LongSciArxiv: Dual Manual Synthetic Datasets and LLM Benchmarking for Long-to-Long Scientific Survey Generation

Anonymous ACL submission

Abstract

Survey generation involves synthesizing comprehensive scientific papers from large collections of research literature. Despite recent advances, this task remains challenging for natural language processing (NLP), especially when both input and output are long. While current large language models (LLMs) support extended context lengths, their ability to produce full-length surveys remains underexplored due to the lack of suitable datasets and benchmarks. We introduce **GenSurvey**, a dataset of 700 human-written surveys paired with reference abstracts. We further create **GenSection**, a synthetic dataset for section-level generation created using chain-of-thought prompting with GPT-4 and refined through human verification. These datasets form **LongSciArxiv**, a dual benchmark designed for real-world tasks in education and research. These task requires models to integrate hundreds of abstracts into coherent surveys exceeding 10,000 words. In our experiments, we evaluate 10 open-source LLMs ranging from 1B to 70B parameters. Results show that mid-sized models such as Mistral 7B and LLaMa3 8B offer the best trade-off between performance and cost. Our findings highlight the complexity of long-to-long generation and the need for scale-aware model design and benchmarking.

1 Introduction

Recent large language models (LLMs) such as GPT-4 (OpenAI, 2024) and Gemini-1.5 (Google et al., 2024) show strong performance in long-context understanding and generation tasks. For example, GPT-4 supports prompts up to 128,000 tokens. As a result, recent work has focused on benchmarking LLMs' ability to handle long inputs and produce extended outputs (Zhang et al., 2024; Wang et al., 2024a; Li et al., 2024a; Köksal et al., 2024). Most studies target summarization (Liu et al., 2024) or story generation (Xie and Riedl, 2024), where models transform text into long and short forms. However, generating coherent and detailed long-form output from large contexts remains a challenge (Wu et al., 2025). Automatically generating scientific surveys requires synthesizing many research abstracts into a coherent, structured long-form document (as illustrated in Figure 1). Despite advances in LLMs, the task remains difficult due to its extreme length, dense content, and structural complexity. First, scientific surveys often exceed the context window or output capacity of most models, requiring incremental generation that risks context loss and structural inconsistency (Hosseini et al., 2025). Second, encoding a large set of reference abstracts demands substantial memory and reasoning, especially under resource constraints (Jiang et al., 2024b). Third, effective survey writing goes beyond fluency. The capability of current LLMs is still insufficient, as they have difficulties in provid-

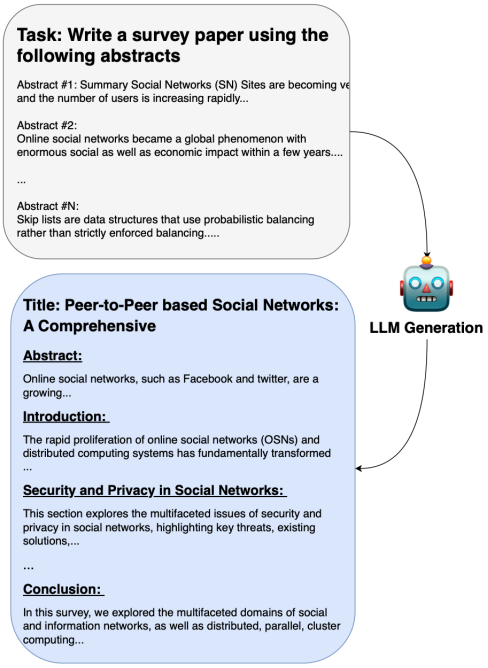

Figure 1: An example of long-to-long scientific text generation.

ing precise topic coverage, coherent organization, and accurate citation usage (Sel et al., 2025). We further discuss challenges in long-form generation in Appendix A.

Currently, Retrieval-Augmented Generation (RAG) offers a solution by integrating external knowledge to improve model accuracy (Lewis et al., 2020; Chen et al., 2024). Recent work also emphasizes the efficiency benefits of RAG and the advantages of using long-context models (Li et al., 2024b). Despite these advances, the application of LLMs, whether with long-context capabilities or RAGs, to synthesize full survey articles from multiple scientific sources is still underexplored. Although few studies have taken initial steps in this direction (Wang et al., 2024b), there are no public benchmarks to evaluate this specific task.

To study these challenges systematically, we introduce two new datasets. First, **GenSurvey** is a dataset of 700 scientific survey papers from Arxiv. All survey papers are written by humans, each longer than 10,000 words. Each paper is linked to about 100 reference abstracts. Then **GenSection** is built upon GenSurvey. It contains more than 4,700 synthetic section-level scientific texts. This dataset is generated using GPT-4 and manually verified by human annotators. Together, these two datasets form **LongSciArxiv** supporting both full-document and section-level evaluation. We fine-tune and evaluate 10 open-source LLMs of varying sizes to establish baseline performance.

The contributions of our study are as follows:

- We introduce **LongSciArvix** containing: **GenSurvey** is the first benchmark for full-document scientific survey generation; and **GenSection** is a high-quality instruction-tuning dataset for modular scientific generation. They support evaluation at both the document and section levels, with human and synthetic references.

- We fine-tune and evaluate 10 open-source LLMs (1B to 70B), including LLaMA3, Mistral, DeepSeek and Qwen2, using multiple training strategies. This enables a comprehensive analysis of model size, method, and efficiency trade-offs.

- We conduct extensive automatic and human evaluations on fluency, structure, and citation accuracy. Our findings show that mid-sized models like Mistral 7B and LLaMA3 8B achieve the best quality-efficiency balance and that larger LLMs benefit more from Reinforcement Learning with Human Feedback optimization in long-form generation tasks.

2 GenSurvey Dataset

2.1 Data Construction

Figure 2 illustrates the construction process of the GenSurvey dataset. The process includes four main steps: data collection, annotation, validation, and dataset splitting.

Data Collection We start by crawling ArXiv[1] for computer science papers with the keyword *"survey"* in their titles. We use PyPDF2[2] to convert each PDF into plain text. For every selected paper, we extract the reference list. We then retrieve the titles and abstracts of each cited reference using publicly available metadata.

Data Annotation Annotators with IT background follow clear instructions for filtering and cleaning data. They remove survey papers that do not meet quality standards. For selected papers, the annotators assign a topic and remove all figures and tables. Appendix B.1 describes the annotation requirements in detail.

Data Validation The annotation team members cross-validate their work. Each survey goes through multiple validation rounds to ensure consistency, completeness, and accuracy. This process helps maintain high-quality annotations for training and evaluation. Appendix B.2 provides the complete validation guideline. The results of the process are 700 survey papers. Each instance includes the full text of a survey and a corresponding set of reference abstracts. These pairs form the foundation of the GenSurvey dataset.

Dataset Splitting We store all the annotated content and metadata in JSON format. Each instance includes seven key attributes, listed in Table 7 (Appendix B.3). We divide the dataset into training, validation, and test sets using a 4:1:2 ratio.

2.2 Dataset Statistic and Analysis

2.3 Dataset Statistics

We summarize the statistics of the GenSurvey dataset in Table 1. To our knowledge, GenSurvey

[1] https://arxiv.org/
[2] https://pypi.org/project/PyPDF2

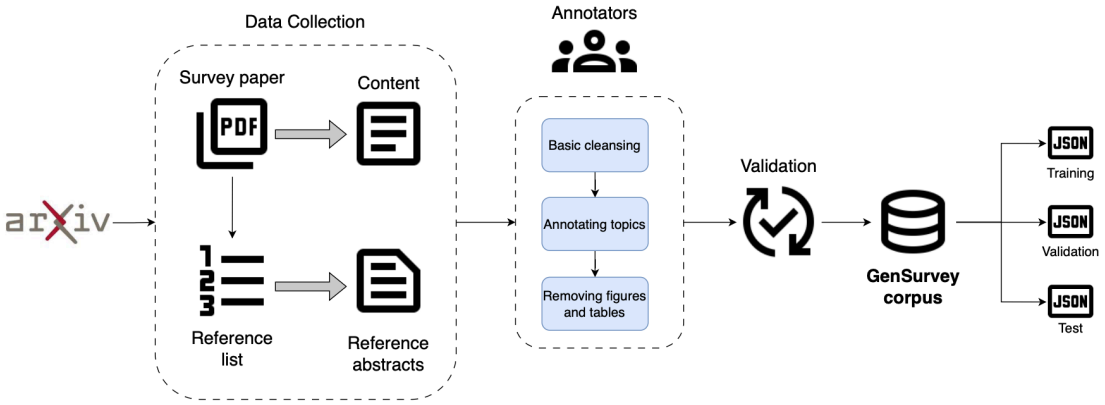

Figure 2: Data creation pipeline of GenSurvey Dataset.

is the first dataset designed for long-to-long generation with both long inputs and long outputs. On average, each input contains 3,307 words. The corresponding survey outputs average 10,942 words. In some cases, the inputs reach 31,639 words and the outputs extend to 42,628 words.

