Efficient Real-time Refinement of Language Model Text Generation

Anonymous ACL submission

Abstract

Large language models (LLMs) have shown remarkable performance across a wide range of natural language tasks. However, a critical challenge remains in that they sometimes generate factually incorrect answers. To address this, while many previous work has focused on identifying errors in their generation and further refining them, they are slow in deployment since they are designed to verify the response from LLMs only after their entire generation (from the first to last tokens) is done. Further, we observe that once LLMs generate incorrect tokens early on, there is a higher likelihood that subsequent tokens will also be factually incorrect. To this end, in this work, we propose **Streaming-VR** (**Streaming V**erification and **R**efinement), a novel approach designed to enhance the efficiency of verification and refinement of LLM outputs. Specifically, the proposed Streaming-VR enables on-the-fly verification and correction of tokens as they are being generated, similar to a streaming process, ensuring that each subset of tokens is checked and refined in real-time by another LLM as the LLM constructs its response. Through comprehensive evaluations on multiple datasets, we demonstrate that our approach not only enhances the factual accuracy of LLMs, but also offers a more efficient solution compared to prior refinement methods.

1 Introduction

Large language models (LLMs) (Achiam et al., 2023; Jiang et al., 2023a; Dubey et al., 2024) have demonstrated significant advancements across various tasks, such as question answering (QA) (Yang et al., 2018; Kwiatkowski et al., 2019; Fan et al., 2019; Min et al., 2020) and its more complex real-world applications supported by information retrieval (IR) (Xiong et al., 2020; Karpukhin et al., 2020; Ni et al., 2021). However, LLMs still face notable limitations like hallucinations, mainly due to the incorrect or outdated knowledge of the model itself (Rawte et al., 2023) and the wrong application and generalization of memorized or retrieved knowledge (Jiang et al., 2024; Xu et al., 2024).

Previous approaches have sought to mitigate these inaccuracies by augmenting LLMs with external knowledge sources (Guu et al., 2020; Lewis et al., 2020; Luo et al., 2023). However, these methods often face challenges in maintaining faithfulness, as they may retrieve information that is either ungrounded or irrelevant to the context. To this end, in the realm of error identification and verification, recent research has highlighted the challenges LLMs face in accurately detecting and correcting mistakes (Peng et al., 2023).

However, traditional methods (Faltings et al., 2021; Yasunaga et al., 2021; Madaan et al., 2024) have a couple of challenges. First, they are inefficient. They focus mainly on identifying and correcting misinformation only after the complete answer has been generated. This approach not only delays error detection but also requires re-evaluating the entire text, which is computationally expensive and time-consuming. Second, cascading errors. LLMs generate text sequentially, predicting one token at a time based on the preceding context. An early error in this sequence can propagate through subsequent tokens, compounding inaccuracies throughout the response. This error propagation makes it even more challenging to correct misinformation effectively, especially when early mistakes lead to increasingly complex or numerous errors to the overall response. These challenges highlight the critical need for intermediate corrections to prevent further inaccuracies throughout the response.

In this work, we propose **Streaming-VR** (**Streaming V**erification and **R**efinement), a method designed to address the issue of error propagation in LLM-generated text. As visually depicted in Figure 1 (b), Streaming-VR evaluates model-generated answers in real-time, identifying the entire token sequence and correcting only if its

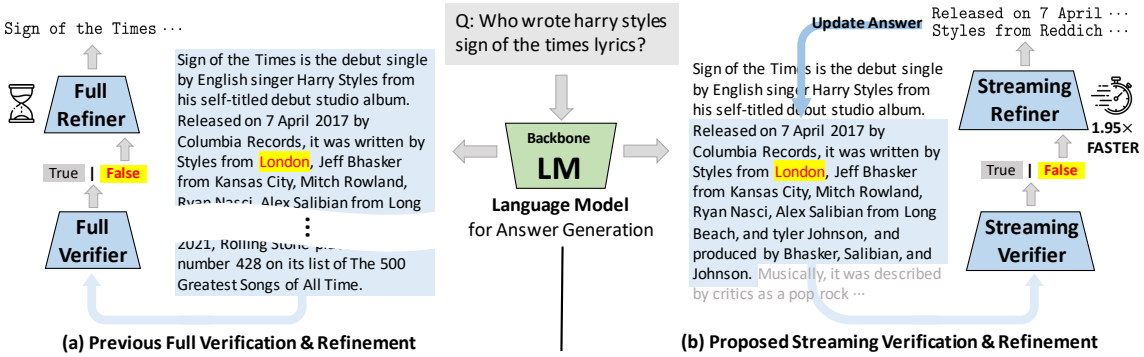


Figure 1: (**a**): Previous verify-and-refine framework after the entire answer generation. (**b**): Our proposed method, **Streaming-VR**, that verifies intermediate answers in sentence-level and refine them if identified as error, with remarkable efficiency.

subset is wrong. By employing an external verification model, Streaming-VR verifies errors during the generation process, detects inaccuracies in newly generated sequence of tokens, and promptly corrects them. Because rectification occurs immediately after verification and runs concurrently with text generation, Streaming-VR significantly enhances efficiency and improves the factual accuracy of model outputs. Our experimental results show that when LLMs generate incorrect tokens early in a sequence, it substantially increases the likelihood of subsequent sentences being factually inaccurate. Specifically, approximately 37.6% of the answers in various settings were found to contain factual inaccuracies caused by error propagation (early erroneous tokens), highlighting the critical importance of employing Streaming-VR.

We validate the effectiveness and efficiency of Streaming-VR experimentally on two benchmark datasets: achieving approximately 39.8% and 31.5% higher efficiency for ASQA (Stelmakh et al., 2022) and QuoteSum (Schuster et al., 2024) in average, respectively. Also, Streaming-VR is approximately 1.95× faster than Full-VR with comparable answer quality. We have employed Mistral 7B (Jiang et al., 2023a), LLaMA-3.1 8B (Dubey et al., 2024), and GPT-4o (Achiam et al., 2023).

2 Related Work

Large Language Models Recent advancements in language models (LMs) (Radford, 2018; Devlin et al., 2019; Liu et al., 2019; Raffel et al., 2023) and LLMs with billions of parameters, have led to significant improvements in performance across various natural language tasks. Since LMs cannot memorize or learn every real-world knowledge, several studies have explored methods to enhance their capabilities by leveraging external knowledge sources like retrieval-augmented generation (Lewis et al., 2020), for knowledge-intensive tasks. Despite the assistance of external knowledge, models often generate incorrect answers due to the failure of factual recall (Jiang et al., 2024) as they may not succeed in retrieving or applying the relevant knowledge appropriately, and generalizing memorized knowledge accurately.

