H-QFS: Hybrid Query-Focused Scientific Document Summarization using Multitask Learning

Anonymous ACL submission

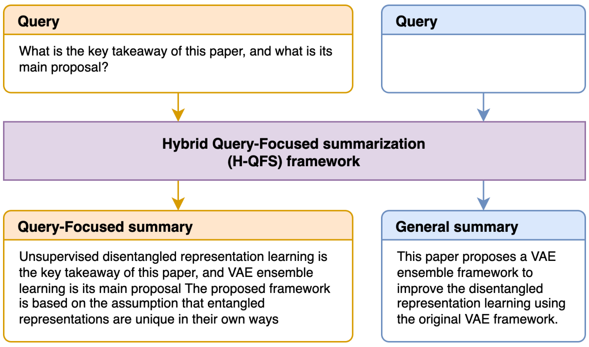
Abstract

Scientific document summarization, typically focused on a single gold summary per paper, often overlooks the diverse perspectives inherent in scholarly content. In response to this limitation, the Multi-Perspective Scientific Document Summarization (MuP) dataset was introduced, providing multiple summaries for each paper. However, no existing work implemented Query-Focused Summarization (QFS), which can specifically generate a summary according to the diverse perspectives of user requirements. To address this gap, our study introduced the Hybrid Query-Focused Summarization (H-QFS) framework which are proficient in both the QFS and the General Summarization (GS) tasks. Given the absence of queries in the MuP dataset, a query-less resource QFS strategy was applied to our framework, using proxy queries generated from summary masking. Furthermore, guided by the intuition that QFS can focus on specific summaries while GS can capture the global information of the entire document, we employed multitask learning of QFS and GS tasks in our H-QFS. This approach aimed to enhance QFS performance while maintaining the GS capacity to summarize the overall content comprehensively. Experimental results showed that the H-QFS framework outperformed existing works in the QFS task, achieving state-of-the-art performance in synthetic-query validation sets. Furthermore, our framework maintained competitive GS performance, showcasing versatility across scenarios. Our contributions include: (1) among the first to propose a framework of QFS for scientific document summarization, (2) proposing and investigating the effectiveness of multitask learning to enhance QFS, and (3) outperforming baselines in the QFS tasks and maintaining competitive performances in the GS task.

1 Introduction

Scientific document summarization, traditionally based on datasets featuring a single gold summary per paper, inadvertently overlooks the inherent richness of diverse perspectives within scientific document. Scholarly content inherently presents multifaceted viewpoints, methodologies, and interpretations. In response to this limitation, Cohan et al. (2022) introduced the Multi-Perspective Scientific Document Summarization (MuP) dataset, aiming to bridge this gap by providing multiple summaries for each scientific paper.

However, previous research endeavors (Kumar et al., 2022; Sotudeh and Goharian, 2022; Urlana et al., 2022; Akkasi, 2022) treated the multi-perspective aspect merely as paper-summary pairs, neglecting to delve into the specificity of these perspectives. This oversight may result in a loss of the unique capability to effectively summarize content from various perspectives.



What is the key takeaway of this paper, and what is its main proposal?

Unsupervised disentangled representation learning is the key takeaway of this paper, and VAE ensemble learning is its main proposal The proposed framework is based on the assumption that entangled representations are unique in their own ways

This paper proposes a VAE ensemble framework to improve the disentangled representation learning using the original VAE framework.

Figure 1: Hybrid Query-focused Summarization (H-QFS) framework: A multitask learning approach capable of generating tailored summaries based on specific user queries and general summaries in their absence.

To bridge this gap, we presented the Hybrid Query-Focused Summarization (H-QFS) framework, a novel approach capable of generating tailored summaries based on specific user queries and general summaries in their absence, as depicted in Figure 1. In the absence of queries in the MuP dataset, we adopted the Mask ROUGE Regression framework (MERGE) (Xu and Lapata, 2020), a query-less resource QFS framework utilizing Unified Masked Representation (UMR) as a proxy query derived from summary masking. Moreover, guided by the idea that QFS and GS can mutually benefit each other, i.e., QFS tasks tend to focus more on the granular perspective while GS focus more on the global information of the entire document, led us to employ multitask learning for both QFS and GS tasks in our H-QFS framework. This strategy aimed to enhance QFS performance while preserving the GS capacity to comprehensively summarize the overall content.

Experimental results showed that our H-QFS framework effectively outperformed existing works in the QFS task, achieving state-of-the-art performance in ROUGE scores in the synthetic-query validation set, an synthesized query to MuP dataset for QFS task validation. Simultaneously, it maintained GS performance, achieving state-of-the-art results in the validation set, and securing the 2nd place in the blind test set.

Our contributions are threefold: (1) we propose a novel framework for QFS in scientific document summarization, (2) propose and investigate the effectiveness of multitask learning to enhance QFS performance named H-QFS framework, and (3) outperform baseline models in the QFS tasks, while maintaining GS capacity to summarize the overall content.

2 Related Work

This section reviews three domains: (1) scientific document summarization, (2) Multi-Perspective Scientific Document Summarization (MuP), and (3) Query-Focused Summarization (QFS). In the following sections, we delved into each domain, emphasizing their contributions and highlighting evolving trends in text summarization research.

2.1 Scientific document summarization

Scientific documents, distinguished by specific characteristics like well-structured hierarchies and domain-specific knowledge requirements, often pose challenges for summarization. Existing studies (Cohan et al., 2018; Xiao and Carenini, 2019; Gidiotis and Tsoumakas, 2020; Cui et al., 2020; Grail et al., 2021) predominantly use arXiv and PubMed datasets for training, relying on abstracts as gold summaries. However, the suitability of abstracts may not align with summarization goals. The LongSumm dataset (Chandrasekaran et al., 2020), introduced in the Scholarly Document Processing (SDP) 2020 shared tasks, addresses the need for longer, in-depth summaries. It includes both extractive and abstractive summaries derived from video recordings and blog posts, respectively. Numerous studies (Li et al., 2020; Sotudeh et al., 2020; Gidiotis et al., 2020; Kaushik et al., 2021; Roy et al., 2021; Ying et al., 2021) participated in this shared task, employing diverse approaches to tackle challenges in scientific document summarization. In abstractive summarization, researchers often use extractive models to highlight key information, especially when dealing with lengthy source documents. However, a notable drawback in these studies is the application of models generating only one summary per source document, potentially overlooking diverse perspectives within the summaries.

2.2 Multi-Perspective Summarization (MuP)

Multi-Perspective Scientific Document Summarization (MuP) debuted in the Scholarly Document Processing (SDP) share task 2022 (Cohan et al., 2022), introducing a novel challenge with multiple gold summaries for each document. Four participating studies included Kumar et al. (2022), employing an extractive-abstractive approach with section identification; Sotudeh and Goharian (2022), using a two-step LED process for salient sentence extraction; Urlana et al. (2022), exploring self-pretrained models with BART performing the best; and Akkasi (2022), applying graph attention networks (GATs) without an abstractive phase. Evaluation results showed Kumar et al. (2022) outperforming in ROUGE-2 and ROUGE-L, while Sotudeh and Goharian (2022) and Urlana et al. (2022) demonstrated competitive performances. Besides automatic evaluations, the MuP share task conducted human evaluations on faithfulness, readability, and coverage. Kumar et al. (2022) scored highest in readability and coverage, closely trailing GATs in faithfulness. However, organizers noted studies treating MuP as a general summarization task, potentially overlooking its unique capability to summarize content from various perspectives, i.e., the original intended objective of the MuP dataset.

