Demystify Verbosity Compensation Behavior of Large Language Models

Yusen Zhang, Sarkar Snigdha Sarathi Das, Rui Zhang
Department of Computer Science and Engineering, Penn State University
{yfz5488, sfd5525, rmz5227}@psu.edu

Abstract

Recent work has revealed Large Language Models (LLMs) often exhibit undesirable behaviors, such as hallucination and toxicity, limiting their reliability and broader adoption. In this paper, we discover an understudied type of undesirable behavior of LLMs, which we term Verbosity Compensation (VC). VC is similar to the hesitation behavior of humans under uncertainty, compensating with excessive words such as repeating questions, introducing ambiguity, or providing excessive enumeration. We present the first work that analyzes Verbosity Compensation, explores its causes, and proposes a simple mitigating approach. Our experiments on five datasets of knowledge and reasoning-based QA tasks with 14 LLMs, reveal three conclusions. 1) A pervasive presence of VC across all models and all datasets. 2) The large performance gap between verbose and concise responses. We also demonstrate that this difference does not naturally diminish as LLM capability increases. 3) Higher uncertainty exhibited by VC responses across all five datasets, suggesting a strong connection between verbosity and model uncertainty. We propose a simple yet effective cascade algorithm that replaces the verbose responses with the other model-generated responses, alleviating the VC of the Mistral model from 63.81% to 16.16% on the Qasper dataset.

1 Introduction

Recent research has highlighted various undesirable behaviors of Large Language Models, such as hallucination (Huang et al., 2023), toxicity (Wen et al., 2023), and ethical bias (Tao et al., 2023), which pose significant risks to users. Among them, the verbose response issue where LLMs respond with excessive words has attracted more and more attention in the LLM era because of unnecessary long output for solving problems (Singhal et al., 2023) and the unavoidable high cost of LLM-generated tokens.

The existing work mainly focuses on the length of the response and its applications. Researchers found that imposing a length constraint in the prompt can improve the performance of LLMs, under chain-of-thought (Chiang and Lee, 2024; Nayab et al., 2024) and machine translation (Briakou et al., 2024) settings. Singhal et al. (2023) found RLHF training favors the lengthy response. However, length is not enough to analyze verbosity as it provides a general overview but fails to capture key fine-grained features such as content structure.

In this paper, we discover a type of undesirable verbosity behavior of LLMs. We term it Verbosity Compensation (VC). Instead of focusing merely on the *length*, we analyze the frequency, types, and their relation to model performance. We also find VC is closely connected to the uncertainty of LLMs, demystifying the mechanism of the VC behavior, and improving the understanding of both VC and uncertainty. Interestingly, VC is similar to the hesitation behavior of humans under uncertainty (Juola, 2008; Brookshire and McNeil, 2014). Figure 1 shows a motivating example. In the first response, LLM generates a concise answer that is correct with low uncertainty. In the second and third responses, instead of generating an answer concisely, such as “16.5”, LLM repeats the question, and produces ambiguity, leading to a VC response with low performance and high uncertainty. VC is harmful and undesired for both users and servers. For the users, VC will lead to confusion and inefficiency (Fowler, 1927; Oppenheimer, 2006). When an LLM enumerates multiple answers, users are unclear about which one is correct. Besides, VC leads to bias among users of different length preferences if verbose answers attain higher/lower scores. For the servers, the verbosity leads to unnecessary higher costs and higher latency because of useless tokens.

To analyze the VC behavior systematically, we unify four long-context question-answering

Proceedings of the 2nd Workshop on Uncertainty-Aware NLP (UncertaiNLP 2025), pages 160–178
November 9, 2025 ©2025 Association for Computational Linguistics

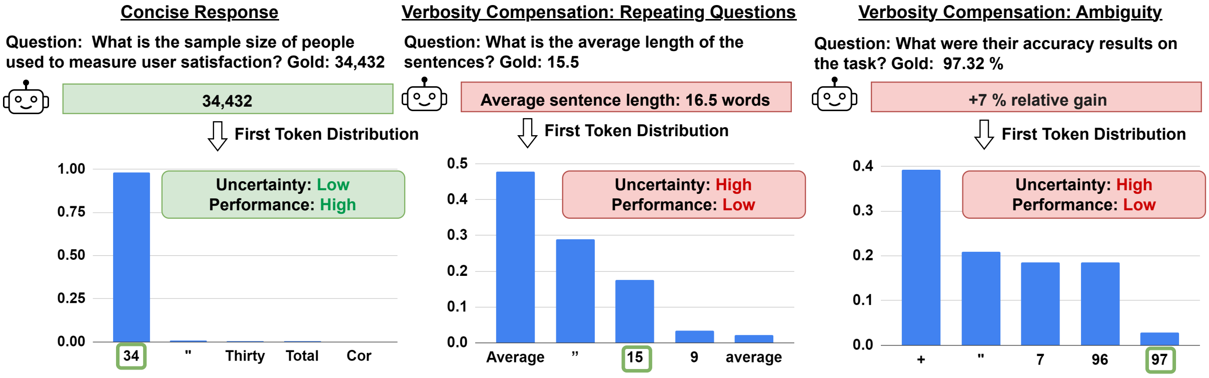


Figure 1: An illustration of comparison between concise and verbose responses. For each example, we ask the model to generate the response as concisely as possible. In the first response, LLM generates a concise answer, while in the second and third responses, LLM performs repeating, and ambiguity, leading to a verbose response with low performance and high uncertainty (Detailed numbers in Appendix B.2).

datasets and a reasoning-based language understanding dataset. We choose short-form QA with several tokens (comprising phrases, names, rather than complete sentences) in the gold answer to ensure the gold label is concise and easy to judge VC behavior in responses. We benchmark 14 LLMs on proposed datasets. Although we find that different models and datasets exhibit diverse distribution, we can categorize VC into five distinct types, including repeating questions, enumerating, ambiguity, verbose details, and verbose format. **The result reveals a pervasive presence of verbosity compensation (VC) across all models and all datasets**. Notably, GPT-4 exhibits a VC frequency of 50.40%. Meanwhile, we found that verbose responses exhibit significantly different recall from concise ones, with a notable drop of 24.72% on the Qasper dataset, **highlighting the urgent need to disentangle verbosity with veracity**.

Next, we measure the uncertainty of model responses using perplexity and Laplacian scores for open and closed-source models. We find that verbose responses exhibit higher uncertainty across all five datasets, suggesting **a strong connection between verbosity and model uncertainty**. Finally, we leverage the connection between performance and VC to develop a routing algorithm that obtains significant improvements over the random selecting baseline and uncertainty-based routing. To mitigate VC in LLMs, we propose a simple yet effective cascade algorithm that replaces verbose responses with responses of larger LLMs. Experiments demonstrate the efficacy of the proposed algorithm through tests on three model combinations: Gemma to Gemini, Mistral to GPT-4, and Llama to Claude. The results show that our approach effectively alleviates the VC of the Mistral model from 63.81% to 16.16% on the Qasper dataset. The insights above can inspire the development of practical applications and effective mitigation strategies. Future work can mitigate the uncertainty of the LLMs by alleviating VC behavior due to the proposed connections between them.

2 Related Work

Verbosity in LLM Responses Recently work has studied the verbosity of LLM-generated contents and its implications. Concise thoughts (Nayab et al., 2024) use prompts to constraint the length of Chain-of-thought reasoning and generate more concise responses with better performance. Ivgi et al. (2024) investigate the fallback behavior of LLM-generated responses when facing uncertainty. Singhal et al. (2023) investigate the correlation between generated length and reinforcement learning from human feedback (RLHF) techniques. Saito et al. (2023) find that LLMs sometimes prefer more verbose answers even if they have similar qualities. By contrast, Huang et al. (2024) find that GPT-4 prefers short responses in faithfulness and coverage when it comes to summarization. Unlike these works, we discover the connection between performance and verbosity compensation behavior in both CoT and general QA settings and connect verbosity to uncertainty. Besides, we use the cascading model to mitigate verbosity while they use prompt engineering.

Uncertainty Quantification of LLMs With the thriving of Large Language Models (LLMs), researchers have begun exploring uncertainty quantification in LLM responses (Geng et al., 2023). For white-box models, researcher have focused on unsupervised methods including entropy (Malinin and Gales, 2020), similarity (Fomicheva et al., 2020; Lin et al., 2022), semantic (Kuhn et al., 2023; Duan et al., 2023), and logits (Kadavath et al., 2022; Chen et al., 2024), whereas for black models, the uncertainty evaluation is based on multiple sampled output of the LLMs (Malinin and Gales, 2020; Lin et al., 2023; Manakul et al., 2023) However, these works aim to improve the evaluation metrics for LLM uncertainty while we focus on connecting uncertainty with verbosity compensation behavior.

Optimisation of LLM API Calls Recently, researchers have proposed to reduce the cost of leveraging a pool of LLMs (Wang et al., 2024) with a cascade algorithm. FragulGPT (Chen et al., 2023) use a cascade algorithm to visit LLMs from weak to strong and use an LLM evaluator to judge if the response is good enough to use (Madaan et al., 2023). (Ramírez et al., 2024) leverage the uncertainty of the prediction as the evaluator to evaluate both cascading and routing structures. Similarly, (Gupta et al., 2024) improve it by using token-level uncertainty. Our work, by contrast, aims at mitigating verbosity compensation which has not been explored before, and our evaluator is the verbosity of the response in the cascade algorithm.