Description	Train	Val	Test	Total
No. of survey articles	400	100	200	700
Avg no. of subject per survey	1.71	1.72	1.95	1.78
Avg no. of reference per survey	98.54	97.27	126.83	106.42
Avg no. of word per survey	11,995	10,264	12,677	10,942
Avg no. of word in input	2891	2844	4188	3307

Table 1: GenSurvey dataset statistic.

More detailed statistics and comparisons are provided in the Appendix B.3. For each survey, the annotators identify 1 to 2 main topics and list approximately 100 references. According to Figure 5 in the Appendix, the most frequent topics are ***Machine Learning*** and ***Networking and Internet Architecture***.

3 GenSection Dataset

Although state-of-the-art LLMs perform well in many downstream tasks, their ability to write scientific surveys remains uncertain. We aim to explore whether LLMs can serve as automatic annotators for long-form generation. Previous work (Tan et al., 2024) shows that LLMs can label raw data using detailed instructions, even in domain-specific tasks. GenSurvey provides fully human-written survey papers for benchmarking. However, training models directly on GenSurvey is difficult because of the extreme length of inputs and outputs. To address this, we construct **GenSection**, a synthetic dataset that breaks down survey generation into smaller instruction–input–output triplets. Each triplet represents one section of a survey. This design enables efficient instruction tuning and supports section-level evaluation. In each triplet, *instruction* defines the title of the section and the goal of writing. The *input* includes relevant reference abstracts. The *output* is the text of the section generated by GPT-4. All outputs are reviewed by domain experts to ensure quality. GenSection complements GenSurvey by supporting scalable model training and providing a synthetic baseline for comparison.

3.1 Data Construction

The construction of the GenSection dataset follows a two-step process: survey structure generation and survey content generation. Figure 3 illustrates the full pipeline. We apply zero-shot chain-of-thought (COT) prompting with GPT-4 (OpenAI, 2024) to first generate a structured outline for a survey, then produce the content for each section. All prompt templates used in both steps are included in Appendix F.

Survey Structure Generation The input consists of abstracts from reference papers and a list of human-annotated topics. GPT-4 generates a structure for the survey, including sections such as *Introduction*, *Body Sections*, and *Conclusion*.

Survey Content Generation We design zero-shot COT prompting again to generate the full text for each section. The content is grounded in the provided abstracts and topic list.

3.2 Data Annotation and Validation

Human annotators validate each generated survey section for coherence, relevance, and alignment with the input abstracts. The validation process consists of two rounds. In the first round, two annotators with a background in IT independently review the outputs generated by GPT-4. They follow the detailed guidelines provided in the Appendix C. In the second round, we evaluate all samples where

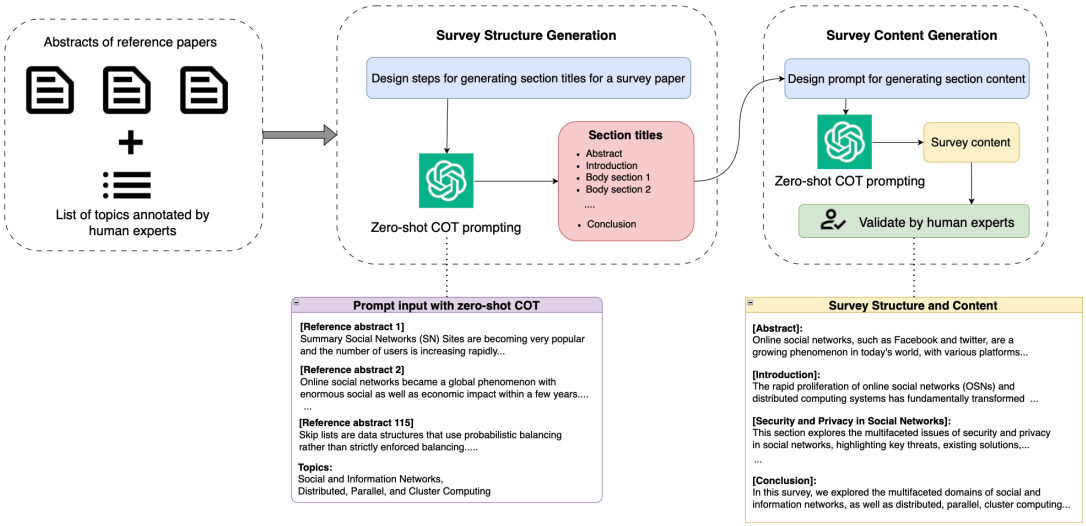

Figure 3: GenSection Dataset Construction Pipeline. The pipeline generates survey structure and content using zero-shot chain-of-thought (COT) prompting.

the annotators disagreed. The final decisions are made using the same set of guidelines. As a result, each instance includes an instruction prompt, a set of reference abstracts, and the validated output for one section of a survey paper.

3.3 Dataset Description

The GenSection dataset contains 4,792 instruction sets derived from 700 survey papers. Each entry corresponds to a single section within a survey. The dataset uses an instruction-following format, where the instruction guides the generation of a specific section using reference abstracts and annotated topics. This structure supports the training of models for section-level generation.

Description	Train	Val	Test	Total
No. of sections	3072	340	1380	4792
Avg no. of section per survey	6	6	6	6
Avg no. of word in sections	5564	5130	4166	4951

Table 2: GenSection dataset statistic.

Table 2 shows the statistics of the GenSection dataset. On average, each survey contains around six sections. The average word count per section is highest in the training set with 5,564 words, slightly lower in the validation set with 5,130 words, and lowest in the test set with 4,166 words. Across the entire dataset, the average length of the section is 4,951 words.

4 Experiments

In this study, we aim to evaluate the effectiveness of large language models (LLMs) on the task of long-to-long scientific text generation using our proposed GenSurvey and GenSection datasets. We organize our experiments around the following research questions:

- **RQ1**: How do different-sized open-source LLMs perform on long-to-long scientific text generation across automatic and human evaluation metrics?
- **RQ2**: What are the trade-offs between model size, output quality, and computational efficiency in handling long-context generation tasks?
- **RQ3**: How effective are different fine-tuning strategies (PEFT and RLHF) and LoRA variants in improving long-form generation performance?
- **RQ4**: How does the performance of fine-tuned open-source models compare to proprietary models such as GPT-4 in terms of fluency, structure, and citation handling?

4.1 Experimental Setup

As a baseline, we conduct the experiments to evaluate the performance of 10 LLMs with varying model sizes on GenSurvey and GenSection datasets. The models used are LLaMa3 (Grattafiori et al., 2024) with sizes 1B, 3B, and 8B; Qwen2 (Yang et al., 2024) with sizes 1.5B, 3B, 7B, and 72B; Mistral 7B (Jiang et al., 2023) and Mixtral of Expert (8x7B) (Jiang et al., 2024a); and Deepseek R1 (DeepSeek-AI et al., 2025) with 70B distilled version.

For the GenSurvey dataset, we use full human-written survey papers as outputs and their corresponding reference abstracts as inputs. Unlike GenSection, we do not split the surveys into smaller

sections. Instead, we preserve the original structure to evaluate document-level scientific generation. Each training instance includes a list of abstracts as input and a complete survey paper as output. This setting introduces a significant challenge for models, particularly those with limited context capacity, since both inputs and outputs are extremely long. This setup allows us to evaluate whether language models can synthesize scientific literature across extended contexts.

