Multi^2: Multi-Agent Test-Time Scalable Framework for Multi-Document Processing

Juntai Cao[1*], **Xiang Zhang**[1*], **Raymond Li**[1], **Jiaqi Wei**[2], **Chuyuan Li**[1], **Shafiq Joty**[3], **Giuseppe Carenini**[1]
[1] University of British Columbia
[2] Zhejiang University
[3] Salesforce Research
{jtcao7, raymondl, chuyuan.li, carenini}@cs.ubc.ca
xzhang23@ualberta.ca, cyou@cs.stonybrook.edu, sjoty@salesforce.com

Abstract

Recent advances in test-time scaling have shown promising results in improving large language model performance through strategic computation allocation during inference. While this approach has demonstrated strong improvements in reasoning tasks, its application to natural language generation tasks, particularly summarization, remains unexplored. Among all of the generation tasks, multi-document summarization (MDS) presents unique challenges by requiring models to extract and synthesize essential information across multiple lengthy documents. Unlike reasoning tasks, MDS demands a more complicated approach to prompt design and ensemble methods, as no single "best-overall" prompt can satisfy diverse summarization requirements. The inherent diversity in summarization needs necessitates exploring how different prompting strategies can be systematically combined to improve performance. We propose a novel framework that harnesses prompt diversity to enhance MDS performance. Our approach generates multiple candidate summaries using carefully designed prompt variations, then ensemble them through sophisticated aggregation methods to produce refined summaries. This prompt diversity enables models to capture different aspects and perspectives of the source documents, leading to more comprehensive and higher-quality summaries. To evaluate our method effectively, we also introduce two new LLM-based metrics: the Preference Alignment Score (PAS) and LLM Atom-Content-Unit score (LLM-ACU), which assess summary quality while addressing the positional bias inherent in automatic evaluations performed by LLMs. Our experiments demonstrate that leveraging prompt diversity significantly enhances summary quality, while also revealing the practical scaling boundaries for MDS tasks.

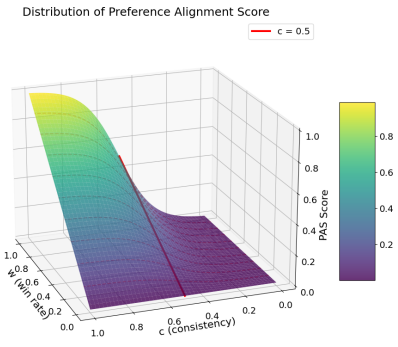

Figure 1: Visualization of the distribution Preference Alignment Score. Applying LLMs' strong language understanding ability, PAS assign higher score to the content which *consistently* gets preferred by the LLM.

1 Introduction

Test-time scaling (or inference-time scaling) has emerged as a promising approach for enhancing LLM's performance beyond traditional architectural or data improvements (OpenAI, 2024). While earlier work focused on relationships between models' capabilities, size, and training resources, recent research demonstrates that strategic compute allocation during inference can yield substantial performance gains. For instance, studies show that increased inference computation produces better results than equivalent investments in pretraining (Snell et al., 2024; Agarwal et al., 2024; Muennighoff et al., 2025).

Research on test-time scaling has largely centered on logical and math reasoning tasks, leaving traditional natural language generation (NLG) tasks relatively unexplored. This gap is particularly notable in summarization, a domain where LLMs have already demonstrated significant advances, generating summaries competitive with human performance (Xiao et al., 2024; Zhang et al.,

*Equal contribution.

Proceedings of The 5th New Frontiers in Summarization Workshop, pages 135–156
November 8, 2025

2024c; Pu et al., 2023). Beyond text generation, LLMs have also been proven effective as judges when guided by well-designed evaluation protocols (Liu et al., 2024b,c). Recent expansions in context window sizes have created new opportunities to study scaling effects on length-constrained tasks like summarization (Liu et al., 2022). However, LLMs still struggle with key challenges including hallucination, incomplete coverage, language inconsistency, and verbosity (Liu et al., 2024b; Belem et al., 2024).

In this paper, we aim to examine LLMs' summarization capabilities and their scaling properties by focusing on the multi-document summarization (MDS) task. MDS requires synthesizing and linking information across lengthy documents, handling information redundancy, maintaining factual consistency, and generating coherent and concise summaries while preserving key details. In addition, MDS demands effective reasoning to determine relevance and priority among diverse pieces of information. These characteristics make MDS particularly time- and labor-intensive (Van Veen et al., 2024). To tackle these challenges, we propose a multi-agent approach that leverages prompt ensemble to scale summarization at test time. While traditional prompt ensemble methods exist - such as (a) applying different sampling strategies to a single prompt (Li et al., 2023), or (b) varying few-shot examples within prompts (Arora et al., 2022), their direct application to summarization presents notable limitations. The first approach merely explores variations in the output space, while the second heavily relies on example-based learning, which is better suited for reasoning tasks. Furthermore, summarization differs fundamentally from reasoning tasks, where specific prompts like "*Let's think step by step*" (Kojima et al., 2022) can effectively guide models through predetermined reasoning patterns (Zhang et al., 2024d). In contrast, no single "optimal" prompt exists for generating summaries that satisfy diverse requirements. Given these distinctions, summarization demands a more sophisticated approach to prompt ensemble techniques.

Therefore, we propose Multi^2 framework (Fig. 2) to address this challenge. After generating multiple summaries through diverse prompts while maintaining consistent requirements, we employ aggregation to construct a comprehensive final summary that leverages the strengths of each summary candidate. While increased inference-time computation generally improves performance, recent studies have also identified an *inverse scaling* phenomenon, where excessive computation at test-time can paradoxically degrade performance (Gao et al., 2022; Stroebl et al., 2024). We also investigate this phenomenon by systematically varying the number of samples and examining its boundaries.

Another challenge in MDS is the reliability of automatic evaluation metrics. Traditional metrics like ROUGE (Lin, 2004) have proven insufficient for capturing summary quality, while more recent LLM-based metrics such as Auto-ACU (Liu et al., 2023b), LLMCompare (Liu et al., 2024b), and LLMRank (Liu et al., 2024c) show limitations, including constraints in contextual understanding for smaller models and persistent positional biases (Wang et al., 2024c). We specifically highlight **positional bias**, where LLMs tend to favor summaries appearing in a particular position (first or second in a pairwise comparison), leading to inconsistencies in evaluation, particularly during test-time scaling. To improve evaluation consistency, we propose two novel metrics: Preference Alignment Score (PAS) and LLM Atom-Content-Unit (LLM-ACU) score. These metrics aim to leverage LLMs' contextual understanding while incorporating mechanisms to mitigate positional bias, ensuring more reliable and robust summary assessment.

In summary, (1) We present the first comprehensive investigation of test-time scaling laws in text summarization, extending the analysis beyond traditionally explored reasoning tasks; (2) We introduce a new framework Multi^2 that enhances summarization performance through prompt ensemble at test time; (3) We enhance two existing evaluation protocols for summarization through strategic modifications and incorporating LLMs, improving quantitative assessment of summary quality and advancing automatic evaluation methodologies for summarization tasks.

2 Prompt Ensemble: A Formal Formulation

Let x denote the input text and $\mathcal{P} = \{p_1, p_2, \ldots, p_N\}$ be a collection of prompts designed to elicit different aspects of information from the underlying language model. For each prompt $p_i \in \mathcal{P}$, the model produces an output y_i according to a generation function f:

$$y_i = f(x, p_i).$$

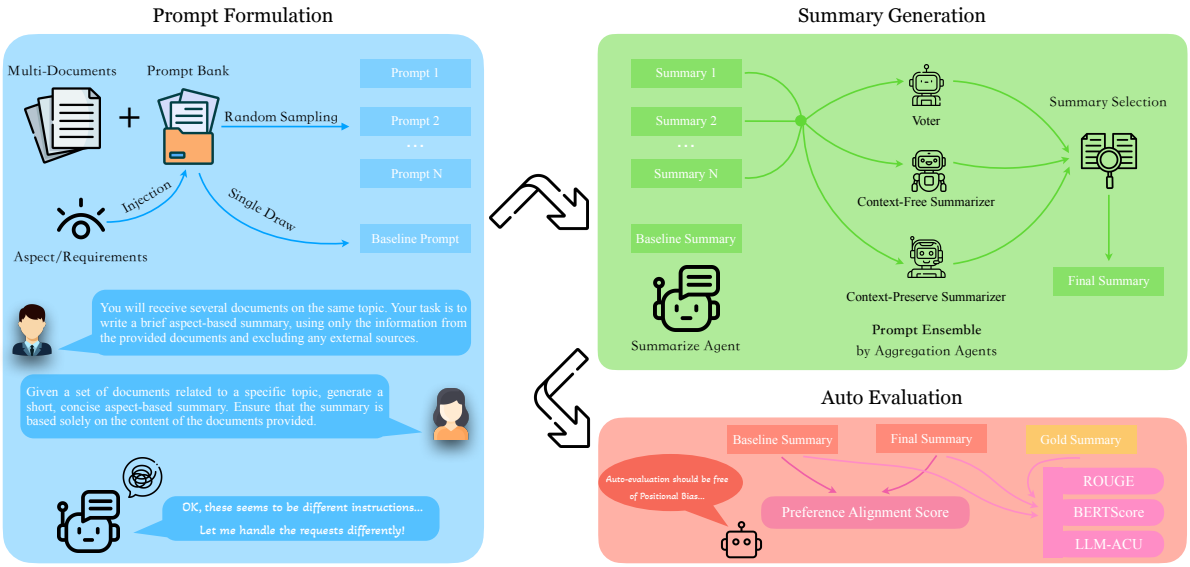

Figure 2: Overview of Multi^2 summarization inference-time scaling framework. Documents are first summarized by independent LLM agents, each guided by a different prompt from a curated prompt bank and constrained by user requirements. The resulting summaries are then processed by an aggregator (Voter, Context-Preserving Summarizer, or Context-Independent Summarizer) to generate the final consolidated summary.

The intuition behind this methodology is that different prompts p_i induce the model to focus on distinct features or details in the input x, thereby generating complementary outputs.

To combine these outputs, we define an aggregation function $g : \mathcal{Y}^N \to \mathcal{Y}$ that fuses the individual outputs $\{y_1, y_2, \ldots, y_N\}$ into a final output y:

$$y = g(y_1, y_2, \ldots, y_N).$$

The aggregation function g can take various forms depending on the specific application, with weighted averaging and majority voting being common implementations. For our MDS task, we implement three distinct formulations of g: content-independent summarization, content-preserving summarization, and voting-based aggregation. The overall system can therefore be formalized as:

$$y = g\big(f(x, p_1), f(x, p_2), \ldots, f(x, p_N)\big).$$

This formulation ensures that the final generated text y benefits from the diverse perspectives provided by the prompt ensemble. Empirical results indicate that the ensemble method consistently outperforms individual prompt-based generations, as it effectively mitigates the shortcomings of any single prompt by incorporating a broader range of contextual insights from the input x.

2.1 Prompt Space Theory

In this section, we formalize the notion of the *prompt space* and analyze its complexity in the context of Chain-of-Thought (CoT) reasoning. The prompt space, denoted as $\mathcal{P}$, represents the set of all possible step templates that a language model (LM) may generate or be guided to generate during the reasoning process. Each template $p \in \mathcal{P}$ is a discrete instruction that dictates how information is to be extracted from the latent representation $h \in R^d$ and subsequently discretized into a sequence of tokens $o = (o_1, o_2, \ldots, o_k)$. In effect, the prompt space forms the interface between the continuous latent space and the discrete textual output (Zhang et al., 2024d).

The latent vector h is assumed to encode m bits of information relevant to the task at hand. When the model follows a given prompt template p, it extracts up to s bits of information per reasoning step. Thus, each template can be viewed as a function

$$p : h \to o, \quad o \in \{0, 1\}^s,$$

where the mapping is constrained by the model's capacity to "read out" a subset of the information encoded in h. The total number of unique ways to extract s bits from m bits is given combinatorially by

$$C(m, s) = \binom{m}{s} = \frac{m!}{s!(m-s)!}.$$

This expression characterizes the *prompt space complexity*, as it represents the number of potential step templates available to the model at each CoT step.

In practice, the prompt space is not uniformly sampled; instead, the LM employs learned heuristics to navigate this enormous space. That is, while the theoretical upper bound $C(m, s)$ may be astronomically high, the effective search space is significantly reduced through task-specific training and, in many cases, human supervision. In an unsupervised setting, the model's intrinsic biases might lead it to select suboptimal templates, thereby increasing the difficulty of navigating the subsequent *answer space* $\mathcal{S}$ – the space of all possible reasoning paths and final outputs.

More formally, let ϕ denote the underlying computation that updates the hidden state:

$$h_{t+1} = \phi(h_t, p),$$

For brevity, we summarize the CoT process as follows: for $t = 1, \ldots, T$,

$$o_t = p_t(h_{t-1}), \quad h_t = \phi(h_{t-1}, p_t).$$

This compact notation encapsulates the iterative extraction of output tokens o_t and the recurrent update of the hidden state h_t via the chosen prompt p_t.

Here, the selection of each $p_t \in \mathcal{P}$ not only determines the immediate output o_t but also has a cascading effect on the evolution of the hidden state h_t and, consequently, the trajectory within the answer space $\mathcal{S}$.

This intricate relationship between the prompt space and the answer space can be seen as a two-tier search problem: first, the model must identify a suitable template p from the high-dimensional prompt space $\mathcal{P}$, and then it must effectively navigate the answer space $\mathcal{S}$ defined by the recurrence $h_t \to h_{t+1}$. Empirical evidence shows that even small deviations in the chosen template p can lead to exponentially larger errors in the final answer, underscoring the sensitivity of the overall reasoning process to prompt selection.

In summary, the prompt space theory emphasizes that the effectiveness of CoT reasoning hinges on the model's ability to manage the combinatorial complexity inherent in extracting relevant information from its latent space. Supervised methods, which incorporate task-specific guidance, can significantly reduce the search complexity from the theoretical bound $C(m, s)$ by constraining the model to a subset of high-quality prompts. This not only simplifies the navigation of the answer space but also enhances the overall reliability of the reasoning process.

3 Multi^2 Framework

3.1 Multi-Agent Summarization

Our Multi^2 test-time scaling framework for MDS is illustrated in Figure 2. The framework operates in two main stages: candidate generation and summary aggregation. In the first stage, input documents are processed by multiple independent LLM agents using randomly selected prompts from a curated prompt bank, simulating real-world summarization scenarios. The generated candidate summaries, along with the original requirements, are then passed to the aggregator module. The aggregator module implements three distinct approaches: vote, context-preserving summarizer (CPS), and context-free summarizer (CFS).

The **vote** agent evaluates all candidate summaries against the original input documents and provides a detailed explanation before selecting the best summary. We explicitly require the agent to complete its reasoning before indicating its final selection, ensuring the choice is constrained by the documented rationale. Instead of selecting the best candidate summary, CPS and CFS aggregate the candidate summaries into a final summary. The **CPS** agent generates a refined summary by consulting both the original documents and the candidate summaries, aiming for completeness and conciseness. In contrast, the **CFS** agent focuses solely on the candidate summaries without access to the original documents, producing a consolidated summary through reference-free synthesis.

3.2 Automatic Evaluation

3.2.1 Positional Bias and Motivation

Recent approaches to automatic evaluation have increasingly leveraged LLMs, either through comparative (pairwise) assessment or direct scoring mechanisms. However, both approaches face challenges. Comparative methods struggle with positional bias, an inherent limitation of LLM judges. While previous research (Liu et al., 2024c) suggested that advanced models (like `gpt-4o`) might mitigate this issue, our experiments in Appendix demonstrate that LLM evaluations remain extremely susceptible to position-dependent variations, especially

on contextual tasks like MDS. Direct scoring approaches face different challenges: defining clear scoring guidelines could be difficult, and ensuring consistent application of grading rubrics across different generations remains challenging. Moreover, the complexity of nuanced scoring - a task challenging even for human evaluators who struggle more with five-point Likert scales than binary preferences makes it particularly difficult for LLMs to provide reliable quantitative assessments.

To address these limitations and enable reliable large-scale evaluation of generated summaries, we propose two novel metrics Preference Alignment Score (PAS) and LLM-ACU score. These metrics are specifically designed to mitigate positional bias, while providing repeatable quantitative measurements for systematic comparison of summary quality.