2.3 Query-Focused Summarization (QFS)

Query-Focused Summarization (QFS) created concise summaries from a document corpus, conditioned on predefined queries or user-specified criteria. Earlier QFS frameworks (Nema et al., 2017;

Baumel et al., 2018; Laskar et al., 2020; Su et al., 2021) predominantly employed a supervised approach, requiring explicit queries in the document-query-summary triplet.

Addressing the limitations of explicit queries during training, Xu and Lapata (2020) proposed the Masked ROUGE Regression framework (MERGE), shifting towards a query-less architecture. This innovative approach transformed generic dataset summaries into proxy queries, forming the Unified Masked Representation (UMR). Drawing inspiration from Fan et al. (2017), Xu and Lapata (2020) discretized summary length into discrete bins and prepend these lengths along with other inputs. Results highlighted the model's ability to learn effectively during training and perform well with user queries during inference.

Inspired by Xu and Lapata (2020)'s work, our study leveraged the query-less advantage of the MERGE framework. In addition, we applied the concept of multitask learning to further enhance the QFS tasks. As a positive side effect, our framework permits usage with or without explicit queries.

3 H-QFS: Hybrid Query-focused summarization framework

We introduced the Hybrid Query-Focused Summarization (H-QFS) framework, a fusion of QFS and GS frameworks. Specifically, H-QFS learns from both QFS & GS in a mulit-task learning manner. We chose Multi-perspective scientific document dataset (MuP) as it provides summaries of multiple perspectives. Specifically, MuP dataset $\{(D, S)\}$ comprises a document D, a set of summaries $S = \{s_1, s_2, ..., s_{|S|}\}$. Due to the absense of specific query, the QFS framework incorporated the Unified Masked Representation (UMR) as a pivotal component for query guidance, and GS framework, replace UMR by a mask token in order to integrate with the QFS framework. H-QFS excels in both QFS and GS scenarios. When presented with an input featuring UMR, the framework is geared towards generating a query-focused summary, while substituting UMR with a mask token enables it to produce a general summary, showcasing its adaptability across a spectrum of summarization tasks.

In particular, the H-QFS framework comprised two main modules: (1) a Ranking module for Extractive summarization and (2) a Summary Generator module for Abstractive summarization, as illustrated in Figure 2. This section provided a detailed exploration of UMR followed by an in-depth discussion of the Ranking module and the Summary Generator module.

3.1 Unified Masked Representation (UMR)

The Unified Masked Representation (UMR), served as a powerful tool for generating query-focused summaries from query-free resources. During the training phase, UMR was derived from the entities-masked summary (UMR$_S$), forming a foundational understanding of salient information a user needs. In the inference phase, UMR was shaped by masking question-words from the user query (UMR$_Q$), aligning the summarization process with the user's query. In the context of our work on scientific document summarization, Packed Levitated Marker (PL-Marker) (Ye et al., 2021), a state-of-the-art entity-relation extraction model for scientific documents, was leveraged. (See Appendix A for more detail)

3.2 Ranking module

The ranking module, also referred to as the extractive summarization module, was tasked with sentence scoring and ranking to identify the most salient sentences in the source paper related to the query. Each paper was segmented to candidate sentences using spaCy (Honnibal et al., 2020). To predict the relevant scores of those candidate sentences, a BERT regression model, characterized by a BERT classification model with a single neuron in the output classification layer, was employed. Specifically, the model was tasked to minimize the MES loss. By giving each candidate sentence together with UMR or mask token, the ranked of sentences was guided by a target score (y), calculated based on ROUGE score (Lin, 2004) metrics.

In the **QFS task**, the target score (y) was calculated from pair of candidate sentence and reference summary, specifically $y = R_2 + \lambda * R_1$, where R_1 and R_2 represented ROUGE-1 and ROUGE-2 F1 scores, respectively, and λ was set to 0.15, following the optimization approach of Xu and Lapata (2020). In the **GS task**, our hypothesis suggested that sentences consistently relevant across multiple summaries should be prioritized. To implement this, we employed a scoring strategy in which the candidate sentence's score (y) was averaged with the scores derived from multiple summaries.

Specific for QFS, the input sample (x) was constructed by concatenating the UMR with each can-

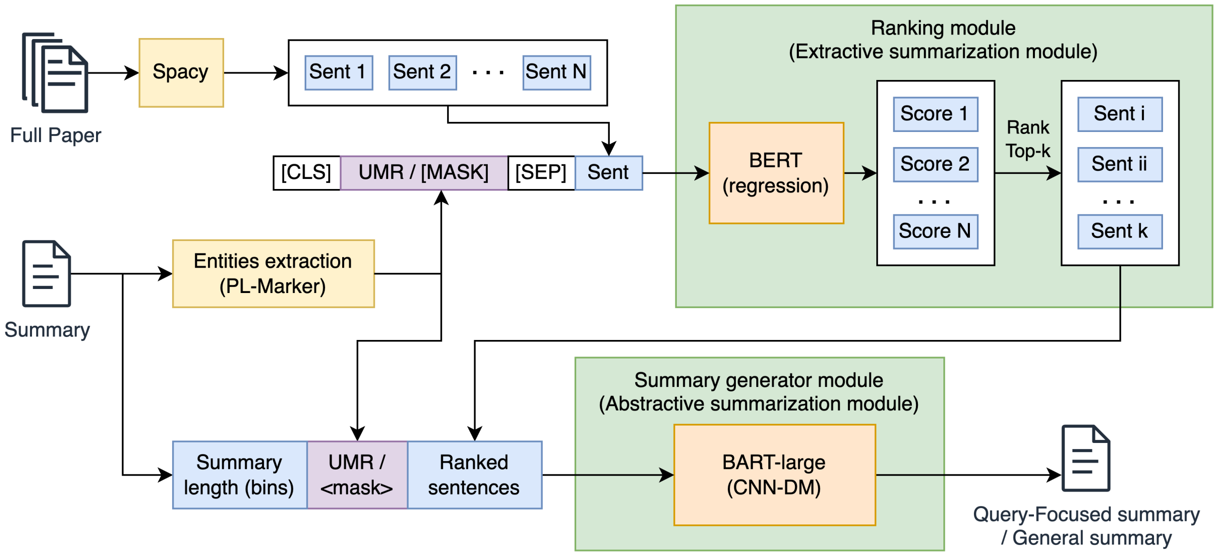


Figure 2: Overview of the H-QFS framework consists of two main modules: (1) Ranking module and (2) Summary Generator module. Each module was guided by UMR in QFS scenarios and a mask token (MASK) in GS scenarios.

didate paper sentence. This concatenation began with a [CLS] token (classification) and was separated by [SEP] tokens (separator). Since the absence of a query in GS task, the input samples (x) of both tasks were added the section and position detail sentence (s_i). The section title of the sentence, denoted by t_n, was prepend and separated by [SEP] tokens. Additionally, to indicate the number of sentence sections, the count of [SEP] tokens was increased by the number of sections (n) to which the sentence belonged. The final input sequence was represented as:

$$\mathrm{x}_i = \begin{cases} \text{concat}([\text{CLS}], \text{UMR}, [\text{SEP}], t_n, n * [\text{SEP}], s_i) & \text{if QFS case} \\ \text{concat}([\text{CLS}], [\text{MASK}], [\text{SEP}], t_n, n * [\text{SEP}], s_i) & \text{else GS case} \end{cases} \tag{1}$$

Furthermore, in the context of score visualization ($y \in [0, 1.15]$), the 90th percentile of the training dataset revealed a score of 0.0475, signifying that a substantial portion of candidate sentences were considered irrelevant. This observation prompted concerns about potential overemphasis on less relevant sentences if the model were trained on the complete dataset. To address this issue, we introduced a **low score sampling technique**, randomly excluding samples with scores below 0.05—specifically, 90 percent of such low-score samples. This approach not only ensured a more balanced distribution of scores for effective model training but also optimized computational efficiency, thereby reducing overall computation time. Further details and the score distribution can be found in Appendix B.