To address this issue, recent research has focused on verifying the relevance and accuracy of retrieved knowledge using separate verification mechanisms (Baek et al., 2023). Additionally, methods for generating answers through on-demand retrieval of external information, employing special retrieval tokens, followed by critiquing the outputs to improve their quality, have been explored (Asai et al., 2024). A dynamic retrieval process that determines both when and what to retrieve during answer generation (Jiang et al., 2023b) has demonstrated notable improvements in knowledge-intensive tasks. This is particularly significant as the retrieve-and-generate paradigm faces significant challenges in generating lengthy texts, primarily due to difficulties in maintaining coherence and consistency. Retrieved knowledge is often fragmented and lacks contextual integration, while static retrieval methods fail to adapt dynamically to evolving text, leading to disjointed or repetitive outputs. Future research could address these issues through iterative retrieval mechanisms that refine knowledge during generation, advanced reasoning capabilities to synthesize information from multiple sources, and hierarchical retrieval strategies (Jeong et al., 2024) that organize information at different levels of granularity and difficulty leveraging an external query complexity classifier.

Language Model Verification and Refinement Other than the misinformation induced by wrong knowledge, LLM itself often generates plausible but incorrect texts (Zhang et al., 2024) (*i.e.*, hallucination). Thus, evaluating the factuality (Thorne et al., 2018; Min et al., 2023) of LLM outputs correcting inaccuracies has emerged as an important topic. Various approaches explore methods to enhance the factual accuracy of model responses and develop robust fact-checking or answer-verifying models. For instance, Dhuliawala et al. (2024) gen-

erates a series of independent questions to check the factual claims made in the model response, followed by synthesizing the answers from the verification step. Beyond evaluating or verifying the faithfulness of LLM answers, answer-correction has also become a prominent area of focus in various fields. Iterative refinement is well known to be helpful for improving generative contents of natural language (Madaan et al., 2024) and code (Faltings et al., 2021; Yasunaga et al., 2021) autonomously, but limited to the final outcome after waiting for the whole generation is done.

On the other hand, Lightman et al. (2023) demonstrates the effectiveness of process supervision by focusing on each step of the reasoning process, and allowing the model to identify and correct errors in the middle. It emphasizes the importance of intermediate verification in complex multi-step reasoning tasks like mathematical problem solving where a single error can derail the entire answer. Also, Welleck et al. (2023) employs an online training procedure for a separate corrector to learn from feedback on intermediate outputs. Nevertheless, LMs are capable of correcting errors only when their locations are identified (Tyen et al., 2024) exactly, which poses a bottleneck in improving self-correction capabilities. Furthermore, Huang et al. (2024) have demonstrated through experimental analyses that current LLMs struggle to self-correct their reasoning without external feedback, often resulting in degraded performance after attempting self-correction. Alternatively, Cobbe et al. (2021); Wang et al. (2023) utilize a trained critique model or verifier to correct errors on responses through their feedback. In addition, Gou et al. (2024) show that verification and correction can be done effectively by interacting with diverse external tools. In contrast to the previous works which have to wait for the entire answer generation or limited to the inherent answering ability, we propose a novel method with external model, that refines the specific intermediate sentence of an answer identified as incorrect, with higher efficiency.

3 Method

3.1 Preliminaries

We begin with preliminaries, formally explaining Large Language Models and traditional verify-and-refine approach, Full Verification and Refinement.

Large Language Models Let us define the process of generating an answer $\boldsymbol{a}$ to a given question $\boldsymbol{q}$ as a function: $\boldsymbol{a} = \texttt{LLM}(\boldsymbol{q})$.

For the real-time sentence-level verification and refinement, we also analyze the individual sentences in answer. To elaborate, an answer $\boldsymbol{a}$ is structured as a sequence of n sentences, expressed as $\boldsymbol{a} = [s_1, s_2, \cdots, s_n]$, where the notation $[\cdot]$ signifies concatenation in the specified order. To facilitate real-time correction of incorrect sentences within intermediate answers, we define the intermediate answer at a certain step t ($t \geq 1$) as $\boldsymbol{a}_{\leq t} = [s_1, \cdots, s_t]$ containing t sentences in total. Note that this can also be expressed as $\boldsymbol{a}_{\leq t} = [\boldsymbol{a}_{\leq t-1}, s_t]$, where s_t is the most recently generated sentence in streaming setup. We initialize $\boldsymbol{a}_{\leq 0}$ as an empty string for coherence.

In QA systems that incorporate external knowledge, such as in retrieval-augmented generation (RAG), or examples as in in-context learning (ICL), the answering process differs slightly. Formally, let $\boldsymbol{d}$ denote the external knowledge or example retrieved from the source $\mathcal{D}$. The retrieval is performed using a dedicated retrieval model `Retriever`, for a given query $\boldsymbol{q}$, defined as: $\boldsymbol{d} = \texttt{Retriever}(\boldsymbol{q}; \mathcal{D})$. This process involves ranking the retrieved data based on its relevance or similarity to the given query. After the related documents are retrieved for RAG or ICL, we now incorporate them as input to the LLMs as: $\boldsymbol{a} = \texttt{LLM}(\boldsymbol{q}, \boldsymbol{d})$.

Full-VR The simplest traditional approach for verifying and refining LLM answers, namely Full-VR (Full Verification and Refinement), is the most common strategy for improving them just by regenerating the entire responses if identified to be incorrect. While many previous works (Yasunaga et al., 2021; Cobbe et al., 2021; Welleck et al., 2023) achieve significant improvements through supplementary techniques, we focus solely on the vanilla setting, for a direct efficiency comparison without any additional methods designed to increase the factual accuracy of answers. And finally, the overall Full-VR pipeline is expressed as follow for a given query $\boldsymbol{q}$ and its answer $\boldsymbol{a} = \texttt{LLM}(\boldsymbol{q})$:

$$\tilde{\boldsymbol{a}} = \begin{cases} \boldsymbol{a} & \textit{if} \quad o = \texttt{True} \\ \texttt{Refiner}(\boldsymbol{a}) & \textit{if} \quad o = \texttt{False} \end{cases}$$

where $o = \texttt{Verifier}(\boldsymbol{a})$ is the verification output, and $\tilde{\boldsymbol{a}}$ is the final output of Full-VR.

3.2 Streaming Verification and Refinement

Our approach is structured in the following steps during the generation of answers: 1) Streaming-

Verification and 2) Streaming-Refinement if necessary; for the sentence identified as error and go back to 1). We formulate the overall framework of Streaming-VR as follow for a given query $\boldsymbol{q}$ and the t-th sentence $s_t \in \texttt{LLM}(\boldsymbol{q})$ in its answer:

$$\tilde{s}_t = \begin{cases} s_t & \textit{if} \quad o_t = \texttt{True} \\ \texttt{Refiner}(s_t) & \textit{if} \quad o_t = \texttt{False} \end{cases}$$

where $o_t = \texttt{Verifier}([\tilde{\boldsymbol{a}}_{\leq t-1}, s_t])$ is the verification output, and $\tilde{s}_t$ is the new sentence output of Streaming-VR at a certain step t. Note that the refinement model, `Refiner` takes into account the whole context of previously verified and (may have been) refined sentences, $\tilde{\boldsymbol{a}}_{\leq t-1} = [\tilde{s}_1, \cdots, \tilde{s}_{t-1}]$. After processing all the sentences by Streaming-VR, the final refined answer output should be in the form as follow: $\tilde{\boldsymbol{a}} = [\tilde{s}_1, \cdots, \tilde{s}_n]$.