3.2.2 Preference Alignment Score

We develop the Preference Alignment Score (PAS) as an enhancement to the LLMCompare (Liu et al., 2024b) method for quantitatively evaluating preference rates of summaries compared to a baseline. LLMCompare employs an LLM judge to evaluate two summaries against the source documents, determining which is superior (1 or 2) or if they are equivalent (tie). The pairwise comparative setup offers utility to practitioners (e.g., evaluation for A/B testing) while eliciting evaluations better aligned with human judgment from automatic evaluators (Wang et al., 2023a; Liu et al., 2024a). To address the inherent positional bias, we implement the metric with two-phase comparison process. First, we use an LLM as judge to obtain preferences with summaries (target and baseline) in their original positions. Then, we swap the positions of the two summaries and obtain a second set of preferences, relabeling them based on their new positions to eliminate labeling bias. From this two-step comparison, we compute the *win rates* (w_1, w_2) of the target summarization method against the baseline in each step, and the *consistency rate* (C) of predictions across both orderings (Figure 3).

Importantly, when evaluating consistency, if either comparison (i.e., before or after the swapping) results in a tie, we consider it consistent with any outcome in the other comparison to avoid over-penalizing borderline cases. The final PAS score is computed as follows:

$$\text{PAS} = W_{\text{pref}} \frac{1}{1 + \exp^{-k(C-0.5)}}, \qquad (1)$$

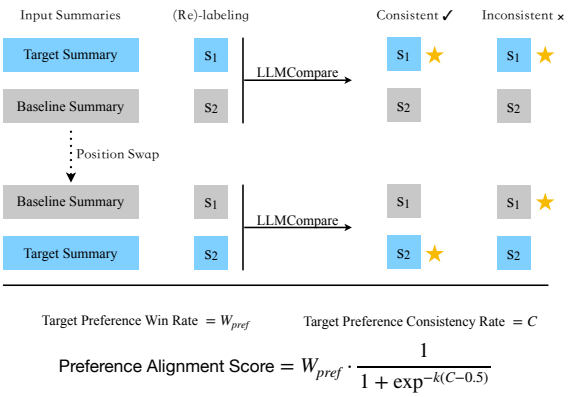

Figure 3: Two-step calculation of PAS based on LLM-Compare.

where W_{pref} refers to *preference rate* calculated from *win rates* (w_1 and w_2); C refers to *consistency score*; k controls sensitivity to consistency variations (default to 10 according to the our experiments on a validation set). In practice, the preference weight W_{pref} can be determined using either max-pooling or averaging:

$$W_{\text{pref}}^{\text{max}} = \max(w_1, w_2) \qquad (2)$$

$$W_{\text{pref}}^{\text{avg}} = \frac{(w_1 + w_2)}{2} \qquad (3)$$

Figure 1 illustrates the distribution of PAS score across different preference weights W and consistency values C.

The PAS score integrates both *preference rate* and *consistency* to ensure robust evaluation. A high PAS score requires both factors to be high, indicating consistent preference for the same summary. When model predictions remain stable, the PAS score correlates directly with the *preference rate*. However, inconsistent predictions yield low PAS scores regardless of preference outcomes, as the metric deliberately penalizes unreliable evaluations.

A low (near-zero) PAS score can result from either (a) summaries that consistently underperform the baseline, or (b) unreliable evaluations due to positional bias. Our framework effectively distinguishes between these scenarios. For instance, if evaluations consistently favor Position 1 regardless of content, the win rate might reach 100%, but the consistency score would approach 0, yielding a very low PAS score (0.06) to correctly identify unreliable evaluation. Conversely, with high consistency, the same win rate produces a PAS score near 1, indicating clear, reliable preference.

By design, PAS scores above 0.5 indicate performance better than baseline, while lower scores signal inferior performance or evaluation inconsistency. PAS deliberately employs a conservative approach to ambiguous cases, assigning low scores when no clear winner emerges due to tied quality or inconsistent judgments. This design choice prioritizes robustness and interpretability over sensitivity, treating both "tie with baseline" and "worse than baseline" scenarios similarly, as both indicate failure to establish consistent advantage.

3.2.3 LLM-ACU Score

Inspired by the Atomic Content Unit (ACU) score (Liu et al., 2023a,b), we propose an LLM-based ACU metric to quantitatively measure the completeness of summaries. The process consists of two phases. First, using few-shot prompting, we guide an LLM to extract ACUs from reference summaries. These ACUs are designed to capture essential factual units that are independently interpretable without references. In the evaluation phase, we present the extracted ACUs alongside the model generated summary and ask an LLM to determine which of the ACUs are entailed in the generated summary. The final score f for a set of summaries S and their corresponding ACU sets $\mathcal{A}$ is computed as the average unnormalized ACU score:

$$f(S, \mathcal{A}) = \frac{1}{|S|} \sum_{s \in S} \frac{e_s}{|\mathcal{A}_s|}, \tag{4}$$

where e_s represents the number of ACUs in the system output that are entailed by the gold standard ACUs $\mathcal{A}_s$ determined by the LLM. Recent work has suggested that fine-tuning primarily enables format adaptation rather than information acquisition in language models (Allen-Zhu and Li, 2024). Therefore, we do not finetune models for extracting ACUs and checking entailment, but instead leverage the advanced language understanding capabilities of LLMs directly for both steps. Therefore, we adopt `gpt-4o` for both ACU extraction and entailment verification.

4 Main Results

Our experimental results are presented in Tables 1 and 2 for preference metric (PAS scores), Tables 3, 4, and 5 for completeness metrics (LLM-ACU, ROUGE and BERTScore), across both MultiNews and OpenASP datasets. We also attach a detailed analyses examining the relationship between summary length and quality in the Appendix.

4.1 Effectiveness of Test-Time Scaling and Metrics Alignment

Our experiments demonstrate significant improvements through test-time scaling across both preference and completeness metrics. On MultiNews, starting from a low preference baseline, all scaling methods show substantial gains in overall quality. For LLM-ACU score specifically, CPS aggregator achieves the strongest performance in information coverage, with `gpt-4o-mini` showing substantial gains from a baseline of 47.13 to 54.64 with 6 samples. Similarly for OpenASP, despite beginning from a stronger preference baseline, scaling with prompt ensemble still provides notable improvements in overall quality. The LLM-ACU score show comparable trends, with CPS improving `gpt-4o-mini`'s coverage from 42.35 to 47.82 using 5 samples. These results consistently demonstrate that scaling at test time can effectively enhance both summarization quality and information coverage across different datasets.

Furthermore, Table 5 demonstrates that ROUGE scores consistently improve as the number of ensembled samples increases across both datasets, while maintaining similar degrees of BERTScore demonstrates the robustness of our approach in scaling summarization performance while preserving semantic fidelity. This trend not only reinforces the effectiveness of our test-time scaling approach from the perspective of traditional metrics, but also validates that our new metrics PAS and LLM-ACU score align well with the established evaluation frameworks.

Analysis of the results reveals two key patterns. First, CPS consistently outperforms both CFS and voting approaches across all experimental conditions, suggesting that **access to source documents during ensemble is crucial for maintaining comprehensive coverage and generating more preferred summaries**. Second, completeness improvements are more pronounced on MultiNews compared to OpenASP, indicating that **general-purpose summarization may benefit more from diverse prompt sampling for information capture**.

4.2 Scaling Boundaries and Inverse Scaling

The scaling limitations manifest differently across ensemble methods. In terms of completeness, voting shows minimal improvement across all sample sizes, suggesting that simple selection-based

# Samples	Baseline		gpt-4o				gpt-4o-mini					
	Max	Avg	CFS		CPS		CFS		CPS		Vote	
			Max	Avg	Max	Avg	Max	Avg	Max	Avg	Max	Avg
2	0.25	0.15	0.69	0.51	0.82	0.62	0.67	0.49	**0.80**	**0.61**	**0.37**	**0.23**
3			0.73	0.55	0.79	0.62	0.72	0.53	0.72	0.54	0.27	0.16
4			0.68	0.50	0.82	0.64	0.73	0.55	**0.80**	0.60	0.27	0.16
5			0.71	0.52	**0.85**	**0.69**	**0.81**	**0.62**	0.78	0.60	0.28	0.17
6			**0.79**	**0.60**	0.81	0.63	0.77	0.57	0.77	0.60	**0.37**	**0.23**

Table 1: PAS scores on Multinews dataset using `gpt-4o` and `gpt-4o-mini` models with context-free summarizer (CFS) and context-preserving summarizer (CPS). The aggregator using Vote is model-invariant. We report PAS score with max-pooled ("Max") and average ("Avg") preference scores (W_{pref}). Baseline shows both max-pooled and average PAS across all samples. Best scores per column are shown in **bold**.