3.3 Summary generator module

After extracting the most relevant sentences through the Ranking Module, the Summary Generator Module, or the Abstractive Summarization Module leverages these pertinent sentences to generate summaries. Illustrated in Figure 2, this module relied on three primary input components: (1) a summary length bin token for summary length control, (2) UMR for query-focused guidance in the QFS task or a mask token for general summarization, and (3) the ranked sentences and important sections from the source paper.

Following the **summary length control** strategy of Xu and Lapata (2020), inspired from Guo et al. (2016), our system employed discrete length bins. The summary length, computed using the BART tokenizer, was discretized into 10 bins with a range of 30 words (tokens) per bin, except for the last length bin, which aggregated all samples with a summary length exceeding 270 tokens.

4 Experimental results and Analysis

4.1 Dataset

In our research, we primarily utilized the Multi-Perspective Scientific Document Summarization (MuP) dataset. The MuP dataset, created for the MuP shared task at SDP 2022, encompasses 1-10 summaries per scientific paper, with 8379, 1060, and 1052 papers in the train, validation, and blind test sets, respectively. The total summaries include 18934, 3604, and 4611 in the respective sets.

For validation in the QFS task, we augmented our dataset with a synthetic query dataset generated from Llama2, a Large Language Model (LLM) (Touvron et al., 2023). The **synthetic-query validation dataset** was created using a 2-shot prompting strategy, incorporating the abstract, summary, and a manually crafted query for each example. This method not only introduced specificity into the summarization process but also enhanced the validation for the QFS task. The resulting synthetic query dataset exhibited an average of 111 words per query and 5.62 sentences per query, offering a rich and diverse set of queries for validation. (Refer to the Appendix for detailed examples of the 2-shot prompting strategy.)

4.2 Implementation details

The implementation utilized a single A6000 GPU for both the ranking and summary generator modules. The **ranking modules** in all frameworks employed a BERT-base-uncased model (Devlin et al., 2018), accessible on Hugging Face [1]. The models underwent fine-tuning with a learning rate of 3×10^{-5} and a batch size of 64 for 3 epochs. Models were saved according to the highest Kendall's Tau ranking (Kendall, 1938). The **summary generator modules** in all frameworks utilized a BART-Large-CNN model, a pretrained BART model (Lewis et al., 2019) previously fine-tuned with CNN on the CNN Daily Mail summarization dataset (Hermann et al., 2015). The model is publicly available on Hugging Face [2]. The models underwent fine-tuning with a dynamic learning rate approach, maximum 5×10^{-5}, 1000 warm-up steps, gradient accumulation every 10 steps, and a batch size of 4. In single-task modules, such as those in the QFS and GS frameworks, the models were fine-tuned for 5 epochs. In contrast, for the summary generator module of the H-QFS framework, operating in a multi-tasking environment, the dataset was doubled, necessitating fine-tuning for 3 epochs. Models were saved according to the highest ROUGE-1 score (Lin, 2004).

4.3 Ranking module performance

In the Ranking Module, the primary objective is to predict scores for candidate sentences, where a higher score indicates the suitability of the candidate sentence for inclusion in the input sequence of the Summary Generator Module. The evaluation process involved two key aspects. First, the correlation between target scores and predicted scores was meticulously examined to assess the model's ability to accurately assign scores. This analysis provided insights into the consistency and reliability of the ranking module. Second, the effectiveness of the module in retrieving relevant sentences was evaluated using the ROUGE-2 Recall score against the gold summary. This metric gauged the module's performance in capturing essential information from the source document, emphasizing its role in selecting sentences with a higher degree of relevance for subsequent summarization.

Framework	Pearson's	Spearman's	Kendall's
QFS (UMR$_S$)	0.868*	0.917**	0.766**
QFS (UMR$_Q$)	0.523∼	0.788*	0.602*
GS	0.790*	0.880*	0.705*
H-QFS			
(QFS task)	0.615∼	0.758*	0.577*
(GS task)	0.792*	0.881*	0.706*
(multi-task)	0.648∼	0.792*	0.609*

Table 1: The score correlation performance, utilizing Pearson's, Spearman's, and Kendall's Tau ranking correlations. The symbols '∼', '*' and '**' represent the interpretation that predicted scores have **moderate**, **strong** and **very strong** correlations, respectively, with reference scores

Score correlation performance In this study, we utilized three well-established correlation measures—Pearson's (Benesty et al., 2009), Spearman's (Spearman, 1961), and Kendall's Tau ranking (Kendall, 1938), to evaluate the score correlation performance. The results, presented in Table 1. For the QFS framework, we conducted validation using (UMR$_S$) and (UMR$_Q$). When utilizing (UMR$_S$), the ranking module exhibited a strong correlation in Pearson's, and notably, very strong correlations in both Spearman's and Kendall's Tau ranking. However, when using (UMR$_Q$), the scoring performance witnessed a significant drop to a moderate correlation in Pearson's, while maintaining strong correlations in both Spearman's and Kendall's Tau ranking. This outcome suggested that the ranking module of the QFS framework exhibits limited adaptability to real queries. Conversely, the GS framework displayed strong correlations across all three measures, affirming the proficiency of the ranking modules in accurately predicting scores compared to the reference scores. In the H-QFS framework, the correlation perfor-

[1] https://huggingface.co/bert-base-uncased

[2] https://huggingface.co/facebook/bart-large-cnn

	R2 recall			
Framework	k=10	k=20	k=30	truncated
Baseline	-	-	-	16.48
QFS (UMR$_S$)	18.84	23.06	25.45	24.59
QFS (UMR$_Q$)	10.74	15.06	17.89	17.01
GS	12.76	17.20	20.00	18.73
H-QFS				
(QFS task)	12.61	16.99	19.85	19.40
(GS task)	12.80	17.26	20.01	18.92
((multi-task)	12.70	17.13	19.93	19.16

Table 2: Retrieval performance represented by ROUGE-2 recall

mance improved in both tasks, underscoring the positive complementarity of multi-task learning for identifying relevant sentences.

Retrieval performance To evaluate the retrieval performance of the Ranking Module, ROUGE-2 (Recall) scores were calculated against the gold summary. The top-k sentences extracted by the ranking module underwent reordering to their original sequence in the paper. Subsequently, these reordered sentences were concatenated to create a candidate sequence, and the ROUGE-2 recall was calculated between the candidate sequence and the reference sequence, which served as the summary. The results aligned with score correlation trends. The QFS framework, using (UMR$_S$), demonstrated notable retrieval performance with ROUGE-2 Recall surpassing the baseline when k=10. However, a significant drop occurred when transitioning to (UMR$_Q$), although it still outperformed the truncated baseline, where all paper contents were truncated to 1024 tokens, when k=30 as presented in Table 2.