In contrast, the GenSection dataset is already structured for instruction-following. Each entry corresponds to a standalone section. We fine-tune models on this dataset directly without additional preprocessing. During evaluation, we concatenate generated sections using their file IDs to reconstruct the full survey paper. Additional implementation details are provided in Appendix D.1.

We train all the models on 8x80GB NVIDIA A100 GPU. We include the details of our training parameter in Table 9 in the Appendix D.2. The baseline models are fine-tuned using LoRA (Hu et al., 2022), DeepSpeed (Aminabadi et al., 2022), and Flash Attention (Dao, 2024). These techniques allow us to train LLMs on 32k of input context, and we set the generation text length to the maximum of the capability of each model. The models are all available on Hugging Face[3].

4.2 Evaluation Metrics

For evaluation, we employ a combination of quantitative and qualitative metrics to assess model performance. We use ROUGE scores (Lin, 2004) (R-1, R-2, R-L) to measure the n-gram overlap between the generated and human reference text. BERTScore (Zhang et al., 2020) is used to assess the contextual similarity between the generated content and the reference texts. To evaluate how well the generated survey content follows the structural organization of human-written surveys, we use Soft Heading Recall (S-H Recall) (Fränti and Mariescu-Istodor, 2023). This metric measures the alignment between the section headings in the generated output and those in the reference. In addition, we conduct a human evaluation to assess the relevance, fluency, coherence, and citation captured in the generated survey content. This evaluation ensures a comprehensive evaluation of the models' abilities. Additional details are provided in Appendix D.4.

[3]We include the list of models on the Appendix D.3.

5 Overall Results

We evaluate the fine-tuned models using the test set from the GenSurvey dataset. Table 3 summarizes the results based on automatic metrics. We use only the GenSurvey test set to ensure that all models are assessed on real-world data. Results from both **GenSurvey** and **GenSection** offer insights into model behavior across different input-output formats.

GenSurvey Dataset Mistral 7B achieves the highest scores on all automatic metrics. It records a ROUGE-1 of 0.778, ROUGE-2 of 0.418, ROUGE-L of 0.265, and a BERTScore of 0.886. These results show strong fluency and semantic alignment in long-form scientific generation. LLaMa3 8B and Qwen2 7B also perform well, confirming the strength of mid-sized models in handling fragmented and context-rich input. DeepSeek R1 Distilled 70B obtains a extremely high S-H Recall of 0.997, demonstrating its ability to preserve document structure despite lower lexical overlap. In contrast, smaller Qwen2 models struggle with both content quality and structural consistency.

GenSection Dataset On GenSection, performance is more uniform across models. The overall ROUGE scores are lower than those on GenSurvey, but the gap between large and small models narrows. LLaMa3 3B achieves the highest BERTScore of 0.876, suggesting it can generate semantically rich section content without full-document context. Mistral 7B again performs consistently well across all metrics. DeepSeek also leads in S-H Recall with a score of 0.997. Meanwhile, Qwen2 1.5B and 3B achieve relatively high BERTScores (0.855) but much lower S-H Recall scores (0.517 and 0.527), indicating poor structural alignment despite semantic relevance.

Comparison Across both datasets, baseline models generally achieve higher ROUGE and BERTScore values on GenSurvey. However, S-H Recall remains consistently high for DeepSeek and Mistral 7B. This suggests that both models effectively capture structural and formatting patterns. The largest gap in structure-aware performance appears in smaller Qwen2 models. These models favor lexical overlap but fail to maintain coherent section structure. Based on Table 3, we conclude that **Mistral 7B and DeepSeek R1 Distilled 70B perform best in automatic evaluations *(RQ1)***.

Model	GenSurvey Dataset					GenSection Dataset				
	R-1 ↑	R-2 ↑	R-L ↑	S-H Recall ↑	BERTScore ↑	R-1 ↑	R-2 ↑	R-L ↑	S-H Recall ↑	BERTScore ↑
DeepSeek R1 Distilled 70B	0.548	0.292	0.193	**0.997**	0.841	**0.407**	**0.111**	0.125	**0.997**	0.709
LLaMa3 1B	0.644	0.318	0.207	0.964	0.871	0.336	0.096	0.117	0.964	0.717
LLaMa3 3B	0.726	0.298	0.216	0.966	0.719	0.336	0.097	0.117	0.966	**0.876**
LLaMa3 8B	0.735	0.389	0.253	0.966	0.876	0.337	0.098	0.116	0.966	0.719
Mistral 7B	**0.778**	**0.418**	**0.265**	0.977	**0.886**	0.343	0.100	0.115	0.977	0.721
Mixtral 8x7B	0.625	0.324	0.219	0.966	0.870	0.314	0.089	0.114	0.754	0.717
Qwen2 1.5B	0.631	0.294	0.211	0.518	0.855	0.338	0.100	0.119	0.517	0.711
Qwen2 3B	0.600	0.279	0.210	0.527	0.855	0.354	0.104	**0.126**	0.527	0.855
Qwen2 7B	0.708	0.379	0.242	0.892	0.879	0.334	0.096	0.115	0.892	0.719
Qwen2 72B	0.695	0.363	0.247	0.936	0.878	0.357	0.104	0.121	0.936	0.721

Table 3: Model performance results on GenSurvey and GenSection datasets. Metrics include ROUGE scores (R-1, R-2, R-L), S-H Recall, and BERTScore. The arrow indicates the higher values is the better. The **bold** text indicates the highest scores while the underline text highlights the second best.

6 Discussion and Ablation Study

6.1 Human Evaluation

To further validate and analyze the generated text from our baseline models, we employ two experts in Computer Science to evaluate outputs from ten models. Due to cost constraints, we randomly select 100 test samples from the test set. As a result, each expert is required to rate a total of 1,000 generated texts across the ten models. The experts use a 5-point Likert scale, where (1) represents the worst and (5) the best quality.

We evaluate the generated text based on three key aspects: (i) *Relatedness*, which measures how well the generated text matches the human-written reference; (ii) *Readability*, which evaluates how structured and coherent the text is; and (iii) *Citation Capture*, which quantifies how accurately the model identifies and includes relevant citations in the generated content. The details on implementing human evaluation are provided in the Appendix E.1.

We calculate the average score from both experts and summarize the results in Table 4. On the **GenSurvey** dataset, LLaMa3 8B achieves the highest score in *Relatedness* and the second highest in *Readability*, while DeepSeek R1 Distilled 70B achieves the best score in *Citation Captured*. LLaMa3 3B shows the strongest *Readability*, despite slightly lower scores in the other dimensions. The Qwen2 models show relatively low performance in all three evaluation aspects. This is especially noticeable in the 1.5B and 3B variants.

On the **GenSection** dataset, DeepSeek R1 leads in *Citation Captured* with a score of 3.500 on average and remains strong in *Readability*. LLaMa3 8B stands out with the highest *Readability* of 3.600 and the best overall *Relatedness* score of 2.984. In contrast, Qwen2 1.5B and 3B again score the lowest in nearly all dimensions. **The human evaluation results indicate that LLaMa3 8B and DeepSeek R1 Distilled 70B consistently generate more relevant, readable, and citation-sensitive scientific text *(RQ1)*.** Mistral 7B also shows stable and competitive performance across all criteria, though it does not rank among the top in any single category. However, human validation findings indicate that even advanced LLMs such as DeepSeek Distilled 70B and Qwen2 72B are still unable to produce scientific surveys that match the depth and coherence of human-written texts. This underscores the difficulty of our benchmarks and highlights the ongoing challenges in long-to-long scientific text generation.

6.2 Performance versus Model Sizes

In our experiments, we record the training time for all baseline models. Figure 8 in Appendix E.2 shows the relationship between model performance (P) and training time in hours (T). The performance score (P) is calculated as the average of all evaluation metrics. These include ROUGE, BERTScore, and S-H Recall on the GenSurvey dataset.

We observe that models with smaller parameter sizes, such as Qwen2 1.5B and LLaMa3 1B, achieve moderate performance while requiring much less training time. Larger models like DeepSeek R1 70B and Qwen2 72B exceed 1,000 minutes of training time, but offer only marginal improvements in performance compared to smaller models. Mistral 7B and LLaMa3 8B achieve a favorable balance between performance and training time. Both models attain high evaluation scores while maintaining moderate computing demands.