The answer verification relies on the verifier's output, $o_t = \texttt{Verifier}(\boldsymbol{a}_{\leq t})$ such that $o_t \in \{\texttt{True}, \texttt{False}\}$. We utilize a fine-tuned LLM to determine whether the input is `True` or `False` by evaluating the factuality of the generated answers in sentence-level. To this end, we augment training data with true- and false-labeled sentences, as there is no proper question answering dataset labeled accurately with unit-level (*e.g.* sentence-level) answers for our streaming-verifier. The augmented sentences are made from the provided reference answer data by rephrasing it for `True` and adding wrong information for `False` by GPT-4o (Achiam et al., 2023) with the specific prompt as in Implementation Details (Appendix A). To suit real-time verification scenarios, we split the answer data into individual sentences using NLTK (Bird and Loper, 2004). These sentences are concatenated incrementally in their original order to form intermediate answers $\{\boldsymbol{a}_{\leq 1}, \cdots, \boldsymbol{a}_{\leq t}\}$, ensuring that `False`-labeled sentences only appear at the end, never in the middle. This design allows the streaming-verifier to focus on determining the factuality of the newly-generated sentence at the end. To further enhance the training process, a special sentence-separation token, `[SEP]` is inserted right before the last sentence in each intermediate answer, formatted as $[s_1, s_2, \cdots, \texttt{[SEP]}, s_t]$ for a certain stage t. This setup allows a model to be trained to verify the last sentence along with the context from the preceding `True`-labeled paragraph in the train set.

To facilitate a real-time scenario with conventional language models, we provide the entire prompt given for answering the test query to the refinement model. Additionally, we only include the retrieved passages or few-shot examples given to the generation prompt, without incorporating any extra information from external knowledge sources for refinement. This strategy ensures that the contextual information relevant to the intermediate generation processes is fully incorporated. Furthermore, as the intermediate answers are refined, they must be updated to reflect the newly refined preceding sentences, thereby enabling a continuous and coherent streaming refinement process.

4 Experiments

4.1 Datasets and Evaluation Metrics

We leverage two different datasets to evaluate the effect of Streaming-VR especially for multi-answer questions which require well-grounded responses to assess the trustworthiness of QA systems.

ASQA (Stelmakh et al., 2022) is a challenging dataset serving as a bridge between factoid and long-form QA tasks by addressing ambiguous questions that can have multiple correct answers depending on their interpretation. It is composed of 4,353 and 948 questions in the train and dev sets, respectively, while the test set is not publicly available. So we use the dev set as our test set here. ASQA provides the reference long-form answers for every questions which are originated from AmbigQA (Min et al., 2020), the ambiguous questions subset of questions from NQ (Kwiatkowski et al., 2019). In this paper, to evaluate the quantitative performance of methods on ASQA, we follow the official metrics and report: Disambiguous-Rouge (DR) as the overall score which combines ROUGE-L (R-L) (Lin, 2004) for text quality and Disambig-F1 (Dis-F1; QA score based on RoBERTa large (Liu et al., 2019)) for factual correctness.

To evaluate the consistent impact of Streaming-VR also in a retrieval-augmented generation (RAG) setting, as in the original ASQA paper (Stelmakh et al., 2022), we perform experiments using retrieved documents. Specifically, we use the top-k documents ranked by semantic similarity between the query and external documents for open-book answer generation on the ASQA dataset. These documents, retrieved from the Wikipedia corpus (2018-12-20 snapshot) using GTR-XXL (Ni et al., 2021), are provided by the LLM citation benchmark ALCE (Gao et al., 2023).

QuoteSum (Schuster et al., 2024) is also a difficult question answering dataset for Semi-Extractive Multi-source Question Answering (SEMQA), a

task designed to assess the comprehensive answering ability by summarizing information from multiple sources. To be specific, SEMQA requires models to generate a comprehensive response that integrate verbatim factual spans extracted from input sources along with supplementary non-factual text connecting them, thereby ensuring a cohesive answer. QuoteSum is made up of 4,009 semi-extractive answers to 1,376 unique questions from PAQ (Lewis et al., 2021) and NQ. For the quantitative evaluation on QuoteSum, we follow the official metrics and report: ROUGE-L, Sem-F1 for answer extraction quality, and overall SEMQA score where they do not require any model-based evaluations.

Building on the original evaluation of QuoteSum (Schuster et al., 2024), we further conduct a quantitative assessment of the variants of few-shot models. Specifically, we use dynamic prompt with top-k examples for each questions in the test set, as provided in the original paper. These examples are retrieved from the training set by selecting the passages whose queries are most similar to the target test query, based on the cosine similarity between their sentence embeddings (Ni et al., 2022).

4.2 Analyses on Efficiency

In addition to evaluating the quality and factual accuracy of model responses, we also measure token count to assess the efficiency of each method. Since our experiments rely on models accessed through the HuggingFace (Wolf et al., 2020) API, it was not feasible to implement simultaneous execution of the verifier and refiner alongside the answering model, as would occur in real-world applications. Consequently, we analyze the inference cost (*i.e.*, the number of tokens) per model for each method. This metric is crucial as the number of refined tokens directly affects to the LLM user's waiting time for response corrections. To quantify the efficiency, we define the efficiency of Streaming-VR relative to Full-VR, taking a cue from the thermal efficiency in thermodynamics, which is formulated as: (Efficiency) $:= \frac{\text{benefit}}{\text{cost}} = 1 - \frac{\mathcal{T}_\text{S}}{\mathcal{T}_\text{F}}$. Here, $\mathcal{T}_\text{S}$ and $\mathcal{T}_\text{F}$ represent the average number of generated tokens in the refinement phase per answer for Streaming-VR and Full-VR, respectively.

It should be emphasized that the tokens being verified are identical for both methods. Consider an answer with N sentences, where each sentence contains $\mathcal{T}_i$ tokens $(i = 1, \ldots, N)$. Full-VR processes all $\sum_{i=1}^{N} \mathcal{T}_i$ tokens in a single step, whereas Streaming-VR verifies sentence segments sequentially, processing $\mathcal{T}_i$ tokens at step i from $i = 1$ to N. Despite this difference in approach, both methods process the same total number of tokens, resulting in identical overall verification costs, irrespective of the number of verifier invocations.