Furthermore, we hypothesized that, without order correction, the BART model can handle and generate a specific summary from the most relevant sentence based on the specific query or the most salient sentence for a general summary. Moreover, with an increased value of k and reordering, some relevant sentences may be moved to the last position and consequently cut out in the truncation process due to the maximum input token limit of the Pre-trained Language Model (PLM). Therefore, an alternative exploration involved considering all ranked sentences without reordering them to the original sequence in the paper. In this scenario, all ranked sentences were concatenated, and the resulting sequence was truncated to 1024 tokens, adhering to the maximum input token limit of the BART-Large model.

The results showed that truncated sentences from all frameworks outperformed the baseline. This finding suggested that the extracted sentences from our ranking module contain more relevant information for the summary compared to the baseline, potentially leading to better results in the summary generator module.

4.4 Summary generator module performance

Within the Summary Generator Module, we conducted a comprehensive experiment to assess the impact of various factors on summarization performance. Specifically, we explored the impact of the top-k sentences, the influence of paper sections, and the module's ability to control the length of generated summaries. The evaluation tasks were bifurcated into two categories: the QFS task and GS task. For the QFS task, the evaluation involved using the generated queries for the Llama 2 model on the validation set. In contrast, the GS task was evaluated without any explicit queries, encompassing both the validation set and the blind test set.

Impact of top-k sentences For this experiment, we systematically varied the values of k from 4 to 30 to investigate the impact of top-k sentences selection on the overall performance of our framework. The results shown that the optimal value for k is determined to be k=10. (for more top-k analysis, please see Appendix D)

QFS task performance In the QFS task, the evaluation involved utilizing the generated queries for the Llama 2 model on the validation set. We systematically explored different combinations of paper sections combined with ranked sentences, as detailed in Table 3. Specifically, the ALiR setting followed the experiment conducted by Kumar et al. (2022). Furthermore, we explored two types of ranked sentences: (1) k=10, representing a sequence of top-k sentences where k=10, determined as the optimal value. These sentences were reordered to align with the original paper order. (2) Ranked, representing a sequence of all ranked sentences without original reordering. This dual exploration aimed to ensure an understanding of the performance implications. The QFS framework demonstrated performance in the AR scenario for both k=10 and ranked types. Notably, the ranked type showed a slight improvement in all ROUGE-1, 2, L, and average scores compared to the k=10 type. On the other hand, H-QFS excelled in the R

Framework	R-1	R-2	R-L	R-avg
Baseline	39.48	11.86	24.25	25.05
GATS (Akkasi, 2022)	35.46	9.53	19.63	21.54
GUIR (Sotudeh and Goharian, 2022)	41.05	12.18	24.61	25.95
QFS framework				
R (k=10)	43.82	17.50	27.40	29.57
R (ranked)	44.09	17.73	27.49	29.77
AR (k=10)	44.17	17.80	27.58	29.85
AR (ranked)	44.29	**17.88**	27.67	29.95
AIR (k=10)	44.05	17.72	27.52	29.76
AIR (ranked)	44.18	17.79	27.55	29.84
ALiR (k=10)	43.88	17.51	27.30	29.56
ALiR (ranked)	44.11	17.67	27.47	29.75
H-QFS				
R (k=10)	**44.83**	17.34	27.63	29.93
R (ranked)	44.76	17.57	**27.85**	**30.06**
AR (k=10)	44.19	17.14	27.43	29.59
AR (ranked)	44.29	17.11	27.50	29.63
AIR (k=10)	44.20	17.04	27.57	29.61
AIR (ranked)	44.60	17.71	27.76	30.02
ALiR (k=10)	44.48	17.13	27.47	29.69
ALiR (ranked)	44.27	17.06	27.58	29.64

Table 3: Summary generator performance in query-focused summarization (QFS) task; **Bold**: 1st place, underline: 2nd place. The section abbreviation indicates that the summary generator utilized (1) R: ranked sentences from the ranking module, (2) AR: ranked sentences prepended with the Abstract section, (3) AIR: concatenation of Abstract, Introduction, and Ranked sentences, and (4) ALiR: concatenation of Abstract, Last 5 sentences of the Introduction, and Ranked sentences.

	Validation set		Blind test set	
Framework	R-1	R-avg	R-1	R-avg
Baseline	39.48	25.05	40.80	25.87
GATS (Akkasi, 2022)	35.46	21.54	33.85	19.66
LTRC (Urlana et al., 2022)	-	-	40.68	26.05
GUIR (Sotudeh and Goharian, 2022)	41.05	25.95	**41.36**	26.24
AINLPML (Kumar et al., 2022)	-	-	41.08	**26.58**
GS framework				
R (k=10)	39.57	25.17	40.65	25.82
R (ranked)	39.56	25.08	40.52	25.69
AR (k=10)	39.65	25.16	40.57	25.77
AR (ranked)	39.71	25.28	40.58	25.89
AIR (k=10)	39.36	24.95	40.22	25.51
AIR (ranked)	39.60	25.08	40.42	25.61
ALiR (k=10)	39.02	24.83	40.88	25.87
ALiR (ranked)	39.13	24.92	40.80	25.90
H-QFS framework				
R (k=10)	**41.63**	26.00	40.16	25.08
R (ranked)	41.39	**26.02**	41.14	25.86
AR (k=10)	41.14	25.80	40.79	25.83
AR (ranked)	40.80	25.57	40.82	25.77
AIR (k=10)	40.81	25.78	40.28	25.11
AIR (ranked)	40.41	25.43	40.91	25.80
ALiR (k=10)	41.46	25.93	40.93	25.85
ALiR (ranked)	41.11	25.82	40.81	25.84

Table 4: Summary generator performance in general summarization (GS) task; **Bold**: 1st place, underline: 2nd place. The section abbreviation indicates that the summary generator utilized (1) R: ranked sentences from the ranking module, (2) AR: ranked sentences prepended with the Abstract section, (3) AIR: concatenation of Abstract, Introduction, and Ranked sentences, and (4) ALiR: concatenation of Abstract, Last 5 sentences of the Introduction, and Ranked sentences.

scenario, especially in the ranked type, which showcased a state-of-the-art performance in ROUGE-L and ROUGE-average. Moreover, in both QFS and H-QFS frameworks for the QFS task, all experiments significantly outperformed existing works on the validation set. This underscores the efficiency of our framework in generating query-specific summaries, reaffirming its potential in advancing the field.

GS task performance Similar to the QFS task, an experiment in the GS task also explored the impact of paper sections and ranked sentence types. In both the validation set and the blind test set, the GS framework demonstrated competitive performance with the baseline. However, the performance was observed to be lower than that of previous works. In contrast, the Hybrid Query-Focused Summarization (H-QFS) framework, which, in the GS scenario, is similar to the GS framework but was multi-task trained together with the QFS task, outperformed previous works in the validation set and competitively performed in the blind test set, securing the second place in ROUGE-1 score, as shown in Table 4. It's noteworthy that, in the validation set, there was summary length control according to the summary, while in the blind test set, the '<len04>' bin was utilized based on our length bin variation for the blind test set.