# Samples	Baseline		gpt-4o				gpt-4o-mini					
	Max	Avg	CFS		CPS		CFS		CPS		Vote	
			Max	Avg	Max	Avg	Max	Avg	Max	Avg	Max	Avg
2	0.51	0.36	0.63	0.50	0.70	0.55	0.73	0.57	0.79	0.63	0.61	0.45
3			0.72	0.57	0.76	0.59	0.75	0.60	0.83	0.69	0.64	0.48
4			0.72	0.55	0.74	0.59	0.77	0.62	0.83	0.71	**0.66**	**0.51**
5			**0.74**	0.59	0.76	**0.61**	**0.82**	**0.67**	**0.86**	**0.72**	0.64	0.48
6			**0.74**	**0.60**	**0.77**	0.60	0.81	0.66	0.85	**0.72**	0.56	0.42

Table 2: PAS scores on OpenASP dataset under the same settings as in Table 1.

LLM-ACU (MultiNews)		gpt-4o		gpt-4o-mini		
# Samples	Baseline	CFS	CPS	CFS	CPS	Vote
2	47.13	48.75	51.00	49.14	52.35	47.44
3		49.25	51.11	50.03	52.88	48.31
4		49.69	51.96	51.02	54.17	48.29
5		**50.86**	**52.70**	50.95	53.90	47.65
6		50.35	52.40	**51.70**	**54.64**	**48.34**

Table 3: Comparison of LLM-ACU scores on MultiNews dataset using different ensemble methods. The vote scores are model-invariant and apply to both models. Baseline indicates single sample performance without prompt ensemble. Best score for each model and aggregation agent is shown in **bold**.

LLM-ACU (OpenASP)		gpt-4o		gpt-4o-mini		
# Samples	Baseline	CFS	CPS	CFS	CPS	Vote
2	42.35	43.05	44.16	44.36	46.07	43.86
3		44.00	45.00	45.04	47.35	44.03
4		43.64	45.51	45.05	47.55	44.47
5		44.07	**46.47**	46.13	**47.82**	44.47
6		**44.66**	46.30	**46.35**	47.46	**45.00**

Table 4: Comparison of LLM-ACU scores on OpenASP dataset under same settings as Table 3.

ensemble may be insufficient for maintaining comprehensive information coverage. The impact of document context during ensemble emerges as a crucial factor. While CFS performs better than voting, it consistently achieves lower completeness scores than CPS, indicating that losing document context during ensemble creates a ceiling on information preservation.

For preference scores, both datasets exhibit saturation points at approximately 5 samples, beyond which additional scaling yields diminishing returns. This inverse scaling phenomenon is particularly evident in MultiNews, where CPS performance peaks at 5 samples before declining at 6 samples, with the preference score nearly dropping to the same level as CFS. Completeness metrics follow a similar pattern, with `gpt-4o`'s scores using CPS plateauing around 5 samples, and `gpt-4o-mini` demonstrating comparable saturation behavior.

These observations suggest that excessive ensemble sizes may introduce noise rather than improvements, and that the choice of ensemble method

Dataset	Model	# Samples	CPS					CFS				
			R1	R2	RL	RLsum	BERTScore	R1	R2	RL	RLsum	BERTScore
MultiNews	**Baseline**	1	36.29	10.57	18.57	19.23	63.27	36.29	10.57	18.57	19.23	63.27
	gpt-4o	2	37.56	10.50	18.44	18.97	63.29	36.33	10.09	18.15	18.38	63.09
		3	37.74	10.57	18.69	19.20	63.26	36.95	10.23	18.36	18.64	63.19
		4	37.81	10.64	18.67	19.29	63.33	36.98	10.33	18.40	**18.74**	63.11
		5	**38.22**	**10.88**	18.85	**19.60**	63.35	37.09	10.35	**18.43**	18.71	63.24
		6	38.17	10.83	**18.88**	19.56	**63.39**	**37.34**	**10.45**	**18.43**	18.75	**63.27**
	gpt-4o-mini	2	39.04	10.78	18.83	20.91	63.22	37.07	10.14	18.25	18.90	**63.18**
		3	39.28	10.87	18.88	21.15	**63.26**	37.53	10.16	18.40	19.29	63.12
		4	39.42	10.86	18.87	21.39	63.19	37.81	10.21	18.40	19.61	63.06
		5	39.45	10.89	18.88	21.46	63.14	38.08	10.39	18.49	19.86	62.98
		6	**39.67**	**11.04**	**18.93**	**21.56**	63.02	**38.34**	**10.49**	**18.68**	**20.22**	63.10
OpenASP	**Baseline**	1	32.47	7.89	15.77	17.11	60.21	32.47	7.89	15.77	17.11	60.21
	gpt-4o	2	33.37	7.83	15.94	17.54	60.46	32.19	7.48	15.60	16.49	**60.31**
		3	33.37	7.87	15.91	17.54	60.50	32.40	7.43	15.63	16.70	60.19
		4	33.66	7.95	16.04	17.86	60.51	32.42	7.49	15.70	16.89	60.27
		5	33.74	8.06	16.02	17.90	**60.54**	32.67	**7.67**	**15.70**	16.95	60.26
		6	**33.98**	**8.08**	**16.08**	**18.00**	60.51	**32.71**	7.61	15.67	**17.03**	60.25
	gpt-4o-mini	2	35.37	8.14	16.20	19.56	60.10	33.19	7.56	15.76	17.79	**60.00**
		3	35.77	8.32	16.23	19.84	**60.16**	33.91	7.65	15.92	18.37	59.94
		4	35.73	8.29	16.26	19.92	60.08	34.30	7.83	15.96	18.66	59.93
		5	35.95	8.30	**16.37**	20.07	60.12	**34.53**	**7.83**	**16.01**	**18.94**	59.91
		6	**36.04**	**8.39**	16.30	**20.15**	60.10	34.52	7.79	15.95	18.87	59.84

Table 5: Comparison of ROUGE and BERTScore scores on MultiNews and OpenASP datasets using different models and ensemble sizes. The BERTScore is computed by DEBERTA-XLARGE-MNLI. Best score for each dataset, model and aggregation method is shown in **bold**.

significantly affects both quality and coverage outcomes. **This highlights the importance of identifying optimal scaling thresholds and maintaining document context throughout the ensemble process.**

4.3 Scaling Effect across Model Sizes

Our experiments with `gpt-4o` and `gpt-4o-mini` reveal interesting patterns in how model size interacts with scaling benefits. In terms of completeness scores, `gpt-4o-mini` often achieves larger relative improvements compared to `gpt-4o` when scaled through prompt ensemble. This suggests that **prompt ensemble can partially compensate for model size limitations in terms of information capture.**

Regarding preference scores, the relationship between model size and performance is more nuanced. While `gpt-4o` generally outperforms `gpt-4o-mini` on MultiNews when using CPS, the smaller model achieves competitive results with CFS. More surprisingly, on OpenASP, `gpt-4o-mini` consistently outperforms its larger version across both CFS and CPS aggregators. This suggests that the **benefits of model scale are not uniform across different summarization tasks, and that scaling smaller models, when combined with appropriate scaling strategies, may sometimes be more effective.** These findings challenge the assumption that larger models necessarily benefit more from inference-time scaling and emphasize the importance of considering both model size and ensemble size in optimization strategies.

5 Conclusion

In this work, we introduced the Multi^2 framework to scale MDS through prompt ensemble, showing that we can leverage computational resources at test time to produce more comprehensive and accurate summaries. Our metrics, PAS score and LLM-ACU score also provide more reliable assessments by effectively mitigating positional bias in summary evaluation. Through systematic analysis, we identified specific scaling boundaries in summarization tasks, offering valuable insights into scaling summarization. Our findings suggest two promising research directions: (1) incorporating test-time search algorithms to dynamically guide prompt ensemble optimization, and (2) extending our evaluation metrics to assess model performance in reasoning tasks. These directions highlight the potential of optimizing LLMs' inference-time behavior across applications where both factual accuracy and logical consistency are crucial.

Limitations

Despite demonstrating that test-time scaling improves summarization quality, our work has several

limitations. First, we restricted our experimental scope to larger general-purpose commercial LLMs rather than including smaller open-source LLMs. This decision was guided by two considerations: (1) our primary objective was to validate the Multi^2 framework's general effectiveness rather than comprehensively benchmarking various LLMs' scaling capabilities, and (2) MDS tasks require robust context understanding typically found in general-purpose, market-proven models rather than smaller models with limited contextual processing ability. Second, we did not conduct human evaluations to compare alignment between our metrics and previous ones. This decision reflects that the baseline metrics we sought to improve have already undergone comprehensive human evaluation and peer review, making additional human studies redundant for our specific research questions.