3 Verbosity Compensation

In this section, we first introduce the definition and quantification of VC, and then we propose the metrics for evaluating the correlation between verbosity compensation and performance, uncertainty, and alleviating it with LLM routing.

3.1 Verbosity Compensation of LLMs

We first formalize the task. A dataset $\mathcal{D}$ consists of multiple data samples where each consists of a source text x, a query q, and a ground truth y. Since this is the first study, we mainly focus on the samples where y mainly contains short phrases for simplicity. A large language model $LLM(*)$ consumes the concatenation of x, q, and an instruction I to produce the response r. We use $|r|$ to represent the tokens in r. For instruction I, we always ask LLM to generate as concisely as possible so that the model is instructed not to generate verbose responses. Since the LLMs have maximum context window sizes L_c, we truncate the source to accommodate diverse context limits (details in A.3).

We define a response r to exhibit verbosity (we use the term verbosity as an alias for VC, and conciseness as an alias for Non-VC) if and only if it contains redundant tokens compared with the ground truth, since we assume the gold label to be concise. To detect VC, we define the verbosity compensation detector $V(x, y, r)$ (abbreviated as the verbosity detector). Using this detector, VC behavior for an LLM is defined as a triple (x, y, r) where $V(x, y, r) = 1$ describes that the VC occurs in the response r. To quantify the frequency of VC, we define it as the ratio of VC responses in each dataset $\sum_{(x,y)\in\mathcal{D}} V(x, y, r)/|\mathcal{D}|$.

3.2 Performance and Verbosity Compensation

A key bias of verbosity compensation is that the performance of the verbose responses is different from the concise ones. To quantify this behavior, we propose two evaluation metrics. One is performance difference (Δ), defined as the average score of the concise responses minus the average score of the verbose responses.

$$\Delta(\mathcal{D}) = \frac{\sum_{(x,y)\in\mathcal{D}}(1 - V(x,y,r)) \times \text{recall}(y,r)}{\sum_{(x,y)\in\mathcal{D}}(1 - V(x,y,r))} - \frac{\sum_{(x,y)\in\mathcal{D}} V(x,y,r) \times \text{recall}(y,r)}{\sum_{(x,y)\in\mathcal{D}} V(x,y,r)}$$

where r is the response generated by LLM and recall(y,r) is defined as $|r \cap y|/|y|$. This metric computes the difference between concise and verbose responses of a model over a dataset. If VC has no influence on the performance, the Δ should be 0. An LLM should show zero Δ because verbosity and performance are naturally independent and thus have no relation with each other. However, if Δ is positive, then it demonstrates that verbosity responses lead to the performance drop for this model on the dataset, and vice versa. To remove the influence of the length difference between verbose and concise responses, we use recall as the scoring function. Compared with precision or F1 scores, scores are higher for verbose responses (or Δ will be smaller) because verbose responses usually contain more tokens than concise ones.

A main problem of Δ is that the recall difference between verbose and concise responses is twisted by the absolute performance of the LLMs. According to the definition, a dataset with lower

Algorithm 1 Cascade Model Selection Algorithm.

```
Input: A list of LLMs $M$, A sample $(x, y, q)$, instruction $I_w$,
  a verbosity detector $V()$.
Output: A response $r$.
  order $M$ by model capability from weak to strong
  for LLM in $M$ do
    $r \leftarrow$ LLM($x \bigoplus q \bigoplus I_w$)
    if $V(x, y, r)$ is false then
      break
    end if
  end for
  return $r$
```

performance naturally has a smaller space for performance difference. An extreme case is that the performance is zero on a dataset and the maximum Δ is zero as well. This impedes the fair comparison between datasets and models because they have diverse absolute performances. Thus, we propose relative performance difference

$$\delta(\mathcal{D}) = \Delta(\mathcal{D}) / \frac{\sum_{(x,y)\in\mathcal{D}} \text{recall}(y, r)}{\sum_{(x,y)\in\mathcal{D}} 1}$$

δ can be seen as the Δ if the absolute performance of the LLMs is scaled to the same number. We use this to compare the influence of performance across datasets and LLMs.

3.3 Verbosity Compensation and Uncertainty

For humans, verbosity compensation usually happens when we feel uncertain about the answers. Thus, for the LLMs, it is natural to speculate verbosity compensation of LLMs is also related to the uncertainty of the model. To test this hypothesis, we evaluate the uncertainty of the LLMs with the tool proposed by Fadeeva et al. (2023). First, we split the samples according to the length of the response $|r|$. Then, we quantify the uncertainty of each split. For open-sourced models, we use perplexity (Fomicheva et al., 2020) for evaluation, and for the close-sourced model, we use the sum of eigenvalues (Lin et al., 2023) of the graph laplacian as the metrics.

3.4 Alleviating Verbosity Compensation with Cascade Model Selection

Although it is difficult to ask a single LLM to generate a concise but correct answer, the verbosity compensation behavior can be mitigated by an ensemble of multiple models. To this end, we propose a **Cas**cade Model **Sel**ection algorithm (CaSel) to increase the chance of getting concise responses. The algorithm is simple and straightforward (Algorithm 1). Given a list of LLMs from weak to strong, we first ask the weak model to generate a response. At any time, if we detect $V(x, y, r) = 1$, we stop the generation of the current sample and redo the generation by a stronger model, and repeat the process. With the power of diverse LLMs, the algorithm can finally obtain a response with less verbosity and better performance.

4 Experiment Setup

4.1 Datasets and Metrics

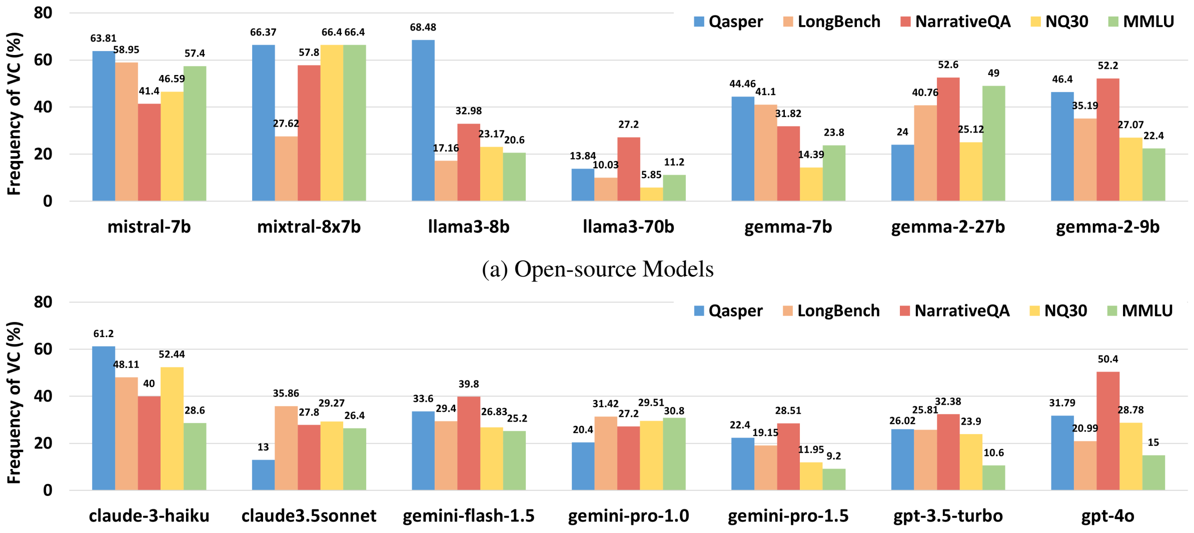
We include two types of datasets. 1) Knowledge-based question answering which aims at extracting knowledge from the given source text that is long or in a particular position. These datasets include **Qasper** (Shaham et al., 2022), **LongBench** (Bai et al., 2023), **NarrativeQA** (Shaham et al., 2022), and **NaturalQuestions_30 (NQ30)** (Liu et al., 2024). and reasoning-based question answering. More details for dataset construction can be found in Appendix A.1. 2) Reasoning-based Question Answering, including a modified **MMLU** (Hendrycks et al., 2021b,a) dataset. **Metrics.** We report recall when measuring verbosity compensation behavior and use F1 score for evaluation of the cascade model performance (Bai et al., 2023).

4.2 Models

We use 14 LLMs in total across all experiments, including both open-source and closed-source models in 6 families: GPT, Claude, Gemini, Llama, Gemma, Mistral. Details are in Appendix A.2. For each model, in addition to the prompt that introduces the task, we also ask them to "generate as concisely as possible, use a single phrase if possible". **Verbosity Detector.** We assume that the gold answer y is concise and clear so that we can compare it with the predicted results to detect verbosity. Specifically, we use an LLM as $V(x, y, r)$. We prompt GPT-3.5-Turbo with definitions and demonstrations of verbosity, as well as the question, prediction, and ground truth. The model needs to generate a binary value showing whether the response is verbose. To evaluate the effectiveness of this detector, we manually annotate 100 samples and compare them with model predictions. 93% of the samples have the same label, demonstrating the effectiveness of the LLM-based detector.