Furthermore, through multi-task learning, performance of H-QFS framework in QFS and GS tasks was improved from QFS and GS framework, respectively. As per our hypothesis, the result showed that the global capturing of the GS task has a positive impact on the performance in QFS task. The H-QFS framework not only excels in handling the QFS task when users want the summary to focus on their query but also maintains performance of GS task when users do not have any query and desire only a general summary.

Summary length control By prepending each length token to the input of the blind test set, we examined the impact of length tokens on the generated summary lengths, as illustrated in Fig. 3. The generated summary lengths exhibit a consistent increase corresponding to the length token, except for the <len01> token. This anomaly is attributed to the scarcity of samples in the <len01> category in the training set. Consequently, the model struggled to learn the appropriate summary length to

	ROUGE avg	
	GS task	QFS task
QFS (AR-ranked)	-	29.95
-Len	-	29.7 (-0.25)
-UMR	-	25.54 (-4.41)
-UMR -Len	-	24.56 (-5.39)
H-QFS (R-ranked)	26.02	30.06
-Len	24.76 (-1.26)	29.46 (-0.6)
-UMR	26.31 (0.29)	27.34 (-2.72)
-UMR -Len	24.72 (-1.30)	25.78 (-4.28)

Table 5: Ablation study for QFS and H-QFS frameworks in both GS and QFS task. The value in parenthesis is the different score between main framework and ablation experiments

generate within this token category. This nuanced observation underscores the importance of an adequately diverse training set to ensure the model's proficiency in adapting to various length specifications during the summarization process. To address this issue, we proposed a solution: combining the <len01> and <len02> tokens together. The summary length for this combined bin can be adjusted to span from 1 to 60 words (tokens). This strategic modification aims to enhance the model's learning and adaptability, particularly in scenarios with limited training data, ensuring more robust performance across a wider spectrum of summary length requirements.

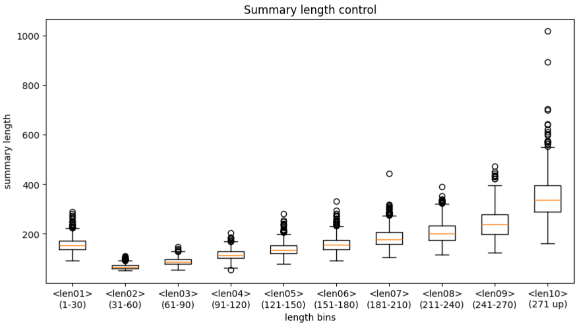


Figure 3: Generated summary length from each length bin

4.5 Ablation study

We conducted an extensive ablation study to assess the importance of each input element in the QFS and H-QFS frameworks for both GS and QFS tasks. Specifically, we removed length control (-Len), UMR (-UMR), and both length control and UMR (-UMR -Len) from the best-performing experiments of the QFS and H-QFS frameworks, which were AR (ranked) and R (ranked), respectively as shown in Table 5.

The results reveal that the QFS framework predominantly relies on UMR guidance, with a notable 4.41-point drop in ROUGE average when UMR is removed. Conversely, the performance drop is only 0.25 when length control is removed. However, when comparing '-UMR' and '-UMR -Len', the performance drop is 0.98. This indicates that without UMR, the model leans more heavily on length control.

For the H-QFS framework in the QFS task, removing UMR results in a performance drop of only 2.72. This suggests that multi-task training with the GS task enhances the robustness of QFS. Even with the '-UMR -Len' configuration dropping 1.56 points from '-UMR', the H-QFS framework continues to perform well compared to the QFS framework.

In the general task of the H-QFS framework, the model predominantly relies on length control because, in the GS task, the UMR of H-QFS is represented by a <mask> token. Interestingly, when UMR is removed in both tasks, it appears that the input data of the QFS task without UMR is quite similar to the input data of the GS task, except for the ranked sentence. This increase in the number of samples in the GS task during the training phase may contribute to improved performance in the GS task when UMR is removed from the H-QFS framework.

5 Conclusion

This study pioneers the application of QFS in Multi-Perspective Scientific Summarization. We introduced the Hybrid Query-Focused Summarization (H-QFS) framework, proficient in generating both query-focused and general summaries. Leveraging this multi-task approach, our framework outperformed existing works in the QFS task, achieving state-of-the-art performance in synthetic-query validation sets. Furthermore, H-QFS maintained strong GS performance, securing the 2nd place in ROUGE 2 and L in validation set and 2nd place in ROUGE 1 in blind test set.

Limitations

One limitation of our study is that the validation for the Query-Focused Summarization (QFS) task relied on a synthetic dataset. This choice was necessitated by the absence of a dedicated query-focused summarization dataset within the scientific document domain. It's important to note that the synthetic dataset used in our validation may potentially overemphasize information derived from the summary, introducing a constraint in the generalizability of our findings.

Acknowledgements

We extend our sincere appreciation to all the dedicated reviewers for their invaluable time, expertise, and constructive feedback. Their thoughtful contributions have greatly enriched the quality of our paper.

References

Abbas Akkasi. 2022. Multi perspective scientific document summarization with graph attention networks (gats). In *Proceedings of the Third Workshop on Scholarly Document Processing*, pages 268–272.

Tal Baumel, Matan Eyal, and Michael Elhadad. 2018. Query focused abstractive summarization: Incorporating query relevance, multi-document coverage, and summary length constraints into seq2seq models. *arXiv preprint arXiv:1801.07704*.

Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. 2009. Pearson correlation coefficient. In *Noise reduction in speech processing*, pages 37–40. Springer.

Muthu Kumar Chandrasekaran, Guy Feigenblat, Eduard Hovy, Abhilasha Ravichander, Michal Shmueli-Scheuer, and Anita de Waard. 2020. Overview and insights from the shared tasks at scholarly document processing 2020: CL-SciSumm, LaySumm and LongSumm. In *Proceedings of the First Workshop on Scholarly Document Processing*, pages 214–224, Online. Association for Computational Linguistics.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. *arXiv preprint arXiv:1804.05685*.

Arman Cohan, Guy Feigenblat, Tirthankar Ghosal, and Michal Shmueli-Scheuer. 2022. Overview of the first shared task on multi perspective scientific document summarization (mup). In *Proceedings of the Third Workshop on Scholarly Document Processing*, pages 263–267.

Peng Cui, Le Hu, and Yuanchao Liu. 2020. Enhancing extractive text summarization with topic-aware graph neural networks. *arXiv preprint arXiv:2010.06253*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Angela Fan, David Grangier, and Michael Auli. 2017. Controllable abstractive summarization. *arXiv preprint arXiv:1711.05217*.

Alexios Gidiotis, Stefanos Stefanidis, and Grigorios Tsoumakas. 2020. Auth@ clscisumm 20, laysumm 20, longsumm 20. In *Proceedings of the first workshop on scholarly document processing*, pages 251–260.

Alexios Gidiotis and Grigorios Tsoumakas. 2020. A divide-and-conquer approach to the summarization of long documents. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:3029–3040.

Quentin Grail, Julien Perez, and Eric Gaussier. 2021. Globalizing bert-based transformer architectures for long document summarization. In *Proceedings of the 16th conference of the European chapter of the association for computational linguistics: Main volume*, pages 1792–1810.