Acknowledgments

The authors thank the anonymous reviewers for their valuable feedback and suggestions. The authors acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC). Nous remercions le Conseil de recherches en sciences naturelles et en génie du Canada (CRSNG) de son soutien.

References

David H. Ackley, Geoffrey E. Hinton, and Terrence J. Sejnowski. 1985. A learning algorithm for boltzmann machines. *Cognitive Science*, 9(1):147–169.

Rishabh Agarwal, Avi Singh, Lei M. Zhang, Bernd Bohnet, Luis Rosias, Stephanie Chan, Biao Zhang, Ankesh Anand, Zaheer Abbas, Azade Nova, John D. Co-Reyes, Eric Chu, Feryal Behbahani, Aleksandra Faust, and Hugo Larochelle. 2024. Many-shot in-context learning. *Preprint*, arXiv:2404.11018.

Zeyuan Allen-Zhu and Yuanzhi Li. 2024. Physics of language models: Part 3.1, knowledge storage and extraction. *Preprint*, arXiv:2309.14316.

Shmuel Amar, Liat Schiff, Ori Ernst, Asi Shefer, Ori Shapira, and Ido Dagan. 2023. OpenAsp: A benchmark for multi-document open aspect-based summarization. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1967–1991, Singapore. Association for Computational Linguistics.

Simran Arora, Avanika Narayan, Mayee F. Chen, Laurel Orr, Neel Guha, Kush Bhatia, Ines Chami, Frederic Sala, and Christopher Ré. 2022. Ask me anything: A simple strategy for prompting language models. *Preprint*, arXiv:2210.02441.

Catarina G. Belem, Pouya Pezeskhpour, Hayate Iso, Seiji Maekawa, Nikita Bhutani, and Estevam Hruschka. 2024. From single to multi: How llms hallucinate in multi-document summarization. *Preprint*, arXiv:2410.13961.

Manik Bhandari, Pranav Narayan Gour, Atabak Ashfaq, and Pengfei Liu. 2020. Metrics also disagree in the low scoring range: Revisiting summarization evaluation metrics. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5702–5711, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. 2024. Large language monkeys: Scaling inference compute with repeated sampling. *Preprint*, arXiv:2407.21787.

Lingjiao Chen, Jared Quincy Davis, Boris Hanin, Peter Bailis, Ion Stoica, Matei Zaharia, and James Zou. 2024. Are more LLM calls all you need? towards the scaling properties of compound AI systems. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

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*.

Yann Dubois, Percy Liang, and Tatsunori Hashimoto. 2024. Length-controlled alpacaeval: A simple debiasing of automatic evaluators. In *First Conference on Language Modeling*.

Günes Erkan and Dragomir R. Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *ArXiv*, abs/1109.2128.

Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. 2019. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1074–1084, Florence, Italy. Association for Computational Linguistics.

Leo Gao, John Schulman, and Jacob Hilton. 2022. Scaling laws for reward model overoptimization. *Preprint*, arXiv:2210.10760.

Shima Gerani, Yashar Mehdad, Giuseppe Carenini, Raymond T. Ng, and Bita Nejat. 2014. Abstractive summarization of product reviews using discourse structure. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1602–1613, Doha, Qatar. Association for Computational Linguistics.

John Giorgi, Luca Soldaini, Bo Wang, Gary Bader, Kyle Lo, Lucy Lu Wang, and Arman Cohan. 2022. Open domain multi-document summarization: A comprehensive study of model brittleness under retrieval. In

Conference on Empirical Methods in Natural Language Processing.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. *Preprint*, arXiv:2111.09543.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Zhengyu Hu, Linxin Song, Jieyu Zhang, Zheyuan Xiao, Tianfu Wang, Zhengyu Chen, Nicholas Jing Yuan, Jianxun Lian, Kaize Ding, and Hui Xiong. 2024. Explaining length bias in llm-based preference evaluations. *Preprint*, arXiv:2407.01085.

Zhi Jin, Sheng Xu, Xiang Zhang, Tianze Ling, Nanqing Dong, Wanli Ouyang, Zhiqiang Gao, Cheng Chang, and Siqi Sun. 2024. Contranovo: A contrastive learning approach to enhance de novo peptide sequencing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 144–152.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Raymond Li, Wen Xiao, Linzi Xing, Lanjun Wang, Gabriel Murray, and Giuseppe Carenini. 2022. Human guided exploitation of interpretable attention patterns in summarization and topic segmentation. *Preprint*, arXiv:2112.05364.

Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen, Jian-Guang Lou, and Weizhu Chen. 2023. Making language models better reasoners with step-aware verifier. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5315–5333, Toronto, Canada. Association for Computational Linguistics.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Gili Lior, Avi Caciularu, Arie Cattan, Shahar Levy, Ori Shapira, and Gabriel Stanovsky. 2024. Seam: A stochastic benchmark for multi-document tasks. *Preprint*, arXiv:2406.16086.

Puyuan Liu, Xiang Zhang, and Lili Mou. 2022. A character-level length-control algorithm for non-autoregressive sentence summarization. *Advances in Neural Information Processing Systems*, 35:29101–29112.

Yang Liu and Mirella Lapata. 2019. Hierarchical transformers for multi-document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5070–5081, Florence, Italy. Association for Computational Linguistics.

Yinhong Liu, Han Zhou, Zhijiang Guo, Ehsan Shareghi, Ivan Vulić, Anna Korhonen, and Nigel Collier. 2024a. Aligning with human judgement: The role of pairwise preference in large language model evaluators. *arXiv preprint arXiv:2403.16950*.

Yixin Liu, Alex Fabbri, Pengfei Liu, Yilun Zhao, Linyong Nan, Ruilin Han, Simeng Han, Shafiq Joty, Chien-Sheng Wu, Caiming Xiong, and Dragomir Radev. 2023a. Revisiting the gold standard: Grounding summarization evaluation with robust human evaluation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4140–4170, Toronto, Canada. Association for Computational Linguistics.

Yixin Liu, Alexander Fabbri, Jiawen Chen, Yilun Zhao, Simeng Han, Shafiq Joty, Pengfei Liu, Dragomir Radev, Chien-Sheng Wu, and Arman Cohan. 2024b. Benchmarking generation and evaluation capabilities of large language models for instruction controllable summarization. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 4481–4501, Mexico City, Mexico. Association for Computational Linguistics.

Yixin Liu, Alexander Fabbri, Yilun Zhao, Pengfei Liu, Shafiq Joty, Chien-Sheng Wu, Caiming Xiong, and Dragomir Radev. 2023b. Towards interpretable and efficient automatic reference-based summarization evaluation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16360–16368, Singapore. Association for Computational Linguistics.

Yixin Liu, Kejian Shi, Alexander R. Fabbri, Yilun Zhao, Peifeng Wang, Chien-Sheng Wu, Shafiq Joty, and Arman Cohan. 2024c. Reife: Re-evaluating instruction-following evaluation. *Preprint*, arXiv:2410.07069.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Yashar Mehdad, Giuseppe Carenini, and Raymond T. Ng. 2014. Abstractive summarization of spoken and written conversations based on phrasal queries. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1220–1230, Baltimore, Maryland. Association for Computational Linguistics.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling. *Preprint*, arXiv:2501.19393.

OpenAI. 2024. Learning to reason with llms. https://openai.com/index/

learning-to-reason-with-llms/ [Accessed: 01/08/2025].

Xiao Pu, Mingqi Gao, and Xiaojun Wan. 2023. Summarization is (almost) dead. *Preprint*, arXiv:2309.09558.

Lin Shi, Chiyu Ma, Wenhua Liang, Weicheng Ma, and Soroush Vosoughi. 2024. Judging the judges: A systematic study of position bias in llm-as-a-judge. *Preprint*, arXiv:2406.07791.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *Preprint*, arXiv:2408.03314.

Benedikt Stroebl, Sayash Kapoor, and Arvind Narayanan. 2024. Inference scaling flaws: The limits of llm resampling with imperfect verifiers. *Preprint*, arXiv:2411.17501.

Ye Tian, Baolin Peng, Linfeng Song, Lifeng Jin, Dian Yu, Haitao Mi, and Dong Yu. 2024. Toward self-improvement of llms via imagination, searching, and criticizing. *Preprint*, arXiv:2404.12253.