4.3 Experimental Results and Analyses

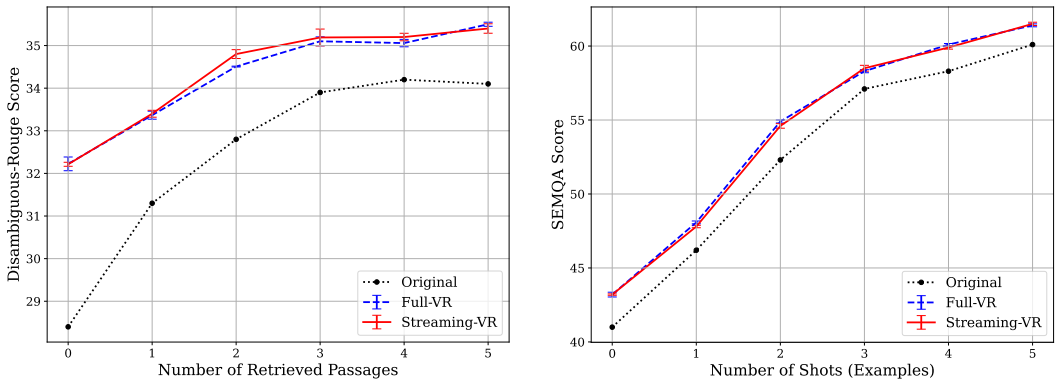
Streaming-VR delivers higher efficiency while maintaining its performance. We conduct a series of experiments on the ASQA and QuoteSum datasets to quantitatively evaluate the efficiency and effectiveness of two approaches: Streaming-VR and Full-VR. For this comparison, we first segment the model-generated answers for each test query into individual sentences, treating these sequentially arranged sentences as distinct intermediate answers. Using Streaming-VR, we verify and refine each intermediate answer in real-time, enabling dynamic adjustments as responses are generated. In contrast, Full-VR serves as the baseline, where the entire answer is verified and refined only after the complete sequence has been generated, processing the output in a single pass from start to finish. Note that for Full-VR, we utilize shared verification results of Streaming-VR: an answer is deemed incorrect if it contains at least one erroneous token in the overall context. By comparing Streaming-VR and Full-VR, we aim to demonstrate the advantages of real-time refinement in improving both answer quality and efficiency.

The main results on ASQA and QuoteSum are summarized in Table 1. Both Full-VR and Streaming-VR employ Mistral 7B as the verifier and GPT-4o as the refiner across three different backbone models (Mistral 7B, LLaMA-3.1 8B, and GPT-4o) for answer generation, as indicated in the method column. Across all response models, the final outcomes after verification-and-refinement converge to similar scores, indicating that the overall quality and faithfulness of the answers are largely determined by the refinement model.

However, we observe a notable performance decline when GPT-4o is used as the backbone for answer generation. Both Full-VR and Streaming-VR with GPT-4o lead to significant drops in Disambig-F1 on ASQA, a key metric for assessing the informativeness of long-form answers, and no other improvements on scores of QuoteSum. These results suggest that GPT-4o, which already generated high-quality answers, may be susceptible to over-correction during the refinement process, reducing the overall effectiveness of the responses.

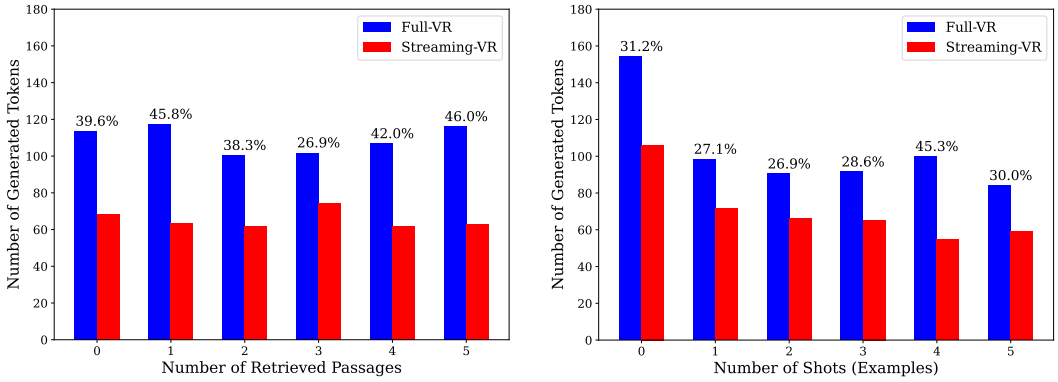
Table 1: Results of Streaming-Verification by **Mistral 7B** and Refinement by **GPT-4o** on **ASQA** and **QuoteSum** for three different backbone response models. $\mathcal{T}_{\text{Ref}}$ indicates the number of newly-generated tokens for refinement.

ASQA										
	Closed-Book					**Open-Book with 5 Passages**				
Method	ROUGE-L	Disambig-F1	DR	$\mathcal{T}_{\text{Ref}}$	Efficiency ↑	ROUGE-L	Disambig-F1	DR	$\mathcal{T}_{\text{Ref}}$	Efficiency ↑
Mistral 7B	33.6	20.7	26.4	–	–	36.4	31.2	33.7	–	–
+ Full-VR	35.3	29.6	32.3	113.6	39.6%	36.6	33.9	35.2	101.8	26.9%
+ **Streaming-VR**	35.2	29.6	32.3	68.6		36.9	33.7	35.3	74.4	
LLaMA-3.1 8B	34.0	23.7	28.4	–	–	36.6	31.7	34.1	–	–
+ Full-VR	35.2	29.4	32.2	117.4	45.8%	37.0	34.2	35.6	106.8	42.1%
+ **Streaming-VR**	35.3	29.4	32.2	63.6		36.8	34.0	35.4	61.9	
GPT-4o	36.6	34.8	35.7	–	–	37.1	35.0	36.0	–	–
+ Full-VR	35.2	29.6	32.3	100.4	38.3%	37.0	33.9	35.4	116.1	46.0%
+ **Streaming-VR**	35.3	29.4	32.2	61.9		36.9	33.9	35.4	62.7	
QuoteSum										
	Zero-Shot					**Five-Shots**				
Method	ROUGE-L	Sem-F1	SEMQA	$\mathcal{T}_{\text{Ref}}$	Efficiency ↑	ROUGE-L	Sem-F1	SEMQA	$\mathcal{T}_{\text{Ref}}$	Efficiency ↑
Mistral 7B	37.5	39.0	38.2	–	–	46.8	51.8	50.1	–	–
+ Full-VR	38.1	39.0	38.5	101.3	25.8%	57.6	49.0	53.1	72.5	24.3%
+ **Streaming-VR**	37.9	39.0	38.4	75.2		57.5	48.9	52.9	54.9	
LLaMA-3.1 8B	43.3	38.9	41.0	–	–	59.1	61.2	60.1	–	–
+ Full-VR	47.6	39.0	43.1	154.3	31.2%	60.7	62.1	61.4	84.1	30.0%
+ **Streaming-VR**	47.5	39.0	43.0	106.1		60.7	62.3	61.5	58.9	
GPT-4o	60.3	39.0	48.5	–	–	65.8	54.7	60.0	–	–
+ Full-VR	60.2	39.0	48.5	60.7	26.7%	65.8	54.7	60.0	78.9	42.2%
+ **Streaming-VR**	60.0	39.0	48.4	44.5		65.3	54.7	59.8	45.6	



(a) Performance on ASQA **(b) Performance on QuoteSum**

Figure 2: Performance comparison on various RAG and ICL settings.