Model	GenSurvey Dataset			GenSection Dataset		
	Relatedness ↑	Readability ↑	Citation Captured ↑	Relatedness ↑	Readability ↑	Citation Captured ↑
DeepSeek R1 Distilled 70B	3.200	2.900	**3.400**	2.120	3.135	**3.500**
LLaMa3 1B	3.010	3.333	2.229	2.041	3.100	2.100
LLaMa3 3B	3.050	**3.500**	2.100	2.051	3.200	2.100
LLaMa3 8B	**3.300**	3.400	2.250	**2.984**	**3.600**	2.360
Mistral 7B	3.250	3.200	2.950	2.750	3.160	3.100
Mixtral 8x7B	3.000	2.619	2.190	2.870	2.870	2.230
Qwen2 1.5B	2.300	1.800	1.800	1.210	1.230	1.000
Qwen2 3B	2.050	1.700	1.600	1.540	1.460	1.500
Qwen2 7B	3.050	2.900	2.250	1.610	2.546	3.230
Qwen2 72B	3.281	2.952	3.190	2.901	2.541	3.360

Table 4: Human evaluation results on GenSurvey and GenSection datasets. The arrow indicates the higher values is the better. The **bold** text indicates the highest scores while the underline text highlights the second best.

The size of each bubble in Figure 8 reflects the size of the model parameter. This visualization reinforces the finding that larger model scale does not lead to a linear gain in efficiency. **These results suggest that small and medium-sized LLMs are practical choices when considering computational cost *(RQ2)*.** LLaMa3 1B, in particular, offers a cost-effective option for resource-constrained environments. Additional analysis is available in Appendix E.2.

6.3 Supervised fine-tuning (SFT) versus Reinforcement Learning from Human Feedback (RLHF)

While supervised fine-tuning (SFT) enables large language models to replicate human-written responses based on instruction–response pairs, its effectiveness depends on the quality and diversity of labeled data. Reinforcement Learning from Human Feedback (RLHF) (Ouyang et al., 2022) extends this paradigm by using preference-based learning signals derived from human judgments. Instead of learning to reproduce specific outputs, RLHF trains models to align with human preferences by optimizing a reward model built from pairwise comparisons. We apply Direct Preference Optimization (DPO) (Rafailov et al., 2023) to three representative models: LLaMa3 1B, Mistral 7B, and DeepSeek R1 Distill 70B. These models are selected based on their strong performance in our earlier analysis. We compare their performance to standard fine-tuning with LoRA. DPO reformulates reinforcement learning as a binary classification problem. It directly optimizes a loss function that favors preferred responses over rejected ones. This approach avoids the need for reward modeling, policy sampling, and extensive hyperparameter tuning.

We implement two configurations. The first applies DPO as a standalone training method. The second uses LoRA for initial fine-tuning, followed by DPO. We compare these results with the baseline that uses LoRA only. Appendix E.3 describes our detailed implementation.

Model	Pipeline	R-1	R-2	R-L	BERTScore	S-H recall
LLaMa 1B (#1 on Efficiency)	LoRA	0.644	0.318	0.207	0.871	**0.964**
	LoRA + DPO	**0.717**	**0.337**	**0.212**	0.870	0.957
	DPO	0.501	0.198	0.178	**0.885**	0.937
Mistral 7B (#1 on Performance)	LoRA	**0.778**	**0.418**	**0.265**	**0.886**	0.977
	LoRA + DPO	0.758	0.386	0.254	0.885	0.989
	DPO	0.716	0.340	0.221	0.882	**0.990**
DeepSeek R1 Distilled 70B (#1 on Human evaluation)	LoRA	0.548	0.292	0.193	0.841	**0.997**
	LoRA + DPO	0.587	0.313	0.202	0.842	0.990
	DPO	**0.593**	**0.317**	**0.203**	**0.844**	0.991

Table 5: Performance comparison of three selected models under different fine-tuning pipelines.

Table 5 presents the results. For LLaMa3 1B, the LoRA+DPO pipeline achieves the highest performance across all metrics. ROUGE scores show significant improvements. For Mistral 7B, the best results come from using LoRA alone, especially in ROUGE and BERTScore. For DeepSeek 70B, DPO without LoRA achieves the top performance.

These results show that DPO benefits larger models by providing a stronger optimization signal. It helps the model generate more human-like scientific output. **Overall, fine-tuning effectiveness depends on model size. LoRA+DPO performs best for smaller models, while DPO alone scales more effectively with larger models in long-form generation *(RQ3)*.**

6.4 LoRA Variants

In recent years, Parameter-Efficient Fine-Tuning (PEFT) methods (Xu et al., 2023) have gained popularity due to their ability to reduce trainable parameters while maintaining model performance. In our main experiments, we use LoRA (Hu et al., 2022) as the baseline fine-tuning approach for training

LLMs.

In this analysis, we extend our investigation to several LoRA variants. These include QLoRA (Dettmers et al., 2023), DoRA (Mao et al., 2024), and PiSSA (Meng et al., 2024). Appendix E.4 provides additional implementation details.

Method	Training time (min)	R-1	R-2	R-L	BERTScore	S-H recall
LoRA	16	0.644	0.318	0.207	0.871	**0.964**
DoRA	40	**0.715**	**0.357**	**0.228**	**0.877**	0.926
QLoRA	12	0.654	0.309	0.208	0.868	0.930
PiSSA	30	0.661	0.298	0.216	0.872	0.931

Table 6: Performance comparison of variants of LoRA methods on LLaMa3 1B.

The results in Table 6 show that DoRA achieves the highest scores across ROUGE-1, ROUGE-2, ROUGE-L, and BERTScore. It outperforms all other methods in content-related metrics, indicating strong performance in long-form text generation. However, its S-H Recall score of 0.926 is lower than that of LoRA and PiSSA. PiSSA is second in the overall ranking. It achieves the highest S-H Recall of 0.931 and performs competitively in other metrics.

LoRA is the most time-efficient method, requiring only 16 minutes of training. It performs well in S-H Recall of 0.964 but shows a lower ROUGE-L and BERTScore, suggesting a trade-off between structural accuracy and textual quality. QLoRA offers balanced results on all metrics. It ranks slightly above LoRA in ROUGE-1, while using the least training time and memory. **These findings suggest that DoRA provides the highest content quality, though at a higher computational cost. PiSSA offers strong structural consistency. LoRA and QLoRA remain practical options when prioritizing efficiency in resource-constrained settings *(RQ3)*.**

6.5 Open-source versus Closed-source LLMs

As proprietary LLMs such as GPT-4 (OpenAI, 2024) demonstrate their superiority in multiple tasks, we compare the performance of our fine-tuned baseline models with zero-shot prompting in GPT-4 to investigate the gap between commercialized LLMs and fine-tuned open-source LLMs. We use GPT-4o[4] version on 200 samples of our GenSurvey test set in this experiments. The prompt for GPT-4o is provided in the Appendix F.

[4] https://platform.openai.com/docs/models/gpt-4o

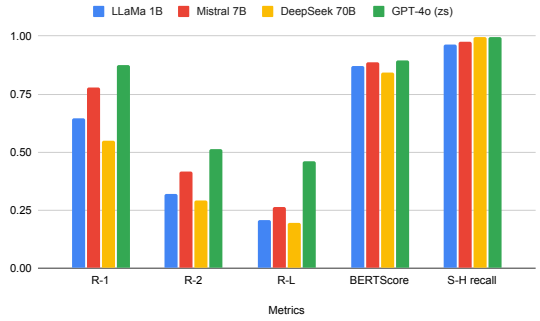

Figure 4: Performance of GPT-4 compared to our best baselines.

Figure 4 shows that GPT-4 outperforms all baseline models in all evaluation metrics. It demonstrates stronger semantic alignment and better structural consistency. However, the performance gap between GPT-4 and Mistral 7B, one of our strongest open-source baselines, remains relatively small.