5 Result and Analaysis

In this section, we analyze verbosity compensation and its connection with performance and uncer-



(a) Open-source Models

(b) Closed-source Models

Figure 2: Frequency of Verbosity Compensation. All models exhibit intensive verbosity compensation behavior. Among them, llama3-70b has the lowest frequency on average (Details in Appendix B.1).

tainty. Then, we evaluate the cascade algorithm.

5.1 Verbosity Compensation

Frequency of Verbosity Compensation Behaviors. Figure 2 shows the frequency of each model on each dataset. As shown, all the models display verbosity compensation behavior on all datasets. On average, 74.19% of the responses are verbose for mistral-7b. The best model is llama3-70b which only contains 13.62% verbose responses. Furthermore, the frequency of VC averaged on seven open-source models is 39.80% which is significantly higher than closed-source models 28.96%.

Five Types of Verbosity Compensation Behaviors. After showing verbosity happens frequently in LLMs, we further conduct a human annotation to inspect verbose response patterns and classify them into five types. Specifically, we choose six combinations of model and dataset (Table 1) and pick out the samples with verbose responses that are not fully correct (recall $\neq 1$, $V(x, y, r) = 1$). By checking all these samples, we classify verbosity compensation behavior into five types (Table 1): Ambiguity indicates not answering precisely; repeating question indicates repeating the tokens in the question or providing unrelated information; enumerating shows answering multiple answers in a row trying to cover the correct answer; verbose detail/format means generating more detailed explanations or format symbols. Then, we annotate the verbosity compensation behaviors and obtain statistics in diverse settings. As shown in Figure 3, the ratio distribution of five types of behavior varies across different models and datasets. Furthermore, the main type of Gemini-1.5-flash is repeating questions on the MMLU dataset (67.86%), and enumerating on the Qasper dataset (47.62%). In contrast, llama-3-70b mainly produces verbose details on the Qasper dataset (32.87%). This shows that different datasets or models have a significantly different distribution of the main type of verbosity behavior.

5.2 Verbosity Compensation and Performance

Verbose and concise responses exhibit significantly different performance. As shown in Table 2 and Table 3, the performance difference ($\Delta \neq 0$) exists on most of the datasets and tasks, including both knowledge/reasoning-based tasks. This demonstrates that when the model performs verbosity compensation, the performance also changes significantly (Supplementary experiments in Appendix C.4, C.6). Among them, most of the datasets and models show lower performance on verbose samples (marked in red). For instance, llama3-70b shows 24.7% performance gap on Qasper dataset. However, *all models cannot disentangle performance with verbosity ($\Delta = 0$), highlighting the urgent need to disentangle verbosity with veracity.*

Correlation with Model Capability. We investigate the influence of model capability on the performance difference between verbose and concise

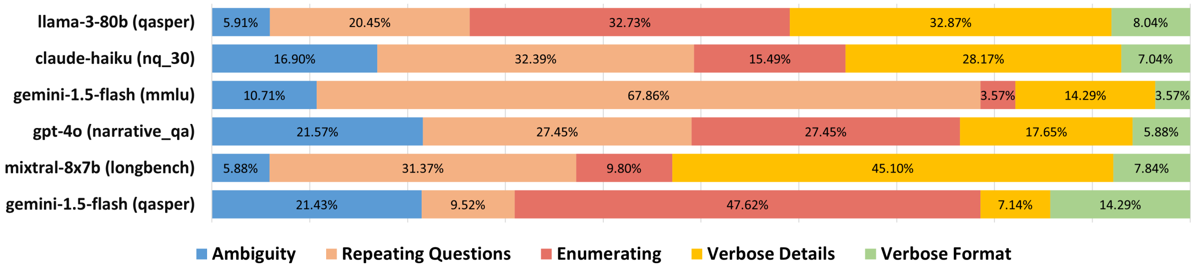


Figure 3: Human annotation of five types of verbosity compensation behavior on five datasets. Different models and datasets show diverse patterns of verbosity types.

Dataset	Question	Gold	Model Prediction	Type
Qasper	What is the size of the dataset?	3029	It is very large	Ambiguty
Longbench	Which genus has more species, Dracula or Pistacia?	Dracula	Pistacia has more species	Repeat
NarrativeQA	What costumes are the teenagers forced to wear?	Bunny costumes	Pig , donkey , rabbit	Enumberate
NQ30	who ran the fastest 40 yard dash in the nfl	Jakeem Grant	Chris Johnson 4.24 seconds	Detail
NarrativeQA	What types of activities occur in ...?	alleged phenomena	“ Disappearances folklore ”	Format

Table 1: Examples of five verbosity compensation types.

		Short (Qasper)				Medium (LongBench)				Long (NarrativeQA)			
	L_c	concise	verbose	Δ	Avg.	concise	verbose	Δ	Avg.	concise	verbose	Δ	Avg.
gemma-7b	4k	45.24	46.76	-1.52	46.51	36.04	18.37	**+17.67**	30.74	15.39	6.70	+8.69	12.66
gemma-2-9b	8k	54.84	49.46	+5.38	52.73	44.86	43.51	+1.36	44.50	29.38	23.05	+6.33	26.81
gemma-2-27b	8k	54.51	48.26	+6.25	53.55	45.97	33.68	+12.30	43.41	32.17	30.86	+1.30	31.87
llama-3-8b	8k	54.36	53.51	+0.85	53.99	36.18	29.00	+7.18	34.64	29.25	19.51	+9.74	25.68
llama-3-70b	8k	52.86	28.74	**+24.12**	49.80	49.98	37.79	+12.19	48.76	34.30	25.91	+8.39	32.06
mistral-7b	8k	63.23	44.84	+18.39	56.42	54.03	37.04	+16.99	46.13	27.60	26.69	+0.91	27.21
mixtral-8x7b	8k	64.12	50.03	+14.10	56.78	2.62	6.24	-3.61	3.40	37.55	28.57	+8.98	33.09
gpt-3.5-turbo	16k	59.81	37.46	+22.34	54.77	53.88	47.02	+6.85	52.21	39.41	27.35	+12.06	35.49
gpt-4o	128k	63.80	44.07	+19.72	58.43	68.83	63.53	+5.30	67.53	59.14	47.12	+12.02	53.25
claude-3-haiku	200k	61.30	56.01	+5.29	58.54	53.02	57.88	-4.86	54.95	50.68	38.50	**+12.18**	46.13
claude-3.5-sonnet	200k	58.36	38.01	+20.35	56.12	59.42	57.36	+2.06	58.85	50.77	56.29	-5.52	52.16
gemini-flash-1.5	1m	62.52	41.64	+20.88	56.00	59.32	58.02	+1.30	59.00	2.51	1.12	+1.39	1.98
gemini-pro-1.0	32k	54.70	35.73	+18.98	51.44	47.85	44.68	+3.18	47.06	22.43	32.40	-9.96	24.89
gemini-pro-1.5	2m	59.40	45.79	+13.61	56.65	64.19	55.75	+8.44	62.97	36.26	41.74	-5.47	37.79
Avg		57.79	44.31	13.48	54.41	48.30	42.13	6.17	46.73	33.35	28.99	4.36	31.50

Table 2: Overall recall comparison between verbose and concise responses. **Bold**/Underline indicate the largest positive/negative performance gap between verbose and concise responses. The verbose responses obtain a significantly different performance than the concise ones, demonstrating the strong relationship between verbosity and performance.

responses δ. We explore two types of model capabilities. One is general capability. We leverage the scores on the leaderboard[1](ELO) as the measurement. The other one is the capability of consuming lengthy input. For this, we investigate the influence of the size of the window context. We define the log context window size as $log(L_c/1000)$ where L_c is the context window size.