Jiafeng Guo, Yixing Fan, Qingyao Ai, and W Bruce Croft. 2016. A deep relevance matching model for ad-hoc retrieval. In *Proceedings of the 25th ACM international on conference on information and knowledge management*, pages 55–64.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Darsh Kaushik, Abdullah Faiz Ur Rahman Khilji, Utkarsh Sinha, and Partha Pakray. 2021. Cnlp-nits@ longsumm 2021: Textrank variant for generating long summaries. In *Proceedings of the Second Workshop on Scholarly Document Processing*, pages 103–109.

Maurice G Kendall. 1938. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93.

Sandeep Kumar, Guneet Singh Kohli, Kartik Shinde, and Asif Ekbal. 2022. Team ainlpml@ mup in sdp 2021: Scientific document summarization by end-to-end extractive and abstractive approach. In *Proceedings of the Third Workshop on Scholarly Document Processing*, pages 285–290.

Md Tahmid Rahman Laskar, Enamul Hoque, and Jimmy Xiangji Huang. 2020. Wsl-ds: Weakly supervised learning with distant supervision for query focused multi-document abstractive summarization. *arXiv preprint arXiv:2011.01421*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Lei Li, Yang Xie, Wei Liu, Yinan Liu, Yafei Jiang, Siya Qi, and Xingyuan Li. 2020. Cist@ cl-scisumm 2020, longsumm 2020: Automatic scientific document summarization. In *Proceedings of the First Workshop on Scholarly Document Processing*, pages 225–234.

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

Preksha Nema, Mitesh Khapra, Anirban Laha, and Balaraman Ravindran. 2017. Diversity driven attention model for query-based abstractive summarization. *arXiv preprint arXiv:1704.08300*.

Sayar Ghosh Roy, Nikhil Pinnaparaju, Risubh Jain, Manish Gupta, and Vasudeva Varma. 2021. Summaformers@ laysumm 20, longsumm 20. *arXiv preprint arXiv:2101.03553*.

Sajad Sotudeh, Arman Cohan, and Nazli Goharian. 2020. On generating extended summaries of long documents. *arXiv preprint arXiv:2012.14136*.

Sajad Sotudeh and Nazli Goharian. 2022. Guir@ mup 2022: Towards generating topic-aware multi-perspective summaries for scientific documents. In *Proceedings of the Third Workshop on Scholarly Document Processing*, pages 273–278.

Charles Spearman. 1961. " general intelligence" objectively determined and measured.

Dan Su, Tiezheng Yu, and Pascale Fung. 2021. Improve query focused abstractive summarization by incorporating answer relevance. *arXiv preprint arXiv:2105.12969*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ashok Urlana, Nirmal Surange, and Manish Shrivastava. 2022. Ltrc@ mup 2022: Multi-perspective scientific document summarization using pre-trained generation models. In *Proceedings of the Third Workshop on Scholarly Document Processing*, pages 279–284.

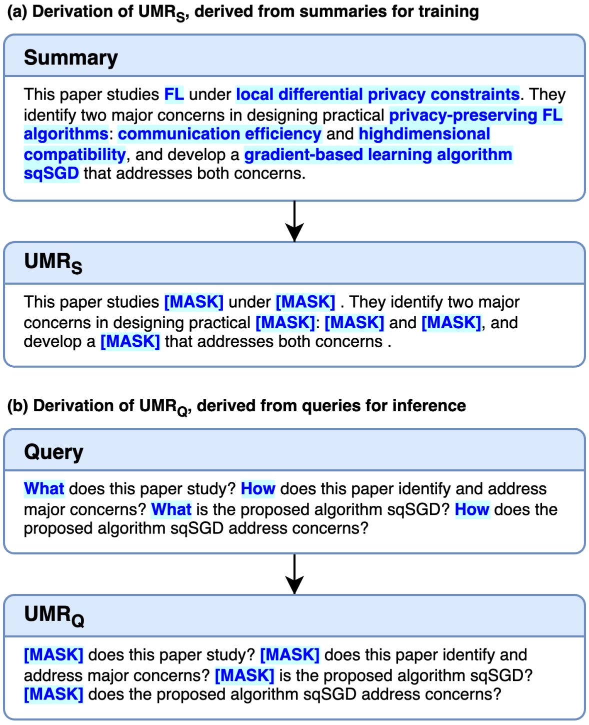
Wen Xiao and Giuseppe Carenini. 2019. Extractive summarization of long documents by combining global and local context. *arXiv preprint arXiv:1909.08089*.

Yumo Xu and Mirella Lapata. 2020. Generating query focused summaries from query-free resources. *arXiv preprint arXiv:2012.14774*.

Deming Ye, Yankai Lin, Peng Li, and Maosong Sun. 2021. Packed levitated marker for entity and relation extraction. *arXiv preprint arXiv:2109.06067*.

Senci Ying, Zheng Yan Zhao, and Wuhe Zou. 2021. Longsumm 2021: Session based automatic summarization model for scientific document. In *Proceedings of the Second Workshop on Scholarly Document Processing*, pages 97–102.

A Unified Masked Representation (UMR)



(a) Derivation of UMR_S, derived from summaries for training

Summary

This paper studies **FL** under **local differential privacy constraints**. They identify two major concerns in designing practical **privacy-preserving FL algorithms**: **communication efficiency** and **highdimensional compatibility**, and develop a **gradient-based learning algorithm sqSGD** that addresses both concerns.

UMR_S

This paper studies **[MASK]** under **[MASK]** . They identify two major concerns in designing practical **[MASK]**: **[MASK]** and **[MASK]**, and develop a **[MASK]** that addresses both concerns .

(b) Derivation of UMR_Q, derived from queries for inference

Query

What does this paper study? **How** does this paper identify and address major concerns? **What** is the proposed algorithm sqSGD? **How** does the proposed algorithm sqSGD address concerns?

UMR_Q

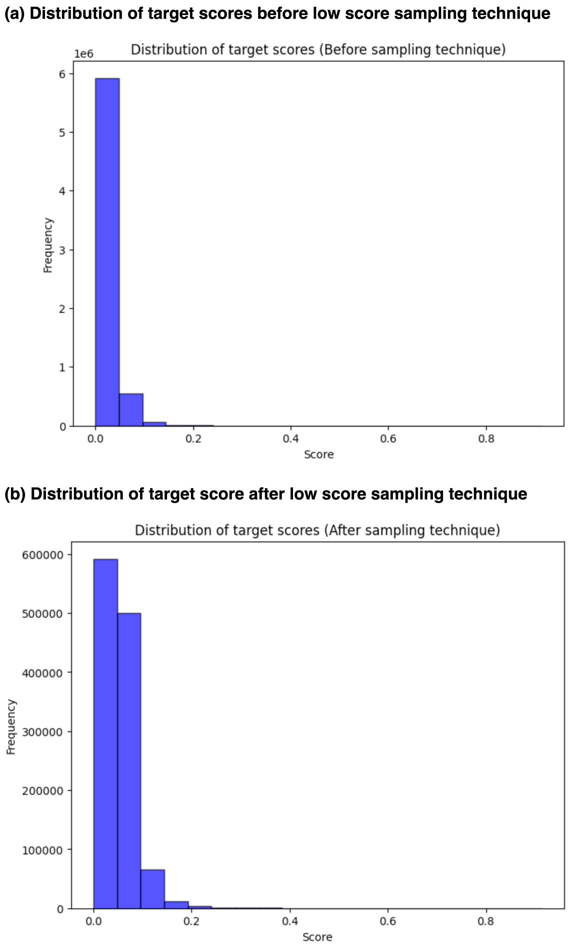
[MASK] does this paper study? **[MASK]** does this paper identify and address major concerns? **[MASK]** is the proposed algorithm sqSGD? **[MASK]** does the proposed algorithm sqSGD address concerns?