Dave Van Veen, Cara Van Uden, Louis Blankemeier, Jean-Benoit Delbrouck, Asad Aali, Christian Bluethgen, Anuj Pareek, Malgorzata Polacin, William Collins, Neera Ahuja, Curtis P. Langlotz, Jason Hom, Sergios Gatidis, John Pauly, and Akshay S. Chaudhari. 2024. Adapted large language models can outperform medical experts in clinical text summarization. *Nature Medicine*.

Chaojie Wang, Yanchen Deng, Zhiyi Lyu, Liang Zeng, Jujie He, Shuicheng Yan, and Bo An. 2024a. Q*: Improving multi-step reasoning for llms with deliberative planning. *arXiv preprint arXiv:2406.14283*.

Junlin Wang, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Zou. 2024b. Mixture-of-agents enhances large language model capabilities. *Preprint*, arXiv:2406.04692.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. 2024c. Large language models are not fair evaluators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9440–9450, Bangkok, Thailand. Association for Computational Linguistics.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. 2023a. Large language models are not fair evaluators. *arXiv preprint arXiv:2305.17926*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023b. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. 2024. Scaling inference computation: Compute-optimal inference for problem-solving with language models. In *The 4th Workshop on Mathematical Reasoning and AI at NeurIPS'24*.

Wen Xiao, Yujia Xie, Giuseppe Carenini, and Pengcheng He. 2024. Personalized abstractive summarization by tri-agent generation pipeline. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 570–581, St. Julian's, Malta. Association for Computational Linguistics.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. *Preprint*, arXiv:2210.03629.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. *ArXiv*, abs/2106.11520.

Eric Zelikman, Georges Harik, Yijia Shao, Varuna Jayasiri, Nick Haber, and Noah D. Goodman. 2024. Quiet-star: Language models can teach themselves to think before speaking. *Preprint*, arXiv:2403.09629.

Dan Zhang, Sining Zhoubian, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. 2024a. Rest-mcts*: Llm self-training via process reward guided tree search. *Preprint*, arXiv:2406.03816.

Di Zhang, Jianbo Wu, Jingdi Lei, Tong Che, Jiatong Li, Tong Xie, Xiaoshui Huang, Shufei Zhang, Marco Pavone, Yuqiang Li, Wanli Ouyang, and Dongzhan Zhou. 2024b. Llama-berry: Pairwise optimization for o1-like olympiad-level mathematical reasoning. *Preprint*, arXiv:2410.02884.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020a. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. *Preprint*, arXiv:1912.08777.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020b. Bertscore: Evaluating text generation with bert. *Preprint*, arXiv:1904.09675.

Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B. Hashimoto. 2024c. Benchmarking large language models for news summarization. *Transactions of the Association for Computational Linguistics*, 12:39–57.

Xiang Zhang, Muhammad Abdul-Mageed, and Laks VS Lakshmanan. 2024d. Autoregressive+ chain of thought= recurrent: Recurrence's role in language models' computability and a revisit of recurrent transformer. *arXiv preprint arXiv:2409.09239*.

Eric Zhao, Pranjal Awasthi, and Sreenivas Gollapudi. 2025. Sample, scrutinize and scale: Effective inference-time search by scaling verification. *Preprint*, arXiv:2502.01839.

Yuchen Zhuang, Xiang Chen, Tong Yu, Saayan Mitra, Victor Bursztyn, Ryan A. Rossi, Somdeb Sarkhel, and Chao Zhang. 2024. Toolchain*: Efficient action space navigation in large language models with a* search. In *The Twelfth International Conference on Learning Representations*.

A Related Work

A.1 Test-time scaling

Test-time scaling strategies can be broadly classified into three categories: repeated sampling, deliberative approaches, and self-refinement. **Repeated sampling** leverages techniques like temperature sampling (Ackley et al., 1985), top-k, and top-p sampling (Holtzman et al., 2020) to generate diverse outputs, which are then enhanced through aggregation strategies such as majority voting (Wang et al., 2023b), weighted majority voting (Li et al., 2023), or best-of-n selection (Cobbe et al., 2021). Recent work (Brown et al., 2024; Wu et al., 2024; Stroebl et al., 2024; Zhao et al., 2025) demonstrates that repeated sampling can significantly expand LLM capabilities across various domains. **Deliberative approaches** incorporate structured reasoning through methods like chain-of-thought prompting (Wei et al., 2023) and tree search. These approaches range from informed search methods (Zhuang et al., 2024; Wang et al., 2024a) to Monte Carlo Tree Search (MCTS) variants (Tian et al., 2024; Zhang et al., 2024b,a). A key characteristic of tree search methods is to use process reward models (PRMs) to guide the search trajectory during generation (Yao et al., 2023; Zelikman et al., 2024). **Self-refinement** (Madaan et al., 2023) enables models to iteratively improve their responses through self-critique and editing. Additionally, all categories of test-time scaling methods can be enhanced through model ensembling (Wang et al., 2024b; Jin et al., 2024; Chen et al., 2024) to combine the strengths of multiple models to achieve better performance.

Yet tree search methods often struggle with the high-dimensional search space created by multiple source documents, making it computationally intensive to explore meaningful trajectories. Self-refinement approaches, which rely on iterative improvements, may lead to information loss as they tend to focus on refining a single perspective rather than maintaining diverse viewpoints from multiple documents. In our work, we adopt the repeated sampling approach to scale MDS at test time, using diverse prompts to generate multiple perspectives that are then consolidated through specialized aggregation methods.

A.2 Multi Document Summarization and Evaluation

Multi-document summarization (MDS) has evolved significantly from traditional methods (Erkan and Radev, 2004; Mehdad et al., 2014; Gerani et al., 2014) to modern approaches powered by deep neural networks (Liu and Lapata, 2019; Zhang et al., 2020a; Giorgi et al., 2022; Li et al., 2022). The advent of LLMs has boosted MDS capabilities even further, with models demonstrating impressive zero- and few-shot performance (Zhang et al., 2024c). Recent work to improve LLMs' summarization abilities has shifted the focus from models' architectural modifications to exploring various prompting strategies (Xiao et al., 2024; Liu et al., 2024b). Despite these advances, MDS continues to face challenges including maintaining cross-document consistency, ensuring factual accuracy, and addressing content incompleteness where key information may be omitted (Belem et al., 2024). In this paper, we propose a test-time approach that addresses these challenges by generating summaries more aligned with user preferences. Traditional evaluation metrics for summarization, such as ROUGE (Lin, 2004), only rely on lexical overlap with reference summaries. These metrics often fail to capture semantic similarity and summary quality adequately (Bhandari et al., 2020). This limitation has led to the development of learned metrics that better align with human judgments (Yuan et al., 2021; Zhang et al., 2020b). The emergence of LLMs has enabled even more sophisticated evaluation approaches. Recent work has explored using LLMs as evaluation agents (Liu et al., 2024b,c), demonstrating their ability to assess multiple quality dimensions including coherence, faithfulness, and informativeness. However, these approaches face challenges such as positional bias and inconsistency across different model sizes (Wang et al., 2024c; Shi et al., 2024). In this paper, we also try to address these limitations by proposing two metrics that remain consistent regardless of position or choice of evaluation model.

B Experiment Setup

Datasets. We evaluate our framework on two datasets: MultiNews (Fabbri et al., 2019) for general-purpose summarization and OpenASP (Amar et al., 2023) for aspect-based summarization. For a balanced comparison, we conduct our experiments on the test sets of both datasets. For MultiNews, we select the first 600 entries from its test set to match the size of OpenASP's test set.

Models. To investigate scaling properties and leverage extended context windows, we evaluate our framework using two state-of-the-art models of different scales: `gpt-4o` and `gpt-4o-mini`. These models enable us to analyze how performance scales with model size while maintaining consistent architectural characteristics.

Prompt Bank. We adapt the prompt collection from Lior et al. (2024) to explore the prompt space. While some prompts in their work were originally designed for extractive summarization, we modified them for abstractive summary generation while preserving their core instructional elements. The prompts are attached in Appendix.