Table 6 shows the correlation on five datasets. Each number in the table is computed based on the 14 data points of 14 LLMs on the corresponding dataset. As shown, for Qasper, LongBench, and NarrativeQA datasets, a strong negative correlation is observed. This indicates that when modeling capability increases, the δ decreases accordingly. In contrast, for MMLU and NQ30, no obvious correlation is observed. The results show that training a stronger model will help avoid the influence of VC on performance for long context questions and answering tasks. However, it does not help MMLU and NQ30. In other words, *simply training a stronger model or extending context window cannot successfully disentangle VC and performance.*

[1] https://lmarena.ai/

		Lost-in-the-Middle (NQ30)				MMLU (Mixed)				All
	L_c	concise	verbose	Δ	Avg.	concise	verbose	Δ	Avg.	Δ
gemma-7b	4k	43.32	37.83	+5.49	42.18	44.59	47.52	<u>-2.93</u>	45.30	5.48
gemma-2-9b	8k	55.82	45.18	+10.64	53.44	63.75	49.07	+14.68	61.67	7.68
gemma-2-27b	8k	54.84	47.81	+7.04	53.79	68.53	45.81	**+22.72**	66.98	9.92
llama-3-8b	8k	49.55	41.75	+7.80	47.92	54.65	47.57	+7.08	53.29	6.53
llama-3-70b	8k	52.08	50.33	+1.75	51.98	60.72	52.88	+7.85	59.92	10.86
mistral-7b	8k	52.89	44.39	+8.51	48.81	64.43	46.25	+18.18	54.55	12.59
mixtral-8x7b	8k	54.86	49.92	+4.94	52.84					6.10
gpt-3.5-turbo	16k	53.90	42.93	**+10.98**	51.43	72.33	50.44	+21.89	69.56	14.83
gpt-4o	128k	63.28	52.30	**+10.98**	60.16	81.00	67.72	+13.29	79.21	12.26
claude-3-haiku	200k	61.17	48.95	+12.22	54.94	61.95	64.49	-2.55	62.61	8.43
claude-3.5-sonnet	200k	57.22	57.72	<u>-0.50</u>	57.34	71.35	56.45	+14.90	67.97	4.46
gemini-1.5-flash	1m	54.69	47.70	+6.99	53.03	58.77	47.17	+11.60	56.60	6.26
gemini-1.0-pro	32k	51.55	45.75	+5.81	50.11	54.15	48.10	+6.06	52.58	4.81
gemini-1.5-pro	2m	57.06	46.29	+10.77	55.84	62.12	54.45	+7.66	61.73	7.00
Avg		55.21	47.52	7.69	52.99	63.61	52.72	10.90	61.37	8.57

Table 3: Overall recall comparison between verbose and concise responses. **Bold**/<u>Underline</u> indicate the largest positive/negative performance gap between verbose and concise responses. Similar to Table 2, the verbose responses obtain a significantly different performance than the concise ones.

Verbosity compensation behavior of Chain-of-Thought reasoning. We further conduct an experiment to demonstrate VC also happens in Chain-of-Thought (CoT) settings. To this end, we pick 100 samples from two datasets, including MMLU and Qasper, and instruct the models to generate a Chain-of-Thought prompt. Also, we ask the model to generate as concisely as possible, where each step contains fewer than 10 tokens. If any step violates this constraint, we regard this response as verbose. Thus, the verbosity evaluator V is set as $\mathbb{1}\left(\bigvee_{s\in r}|s|>10\right)$. Based on the definition, we do not restrict the number of steps of Chain-of-Thought reasoning; a short response can be verbose as well if the length of a single step is too long.

Table 4 shows the comparison between the concise and verbose responses of two models on two datasets (Length statistics of responses in Appendix C.7). All settings display significant Δ. For gpt-turbo-3.5, the recall gap can be as large as 24.54% on MMLU dataset. *This shows that verbosity compensation can also happen in generating longer responses (Appendix C.2), such as Chain-of-Thought reasoning samples.*

5.3 Uncertainty and Verbosity Compensation

Uncertainty Evaluation. The results are shown in Figure 4. As shown in the figure, all four models show larger uncertainty when the length of the responses increases. Especially, when the length is around three tokens, the uncertainty increases shapely. These results demonstrate that 1) when LLMs generate longer responses, they are more uncertain about the sample, and 2) *when verbosity compensation happens ($V(x, y, r) = 1$), LLMs usually are more uncertain about the sample than generating concise results.*

Uncertainty and Length of Response r. We further explore the reason why uncertainty and VC are connected. We conduct a qualitative study and plot the distribution of the softmax score of the first tokens of confident and uncertain responses in Figure 1. As can be seen, for the uncertain response, the probability distribution is more flattened, and the tokens carrying much information do not stand out (ranked high) among the candidates. The model selects the one without critical information but is safer to generate, repeating the question or being off-topic and verbose. Besides, these tokens usually cannot end a sentence grammatically, such as "Avergae" or "+", the model needs to continue generations making the response longer.

5.4 Cascade Model Selection for Mitigating Verbosity Compensation

Reducing Frequency of Verbosity Compensation. Table 5 shows the comparison of using the proposed algorithm. As shown in the table, comparing the cascading algorithm and individual models, the frequency of VC decreases greatly for all settings. For instance, Mistral → GPT decreases the frequency from 63.81% (Mistral) and 31.79% (GPT) to 16.60%. It worth noting that, applying the algorithm greatly reduce the frequency of VC on

		Qasper				MMLU			
	L_c	concise	verbose	Δ	Avg.	concise	verbose	Δ	Avg.
gemma-2-9b	8k	35.82	22.73	13.09	30.12	60.63	50.00	10.62	58.42
gpt-3.5-turbo	16k	69.05	47.81	21.24	61.06	80.95	56.41	24.54	68.32

Table 4: Recall difference of Chain-of-Thought generation. Both models perform worse when they generate verbose answers, demonstrating VC also happens on CoT settings.

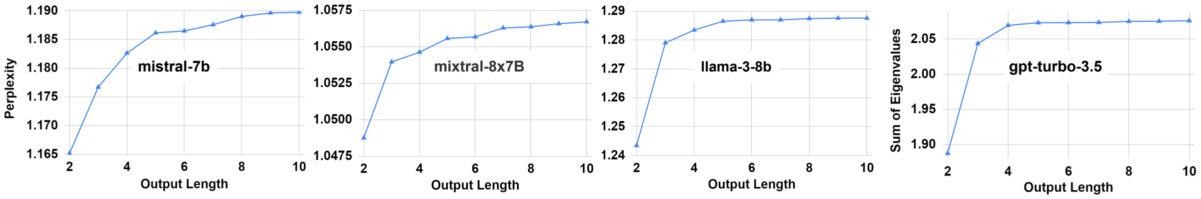


Figure 4: Uncentainty quantification of three open-sourced and one close-sourced models. The scores are averaged across all five datasets. The uncertainty increases with the increasing length of the generated output for all models.

	Qasper	LongB	NQA	NQ30	MMLU	Avg.
mistral-7b	63.81	58.95	41.40	46.59	57.40	74.19
gpt-4o	31.79	20.99	50.40	28.78	15.00	29.39
mistral → gpt	**16.60**	**14.48**	**21.00**	**18.54**	**10.20**	**16.16**
llama3-8b	68.48	17.16	32.98	23.17	20.60	32.48
claude-3.5-sonnet	13.00	35.86	27.80	29.27	26.40	26.47
lllama → claude	**8.20**	**11.80**	**14.60**	**11.71**	**7.80**	**10.82**
gemma-2-9b	46.40	35.19	52.20	27.07	22.40	36.65
gemini-pro-1.5	22.40	19.15	28.51	11.95	9.20	18.24
gemma → gemini	**15.80**	**11.14**	**18.20**	**8.29**	**4.60**	**11.61**

Table 5: Frequency of Verbosity Compensation using diverse cascade models. A → B indicates combining two models using a cascade algorithm. All settings greatly reduce the frequency of VC compared with both strong and weak models.

Dataset	ELO	Log Len
Qasper	0.09	-0.26
LongBench	-0.34	-0.53
NarrativeQA	-0.33	-0.61
MMLU	-0.05	0.13
NQ14	0.06	0.02

Table 6: Correlation between model capability and δ. Details in Appendix B.3.

both weak model and strong models. We also compare the latency of multiple LLMs in Appendix C.5.

Using Cascade Model Selection for LLM Routing. Inspired by the lower performance of the more verbose responses (Appendix B.4), we modify the CasSel to form a model routing algorithm (details in Appendix A.4). Figure 5 shows the performance of the proposed algorithm. As shown, the performance of routing is better than the baseline (Appendix C.1). Furthermore, the routing results from Gemma-2 to Gemini-1.5 are better than the individual performance of both models. This indicates that *the routing algorithm improves the performance for all settings and can surpass the performance of stronger models with less cost.*

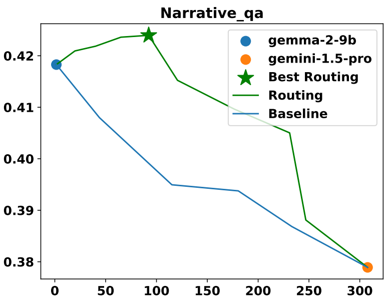


Figure 5: Routing performance of diverse models and datasets. X-axis (unit 10^{-3} dollars per sample) is the average cost. The Y-axis is the F-1 score averaged across the samples on one dataset. Routing performance (green line) is higher than the linear combination of the baseline models (blue line).

6 Conclusion

In this paper, we define VC and propose a comprehensive benchmark to evaluate 14 LLMs, revealing they suffer significantly from five types of VC. We conduct a rigorous analysis and connect VC to 1) model performance and 2) model uncertainty, shedding light on future applications and mitigation. We propose a simple but effective cascade approach to mitigate verbosity compensation in LLMs, and our extensive experiments show it is highly effective.

Ethics Statement

We include five datasets from the existing sources which we do not annotate or incorporate external resources. Thus, the dataset will not be harmful as long as the datasets themselves keep high quality. We also annotate some of the model-predicted results to classify the model results. However, the annotation is a classification task that is free of harmful content generation. Our work shows the negative part of verbosity responses, however, we do not mean verbosity is always unnecessary or harmful. Sometimes it might be helpful for the need of confirmation, or providing more context to the users.