Figure 4: Unified Masked Representation (UMR)

As illustrated in Figure 4, (UMR_S) was derived from the entities-masked summary, while (UMR_Q) was shaped by masking question-words from the user query. Our work utilized PL-Marker (Ye et al., 2021), a state-of-the-art model in scientific entity-relation extraction.

B Low score sampling technique

From score visualization, in the training dataset, more than 90 percent of all samples have their scores lower than 0.05 (while the score range is from 0 to 1.15, with a maximum score of 0.91). This observation indicates that if we train on all samples, the model may excessively focus on less relevant sentences. To address this, we implemented a **low score sampling technique** by randomly removing samples with scores lower than 0.05. Specifically, we sampled out 90 percent of low-score samples, and the resulting distribution is illustrated in Figure 5. This not only ensures a more balanced distribution of scores for effective


(a) Distribution of target scores before low score sampling technique

(b) Distribution of target score after low score sampling technique

Figure 5: Distribution of target score in Ranking module (a) before sampling technique (b) after sampling technique

model training but also optimizes computational efficiency, reducing computation time.

C Synthetic query dataset

While the MuP dataset contains multiple summaries for each source paper, a crucial aspect absent is the inclusion of explicit queries. Recognizing the significance of queries in the Query-Focused Summarization (QFS) task, we employed the llama2 model (Touvron et al., 2023), a Large Language Model (LLM), utilizing a 2-shot prompting strategy for query creation. Each example in our query dataset comprises the abstract of the paper, its summary, and a manually crafted query, as illustrated in Figure 6. This method not only introduces specificity into the summarization process but also enhances the validation process for the QFS task.

The resulting query dataset exhibits an average of 111 words per query and 5.62 sentences per query, providing a rich and diverse set of queries for validation. This approach ensures that the queries align with the content of the papers, contributing to the effectiveness of the subsequent QFS framework.

D Impact of top-k sentences

For this experiment, we systematically varied the values of k from 4 to 30 to investigate the impact of top-k sentences selection on the overall performance of our framework. The results, depicted in Fig. 6, reveal an interesting trend: the module's performance shows improvement with increasing k values until reaching k=10, after which there is a slight decrease in performance.

While the retrieval performance increases with higher k values, it's important to note that the real summary generator, BART, has an input token limitation set at 1024 tokens. As a consequence, excessively high values of k may lead to the removal of important sentences located in the middle or late in the original paper during the truncation process. Therefore, considering both performance and the token limitation, the optimal value for k is determined to be k=10.

E Full experimental results

The full experimental results of summary generator modules GS task, and ablation study were shown in Table 7 and Table 8 respectively. In addition, the output generated from our frameworks were illustrated in Table 9.

Input
Instruction: Query-Focus Summarization (QFS) aims to create a short summary from a set of documents that answers a specific query. However, My dataset contain only document and summary, and exclude query. Let generate query based on Abstract of a paper and Summary of the paper as following example.
An example: [Abstract] Unsupervised dataset alignment estimates a transformation that maps two or more ... [/Abstract] [Summary] This paper proposes a flow-based method for the unsupervised data set alignment problem. ... [/Summary] [Query] What is the main proposal of this paper ? How does the paper address the minimization problem over density models, and what theoretical insight does it provide? What is the novel regularizer introduced by the authors? ... [/Query]
An example: [Abstract] ... [/Abstract] [Summary] ... [/Summary] [Query] ... [/Query]
Query generation: [Abstract] The principle of optimism in the face of (aleatoric and epistemic) uncertainty has been utilized ... [/Abstract] [Summary] This work introduces max-Q Entropy Search (MQES) exploration principle for ... [/Summary] [Query]
Output
What does this paper study? How does this paper identify and address major concerns? What is the proposed algorithm sqSGD? How does the proposed algorithm sqSGD address concerns? What is the gradient subsampling strategy proposed by the authors? How does it achieve better performance? What is the randomized rotation proposed by the authors? What role does it play in the preprocessing step of the proposed algorithm? In what ways does the proposed algorithm sqSGD successfully learn large models like LeNet and ResNet with local privacy constraints? How does the performance of sqSGD significantly dominate that of baseline algorithms? [/Query]

Table 6: A 2-shot learning strategy for query synthesis

		Validation set				Blind test set			
Framework	Experiment	R-1	R-2	R-L	R-avg	R-1	R-2	R-L	R-avg
Baseline		39.48	11.86	24.25	25.05	40.80	12.33	24.48	25.87
GATS (Akkasi, 2022)		35.46	9.53	19.63	21.54	33.85	7.40	17.74	19.66
LTRC (Urlana et al., 2022)		-	-	-	-	40.68	12.47	24.99	26.05
GUIR (Sotudeh and Goharian, 2022)		41.05	**12.18**	**24.61**	25.95	**41.36**	12.52	24.83	26.24
AINLPML (Kumar et al., 2022)		-	-	-	-	41.08	**13.29**	**25.36**	**26.58**
GS	R (k=10)	39.57	11.83	24.10	25.17	40.65	12.33	24.49	25.82
	R (ranked)	39.56	11.73	23.94	25.08	40.52	12.19	24.36	25.69
	AR (k=10)	39.65	11.80	24.05	25.16	40.57	12.27	24.47	25.77
	AR (ranked)	39.71	11.95	24.20	25.28	40.58	12.40	24.68	25.89
	AIR (k=10)	39.36	11.59	23.91	24.95	40.22	12.11	24.20	25.51
	AIR (ranked)	39.60	11.64	24.00	25.08	40.42	12.13	24.27	25.61
	ALiR (k=10)	39.02	11.55	23.91	24.83	40.88	12.31	24.43	25.87
	ALiR (ranked)	39.13	11.65	23.98	24.92	40.80	12.28	24.62	25.90
H-QFS	R (k=10)	**41.63**	12.00	24.35	26.00	40.16	11.64	23.43	25.08
	R (ranked)	41.39	12.12	24.56	**26.02**	41.14	12.29	24.15	25.86
	AR (k=10)	41.14	11.91	24.34	25.80	40.79	12.22	24.49	25.83
	AR (ranked)	40.80	11.73	24.19	25.57	40.82	12.29	24.19	25.77
	AIR (k=10)	40.81	11.98	24.54	25.78	40.28	11.76	23.30	25.11
	AIR (ranked)	40.41	11.75	24.13	25.43	40.91	12.30	24.20	25.80
	ALiR (k=10)	41.46	11.96	24.37	25.93	40.93	12.24	24.38	25.85
	ALiR (ranked)	41.11	11.96	24.40	25.82	40.81	12.24	24.47	25.84

Table 7: General Summarization (GS) task performance in term of ROUGE 1, 2, L, and average; **Bold**: 1st place, underline: 2nd place