Implementation Details. We establish our baseline using summaries generated by `gpt-4o` with a single prompt randomly selected from our prompt bank using a fixed random seed. We scale summarization by applying different aggregation methods to the generated summaries. For voting-based aggregation, we exclusively use `gpt-4o`, since this method operates independently of the generator model and focuses on the well-defined task of selecting the optimal summary from available candidates, rather than producing new text. In contrast, generative aggregation methods synthesize entirely new summaries. To ensure experimental rigor, we execute each configuration with the default temperature setting at 0.8. Our experimental design focuses on two primary variables: (1) inference model size and (2) scaling factor, determined by the number of ensembled samples.

Model & Cost Analyses. The specific model versions used in our experiments are listed in Table 6. The total computational cost for all experiments was approximately $1,000 USD.

Evaluation Protocols. Our experimental evaluation employs multiple complementary metrics: ROUGE Score (Lin, 2004) and BERTScore (Zhang et al., 2020b) serves as the traditional measures for lexical overlap and context similarity against the gold summary, while PAS score quantifies user preference compared to the baseline system, and LLM-ACU score assesses information coverage. We use DEBERTA-XLARGE-MNLI (He et al., 2021) for BERTScore to align with human preference. For LLM-based metrics, we employ `gpt-4o` as our universal evaluator due to its advanced capabilities.

Model Name	Version
gpt-4o	2024-08-06
gpt-4o-mini	2024-07-18
claude-3-sonnet	2024-06-20

Table 6: LLM Versions

C Positional Bias in Automatic Evaluation

In this section, we analyze the positional bias and consistency of two mainstream LLMs (`gpt-4o` and `claude-3.5-sonnet`).

Tables 7 and 8 demonstrate a clear positional bias in both models' evaluations, though in opposing directions. `gpt-4o` shows a strong preference for summaries presented in the first position, with notably higher win ratios across both datasets. Conversely, `claude-3.5-sonnet` exhibits a preference for summaries in the second position, though this bias is relatively less pronounced in the MultiNews dataset. This positional bias is further confirmed in Table 9, where the inconsistency ratios tell a similar story. The discrepancy percentages indicate that `claude-3.5-sonnet` generally achieves better consistency on MultiNews, though both models show comparable discrepancy rates on OpenASP. While claude demonstrates marginally better consistency metrics overall, we opted to use `gpt-4o` in our final implementation due to practical considerations regarding speed and computational budget constraints. Since our evaluation framework incorporates both consistency and preference metrics, the choice between these models does not significantly impact the validity of our methodology or results.

These findings suggest that positional bias is still an inherent challenge in current language models when performing comparative evaluations, regardless of the specific model architecture or training

approach. This observation underscores the importance of implementing appropriate debiasing strategies in evaluation frameworks.

Model	Dataset	Sum1 Win	Sum2 Win
GPT	MultiNews	456	92
Claude	MultiNews	262	336
GPT	OpenASP	355	177
Claude	OpenASP	186	401

Table 7: Model Preference Analysis - Number of wins when comparing summaries in order {Sum1, Sum2}.

Model	Dataset	Sum2 Win	Sum1 Win
GPT	MultiNews	468	86
Claude	MultiNews	285	308
GPT	OpenASP	384	174
Claude	OpenASP	188	396

Table 8: Model Preference Analysis - Number of wins when comparing summaries in order {Sum2, Sum1}.

Model/Dataset	Disc.(%)	Pref Pos	Inc. Ratio
GPT/MultiNews	56.00%	1	333:3
Claude/MultiNews	16.67%	2	27:73
GPT/OpenASP	30.03%	1	174:5
Claude/OpenASP	34.72%	2	6:217

Table 9: Model Consistency Analysis - Comparing discrepancy rates, positional bias, and inconsistency ratios between `gpt-4o` and `claude-3.5-sonnet`.

D Impact of Summary Length

In this section, we investigate the relationship between summary quality and length. Tables 10 and 11 present CAP scores, ROUGELSum scores, and the lengths of generated summaries.

For capable models like `gpt-4o`, we observe that despite improvements in CAP and ROUGELsum scores, summary length remains relatively stable. Notably, the highest-quality summaries are not necessarily the longest ones, demonstrating that sophisticated models can effectively distill core ideas into concise text.

In contrast, for less capable models like `gpt-4o-mini`, preferred and more complete summaries consistently tend to be longer, with summary length increasing proportionally with the number of ensembled samples. This suggests that smaller models may require more text to adequately capture information compared to their larger counterparts.

Moreover, previous work (Hu et al., 2024; Dubois et al., 2024) reveals LLM evaluation mechanisms tend to favor long summaries. This raises an important question: "do longer summaries actually contain more useful information?" To investigate this, we study the relationship between generation length and summary quality using the general-purpose MDS dataset MultiNews.

The results in Table 12 demonstrate how different configurations of our framework affect summary length and the associated computational costs. While the summary length increases substantially from baseline to our most comprehensive setting (from 129.4 to 201.17 words), the computational cost grows more slowly, suggesting efficient information packaging. The CPS aggregator consistently produces longer summaries than CFS, particularly with `gpt-4o-mini`, indicating its effectiveness in capturing diverse information from source documents without introducing excessive computational overhead.

E Prompts

E.1 Summarization Prompts

In Tables 13 and 14, we present the prompt bank used for the MultiNews dataset. Similarly, Tables 15 and 16 contain the prompt bank for the OpenASP dataset. These prompts were adapted and modified from the work of Lior et al. (2024). We utilized the same few-shot examples as provided in their benchmark.

E.2 Ensemble Prompts

We present our summary ensemble prompts for general purpose MDS (for datasets like MultiNews) in Table 17, and for aspect- (or query-) based MDS (for datasets like OpenASP) in Table 18.

MultiNews		CPS			CFS		
Model	**# Samples**	**CAP**	**RLsum**	**Gen_len**	**CAP**	**RLsum**	**Gen_len**
gpt-4o	2	0.82	18.96	155.53	0.69	18.40	138.82
	3	0.79	19.22	158.26	0.73	18.63	145.06
	4	0.82	19.27	161.86	0.68	18.75	147.26
	5	**0.85**	**19.61**	163.14	0.71	18.72	147.60
	6	0.81	19.57	163.58	0.79	18.71	158.25
gpt-4o-mini	2	0.80	20.92	184.98	0.61	18.89	150.58
	3	0.72	21.15	190.76	0.54	19.29	159.40
	4	**0.80**	21.37	196.85	0.60	19.60	165.36
	5	0.78	21.45	191.18	0.60	19.86	170.03
	6	0.77	**21.54**	201.07	0.60	20.21	172.44

Table 10: CAP scores, ROUGELsum scores, and generation lengths on MultiNews dataset for different models and ensemble sizes. The highest CAP and ROUGELsum scores are marked in **bold**.

OpenASP		CPS			CFS		
Model	**# Samples**	**CAP**	**RLsum**	**Gen_len**	**CAP**	**RLsum**	**Gen_len**
gpt-4o	2	0.70	17.51	198.27	0.63	16.49	167.58
	3	0.76	17.54	187.06	0.72	16.69	172.25
	4	0.74	17.86	191.66	0.72	16.89	173.33
	5	0.76	17.89	194.89	0.74	16.92	192.11
	6	**0.77**	**18.00**	194.59	0.74	17.01	178.73
gpt-4o-mini	2	0.79	19.56	196.67	0.73	17.79	234.36
	3	0.83	19.83	209.07	0.75	18.39	245.63
	4	0.83	19.93	216.40	0.77	18.65	251.47
	5	**0.86**	20.07	222.88	0.82	18.94	256.02
	6	0.85	**20.14**	224.05	0.81	18.87	257.55

Table 11: CAP scores, ROUGELsum scores, and generation lengths on OpenASP dataset for different models and ensemble sizes. The highest CAP and ROUGELsum scores are marked in **bold**.

Experiment	# Words	Word/ACU
Baseline	129.4	17.03
gpt-4o/CFS	147.61	18.42
gpt-4o/CPS	163.15	19.51
gpt-4o-mini/CFS	172.45	20.74
gpt-4o-mini/CPS	201.17	22.63

Table 12: Summary length and word cost per ACU across different model configurations on MultiNews dataset. Length shows the average number of words in generated summaries, while Cost measures the average number of words needed to capture each ACU.