Limitations

In this paper, we mainly show the negative effects of verbose responses on question-answering tasks. However, recent research has shown that the model can benefit from long reasoning chains (Guo et al., 2025). In this case, it is difficult to judge whether the long reasoning is verbose. Thus, future work can extend the proposed settings to diverse long-response scenarios and develop smarter verbosity detection. Another limitation is the mitigation algorithm requires multiple models to collaborate. In the future, researchers can propose to use a single model to mitigate VC, via fine-tuning or other techniques.

References

AI Anthropic. 2024. The claude 3 model family: Opus, sonnet, haiku. *Claude-3 Model Card*.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2023. Longbench: A bilingual, multitask benchmark for long context understanding. *arXiv preprint arXiv:2308.14508*.

Eleftheria Briakou, Zhongtao Liu, Colin Cherry, and Markus Freitag. 2024. On the implications of verbose llm outputs: A case study in translation evaluation. *arXiv preprint arXiv:2410.00863*.

Robert H Brookshire and Malcolm R McNeil. 2014. *Introduction to neurogenic communication disorders*. Elsevier Health Sciences.

Chao Chen, Kai Liu, Ze Chen, Yi Gu, Yue Wu, Mingyuan Tao, Zhihang Fu, and Jieping Ye. 2024. Inside: Llms' internal states retain the power of hallucination detection. *arXiv preprint arXiv:2402.03744*.

Lingjiao Chen, Matei Zaharia, and James Zou. 2023. Frugalgpt: How to use large language models while reducing cost and improving performance. *arXiv preprint arXiv:2305.05176*.

Cheng-Han Chiang and Hung-yi Lee. 2024. Overreasoning and redundant calculation of large language models. *arXiv preprint arXiv:2401.11467*.

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. 2021. A dataset of information-seeking questions and answers anchored in research papers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4599–4610, Online. Association for Computational Linguistics.

Jinhao Duan, Hao Cheng, Shiqi Wang, Chenan Wang, Alex Zavalny, Renjing Xu, Bhavya Kailkhura, and Kaidi Xu. 2023. Shifting attention to relevance: Towards the uncertainty estimation of large language models. *arXiv preprint arXiv:2307.01379*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Ekaterina Fadeeva, Roman Vashurin, Akim Tsvigun, Artem Vazhentsev, Sergey Petrakov, Kirill Fedyanin, Daniil Vasilev, Elizaveta Goncharova, Alexander Panchenko, Maxim Panov, Timothy Baldwin, and Artem Shelmanov. 2023. LM-polygraph: Uncertainty estimation for language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 446–461, Singapore. Association for Computational Linguistics.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. 2020. Unsupervised quality estimation for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:539–555.

Henry Watson Fowler. 1927. *A dictionary of modern English usage*. Clarendon Press.

Jiahui Geng, Fengyu Cai, Yuxia Wang, Heinz Koeppl, Preslav Nakov, and Iryna Gurevych. 2023. A survey of language model confidence estimation and calibration. *arXiv preprint arXiv:2311.08298*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Neha Gupta, Harikrishna Narasimhan, Wittawat Jitkrittum, Ankit Singh Rawat, Aditya Krishna Menon, and Sanjiv Kumar. 2024. Language model cascades: Token-level uncertainty and beyond. *arXiv preprint arXiv:2404.10136*.

Dan Hendrycks, Collin Burns, Steven Basart, Andrew Critch, Jerry Li, Dawn Song, and Jacob Steinhardt. 2021a. Aligning ai with shared human values. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021b. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Kung-Hsiang Huang, Philippe Laban, Alexander Fabbri, Prafulla Kumar Choubey, Shafiq Joty, Caiming Xiong, and Chien-Sheng Wu. 2024. Embrace divergence for richer insights: A multi-document summarization benchmark and a case study on summarizing diverse information from news articles. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 570–593, Mexico City, Mexico. Association for Computational Linguistics.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*.

Maor Ivgi, Ori Yoran, Jonathan Berant, and Mor Geva. 2024. From loops to oops: Fallback behaviors of language models under uncertainty. *arXiv preprint arXiv:2407.06071*.

Patrick Juola. 2008. Assessing linguistic complexity. *Language Complexity: Typology, Contact, Change. John Benjamins Press, Amsterdam, Netherlands*.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The NarrativeQA reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *arXiv preprint arXiv:2302.09664*.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. 2023. Generating with confidence: Uncertainty quantification for black-box large language models. *arXiv preprint arXiv:2305.19187*.

Zi Lin, Jeremiah Zhe Liu, and Jingbo Shang. 2022. Towards collaborative neural-symbolic graph semantic parsing via uncertainty. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 4160–4173, Dublin, Ireland. Association for Computational Linguistics.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Aman Madaan, Pranjal Aggarwal, Ankit Anand, Srividya Pranavi Potharaju, Swaroop Mishra, Pei Zhou, Aditya Gupta, Dheeraj Rajagopal, Karthik Kappaganthu, Yiming Yang, et al. 2023. Automix: Automatically mixing language models. *arXiv preprint arXiv:2310.12963*.

Andrey Malinin and Mark Gales. 2020. Uncertainty estimation in autoregressive structured prediction. *arXiv preprint arXiv:2002.07650*.

Potsawee Manakul, Adian Liusie, and Mark JF Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *arXiv preprint arXiv:2303.08896*.

Sania Nayab, Giulio Rossolini, Giorgio Buttazzo, Nicolamaria Manes, and Fabrizio Giacomelli. 2024. Concise thoughts: Impact of output length on llm reasoning and cost. *arXiv preprint arXiv:2407.19825*.

Daniel M Oppenheimer. 2006. Consequences of erudite vernacular utilized irrespective of necessity: Problems with using long words needlessly. *Applied Cognitive Psychology: The Official Journal of the Society for Applied Research in Memory and Cognition*, 20(2):139–156.

Guillem Ramírez, Alexandra Birch, and Ivan Titov. 2024. Optimising calls to large language models with uncertainty-based two-tier selection. *arXiv preprint arXiv:2405.02134*.

Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Keita Saito, Akifumi Wachi, Koki Wataoka, and Youhei Akimoto. 2023. Verbosity bias in preference labeling by large language models. *arXiv preprint arXiv:2310.10076*.

Uri Shaham, Elad Segal, Maor Ivgi, Avia Efrat, Ori Yoran, Adi Haviv, Ankit Gupta, Wenhan Xiong, Mor Geva, Jonathan Berant, and Omer Levy. 2022.

SCROLLS: Standardized CompaRison over long language sequences. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 12007–12021, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Prasann Singhal, Tanya Goyal, Jiacheng Xu, and Greg Durrett. 2023. A long way to go: Investigating length correlations in rlhf. *arXiv preprint arXiv:2310.03716*.

Yan Tao, Olga Viberg, Ryan S Baker, and Rene F Kizilcec. 2023. Auditing and mitigating cultural bias in llms. *arXiv e-prints*, pages arXiv–2311.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024a. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024b. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. MuSiQue: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

Can Wang, Bolin Zhang, Dianbo Sui, Zhiying Tum, Xiaoyu Liu, and Jiabao Kang. 2024. A survey on effective invocation methods of massive llm services. *arXiv preprint arXiv:2402.03408*.

Jiaxin Wen, Pei Ke, Hao Sun, Zhexin Zhang, Chengfei Li, Jinfeng Bai, and Minlie Huang. 2023. Unveiling the implicit toxicity in large language models. *arXiv preprint arXiv:2311.17391*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Huaixiu Steven Zheng, Swaroop Mishra, Hugh Zhang, Xinyun Chen, Minmin Chen, Azade Nova, Le Hou, Heng-Tze Cheng, Quoc V Le, Ed H Chi, et al. 2024. Natural plan: Benchmarking llms on natural language planning. *arXiv preprint arXiv:2406.04520*.

A Implementation Details

A.1 Details of Dataset Construction

The principles of constructing datasets are twofold. First, the *quality* of samples needs to be high. The questions are picked from existing human-annotated datasets, with clear answers. We also filter out Yes/No, True/False, or multi-choice questions to ensure the answer cannot be simply chosen from a set of candidate answers. Second, the dataset should be *challenging* enough for LLMs with moderate performance levels. Otherwise, if the performance is close to 100 percent, the model is too certain about the answer and the phenomena is difficult to observe. Noting that most of the benchmark datasets LLMs already obtain performance higher than 90%,

Knowledge-based question answering. Firstly, we use long-context question-answering tasks whose difficulty resides in picking out useful information across long context and gathering them to answer the question. The distractor paragraphs will also incorporate the difficulty of recognizing the needed information. Specifically, we collect the three long-form question-answering datasets as our evaluation benchmark for long-context QA. These datasets display three levels of lengths, including short (**Qasper**), medium (**LongBench**), and long (**NarrativeQA**). Qasper (Dasigi et al., 2021) is a question-answering dataset over NLP papers. It also contains extractive, abstractive, yes/no, and unanswerable questions. The average length of the source text is 4119.85 words. We also incorporate three datasets from LongBench (Bai et al., 2023) to form a new dataset. We directly name it LongBench. It include HotpotQA (Yang et al., 2018), MuSiQue (Trivedi et al., 2022), and 2WikiMultihopQA (Ho et al., 2020). The average length of the source text is 9522.36 words. NarrativeQA (Kočiský et al., 2018) is a QA dataset over entire books or movie transcripts. The answers can be abstract or extractive, yes/no, and unanswerable, and the average length is 70340.45 words.