(a) Efficiency on ASQA **(b) Efficiency on QuoteSum**

Figure 3: Efficiency comparison on various RAG and ICL settings. The numbers on top of the bars are their efficiency values.

This finding highlights a broader trend: refining answers with the same model used for generation—even a powerful model like GPT-4o—may not improve performance and can even degrade it. For the applications like large-scale data analysis or high-frequency user requests handling thousands or millions of queries daily, or individual users requiring detailed lengthy responses, relying on expensive models like GPT-4o for both generation and refinement can quickly exceed budgetary constraints. Therefore, Streaming-VR, which uses a more cost-effective model for response generation and GPT-4o solely for refinement, emerges as a more practical and economical solution.

Table 2: Statistics on number of tokens. Here, $\mathcal{T}_{\text{Gen}}$ is the number of generated tokens during the initial answer generation, $\mathcal{T}_{\text{Ver}}$ is the total number of tokens verified by the streaming verifier, and $\mathcal{T}_{\text{Ref}}$ is the number of generated tokens during the answer refinement phase by the streaming refiner. We report the average number of tokens per answer.

ASQA						
	Closed-Book			Open-Book w/ 5 Psgs		
Method	$\mathcal{T}_{\text{Gen}}$	$\mathcal{T}_{\text{Ver}}$	$\mathcal{T}_{\text{Ref}}$	$\mathcal{T}_{\text{Gen}}$	$\mathcal{T}_{\text{Ver}}$	$\mathcal{T}_{\text{Ref}}$
Mistral 7B	143.8	–	–	116.1	–	–
+ Full-VR	–	143.8	113.6	–	116.1	101.8
+ Streaming-VR	–	143.8	68.6	–	116.1	74.4
LLaMA-3.1 8B	101.2	–	–	66.9	–	–
+ Full-VR	–	101.2	117.4	–	66.9	106.8
+ Streaming-VR	–	101.2	63.6	–	66.9	61.9
GPT-4o	100.4	–	–	60.2	–	–
+ Full-VR	–	100.4	107.6	–	60.2	116.1
+ Streaming-VR	–	100.4	61.9	–	60.2	62.7
QuoteSum						
	Zero-Shot			Five-Shots		
Method	$\mathcal{T}_{\text{Gen}}$	$\mathcal{T}_{\text{Ver}}$	$\mathcal{T}_{\text{Ref}}$	$\mathcal{T}_{\text{Gen}}$	$\mathcal{T}_{\text{Ver}}$	$\mathcal{T}_{\text{Ref}}$
Mistral 7B	120.4	–	–	92.5	–	–
+ Full-VR	–	120.4	101.3	–	92.5	72.5
+ Streaming-VR	–	120.4	75.2	–	92.5	54.9
LLaMA-3.1 8B	161.3	–	–	83.6	–	–
+ Full-VR	–	161.3	154.3	–	83.6	84.1
+ Streaming-VR	–	161.3	106.1	–	83.6	58.9
GPT-4o	58.5	–	–	65.2	–	–
+ Full-VR	–	58.5	60.7	–	65.2	78.9
+ Streaming-VR	–	58.5	44.5	–	65.2	45.6

To assess the consistent efficacy of Streaming-VR across various settings of RAG and ICL for answer generation, we conduct additional experiments as visualized for performance in Figure 2 and for efficiency in Figure 3. The answers are generated by LLaMA-3.1 8B, verified by Mistral 7B and refined by GPT-4o on both datasets. The results show Streaming-VR's competitive performance compared to Full-VR. Streaming-VR consistently outperforms the initial answers without refinement and achieves comparable results to Full-VR. It also illustrates that Streaming-VR delivers results on par with Full-VR across all retrieved passage and example shot counts, offering performance improvements over the unrefined original response outputs of language model.

Streaming-VR enhances token efficiency In terms of efficiency, Streaming-VR offers substantial advantages over Full-VR across all models and both closed-book and open-book settings. While Full-VR refines the entire response, generating more tokens for error correction with unnecessary token refinement, Streaming-VR operates at the sentence level, refining only those sentences identified as inaccurate, resulting in significantly fewer tokens being produced. The key to Streaming-VR's efficiency lies in its ability to minimize error propagation during the generation process. By addressing inaccuracies early at the sentence level, it reduces the need for extensive revisions in subsequent stages with inefficiencies. This streamlined process leads to token savings of 39.8% for ASQA and 31.5% for QuoteSum on average.

Table 3: Result of Streaming-VR with **LLaMA-3.1 8B** as a response model. Models are indicated as Streaming-`{Verifier}{Refiner}`, where M, L and G stand for Mistral-7B, LLaMA-3.1 8B and GPT-4o, respectively.

ASQA						
	Closed-Book			Open-Book w/ 5 Psgs		
Method	R-L	Dis-F1	DR	R-L	Dis-F1	DR
LLaMA-3.1 8B	34.0	23.7	28.4	36.6	31.7	34.1
+ Streaming-MM	34.5	23.7	28.6	36.2	30.5	33.2
+ Streaming-ML	34.2	24.3	28.8	36.8	31.1	33.8
+ Streaming-MG	35.3	29.4	32.2	36.8	34.0	35.4
+ Streaming-LG	35.2	28.3	31.6	36.8	33.8	35.3
+ *Self-VR*	34.2	23.3	28.2	36.6	31.1	33.7
QuoteSum						
	Zero-Shot			Five-Shots		
Method	R-L	Sem-F1	SEMQA	R-L	Sem-F1	SEMQA
LLaMA-3.1 8B	43.3	38.9	41.0	59.1	61.2	60.1
+ Streaming-MM	39.6	38.9	39.3	58.0	61.2	59.6
+ Streaming-ML	45.2	39.0	41.9	59.9	61.7	60.8
+ Streaming-MG	47.5	39.0	43.0	60.7	62.3	61.5
+ Streaming-LG	47.7	39.0	43.1	61.0	62.3	61.6
+ *Self-VR*	42.4	38.9	40.6	57.4	61.2	59.3

Streaming-VR enhances time efficiency too We further provide a comprehensive analysis of the overall inference costs extending our evaluation beyond token efficiency in Table 2. Streaming-VR consistently produces a significantly smaller number of tokens than Full-VR, as it skips unnecessary modifications to sentences that are already correct. However, beyond the token counts, for practical deployment in real-world applications, latency plays a critical role in assessing efficiency. Specifically, latency is directly influenced by various factors other than token counts, such as the number of model invocations, execution time per call, and whether the models operate in parallel. Compared to the naïve method (purely sequential generation without any verification or refinement), sentence-level correction introduces some inherent delay, as each sentence is verified and refined before proceeding to the next sentence. However, this delay is mitigated by Streaming-VR's streamlined correction mechanism, where incorrect sentence is processed in parallel with sentence generation and verification. As a result, per-sentence verification and refinement does not accumulate linearly, keeping overall latency manageable. Importantly, the external verifier and refiner in Streaming-VR are invoked only once and they operate in parallel.