No.	Prompt
1	In this task, you are presented with multiple news articles about related topics. Your job is to generate a summary that integrates information from the provided articles. Your summary should be short and concise, that includes content only from the provided articles, avoiding any external data sources.
2	Please provide a brief summary by synthesizing only the key points from the articles provided. Focus on the main arguments and conclusions without incorporating any information from outside these texts. Keep your summary concise and directly related to the content of the documents.
3	Generate a concise summary using only the information from the provided articles. Your summary should distill the most essential information, capturing the core insights without adding any external content. Aim for brevity and clarity in your summarization.
4	Please sift through the provided articles and distill their essence into a sharp, concise summary. Focus solely on the facts and key points within these texts, avoiding any embellishment or reference to external information. Your summary should read like a bullet-point list of the most critical insights.
5	You are presented with multiple news articles about related topics. Summarize the contents in a way that captures the key information in a narrative form, but strictly using the details mentioned in the provided documents. Keep it engaging yet brief.
6	Imagine you're preparing a brief for a decision-maker who has limited time. Summarize the provided documents by extracting only the most essential information. Present this in a clear, straightforward manner, focusing on the key facts and figures.
7	Using only the details from the articles I've given you, craft a summary that distills the most important information. Avoid any interpretations or external data, and keep your summary short and direct. Emphasize the main arguments, data points, and conclusions.
8	Operate as an information synthesizer: Draw the essence from multiple articles, focusing solely on the information contained within them. Your summary should be a tight, focused digest of the articles, free from any influence of external data.
9	Scan through the provided articles and compile a summary that highlights only the most significant facts and figures, ensuring the exclusion of all external references. Aim for clarity and brevity.
10	Operate as an academic summarizer: Imagine you are creating a summary for an academic review. Extract and emphasize the most pertinent information, ensuring your summary remains true to the original texts and free of external content.

Table 13: Summarization Prompt Bank for MultiNews Dataset (Part 1)

No.	Prompt
11	Condense the provided information into a compact summary that emphasizes the main points and crucial data from the documents. Exclude any external information to maintain the integrity of the sources.
12	From the provided articles, pull out the core messages and data points. Shape these into a brief, clear summary that directly reflects the content of the documents without any external additions.
13	Compile a concise summary from the news articles given, focusing only on the information contained within. Your summary should integrate the main points without adding any outside information.
14	Create a succinct summary by focusing exclusively on the details provided in the articles. Avoid using any external sources and ensure the summary remains clear and to the point.
15	Produce a brief summary that distills the essential facts from the provided articles. Keep your summary strictly to the content presented in the documents, avoiding external influences.
16	Develop a concise summary using only the information from the articles provided. Emphasize the main points and conclusions while avoiding the inclusion of any external data.
17	Prepare a short, integrated summary by synthesizing key points from the given news articles. Ensure that no external content is included and that the summary is clear and direct.
18	Your task is to distill the primary information from the provided articles into a concise summary. Make sure to exclude any external sources and focus strictly on the given texts.
19	Summarize the provided articles by extracting only the key information and conclusions. Your summary should be brief and must not incorporate any external data.
20	Generate a clear and brief summary using just the information from the provided articles. Focus on distilling the essential points and data without referencing external content.

Table 14: Summarization Prompt Bank for MultiNews Dataset (Part 2)

No.	Prompt
1	In this task you are required to generate an aspect-based summary of a set of documents related the same topic. Please write a short, concise aspect-based summary, only summarize content from the above documents, avoiding any external data sources.
2	Your goal is to create a short, concise aspect-based summary of the given documents. Summarize the key points accurately, using only the information from these documents and excluding any external sources.
3	Produce a brief, aspect-based summary of the collection of documents on the same topic. Ensure your summary is concise and derived only from the provided documents, avoiding any external data sources.
4	Your task is to generate a detailed yet concise aspect-based summary from a collection of documents that focus on the same topic. Begin by thoroughly examining each document to understand the main aspects and themes. Then, synthesize this information into a coherent summary that highlights the significant points.
5	Given a set of documents related to a specific topic, generate a short, concise aspect-based summary. Ensure that the summary is based solely on the content of the documents provided.
6	You will receive several documents on the same topic. Your task is to write a brief aspect-based summary, using only the information from the provided documents and excluding any external sources.
7	You are tasked with generating an aspect-based summary of several documents. Summarize the content briefly and accurately, using only the information from the documents give.
8	In this task, you are required to create an aspect-based summary of a set of documents all related to the same topic. Carefully read through each document and identify the key aspects discussed. Summarize these aspects in a concise manner, ensuring that your summary captures the essential points.
9	You are tasked with producing an aspect-based summary for a series of documents related to the same topic. Start by analyzing each document to identify the critical aspects covered. Your goal is to condense this information into a clear and concise summary.
10	Generate a concise aspect-based summary of the given documents. Focus on summarizing the content based solely on the information from these documents, avoiding any external sources.

Table 15: Summarization Prompt Bank for OpenASP Dataset (Part 1)

No.	Prompt
11	Create a concise aspect-based summary for the provided set of documents. Focus on the main aspects and themes discussed in these documents, ensuring that your summary is based entirely on the content of the provided documents.
12	Produce a short and precise aspect-based summary of the given documents. Identify the key aspects discussed in these documents and synthesize a concise summary based solely on the provided content.
13	You will receive a collection of documents focused on the same topic. Your task is to create an aspect-based summary that highlights the key aspects discussed in these documents. Ensure your summary is brief and does not include any external information.
14	You are provided with multiple documents related to a single topic. Your task is to generate an aspect-based summary that captures the main aspects discussed in these documents. Ensure your summary is concise and solely based on the provided texts.
15	You are tasked with generating an aspect-based summary of several documents on the same topic. Carefully review each document, identify the main aspects, and write a brief summary that captures these aspects using only the provided documents.
16	Your role is to create an educational summary for students using a collection of documents on the same topic. Focus on the main aspects that would help students understand the core concepts discussed in the documents.
17	Imagine you are preparing a briefing for a busy executive who needs to understand the key aspects of several documents quickly. Summarize the most important points from these documents in a concise manner.
18	As an advanced AI tasked with summarizing documents, your goal is to generate an aspect-based summary. Think of yourself as a summarization expert, extracting the most critical aspects from the documents provided.
19	Imagine you are a journalist tasked with writing a summary article based on a series of documents related to a single topic. Identify the key aspects discussed in these documents and compose a brief, coherent summary.
20	Your task is to act as a knowledge distiller, creating a concise aspect-based summary from a series of documents on the same topic. Focus on identifying and summarizing the critical aspects discussed in these documents.
21	You are an AI assistant tasked with providing a summary for a set of documents related to a specific topic. Focus on the key aspects and themes discussed in these documents. Create a summary that captures these aspects in a concise manner, ensuring that your summary is based solely on the provided documents and excludes any external information.

Table 16: Summarization Prompt Bank for OpenASP Dataset (Part 2)

Ensemble Type	Content
Vote	Provide your explanation, then select the best summary of the given documents based on clarity, accuracy, conciseness, and completeness. Documents: {doc} Summary 1: {sum1} Summary 2: {sum2} ... Explanation: "Your explanation here" Decision: [1-5]
CIS	Take all provided summaries into account and generate a better, cohesive summary. Combine and refine the content from the summaries to ensure clarity, accuracy, conciseness, and completeness. Provide the final summary directly. Summary 1: {sum1} Summary 2: {sum2} ... Final revised summary:
CPS	Take all provided summaries into account and generate a better, cohesive summary of the given documents. Combine and refine the content from the summaries to ensure clarity, accuracy, conciseness, and completeness. Provide the final summary directly. Documents: {doc} Summary 1: {sum1} Summary 2: {sum2} ... Final revised summary:

Table 17: Ensemble Prompts for General MDS

Ensemble Type	Content
Vote	Provide your explanation, then select the best summary of the given documents based on clarity, accuracy, conciseness, and completeness, focusing on the specified aspect. Example Response: Explanation: "Your explanation here" Decision: 1 (or 2 or 3 or 4 or 5) Aspect: {query} Documents: {doc} Summary 1: {sum1} Summary 2: {sum2} ... Response:
CIS	Take all provided summaries into account and generate a better, cohesive summary, focusing on the specified aspect. Combine and refine the content from the summaries to ensure clarity, accuracy, conciseness, and completeness. Provide the final summary directly. Aspect: {query} Summary 1: {sum1} Summary 2: {sum2} ... Final revised summary:
CPS	Take all provided summaries into account and generate a better, cohesive summary of the given documents, focusing on the specified aspect. Combine and refine the content from the summaries to ensure clarity, accuracy, conciseness, and completeness. Provide the final summary directly. Aspect: {query} Documents: {doc} Summary 1: {sum1} Summary 2: {sum2} ... Final revised summary:

Table 18: Ensemble Prompts for Aspect-based MDS