root cause: reasoning models tend to generate lengthier responses (on avg. 33% longer than non-reasoning LLMs). As we find in Appendix F, longer assistant responses tend to contain more assumptions, which can derail the conversation by confusing the model on what requirements were posed by the user vs. its own previous turn responses.

6.2 Aptitude vs. Reliability Analysis

Results presented in Table 1 present averaged performance degradation ($\overline{P}$). We now report on the aptitude and reliability analysis based on metrics A and U. Figure 6b visually summarizes the results of the reliability analysis we conducted on the 15 LLMs included in our simulation experiment. First, looking at the two single-turn settings, we see that models that are more able (higher A) tend to be more reliable (lower U). For instance, the two most able models (GPT-4.1 and Gemini 2.5 Pro) achieve the lowest unreliability. At the lower end, the two models with the lowest aptitude (Llama3.1-8B-Instruct and OLMo-2-13B) are also the most unreliable. In summary, **in single-turn settings, models with higher aptitude tend to be more reliable.** This fact is known in the community, with arguments made