LLMs are proven to show difficulties in understanding the information in the middle of the context (Liu et al., 2024), known as lost-in-the-middle. We pick the most challenging split of the dataset in the original work, where the gold answer is in the middle of 30 documents for a QA pair in the Natural Question dataset. We call this **NaturalQuestions_30 (NQ30).** dataset. The average length of input of NQ30 is 3602.13.

Reasoning-based question answering We modify the multi-choice answering samples in **MMLU** (Hendrycks et al., 2021b,a) so that the options work as hints to the question. In this way, the model needs to generate the answer based on the hint rather than picking out the correct option, increasing the difficulty because of the flexibility of open-ended question answers.

For each dataset, we sample 600 instances from them to form our datasets.

A.2 Details of Large Language Models

We include 2 models from Mistral AI[2], among them, **mistral-7b** is its first proposed dense model while **mixtral-8x7b** enhances the 7b model by incorporating a sparse mixture of experts. Gemini (Team et al., 2023; Reid et al., 2024) is a family of LLMs proposed by Google from which three versions of LLMs are selected, including **gemini-pro-1.0**, **gemini-flash-1.5**, and **gemini-flash-1.5**. Built from the research and technology used to create Gemini models, Gemma (Team et al., 2024a,b) is a family of lightweight, open models. We include **gemma-7b**, **gemma-2-9b**, and **gemma-2-27b** for experiments. LlaMA 3 (Dubey et al., 2024) is a family of LLMs with dense Transformer structure. We include **llama-3-8b** and **llama-3-70b** for experiments. Claude (Anthropic, 2024) is a family of large language models developed by Anthropic. We include two models in ascending order of capability: **claude-3-haiku**, **claude-3.5-sonnet**. We also include two versions of GPT models[3], including **gpt-3.5-turbo** and **gpt-4o** in experiments.

During experiments, we use the default parameters of all 14 LLMs. We run gemma, llama, and mistral models from Huggingface[4] on 8 A100 GPUs. For gpt, claude, and gemini models, we run with the official API from the official website. For all datasets, we use the same prompt shown in Table 7. We design a reinforced prompt to ensure LLM understands concise responses are required. Thus, we reinforce the prompt by repetition, and explanation, especially for the weaker models, making a fairer comparison by avoiding failing to understand instructions. We evaluate the robustness of VC against diverse prompts in Apendix C.3.

A.3 Input Chunking Algorithm

Before we feed the input into the model, we first chunk the source so that the model can consume it. As shown in Algorithm 2, we first split the source into sentences and fed as many sentences as possible to LLMs.

Algorithm 2 Input Chunking Algorithm.

```
Input: Source input $x$, query $q$, LLM window size $k$, instruction $I_w$.
Output: A chunk $c$ that LLM can consume.
  Split the source $x$ into sentences $\{s_1, s_2, \cdots, s_n\}$
  Initialize $c \leftarrow$ empty string
  Initialize length budgets $B \leftarrow k -$ count_token$(q) -$ count_token$(I_w)$.
  for $s$ in $s_1, s_2, \cdots, s_n$ do
    if count_token(c) + count_token(s) > B then
      break
    end if
    $c \leftarrow c \bigoplus s$  // $\bigoplus$ indicates concatenating two strings with a blank.
  end for
  return c
```

A.4 LLM Routing Algorithm

Model routing aims to send the sample to the proper model among a diverse collection of LLMs to generate the result so that under the same amount of API cost, the performance is better than other baselines, such as randomly choosing which model to use. We develop an LLM routing algorithm by modifying the proposed model selection algorithm. Different from model selection, we define two numbers p_c and p_v as the possibility of selecting a stronger model for concise and verbose responses. In this way, the cost is controllable to fulfill the diverse budget needs of users. It is worth noting that $V(x, y, r)$ is not available because y is not given in the routing setting. Thus, we propose a heuristic to approximate gold $V(x, y, r)$. We first sample 100 instances from the training set of the original dataset and compute the average length of the gold labels R. Then, we simply classify a response as verbose if it contains more than R tokens, represented as $V(x, y, r) = |r| > R$. Algorithm 3 shows the pseudo-code of LLM Routing. Different from the cascade algorithm for mitigating VC, this algorithm contains two probabilities that are used to control the budget of a single call. The algorithm mimics the real cost by counting tokens in

[2] https://docs.mistral.ai/getting-started/models/

[3] https://openai.com/

[4] https://huggingface.co/

You are given an article and a question. Answer the question as concisely as you can, using a single phrase if possible. Article: {Source Documents} Question: {Question q} Using a single phrase rather than a sentence. Do not repeat any question-related information or explain the answer. The answer is:

Table 7: Prompt of all models on all datasets.

the input and output, timing by the cost per token. We collect the cost of each model from website[5] and use it collected cost to ensure the fairness of comparison. The full name of all models and the price we use in LLM routing algorithm is shown in Table 8. We run each p_v, p_c setting ten times and compute the average to obtain the green lines and we run ten times that we randomly choose a weaker or stronger model with different probability to draw the blue line serving as the baseline. Specifically, for the stars in each figure, $p_v = 1$ and $p_c = 0$, degenerate to the proposed model selection algorithm.

Algorithm 3 Cascade Model Selection Algorithm for LLM Routing.

```
Input: A list of LLMs $M$, A sample $(x, y, q)$, instruction $I_w$, a verbosity detector $V()$, possibility for routing on concise responses $p_c$, possibility for routing on verbose responses $p_v$.
Output: A response $r$.
  order $M$ by model capability from weak to strong
  Set $p_c$ to 1 if $p_v \neq 1$ {We ensure routing on verbose responses first.}
  for LLM in $M$ do
      $r \leftarrow$ LLM($x \bigoplus q \bigoplus I_w$)
      if $V(x, y, r)$ is false then
          $prob \leftarrow$ A random number from 0 to 1
          if $prob \geq p_c$ then
              break {Do not route for concise responses with $1 - p_c$ probability}
          end if
      else
          $prob \leftarrow$ A random number from 0 to 1
          if $prob \geq p_v$ then
              break {Do not route for verbose responses with $1 - p_v$ probability}
          end if
      end if
  end for
  return $r$
```

Figure 6 shows the performance of the different datasets with three routing settings: Mistral 7b → GPT-4o, Gemma2 9b → Gemini-1.5-pro, and LLaMA-3-8b → Claude-3.5-sonnet. As shown, the performance of routing is better than the baselines for all models, datasets, and settings. Furthermore, the routing results from Gemma-2 to Gemini-1.5 are better than the individual performance of both models.

B Details of Experimental Results

B.1 Frequency of Verbosity Compensation

Table 9 shows the detail numbers of frequency of verbosity compensation behavior.

B.2 Uncertainty Verses Length

Table 10 shows some examples of verbose and concise responses and the distribution of the first token.

B.3 Model Capability and Relative Delta

Figure 7 plots the Correlation between model window size and δ, visualizing the negative correlation score in Table 6. The models with the stronger capability to consume lengthy input obtain lower relative delta, indicating verbosity compensation is better avoided. Also, the decreasing speed of the tendency line ranks as follows: Long (NarrativeQA), Medium (LongBench), and Short (Qasper). This means that the effectiveness of the length capability on disentangling verbosity and performance is more significant when the task has a longer input.

B.4 Truncation Principle

We conducted an experiment on Qasper dataset with llama-3-8b and found that When the response is verbose, only keep the first 4 tokens, then stop the generation. The recall only drops from 44.93% to 43.13%. In other words, if the gold answer is not in the first 4 tokens, then the model is not likely to generate it in the rest of the tokens.

[5] https://artificialanalysis.ai/models

	Input Cost	Output Cost	Model Full Name
mistral-7b	0.17	0.2	mistralai/Mistral-7B-Instruct-v0.3
mixtral-8x7b	0.24	0.24	mistralai/Mixtral-8x7B-Instruct-v0.1
llama3-8b	0.05	0.08	meta-llama/Meta-Llama-3-8B-Instruct
llama3-70b	0.59	0.79	meta-llama/Meta-Llama-3-70B-Instruct
gemma-7b	0.07	0.07	google/gemma-7b-it
gemma-2-27b	0.8	0.8	googlegemma-2-27b-it
gemma-2-9b	0.2	0.2	google/gemma-2-9b-it
claude-3-haiku	0.25	1.25	claude-3-haiku-20240307
claude-3.5-sonnet	3	15	claude-3-5-sonnet-20240620
gemini-flash-1.5	0.35	1.05	gemini-1.5-flash
gemini-pro-1.0	0.5	1.5	gemini-1.0-pro
gemini-pro-1.5	3.5	10.5	gemini-1.5-pro
gpt-3.5-turbo	0.5	1.5	gpt-3.5-turbo-0125
gpt-4o	5	15	gpt-4o-2024-05-13

Table 8: The full name and the cost of tokens for each model. The unit of input/output cost is dollar per one million tokens.