We measured the latency of each method by timing the answer refinement process when an answer is determined to be incorrect. Specifically, using Mistral 7B as the verifier and LLaMA-3.1 8B as

the refiner, Streaming-VR requires only an average of 3.07s, whereas Full-VR takes an average of 5.98s to refine the entire answer per question. This makes Streaming-VR approximately 1.95× faster than Full-VR while preserving answer factuality. This analysis underscores the novelty of Streaming-VR across the entire pipeline. For more detailed explanation along with mathematical descriptions on general cases, please refer to Appendix C.

Verification models don't need to be bigger The results in Table 3 show that verifier models can be effective without being large. On both tasks, Streaming-MG performs comparably to Streaming-LG, demonstrating that smaller models can still deliver significant performance gains. These findings highlight that the choice of verifiers is very robust in Streaming-VR, leading to choose smaller models that are resource-efficient and effective, making them particularly valuable for real-world applications with limited computational resources.

Refinement models need to be bigger The results in Table 3 highlight the critical role of a larger and more advanced model for refinement after verification, even when the verifier is relatively small. Using Mistral 7B as verifier and refiner (Streaming-MM) results in no improvement or even degraded performance across datasets and settings.

In contrast, larger refiners yield significant gains. With LLaMA-3.1 8B as the refiner (Streaming-ML), there is a modest Dis-F1 improvement for closed-book setting on ASQA, though handling multiple passages remains challenging. On QuoteSum, Streaming-ML achieves notable improvements in both zero- and five-shot settings, while Streaming-MM reduces answer quality. The most substantial boost comes from GPT-4o as the refiner (Streaming-MG), whose advanced reasoning capabilities drive superior performance in both RAG and ICL settings. These results confirm the importance of using a refiner larger than the response model for coherent and accurate answers.

LLMs still struggle with intrinsic self-correction Additionally, we conduct some experiments to evaluate the efficacy of self-verification and self-refinement within the Streaming-VR pipeline, utilizing only LLaMA-3.1 8B for backbone, verifier and refiner models. In Table 3, the rows of *Self-VR* (Self-Verification and Refinement; *i.e.,* Streaming-LL) illustrate that LLMs continue to face challenges with intrinsic self-correction with some performance drops. This result strengthens the conclusions drawn by Huang et al. (2024), which also have demonstrated that intrinsic self-correction, an approach that model attempts to rectify its initial responses using only its inherent capabilities without external feedback, degrades the response quality.

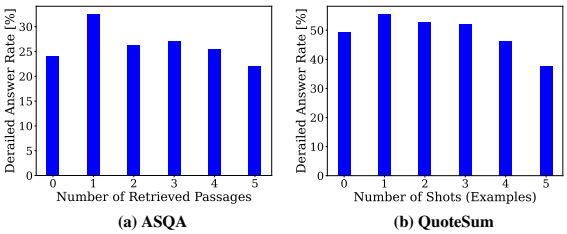


Figure 4: The ratio of derailed answers to incorrect answers. **(a)**: The rate of derailed answers on **ASQA**. **(b)**: The rate of derailed answers on **QuoteSum**.

Errors in the middle derail the entire answer As Zhang et al. (2024) point out that the mistakes or hallucinations in the middle of answer can skew the whole response, we report the statistics of model-generated answers with the rate of derailed answers on each dataset. Specifically, the answers are generated by LLaMA-3.1 8B and verified the finetuned streaming verifier as before. The rate of derailed answers is the ratio of the number of 'answers composed of false sentences in sequence from the first erroneous sentence to the last one' to 'false answers if at least one of their sentences is identified as false'. Results in Figure 4 for ASQA and QuoteSum are 26.3% and 48.9% on average across different settings for RAG and ICL, respectively. Therefore, they highlight the importance of Streaming-VR to prevent derailed responses.

5 Conclusion

In this paper, we introduce Streaming-VR, a novel approach aimed at improving the accuracy and efficiency in language model text generation. Unlike traditional methods solely relying on the final response, Streaming-VR performs real-time verification and correction of erroneous token sequences as they are being produced, with external models simultaneously with answer generation. This prevents error propagation in the early stage and reduces the errors at the end by minimizing the likelihood of compounding inaccuracies, then significantly enhances the efficiency of answer refinement. Extensive experiments for two different QA datasets have clearly demonstrated that Streaming-VR consistently achieves remarkably higher efficiency without compromising response quality.

Limitations

Despite the improvements introduced by our method, Streaming-VR, which enhances both the efficiency and effectiveness of verification and refinement in language model text generation by intervening during intermediate answer generation, there remain promising opportunities for enhancing the answer verifier. Specifically, the primary challenge is the lack of dedicated datasets for answer verification, particularly those suited for real-time scenarios. To address this, we automatically augmented data by paraphrasing sentences or introducing errors by an LLM. However, while effective, this approach carries the risk of mislabeling. Therefore, future work could focus on developing new datasets that are carefully annotated with a diverse range of answers ensuring more accurate verification and reducing the risk of incorrect labeling. Additionally, we can further extend these datasets to include fine-grained labels for multiple classes, rather than just binary ones, to accommodate different types of errors and apply adaptive strategies for subsequent refinement after verification.

Ethics Statement

In our research, we use publicly available question-answering (QA) datasets to evaluate the effectiveness and applicability of Streaming-VR in real-world scenarios. The language model we employ may inadvertently reflect biases embedded in its training data, resulting in outputs that perpetuate racism, sexism, or other forms of discrimination. Such biases can manifest even in contexts that appear neutral, highlighting the need for proactive bias detection and mitigation strategies. Moreover, harmful inputs might lead to the retrieval of offensive information or the generation of inappropriate responses by the language models. This presents a significant risk that we must recognize and address. To mitigate these issues, it is crucial to develop methods for detecting and managing offensive, inappropriate, or biased content in both user inputs and the documents retrieved within our retrieval-augmented framework. We view this as a critical area for future research because minimizing the risk of biased or harmful outputs is essential for the safe and ethical deployment of QA systems.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*.

Jinheon Baek, Soyeong Jeong, Minki Kang, Jong Park, and Sung Hwang. 2023. Knowledge-augmented language model verification. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1720–1736, Singapore. Association for Computational Linguistics.

Steven Bird and Edward Loper. 2004. NLTK: The natural language toolkit. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 214–217, Barcelona, Spain. Association for Computational Linguistics.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2024. Chain-of-verification reduces hallucination in large language models. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 3563–3578, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

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. *Preprint*, arXiv:2407.21783.