In the S-H Recall metric, which measures similarity between the section structure of human and generated outputs, all models achieve high scores. DeepSeek R1 matches GPT-4 in this aspect, indicating that large open-source models can effectively capture structural patterns. **While GPT-4 leads in semantic quality, the strong results of Mistral 7B and DeepSeek R1 highlight the potential of open-source models to generate high-quality survey content *(RQ4)*.**

7 Conclusion

In this paper, we propose **LongSciArxiv** containing two datasets, GenSurvey and GenSection, designed for long-to-long scientific text generation. While **GenSurvey** is a fully human-written dataset, **GenSection** provides an alternative approach for enabling shorter-context models to generate survey sections. We fine-tuned multiple open-source LLMs of varying sizes and conducted comprehensive experiments to evaluate their performance. Our results demonstrate that generating high-quality, coherent survey papers remains a challenging task, even for the most advanced models. These findings highlight the complexity of long-to-long text generation and the need for further advancements in this area. We also show that methods such as DPO can improve output quality depending on model size. Furthermore, strong results from models like Mistral 7B and DeepSeekR1 illustrate the growing potential of open-source alternatives to closed LLMs.

Limitations

While our study presents a comprehensive empirical benchmark for long-to-long scientific generation, it has several limitations that offer directions for future work:

- **Computational constraints.** Due to limited GPU resources, we did not experiment with extremely large-scale models (e.g., DeepSeek R1 168B, LLaMa3 405B), which may offer improved performance. Our focus remains on accessible, mid-sized models (1B–8B) relevant to resource-constrained settings.

- **Evaluation scope.** Our human evaluation involved expert annotators in main experiments due to cost. While inter-rater agreement was maintained, broader annotator pools or multi-aspect scoring (e.g. citation granularity) would yield more robust conclusions.

- **Synthetic supervision bias.** GenSection relies on GPT-4 for synthetic section generation, which may introduce stylistic or structural biases. Despite human validation, future work may incorporate more diverse annotator strategies or adversarial filtering to avoid overfitting to GPT-4 outputs.

- **Domain specificity.** Both GenSurvey and GenSection are constructed from computer science papers on arXiv. While this ensures topic coherence, it limits generalizability to domains like medicine, law, or humanities. Expanding to other scientific disciplines is a valuable next step.

- **Planning and discourse modeling.** While models are evaluated on structure (via S-H Recall), we do not explicitly assess or train on discourse coherence or plan-based organization. As suggested by LongEval (Wu et al., 2025), incorporating discourse planning objectives could enhance model structure adherence.

- **Long-context limitations.** As highlighted in recent work (Hosseini et al., 2025), LLMs may appear to support long contexts but struggle with retaining and reasoning over distant dependencies. This limitation persists in our setup but is not explicitly measured. Although baseline models are able to handle most of the long context input, as described in Appendix D.1, in the case of extremely long context input, we have to trim down to 32k tokens.

Ethics Statement

The surveys and abstracts are open accessed on ArXiv. The copyright of the data (survey papers and abstract from references) remains to the original authors. Our datasets will be provided under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. All manual annotation is performed by research team members and students in the University. Annotators received detailed annotation guidelines before starting their tasks and received fair compensation after completion of the task. Personal information was not collected from any annotators or any stage of data collection. All the models we used in this paper adhere to the copyrights and licenses.

References

Reza Yazdani Aminabadi, Samyam Rajbhandari, Minjia Zhang, Ammar Ahmad Awan, Cheng Li, Du Li, Elton Zheng, Jeff Rasley, Shaden Smith, Olatunji Ruwase, and Yuxiong He. 2022. Deepspeed inference: Enabling efficient inference of transformer models at unprecedented scale. *Preprint*, arXiv:2207.00032.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2024. LongBench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3119–3137, Bangkok, Thailand. Association for Computational Linguistics.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *Preprint*, arXiv:2004.05150.

Ilias Chalkidis, Xiang Dai, Manos Fergadiotis, Prodromos Malakasiotis, and Desmond Elliott. 2022. An exploration of hierarchical attention transformers for efficient long document classification. *Preprint*, arXiv:2210.05529.

Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2024. Benchmarking large language models in retrieval-augmented generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17754–17762.

Krzysztof Marcin Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Quincy Davis, Afroz Mohiuddin, Lukasz Kaiser, David Benjamin Belanger, Lucy J Colwell, and Adrian Weller. 2021.

Rethinking attention with performers. In *International Conference on Learning Representations*.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy. Association for Computational Linguistics.

Tri Dao. 2024. Flashattention-2: Faster attention with better parallelism and work partitioning. In *The Twelfth International Conference on Learning Representations*.

DeepSeek-AI, Daya Guo, and et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient finetuning of quantized LLMs. In *Thirty-seventh Conference on Neural Information Processing Systems*.

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

Zican Dong, Tianyi Tang, Lunyi Li, and Wayne Xin Zhao. 2023. A survey on long text modeling with transformers. *Preprint*, arXiv:2302.14502.

Pasi Fränti and Radu Mariescu-Istodor. 2023. Soft precision and recall. *Pattern Recognition Letters*, 167:115–121.

Gemini Team Google, Petko Georgiev, and et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *Preprint*, arXiv:2403.05530.

Aaron Grattafiori, Abhimanyu Dubey, and et al. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Peyman Hosseini, Ignacio Castro, Iacopo Ghinassi, and Matthew Purver. 2025. Efficient solutions for an intriguing failure of LLMs: Long context window does not mean LLMs can analyze long sequences flawlessly. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 1880–1891, Abu Dhabi, UAE. Association for Computational Linguistics.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Luyang Huang, Shuyang Cao, Nikolaus Parulian, Heng Ji, and Lu Wang. 2021. Efficient attentions for long document summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1419–1436, Online. Association for Computational Linguistics.

Albert Q. Jiang, Alexandre Sablayrolles, and et al. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Albert Q. Jiang, Alexandre Sablayrolles, and et al. 2024a. Mixtral of experts. *Preprint*, arXiv:2401.04088.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2024b. LongLLMLingua: Accelerating and enhancing LLMs in long context scenarios via prompt compression. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1658–1677, Bangkok, Thailand. Association for Computational Linguistics.

Abdullatif Köksal, Timo Schick, Anna Korhonen, and Hinrich Schütze. 2024. Longform: Effective instruction tuning with reverse instructions. *Preprint*, arXiv:2304.08460.

J. Richard Landis and Gary G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174.

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

Tianle Li, Ge Zhang, Quy Duc Do, Xiang Yue, and Wenhu Chen. 2024a. Long-context llms struggle with long in-context learning. *Preprint*, arXiv:2404.02060.

Zhuowan Li, Cheng Li, Mingyang Zhang, Qiaozhu Mei, and Michael Bendersky. 2024b. Retrieval augmented generation or long-context LLMs? a comprehensive study and hybrid approach. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 881–893, Miami, Florida, US. Association for Computational Linguistics.

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

Ran Liu, Ming Liu, Min Yu, He Zhang, Jianguo Jiang, Gang Li, and Weiqing Huang. 2024. SumSurvey: An abstractive dataset of scientific survey papers for long document summarization. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9632–9651, Bangkok, Thailand. Association for Computational Linguistics.

Yulong Mao, Kaiyu Huang, Changhao Guan, Ganglin Bao, Fengran Mo, and Jinan Xu. 2024. DoRA: Enhancing parameter-efficient fine-tuning with dynamic rank distribution. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11662–11675, Bangkok, Thailand. Association for Computational Linguistics.

Mary McHugh. 2012. Interrater reliability: The kappa statistic. *Biochemia medica : časopis Hrvatskoga društva medicinskih biokemičara / HDMB*, 22:276–82.

Fanxu Meng, Zhaohui Wang, and Muhan Zhang. 2024. PiSSA: Principal singular values and singular vectors adaptation of large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

OpenAI. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

Jack W. Rae, Anna Potapenko, Siddhant M. Jayakumar, Chloe Hillier, and Timothy P. Lillicrap. 2020. Compressive transformers for long-range sequence modelling. In *International Conference on Learning Representations*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Bilgehan Sel, Ruoxi Jia, and Ming Jin. 2025. LLMs can plan only if we tell them. In *The Thirteenth International Conference on Learning Representations*.