	General summarization task				QFS task			
	R1	R2	RL	R-avg	R1	R2	RL	R-avg
QFS (AR-ranked)	-	-	-	-	44.29	17.88	27.67	29.95
-Len	-	-	-	-	44.01 (-0.28)	17.67 (-0.21)	27.41 (-0.26)	29.7 (-0.25)
-UMR	-	-	-	-	40.66 (-3.63)	11.74 (-6.14)	24.21 (-3.46)	25.54 (-4.41)
-UMR -Len	-	-	-	-	38.73 (-5.56)	11.30 (-6.58)	23.65 (-4.02)	24.56 (-5.39)
H-QFS (R-ranked)	41.39	12.12	24.56	26.02	44.76	17.57	27.85	30.06
-Len	39.36 (-2.03)	11.3 (-0.82)	23.61 (-0.95)	24.76 (-1.26)	43.93 (-0.83)	16.99 (-0.58)	27.47 (-0.38)	29.46 (-0.6)
-UMR	42.25 (0.86)	12.2 (0.08)	24.48 (-0.08)	26.31 (0.29)	43.12 (-1.64)	13.41 (-4.16)	25.49 (-2.36)	27.34 (-2.72)
-UMR -Len	39.30 (-2.09)	11.28 (-0.84)	23.58 (-0.98)	24.72 (-1.30)	39.83 (-4.93)	12.73 (-4.84)	24.77 (-3.08)	25.78 (-4.28)

Table 8: Ablation study for QFS and H-QFS frameworks in both GS and QFS task. The value in parenthesis is the different score between main framework and ablation experiments

Gold target: This paper focuses on deep reinforcement learning methods and discusses the presence of inductive biases in the existing RL algorithm. Specifically, they discuss biases that take the form of domain knowledge or hyper-parameter tuning. The authors state that such biases rise the tradeoff between generality and performance wherein strong biases can lead to efficient performance but deteriorate generalization across domains. Further, it motivates that most inductive biases has a cost associated to it and hence it is important to study and analyze the effect of such biases.
Query: How does this paper discuss the trade off between generality and performance in deep RL? How does this paper address the issue of injecting domain specific inductive biases in deep RL? How does this paper study the presence of different inductive biases in RL algorithms?
Length control: <len04> (91-120 words)
QFS†∗: Does this paper discuss the trade off between generality and performance in deep RL? How does this paper address the issue of injecting domain specific inductive biases in deepRL? What does this study the presence of different inductive bias in RL algorithms? The main benefit of having fewer domain-specific components.
GS: This paper studies the impact of inductive biases on generalization in reinforcement learning. In particular, the authors consider the effect of different types of biases, including domain knowledge and pretuned hyperparameters, on the generalization ability of deep RL algorithms. The authors compare the performance of two RL algorithms, AlphaZero and AlphaGo, with and without domain-specific biases, and show that the performance improves with fewer domain specific biases.
H-QFS (QFS)†∗: This paper studies the trade-off between generality and performance when we inject inductive biases into deep reinforcement learning (RL) algorithms. In particular, the authors consider two ways of injecting inductive bias: 1) sculpting the agent's objective (e.g., clipping and discounting rewards), 2) crafting the agent-environment interface. The authors evaluate the performance of the proposed methods on the Atari games.
H-QFS (GS)∗: This paper re-examines several domain-specific components that modify the agent's objective and environmental interface. The authors argue that inductive biases may mask the generality of other parts of the system as a whole; if a learning algorithm tuned for a specific domain does not generalize out of the box to a new domain, it can be unclear whether the underpinning learning algorithm is lacking something important. They then investigate the main benefit of having fewer domain specific components, by comparing the learning performance of the two systems on a different set of continuous control problems.

Table 9: Generated summary from our frameworks (†: involve query guidance in summary generator, ∗: involve length control in summary generator)

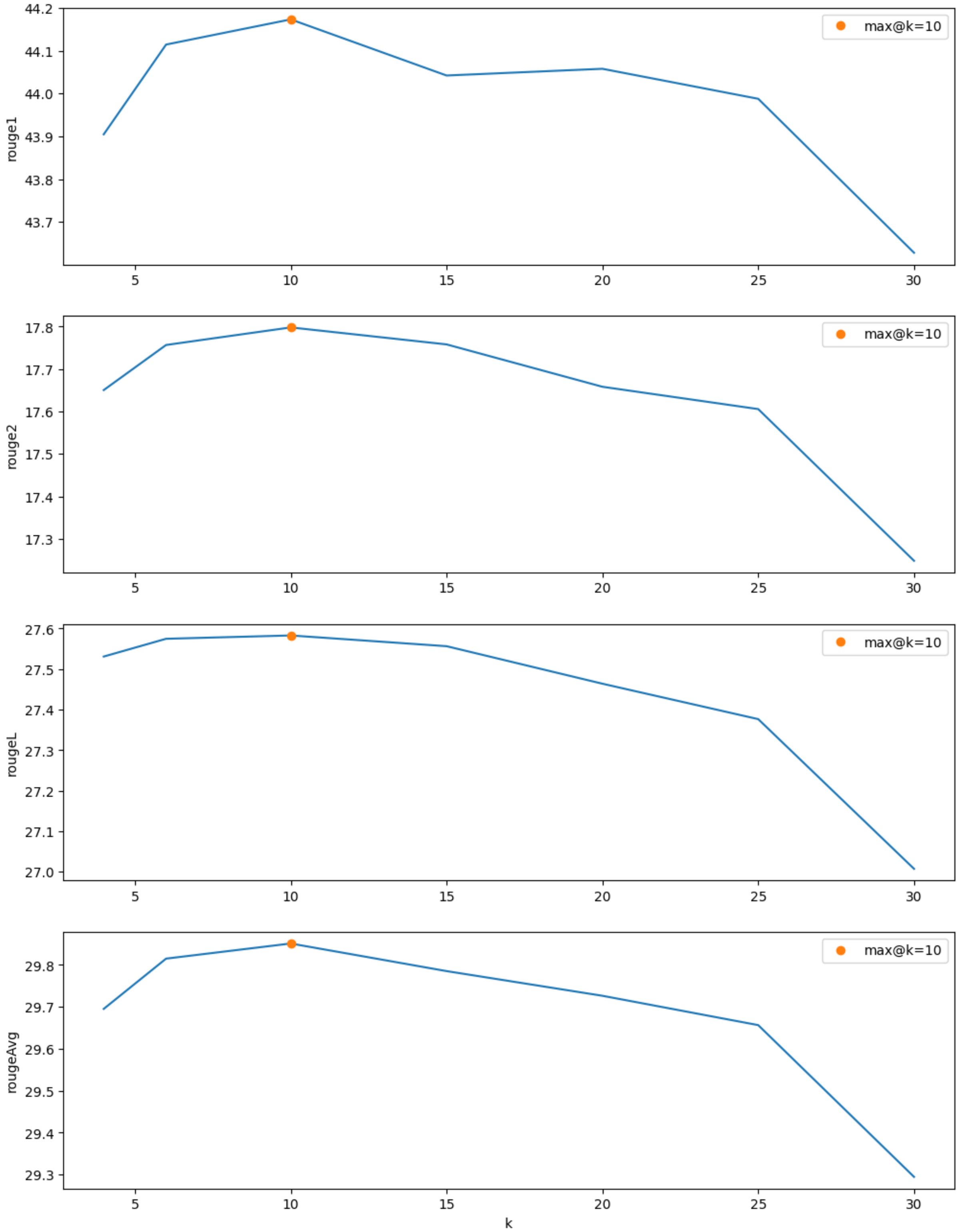


Figure 6: Performance of summary generator while vary k