Felix Faltings, Michel Galley, Gerold Hintz, Chris Brockett, Chris Quirk, Jianfeng Gao, and Bill Dolan. 2021. Text editing by command. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5259–5274, Online. Association for Computational Linguistics.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567, Florence, Italy. Association for Computational Linguistics.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023. Enabling large language models to generate text with citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6465–6488, Singapore. Association for Computational Linguistics.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yujiu Yang, Nan Duan, Weizhu Chen, et al. 2024. Critic: Large language models can self-correct with tool-interactive critiquing. In *The Twelfth International Conference on Learning Representations*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2024. Large language models cannot self-correct reasoning yet. In *The Twelfth International Conference on Learning Representations*.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong Park. 2024. Adaptive-RAG: Learning to adapt retrieval-augmented large language models through question complexity. 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 7036–7050, Mexico City, Mexico. Association for Computational Linguistics.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023a. Mistral 7B. *Preprint*, arXiv:2310.06825.

Che Jiang, Biqing Qi, Xiangyu Hong, Dayuan Fu, Yang Cheng, Fandong Meng, Mo Yu, Bowen Zhou, and Jie Zhou. 2024. On large language models' hallucination with regard to known facts. 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 1041–1053, Mexico City, Mexico. Association for Computational Linguistics.

Zhengbao Jiang, Frank Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023b. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7969–7992, Singapore. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural Questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Patrick Lewis, Yuxiang Wu, Linqing Liu, Pasquale Minervini, Heinrich Küttler, Aleksandra Piktus, Pontus Stenetorp, and Sebastian Riedel. 2021. PAQ: 65 million probably-asked questions and what you can do with them. *Transactions of the Association for Computational Linguistics*, 9:1098–1115.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized bert pretraining approach. *Preprint*, arXiv:1907.11692.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. *Proceedings of the International Conference on Learning Representations (ICLR)*, arXiv:1711.05101.

Hongyin Luo, Yung-Sung Chuang, Yuan Gong, Tianhua Zhang, Yoon Kim, Xixin Wu, Danny Fox, Helen Meng, and James Glass. 2023. SAIL: Search-augmented instruction learning. *Preprint*, arXiv:2305.15225.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon,

Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2024. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. AmbigQA: Answering ambiguous open-domain questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5783–5797, Online. Association for Computational Linguistics.

Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith Hall, Daniel Cer, and Yinfei Yang. 2022. Sentence-T5: Scalable sentence encoders from pretrained text-to-text models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1864–1874, Dublin, Ireland. Association for Computational Linguistics.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Y. Zhao, Yi Luan, Keith B. Hall, Ming-Wei Chang, and Yinfei Yang. 2021. Large dual encoders are generalizable retrievers. *Preprint*, arXiv:2112.07899.

Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, et al. 2023. Check your facts and try again: Improving large language models with external knowledge and automated feedback. *arXiv preprint arXiv:2302.12813*.

Alec Radford. 2018. Improving language understanding by generative pre-training. Technical report, OpenAI.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer. *Preprint*, arXiv:1910.10683.

Vipula Rawte, Swagata Chakraborty, Agnibh Pathak, Anubhav Sarkar, S.M Towhidul Islam Tonmoy, Aman Chadha, Amit Sheth, and Amitava Das. 2023. The troubling emergence of hallucination in large language models - an extensive definition, quantification, and prescriptive remediations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2541–2573, Singapore. Association for Computational Linguistics.

Tal Schuster, Adam Lelkes, Haitian Sun, Jai Gupta, Jonathan Berant, William Cohen, and Donald Metzler. 2024. SEMQA: Semi-extractive multi-source question answering. 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 1363–1381.

Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. 2022. ASQA: Factoid questions meet long-form answers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8273–8288, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

Gladys Tyen, Hassan Mansoor, Victor Carbune, Peter Chen, and Tony Mak. 2024. LLMs cannot find reasoning errors, but can correct them given the error location. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 13894–13908, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Tianlu Wang, Ping Yu, Xiaoqing Ellen Tan, Sean O'Brien, Ramakanth Pasunuru, Jane Dwivedi-Yu, Olga Golovneva, Luke Zettlemoyer, Maryam Fazel-Zarandi, and Asli Celikyilmaz. 2023. Shepherd: A critic for language model generation. *arXiv preprint arXiv:2308.04592*.

Sean Welleck, Ximing Lu, Peter West, Faeze Brahman, Tianxiao Shen, Daniel Khashabi, and Yejin Choi. 2023. Generating sequences by learning to self-correct. In *The Eleventh International Conference on Learning Representations*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*.

Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. 2024. Hallucination is inevitable: An innate limitation of large language models. *arXiv preprint arXiv:2401.11817*.

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.

Michihiro Yasunaga, Jure Leskovec, and Percy Liang. 2021. LM-critic: Language models for unsupervised grammatical error correction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7752–7763, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Muru Zhang, Ofir Press, William Merrill, Alisa Liu, and Noah A. Smith. 2024. How language model hallucinations can snowball. In *Forty-first International Conference on Machine Learning*.

A Implementation Details

Models In our experiments, we employ two open-source LLMs Mistral 7B (Jiang et al., 2023a) and LLaMA-3.1 8B (Dubey et al., 2024) via Hugging Face (Wolf et al., 2020) API and GPT-4o (Achiam et al., 2023) which is accessible via OpenAI API, representing relatively small, medium, and large models, respectively. Here, these models are never fine-tuned or further trained except for their roles in verification. For the overall Streaming-VR pipeline, LLaMA-3.1 8B functions as the primary backbone language model to generate answers for given queries, while all three models are employed for verification or refinement for experiments.

Streaming Verifier We fine-tuned Mistral 7B and LLaMA-3.1 8B as verifiers using augmented training data derived from the ASQA and QuoteSum datasets. Each verifier was trained for five epochs on its respective training set, with a learning rate of 1e-5 and the AdamW (Loshchilov and Hutter, 2019) optimizer. To generate augmented data for false-labeled sentences, we embedded fake information into true sentences using GPT-4o, adjusting the temperature to 0.3, 0.5, and 0.7 to create diverse forms of inaccuracies. Rather than synthesizing entirely new sentences with large language models, which risk introducing unrelated hallucinations, we adopted this targeted augmentation strategy as a more reliable approach. This method proved highly effective in training verifiers to identify hallucinations, delivering exceptional results that highlight the importance of Streaming-VR in improving efficiency while preserving answer quality. The specific prompt used to generate incorrect information for each sentence (`Sentence`) in the provided answer (`Answer`) to a given question (`Question`) is detailed below.

```
You will be given a question (Q) with
its corresponding answer paragraph (A)
that may be incomplete and a sentence
(S) following the paragraph.\n \n Q:
{Question}\n A: {Answer}\n S: {Sentence}\n
You should modify the given sentence S, into
a plausible lie by inserting some wrong
information. Just return only the modified
'sentence (S)' itself.
```

The final test results of the finetuned verifiers used in the experiments, including a Random baseline that selects verification results arbitrarily, are presented in Table 4. For the entire pipeline, we establish a constraint that prohibits the use of any other verifier models larger than the answer generation model. This decision is based on the principle that the verifier should not exceed the capabilities of the response model, as the verifier serves merely as a supplementary tool for identifying mistakes. This reflects our considerations regarding computational costs and efficiency.