Zhen Tan, Dawei Li, Song Wang, Alimohammad Beigi, Bohan Jiang, Amrita Bhattacharjee, Mansooreh Karami, Jundong Li, Lu Cheng, and Huan Liu. 2024. Large language models for data annotation and synthesis: A survey. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 930–957, Miami, Florida, USA. Association for Computational Linguistics.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550, Online. Association for Computational Linguistics.

Chonghua Wang, Haodong Duan, Songyang Zhang, Dahua Lin, and Kai Chen. 2024a. Ada-LEval: Evaluating long-context LLMs with length-adaptable benchmarks. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3712–3724, Mexico City, Mexico. Association for Computational Linguistics.

Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. 2020. Linformer: Self-attention with linear complexity. *Preprint*, arXiv:2006.04768.

Yidong Wang, Qi Guo, Wenjin Yao, Hongbo Zhang, Xin Zhang, Zhen Wu, Meishan Zhang, Xinyu Dai, Min zhang, Qingsong Wen, Wei Ye, Shikun Zhang, and Yue Zhang. 2024b. Autosurvey: Large language models can automatically write surveys. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Siwei Wu, Yizhi Li, Xingwei Qu, Rishi Ravikumar, Yucheng Li, Tyler Loakman, Shanghaoran Quan, Xiaoyong Wei, Riza Batista-Navarro, and Chenghua Lin. 2025. Longeval: A comprehensive analysis of long-text generation through a plan-based paradigm. *Preprint*, arXiv:2502.19103.

Kaige Xie and Mark Riedl. 2024. Creating suspenseful stories: Iterative planning with large language models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2391–2407, St. Julian's, Malta. Association for Computational Linguistics.

Lingling Xu, Haoran Xie, Si-Zhao Joe Qin, Xiaohui Tao, and Fu Lee Wang. 2023. Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment. *Preprint*, arXiv:2312.12148.

An Yang, Baosong Yang, and et al. 2024. Qwen2 technical report. *Preprint*, arXiv:2407.10671.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems*, 33.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Hao, Xu Han, Zhen Thai, Shuo Wang, Zhiyuan Liu, and Maosong Sun. 2024. ∞Bench: Extending long context evaluation beyond 100K tokens. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15262–15277, Bangkok, Thailand. Association for Computational Linguistics.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, and Zheyan Luo. 2024. LlamaFactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 400–410, Bangkok, Thailand. Association for Computational Linguistics.

A Related Work on Long-form Text Generation

Long-form text is typically defined as documents that span thousands of tokens, far exceeding the length of short texts or passages that fit within standard model contexts (Dong et al., 2023). Such lengths surpass the fixed input limits of typical Transformer-based language models (e.g. BERT is capped at 512 tokens (Devlin et al., 2019)), and they exacerbate the $O(n^2)$ time and memory complexity of self-attention, making naive processing of long documents infeasible. To address these challenges, researchers have explored multiple architectural innovations. Hierarchical models decompose a document into smaller units (e.g. sentences or paragraphs), which are encoded separately and then aggregated by higher-level encoders to capture long-range dependencies (Chalkidis et al., 2022). Efficient Transformers with sparse or structured attention patterns limit the attention scope to a local window and select global tokens, reducing complexity while preserving context (e.g., Longformer (Beltagy et al., 2020), BigBird (Zaheer et al., 2020)). Other models extend Transformer memory via recurrence. Transformer-XL introduces segment-level recurrence to carry forward states across chunks, and Compressive Transformer further condenses past activations to retain longer-term context (Dai et al., 2019; Rae et al., 2020). Additionally, low-rank approximation and kernel-based attention methods (such as Linformer and Performer) achieve linear or near-linear scaling, enabling processing of sequences with thousands of tokens (Wang et al., 2020; Choromanski et al., 2021). These advances substantially expand the range of text lengths that can be modeled, though fully capturing global coherence and long-range dependencies in very lengthy documents remains an open challenge (Dong et al., 2023).

B GenSurvey Details

B.1 Data Annotation Guidelines

Three annotators with an IT background are tasked with reviewing and annotating survey papers according to the following guidelines:

1. **Data Filtering:** Filter out survey papers that has converting error and exclude papers that are incomplete or fail to provide sufficient content.

2. **Basic Cleaning:** Remove formatting issues and irrelevant metadata

3. **Topic Annotation:** Carefully read through the content of the survey papers and annotate the primary topics for each paper. The annotated topics should accurately reflect the central themes of the survey. Annotators should annotate at most 4 topics for each survey paper.

4. **Removal of Figures and Tables:** Remove all figures, tables, and non-text content from the survey papers. Only the textual content should remain for the annotation process.

These steps ensure the quality and consistency of the annotated data. Annotators also need to collect the abstracts of references extracted from the reference list in each survey paper. If the abstract is not available or the reference paper is not accessible, the annotator will skip and move to the next one. As a result, the process helps surface common patterns and edge cases that ensures dataset quality and downstream modeling.

B.2 Data Validation Guidelines

In this phase, the annotation team performs cross-validation of the quality of the data annotated by their peers. Each survey should be reviewed by at least two annotators to ensure objectivity and quality control. The following guidelines are to be followed by annotators during the cross-validation process:

1. **Check for Consistency:**
   - Flag any inconsistencies where the same type of content is annotated differently (e.g., the same topic being annotated with different labels).
   - Ensure that the style and terminology used in the annotations are consistent across all documents.

2. **Check for Completeness:**
   - Review each annotated survey to ensure that all relevant sections have been well-structured.
   - Ensure that no important content has been omitted. If any section seems to lack an annotation, flag it for review.

3. **Check for Accuracy:**
   - Ensure that the annotated topics accurately reflect the content of the survey section..
   - Identify and correct any annotations that are not aligned with the content of the original survey.

4. **Flag Incomplete or Ambiguous Annotations:**
   - If annotators come across any ambiguous, incomplete, or unclear annotations, highlight them and make a note for further discussion with the annotator who performed the original annotation.
   - For sections that are difficult to interpret or if the annotation seems incorrect, request clarification or further feedback from the original annotator.

5. **Provide Feedback to Annotators:**
   - After reviewing the annotations, provide feedback to the original annotator, pointing out any inconsistencies, missing content, or errors.
   - Offer constructive suggestions for improving the annotations, especially if you find patterns in errors across multiple papers.
   - Ensure that feedback is clear and actionable, to help the original annotator make the necessary revisions.

These steps ensure that the annotations are consistent, complete, and accurate. The cross-validation process is critical for maintaining the quality and reliability of the data, which is essential for the subsequent training and evaluation of models.

B.3 Data Statistic and Analysis

We also investigated the annotated topics in the GenSurvey dataset as summarized in Figure 5. The chart shows the frequency of different topics, with "Machine Learning" being the most frequent, appearing 174 times, followed by "Cryptography and Security" with 90 times, and "Computer Vision and Pattern Recognition" with 89 times. Topics such as "Distributed, Parallel, and Cluster Computing" with 65 times, and "Artificial Intelligence" with 65 times also appear frequently. In contrast, more specialized topics, such as "Physics and Society"

Key	Value
title	A formula for a quartic integral: a survey of old proofs and some new ones
article_id	arXiv:0707.2118
subject	["Classical Analysis and ODEs"]
abstract	We discuss several existing proofs of the value of a quar_x0002_tic integral and present a new proof that evolved from rational Landen ...
content	1. Introduction The evaluation of definite integrals has attracted the scientific community, both professional and amateurs, for a long ...
reference	[[1] B. Berndt. Ramanujan's Notebooks, Part I. Springer_x0002_Verlag, New York, 1985., [2] G. Boros and V. Moll. An integral hidden in Gradshteyn and Ryzhik. Jour. Comp.Applied Math., 106:361–368, 1999., ...]
reference_content	[{ reference_num: [2], reference_title: An integral hid_x0002_den in Gradshteyn and Ryzhik, reference_abstract: We provide a closed-form expression for the integral ... }, ...]

Table 7: An example of GenSurvey data.