	L	Qasper	LongB	NQA	NQ30	MMLU	Avg.
mistral-7b	8k	63.81	58.95	14.20	46.59	57.40	74.19
mixtral-8x7b	8k	66.37	**4.38**	57.80	66.40	66.40	52.27
llama3-8b	8k	68.48	17.16	32.98	23.17	20.60	32.48
llama3-70b	8k	13.84	10.03	27.20	**5.85**	11.20	**13.62**
gemma-7b	4k	44.46	41.10	31.82	14.39	23.80	31.11
gemma-2-27b	8k	24.00	40.76	52.60	25.12	49.00	38.30
gemma-2-9b	8k	46.40	35.19	52.20	27.07	22.40	36.65
claude-3-haiku	200k	61.20	48.11	40.00	52.44	28.60	46.07
claude-3.5-sonnet	200k	**13.00**	35.86	27.80	29.27	26.40	26.47
gemini-flash-1.5	1m	33.60	29.40	39.80	26.83	25.20	30.97
gemini-pro-1.0	32k	20.40	31.42	27.20	29.51	30.80	27.87
gemini-pro-1.5	2m	22.40	19.15	28.51	11.95	**9.20**	18.24
gpt-3.5-turbo	16k	26.02	25.81	32.38	23.90	10.60	23.74
gpt-4o	128k	31.79	20.99	50.40	28.78	15.00	29.39
Avg		34.53	31.71	44.11	31.98	31.14	34.69

Table 9: Frequency of Verbosity Compensation. All models have verbosity compensation behavior. Among them, llama3-70b has the lowest frequency on average.

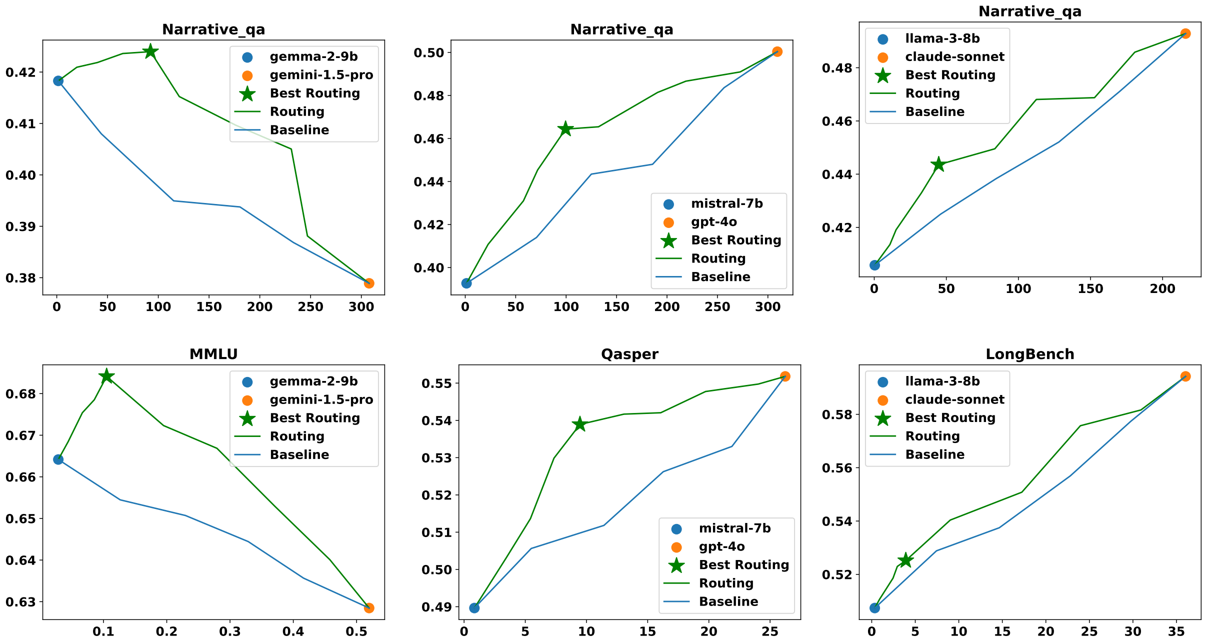


Figure 6: Routing performance of diverse models and datasets. X-axis (unit 10^{-3} dollars) is the average cost of running one sample. The Y-axis is the F-1 score averaged across the samples on one dataset. Routing performance (green line) is higher than the linear combination of the baseline models (blue line) with all datasets and models.

C Supplementary Experiments

C.1 Comparison with Uncertainty-based Routing Algorithm

We further conduct an analysis to compare the performance of the proposed routing algorithm with the uncertainty-based routing algorithm in addition to the random baselines. For the uncertainty-based routing algorithm, we first use perplexity as the metric to rank the uncertainty of the responses generated by a small model. We select top K% uncertain samples and replace them with the responses generated by the larger model. We select K from a set of $\{0, 10, 20, \cdots, 100\}$ and connect them to draw the curve in Figure 8. As can be seen, although the uncertainty-based routing algorithm can obtain a better performance than the random baseline, it is still worse than the proposed algorithm by comparing the AUC of the figure (Area Under the Curve), demonstrating the effectiveness of the proposed algorithm.

C.2 Verbosity Compensation in Trip Planning Dataset

To further demonstrate that VC generally occurs in diverse open-ended tasks with diverse response lengths, we run a trip planning dataset from the Natural-Plan benchmark (Zheng et al., 2024) using two Llama-3 models and test VC frequency and performance gaps. The task is to find the itinerary regarding the order of visiting N cities. We randomly select 500 data points from the dataset to form our dataset. For the prompt design, we follow the zero-shot prompt in the original paper and add one sentence "Answer as concisely as possible, each step contains less than 10 words". For the verbosity detector follows our CoT setting: $V(x, y, r) = \mathbb{1}\left(\bigvee_{s \in S} |s| > 10\right)$. The results are shown in Table 11. VC also occurs frequently in trip planning, demonstrating the general presence of VC in both short- and long-response open-ended tasks.

C.3 Robustness of Verbosity Compensation against Prompt Choices

As shown in Table 7 We design a reinforced prompt to ensure LLM understands concise responses are required. Thus, we reinforce the prompt by repetition, explanation, etc., especially for the weaker models, making a fairer comparison by avoiding failing to understand instructions.

We further experiment with multiple possible prompts to show VC is not overfitting to certain prompt settings. We aim to show that as long as the model knows to generate as concise as possible, we can observe significant VC behaviors.

Three Examples of Verbose Responses	
Question	What is the average length of the sentences?
Gold	15.5
Vanilla	Average sentence length: 16.5 words
Distribution	[['Average', 0.47828879952430725], ['"', 0.29009681940078735], ['15', 0.17595264315605164], ['9', 0.034647125750780106], ['average', 0.021014543250203133]]
Question	What were their accuracy results on the task?
Gold	97.32 %
Vanilla	+7 % relative gain
Distribution	[['+', 0.39178282022476196], ['"', 0.20970624685287476], ['7', 0.1850651204586029], ['96', 0.1850651204586029], ['97', 0.028380654752254486]]
Question	What's the size of the previous largest OpenIE dataset?
Gold	3,200 sentences
Vanilla	" 13 times larger "
Distribution	[['"', 0.5911211967468262], ['13', 0.3585331439971924], ['"L', 0.02022705413401127], ['Three', 0.01785031333565712], ['3', 0.012268329039216042]]
Three Examples of Concise Responses	
Question	What is the sample size of people used to measure user satisfaction?
Gold	34,432
Vanilla	34,432 users
Distribution	[['34', 0.9797704219818115], ['"', 0.008476674556732178], ['Thirty', 0.006601640954613686], ['Total', 0.004004097543656826], ['Cor', 0.0011471931356936693]]
Question	Which news organisations are the headlines sourced from?
Gold	BBC and CNN
Vanilla	BBC and CNN
Distribution	[['BBC', 0.9247239232063293], ['"', 0.040629532188177711], ['"B', 0.027924243360757828], ['B', 0.003779135411605239], ['"C', 0.0029431935399770737]]
Question	which datasets did they experiment with?
Gold	Europarl MultiUN
Vanilla	Europarl MultiUN
Distribution	[['Eu', 0.9808066487312317], ['Euro', 0.009615491144359112], [' Europ', 0.0074885510839521885], ['"', 0.0014745831722393632], ['European', 0.000614697695709765]]

Table 10: Distribution of the first token generated by llama-3-8b on Qasper dataset. When generating the verbose responses, the model is uncertain about the gold answer. Thus, starting with some commonly seen tokens. In contrast, the concise responses directly start with gold answers with high confidence.

	concise	verbose	Δ	Avg.	VC Freq.
llama-3-8b	15.18	3.62	11.56	9.22	**51.49**
llama-3-70b	21.81	4.87	**16.94**	19.63	12.87

Table 11: VC frequency and performance gaps on trip planning dataset.