Table 4: The results of Streaming Verifier finetuned on train set for each dataset. We report the test accuracy of verifiers along with random classifier.

Method	ASQA	QuoteSum
Random	49.6	50.3
Mistral 7B	86.8	81.7
LLaMA-3.1 8B	86.7	93.0

B Additional Experimental Results

Table 5: Result of baselines on ASQA

	ASQA					
	Closed-Book			Open-Book w/ 5 Psgs		
Method	R-L	Dis-F1	DR	R-L	Dis-F1	DR
Self-RAG 7B	22.5	13.0	17.1	32.6	27.5	29.9
Self-RAG 13B	21.0	15.1	17.9	34.1	29.7	31.8
Mistral 7B	33.6	20.7	26.4	36.4	31.2	33.7
+ Streaming-VR	35.2	29.6	32.3	36.9	33.7	35.3

Additional Baselines In Table 5, we present additional results for other baselines on ASQA, Self-RAG (Asai et al., 2024), one of the most representative methods with their trained models publicly available. Self-RAG performs on-demand retrieval of external information via a specialized retrieval token, followed by a critique of the generated output to refine it. When we compare the baseline results with the similar model size, Mistral 7B demonstrates significantly superior performance to Self-RAG even without the help of refinement.

C Time-Efficiency of Streaming-VR

Streaming-VR is always faster than Full-VR The latencies of Full-VR (t_{F}) and Streaming-VR (t_{S}) can be calculated as follow:

$$t_{\text{F}} = t_{\text{Ver}} + \mathcal{T}^{\text{F}}_{\text{Ref}} \times t_{\text{Ref}}$$

$$t_{\text{S}} = N \times t_{\text{Ver}} + \mathcal{T}^{\text{S}}_{\text{Ref}} \times t_{\text{Ref}}$$

Here, we assume that the inference time of verification model per call, t_{Ver}, is the same regardless of the input length for simple comparison, and t_{Ref} denotes the token generation time per token of the refiner. Since the verifier does not generate tokens,

it follows that $t_{\text{Ver}} < t_{\text{Ref}}$. Furthermore, as shown in Table 2, with an average of seven sentences per answer ($N = 4$), we observe that $N + \mathcal{T}^{\text{S}}_{\text{Ref}} < \mathcal{T}^{\text{F}}_{\text{Ref}}$. Consequently, we can conclude that $t_{\text{S}} < t_{\text{F}}$ due to the following inequality:

$$t_{\text{S}} < \left(N + \mathcal{T}^{\text{S}}_{\text{Ref}}\right) \times t_{\text{Ref}} < \mathcal{T}^{\text{F}}_{\text{Ref}} \times t_{\text{Ref}} < t_{\text{F}}$$

When Streaming-VR is applied in real-world scenarios, where the verifier and refiner operate simultaneously alongside the answering model, the latency of Streaming-VR is updated to $t^{\text{real}}_{\text{S}}$ as:

$$t^{\text{real}}_{\text{S}} = \max\left\{t_{\text{Ver}}, \frac{\mathcal{T}^{\text{S}}_{\text{Ref}}}{N} \times t_{\text{Ref}}\right\}$$

because it only depends on the time for the last sentence to be verified (and may be refined). So it demonstrates that Streaming-VR achieves significantly lower latency compared to Full-VR, as $t^{\text{real}}_{\text{S}} \ll t_{\text{S}} < t_{\text{F}}$. As a result, comparing the number of tokens generated during refinement is sufficient to analyze the overall latency of both methods.

D GPU memory

In a practical deployment setting, Full-VR and Streaming-VR exhibit the same maximum GPU memory usage. To be specific, as the maximum number of tokens that Full-VR and Streaming-VR should handle are exactly the same, which depends on the generated (and refined) answer from the original response model. Quantitatively, when using LLaMA-3.1 8B for answer generation and Mistral 7B for verification, it requires approximately 43,900 (16,035 + 27,865) MB of GPU memory.

E Case Study

We have conducted case studies to examine the advantages of Streaming-VR in greater detail. In Table 6, we present the results of various methods for the query: *"Who does sam marry in foyle's war?"*, a British detective drama television series broadcast on ITV. The answer to the question is *Adam Wainwright*. However, this question introduces ambiguity because the answer could refer either to *Max Brown*, who portrayed *Adam Wainwright* in Season 6, or *Daniel Weyman*, who was recast as *Adam Wainwright* in Seasons 7 and 8.

As shown in Table 6, the original answer generated by the naïve LLaMA-3.1 8B model reflects the model's difficulty in understanding due to grammatical errors and ambiguous information. However, when using the methods that enable the refinement through the verification (with red to be `False` and blue to be `True`), Full-VR still contains an error regarding the name of the broadcasting center, whereas Streaming-VR successfully corrects every error in the answer. Also, we observe that the answer generated and refined by Streaming-VR preserves the structure of the original answer, whereas Full-VR generates an entirely new response, significantly altering the sentence structure.

Table 6: Case Study using LLaMA-3.1 8B as the original answer generation model, Mistral 7B as the verifier and GPT-4o as the refiner.

Q: Who does sam marry in foyle's war?	
Method	**Answer**
LLaMA-3.1 8B	In the **BBC** drama series Foyle's War, there are actually two main characters who get married. The first one is Sam Stewart, played by Michael Kitchen, who is the main character, David Foyle. However, Sam Stewart is a love interest and also a major character in the series. Sam Stewart marries a **woman named Kate Baxter**, but then also falls in love with a **woman named Samantha Stewart** who is a secretary at the police station and they get married.
+ Full-VR	In the **BBC** drama series Foyle's War, Sam Stewart, played by Honeysuckle Weeks, is a major character who works as a driver for Detective Chief Superintendent Christopher Foyle. Sam Stewart marries **Adam Wainwright**, a former Bletchley Park codebreaker who becomes a local politician. Their relationship evolves as they navigate post-war challenges, with Sam balancing her duties as a wife and her continuing involvement in Foyle's work.
+ Streaming-VR	In the **ITV** drama series Foyle's War, Sam marries Adam Wainwright, a former Bletchley Park codebreaker. Sam Stewart, played by Honeysuckle Weeks, is the driver for the main character, Christopher Foyle. However, Sam Stewart is a former driver for Christopher Foyle and also a major character. Sam Stewart marries a **man named Adam Wainwright**, a former Bletchley Park codebreaker, and they start a life together after the war ends, with Sam continuing her role as a prominent character.