Dataset	Task	Source Type	Output Length	Human-written?
AutoSurvey (Wang et al., 2024b)	Survey generation	Titles + metadata	∼10,000 words	✗
SciFact (Wadden et al., 2020)	Factual verification	Abstracts + claims	∼300 words	✓
GovReport (Huang et al., 2021)	Report summarization	Gov. documents	∼1,600 words	✓
LongBench (Bai et al., 2024)	Multitask long-context eval	QAs, code, etc.	>6,700 words	Mix
LongEval (Wu et al., 2025)	Long-form generation (plan-based)	arXiv, Wiki, Blogs	2,500–5,000 words	✓
GenSurvey (Ours)	Survey generation	Abstracts abd Topics	>10,000 words	✓
GenSection (Ours)	Section-level survey writing	Abstracts abd Topics	∼4000 words	Mix

Table 8: Comparison of GenSurvey and GenSection with prior long-form generation datasets.

and "Multimedia", appear less often, with only 11 instances each. This distribution suggests that the dataset is heavily focused on topics related to computer science and technology, particularly machine learning, security, and computer vision, while other scientific disciplines are less represented.

Comparison with Existing Datasets Table 8 presents a comparative analysis of our proposed datasets with existing benchmarks used for long-context or long-form generation. AutoSurvey (Wang et al., 2024b) targets survey creation using titles and metadata, generating outputs leveraging multiple LLMs. SciFact (Wadden et al., 2020) and GovReport (Huang et al., 2021) focus on compact summaries derived from scientific or governmental texts, but their outputs are relatively brief. LongBench (Bai et al., 2024) offers a multitask evaluation framework for long-context tasks such as QA, summarization, and code interpretation, resulting in mostly short outputs not tailored for complete document generation. LongEval (Wu et al., 2025), a new standard for evaluating long-text generation on arXiv, Wikipedia, and blogs, employs direct and plan-based models. It emphasizes structured generation and domain scoring (e.g., methodology, experimental details), with moderate document lengths (2,500–5,000 words), aimed at general long-form content rather than scientific text. In contrast, **GenSurvey** and **GenSection** uniquely address long-to-long scientific text generation. GenSurvey offers human-written, comprehensive scientific surveys exceeding 10,000 words. GenSection supplements this with over 4,000 section-level instances, providing a modular, instruction-tuned framework validated by human evaluation.

B.4 Annotator payment

Each annotator receives 2 USD for collecting one survey paper and its corresponding references abstracts. For topic annotation and cross-validation, each annotator receives 0.2 USD per survey paper.

C GenSection Details

We employ two Master students to validate the generated text based on two criteria: **Relevance** and **Structure**, using the provided abstracts and topics.

Relevance. Determine whether the content of the generated section is meaningfully related to the provided reference abstracts and topic. They need to annotate one of the following labels:

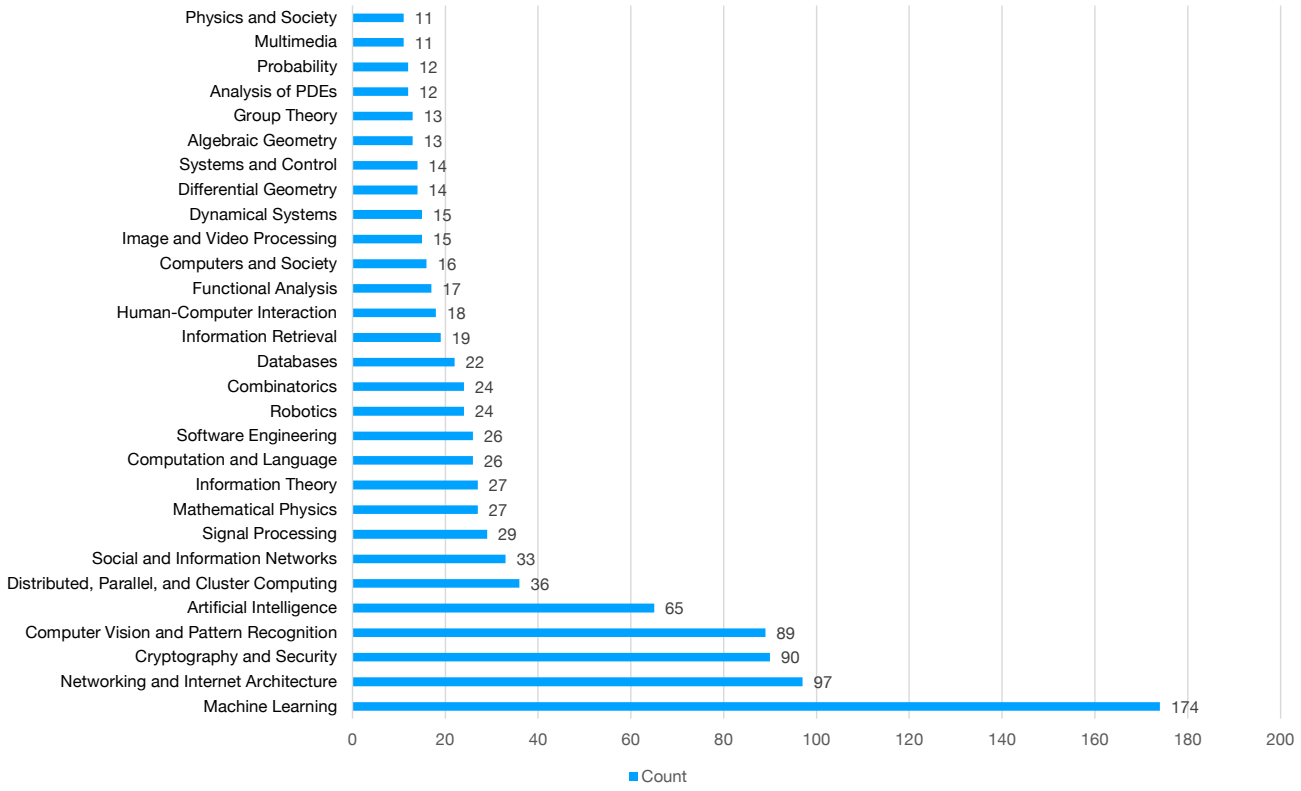

Figure 5: Topic Distribution of GenSurvey dataset.

1. **Relevant**: The section clearly and accurately reflects the content of the input abstracts and topic.

2. **Not Relevant**: The section is unrelated or only marginally connected to the provided abstracts and topic.

3. **Not Enough Information (NEI)**: The input abstracts and topic do not provide sufficient detail to make a confident judgment.

Structure. Evaluate how well the section is organized and presented in terms of scientific writing. The annotators select one of the following labels:

1. **Good Structure**: The section has a clear and coherent flow, follows academic writing conventions, and is logically structured.

2. **Moderate Structure**: The section is generally well-formed but may contain some organizational issues or inconsistencies.

3. **Bad Structure**: The section lacks logical flow, contains disorganized content, or does not resemble a well-written scientific section.

We then calculate Cohen's Kappa (κ) (McHugh, 2012), which is commonly used to measure the inter-rater agreement between two raters. Cohen's Kappa (κ) is calculated as:

$$\kappa = \frac{P_o - P_e}{1 - P_e}$$

In this formula, P_o is the proportion of times the two raters agree, and P_e is the probability of agreement by chance. The coefficients between the two raters are $\kappa_{\text{Relevant}} = 0.79$ and $\kappa_{\text{Structure}} = 0.56$. According to (Landis and Koch, 1977), $0.61 < \kappa_{\text{Relevant}} < 0.80$ indicates substantial agreement, while $0.41 < \kappa_{\text{Structure}} < 0.60$ indicates moderate agreement. Additionally, we observe that for the marking of Relevance and Structure, the two raters have 70% agreement on *Relevant* and 59% agreement on *Good Structure*, respectively.

Annotator payment Each rater receives 0.2 USD for each section rated.

D Experimental Settings

D.1 Implementation Details

Each training sample in the GenSurvey dataset is formalized as a triplet (I, X, Y), where:

- I is a simple instruction guiding the model to generate the full survey (e.g., "Write the survey using the following abstracts").

- $X = \{a_1, a_2, \ldots, a_n\}$ is the set of abstracts from n reference papers, where each a_i is a short text representing the abstract of one reference paper.

- Y is the corresponding full survey text written by a human, composed based on the information synthesized from the abstracts in X.