Table 12 shows the performance gap on MMLU and Qasper datasets using Llama-3-8b with different prompt designs. As can be seen, compared with the original prompt, the variation of the prompt can also observe a significant Δ over both datasets. This demonstrates the robustness of VC against the choice of prompts. It is worth noting that, "Answer as concise as possible" yields the highest scores on two datasets, as well as the highest Δ, demonstrating a simpler prompt with less constraint might generate a larger performance gap between concise and verbose responses.

C.4 Evaluation of Verbosity and Performance on Same Test Instances

As shown in Table 2, and Table 3, the performance of concise and verbose samples is based on the split of the dataset. There is no overlap between the samples in the concise and verbose split. To prevent the influence of bias in different instances, we conduct an analysis that fixes the test instances and compares different models. Specifically, for each instance, we calculated the ratio of LLMs exhibiting VC behavior and reported the averaged ratio across datasets in Table 13. This approach

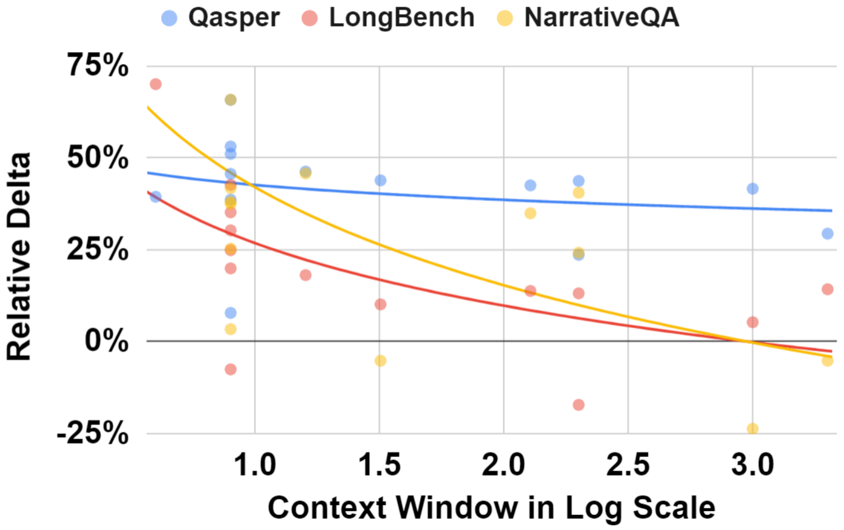


Figure 7: Correlation between model window size and δ. Results show that the model with a longer context window shows less δ on Qasper, LongBench, and NarrativeQA dataset.

MMLU				Qasper			
concise	verbose	Δ	Avg.	concise	verbose	Δ	Avg.
Prompt in Table 7							
58.4	44.82	13.57	55.6	58.99	54.6	4.39	55.98
Using a single phrase rather than a sentence.							
55.13	43.43	11.71	52.70	54.22	48.11	6.11	51.30
Answer as concise as possible.							
68.04	50.26	**17.78**	61.07	70.17	60.44	**9.73**	63.63

Table 12: Comparison between original and other variations of the prompts. VC consistently occurs, demonstrating the robustness of the VC against prompts.

	concise		verbose		overall		
	Recall	Support	Recall	Support	Δ	VC Freq.	Avg. Recall
Qasper	61.85	2272	45.63	389	**16.22**	32.46	56.59
LongBench	50.31	1912	44.22	375	6.10	30.42	48.46
NarrativeQA	38.09	2540	31.67	355	6.42	**36.29**	35.76
MMLU	65.09	1694	51.47	475	13.62	24.20	**61.79**
NQ30	53.34	1516	44.89	362	8.45	26.41	51.10

Table 13: Overall recall comparison between verbose and concise responses. Each dataset contains the prediction from all 14 LLMs.

also increases the robustness of our findings, as the support (number of samples) for each dataset is 14 times higher than when using a single model. As shown in the table, the performance δ is still pervasive for all five datasets. Specifically, on the Qasper dataset, the Δ reaches 16.22%

C.5 Latency Comparison of CaSel Algorithm and Individule Models

We conduct an analysis to compare the useless token generated and the time cost of individual models and the CaSel algorithm on two datasets using Mistral-7b and GPT-4o. To assess the number of useless tokens generated, given a re-

	Qasper					NarrativeQA				
	# Mistral	# GPT	# Total	VC Freq.	Infer. Time	# Mistral	# GPT	# Total	VC Freq.	Infer. Time
Mistral-7b	663	N/A	663	63.81	**0.80**	596	N/A	596	41.40	**1.22**
GPT-4o	N/A	207	207	31.79	1.27	N/A	327	327	50.40	14.86
Mistral → GPT	0	86	**86**	**16.60**	1.21	0	93	**93**	**21.00**	5.93

Table 14: Comparison of the number of generated useless tokens and inference time. # Mistral/GPT indicates the number of useless tokens generated by Mistral-7b and GPT-4o on the dataset. # Total is the sum of # Mistal/GPT, showing the total number of useless tokens. Infer. Time is the running time of the algorithm per sample (Unit: second). CaSel (Mistral → GPT) generated the fewest number of useless tokens and maintained the lowest VC frequency. The inference time is higher than the small model but still lower than the larger model.

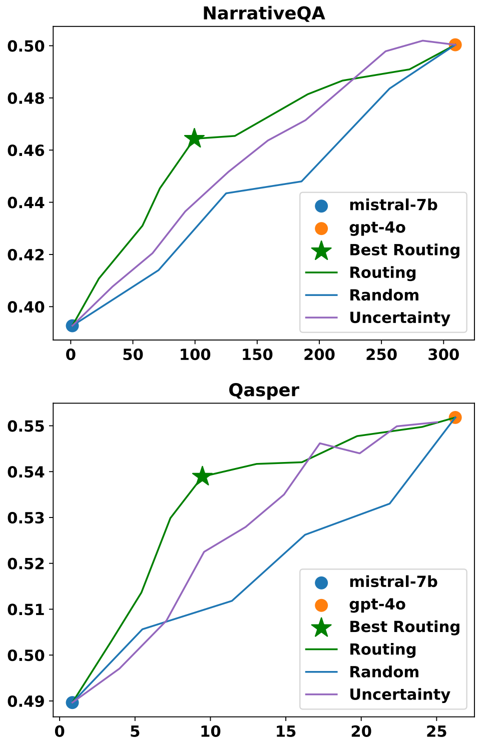


Figure 8: Routing performance of Mistral-7b to GPT-4o. X-axis (unit 10^{-3} dollars) is the average cost of running one sample. The Y-axis is the F-1 score averaged across the samples on one dataset. Routing performance (green line) is higher than the random baseline models (blue line) and uncertainty-based baseline (purple .

sponse r, we first define the useless tokens as the part with longer than gold answer in response r: $\sum_{i=1}^{N} \max(0, |r_i| - |y|)$, where N is the number of samples in a dataset. As shown in Table 14, with our proposed cascade algorithm, the total inference time might be higher than using a small model (0.79 vs. 1.21 seconds per sample) and lower than using a large model (14.86 vs. 5.93 seconds per sample), but the number of useless tokens generated is much less. On the other hand, by using the proposed algorithm, the useless tokens generated decrease from 596/327 to 93, mitigating the VC rate from 41.40% to 21.00% on the NarrativeQA dataset, demonstrating that useless tokens greatly decrease by using the proposed algorithm.

C.6 The Influence of the Digits in Responses

We analyze the performance and VC frequency of the samples with and without numbers using llama-3-8b on the Qasper and NarrativeQA dataset. The results are shown in Table 15. Although the model is easier to perform better on the sample without numbers, the VC frequency is relatively lower for the responses with digits. To understand the reason, we further inspect the Qasper dataset, we find that the samples with numbers are not as open-ended as the ones without numbers, meaning that the search space of the answers with numbers is smaller. This leads to a lower VC frequency and is easier to answer.

C.7 Response Length of Chain-of-Thought Experiments

Our evaluation is not limited to short gold answers. To demonstrate the generalization of the proposed VC behavior, we run the experiments on Chain-of-Though settings where the responses can contain more than 300 words. Table 16 shows the statistics of Chain-of-Thought experiments. The average response length can reach more than 50 words, and the VC behavior is still pervasive.

	Qasper				NarrativeQA			
	concise	verbose	Avg.	VC Freq.	concise	verbose	Avg.	VC Freq.
w/o digits	58.99	53.66	**56.18**	52.63	33.39	18.18	**27.21**	40.66
w/ digits	58.97	57.73	58.40	**45.83**	56.25	10.00	38.46	**38.46**

Table 15: Comparison between responses with digits and without digits. The responses with digits show lower verbosity compensation frequency.

	MMLU				Qasper			
	VC Freq.	Min Len.	Max Len.	Avg Len.	VC Freq.	Min Len.	Max Len.	Avg Len.
gpt-3.5-turbo	51.49	3	90	26.24	37.62	4	81	23.38
gemma-2-9b	20.79	9	107	27.92	43.56	18	103	37.08
llama-3-8b	43.56	15	333	57.14	44.15	20	185	50.15

Table 16: Lengths of the generated responses under chain-of-thought setting. The maximum length of the generated results can reach more than 300 words demonstrating that VC occurs in long response settings.