The training objective is to learn a function $f_\theta : X \rightarrow Y$, parameterized by θ, such that the model generates a coherent and comprehensive survey $\hat{Y} = f_\theta(X)$ given the set of abstracts X, and $\hat{Y} \approx Y$.

Meanwhile, each training instance in the GenSection dataset is represented as a triplet (I, X, Y'), where:

- I is the instruction that specifies the goal of the section to be generated (e.g., "Write the introduction section for a survey on Graph Neural Networks using the following abstract").

- $X = \{a_1, a_2, \ldots, a_n\}$ is the set of abstracts corresponding to the reference papers for the survey.

- Y' is GPT-generated content for the section described in I.

The model is trained to learn a function $f_\theta : (I, X) \rightarrow Y'$, where f_θ maps the instruction and reference abstracts to the desired section output. During evaluation, multiple predictions $\hat{Y'}_1, \hat{Y'}_2, \ldots, \hat{Y'}_m$ are generated for sections belonging to the same survey (identified by a shared file ID), and concatenated to form the full survey prediction:

$$\hat{Y'}_{\text{survey}} = \texttt{Concat}(\hat{Y'}_1, \hat{Y'}_2, \ldots, \hat{Y'}_m)$$

where `Concat` denotes the sequential concatenation operation based on the order of the sections.

Handling Long Inputs. In generating full documents on GenSurvey, we input the entire set of reference abstracts. To remain within model context limits, we choose models that handle lengthy inputs without truncation. We use LLaMA 3.1 models (1B, 3B, 8B), which support up to 128K tokens[5], Qwen2 models (1B, 3B, 7B) with a capacity of up to 131K tokens[6], and Mistral 7B and Mixtral 8x7B, which handle 32K tokens[7]. Additionally, we use the DeepSeek R1 Distilled LLaMA 70B, accommodating up to 128K tokens[8]. In GenSection, section-level generation inputs are about to 4,000 tokens, suiting models of different capacities. This framework permits evaluation within models' supported architectures without input trimming or heuristic adjustments.

D.2 Training Parameters

In our experiments, we utilize several advanced techniques to optimize model performance and efficiency. We apply LoRA (Hu et al., 2022), specifically using a rank of 256, to reduce the number of trainable parameters while maintaining model performance. This method allows us to fine-tune large models with fewer computational resources. We conduct our experiment using LLaMa-Factory (Zheng et al., 2024) library.

We use DeepSpeed (Aminabadi et al., 2022) with ZeRO Stage 3 (z3) to optimize memory and computational efficiency during training. Key settings include automatic adjustments for *train_batch_size* and *train_micro_batch_size_per_gpu* based on available resources, and gradient accumulation with *gradient_accumulation_steps* set to "auto." Loss scaling is dynamically managed with an initial scale of 0, a scale window of 1000, and an initial scale power of 16. Stage 3 enables efficient parameter management with settings like *contiguous_gradients* and *stage3_max_live_parameters* set to large values for improved memory allocation. Additional configurations, such as *overlap_comm* set to false and *stage3_gather_16bit_weights_on_model_save*, further enhance training efficiency and model storage. This setup ensures optimal performance while minimizing memory usage.

Furthermore, we use FlashAttention 2 (Dao, 2024) with *bfloat16* (BF16) precision for efficient memory utilization and optimized computation during model training. By default, FlashAttention

[5] https://console.groq.com/docs/model/llama-3.1-8b-instant

[6] https://huggingface.co/docs/transformers/main/en/model_doc/qwen2

[7] https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2

[8] https://console.groq.com/docs/model/deepseek-r1-distill-llama-70b

2 operates in *flash* mode with contiguous memory layouts, reducing fragmentation and improving memory access. It automatically adapts to varying batch sizes and sequence lengths, ensuring efficient use of available GPU memory. The default settings focus on mixed-precision training, maximizing throughput while maintaining computational efficiency. FlashAttention 2 is designed to fully leverage GPU capabilities, minimizing latency for attention operations, and is particularly well suited for processing long sequences without exceeding memory limits.

D.3 Model List

Below is the pre-trained models used in our experiments:

https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Llama-70B
https://huggingface.co/meta-llama/Llama-3.2-1B-Instruct
https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct
https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct
https://huggingface.co/Qwen/Qwen2-1.5B-Instruct
https://huggingface.co/Qwen/Qwen2.5-3B-Instruct
https://huggingface.co/Qwen/Qwen2.5-7B-Instruct
https://huggingface.co/Qwen/Qwen2.5-72B-Instruct
https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2
https://huggingface.co/mistralai/Mixtral-8x7B-Instruct-v0.1

D.4 Metrics

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) (Lin, 2004) score is a set of metrics used to evaluate the quality of summaries by comparing n-gram overlaps between the generated text and reference text. The most commonly used ROUGE metric is ROUGE-N, which measures the overlap of n-grams (typically unigrams or bigrams) between the generated and reference text. The ROUGE-N score is computed as follows:

$$\text{ROUGE-N} = \frac{\sum_{n\text{-gram} \in \text{Generated}} \text{Count}_{\text{match}}(n\text{-gram})}{\sum_{n\text{-gram} \in \text{Reference}} \text{Count}(n\text{-gram})} \tag{1}$$

Where $\text{Count}_{match}(n-gram)$ refers to the number of matching n-grams between the generated and reference texts. $\text{Count}(n-gram)$ refers to the total number of n-grams in the reference text.

ROUGE scores are typically reported for multiple n-gram sizes (ROUGE-1 for unigrams, ROUGE-2 for bigrams) and can be extended to measure recall, precision, and F1-score. In out experiment, we use F1-score for ROUGE.

BERTScore (Zhang et al., 2020) is a metric used to evaluate the quality of text generation by comparing the contextual similarity between the generated text and a reference text. It leverages BERT embeddings to measure semantic similarity at the token level, rather than relying on exact n-gram matching like traditional metrics such as ROUGE. The BERTScore is calculated by computing cosine similarity between token embeddings from the generated and reference texts. The formula for BERTScore is as follows:

$$\text{BERTScore} = \frac{1}{|T_{\text{generated}}|} \sum_{t \in T_{\text{generated}}} \max_{r \in T_{\text{reference}}} \text{cosine\_sim}(t, r) \tag{2}$$

Where $T_{\text{generated}}$ and $T_{\text{reference}}$ are the token sets from the generated and reference texts, respectively, and $\text{cosine\_sim}(t, r)$ is the cosine similarity between token embeddings t and r. This approach ensures that BERTScore captures contextual and semantic relationships between words, making it more suitable for tasks like document summarization and translation.

Soft Heading Recall is calculated based on the Soft Recall (Fränti and Mariescu-Istodor, 2023). Soft Heading Recall (S-H Recall) evaluates the structural alignment between the generated and reference survey. It measures the similarity between the generated and reference chapter titles while penalizing the similarity of titles within the generated survey itself. The formula for S-H Recall is defined as follows:

$$\text{Sim}(t_i, t_j) = \cos(\text{embed}(t_i), \text{embed}(t_j))$$

Where t_i and t_j represent section titles from the generated and reference surveys, respectively, and $\text{embed}(t_i)$ and $\text{embed}(t_j)$ are their corresponding embeddings.

The total number of chapters is denoted by $|T|$, and the formula for calculating S-H Recall is:

Model	Learning Rate	Epochs	Batch Size	LoRA Rank	Dropout
Deepseek R1 Distilled 70B	1e-05	2	32k	256	0.0
LLaMa3 1B	5e-05	6	32k	256	0.0
LLaMa3 3B	3e-05	6	32k	256	0.1
LLaMa3 8B	3e-05	6	32k	256	0.1
Mistral 7B	3e-05	6	32k	256	0.1
Mixtral 8x7B	2e-05	6	32k	256	0.1
Qwen2 1.5B	5e-05	6	32k	256	0.0
Qwen2 3B	3e-05	6	32k	256	0.1
Qwen2 7B	3e-05	6	32k	256	0.1
Qwen2 72B	1e-05	6	32k	256	0.0

Table 9: Training hyperparameters for each LLM used in our experiments.

$$\text{card}(T) = \sum_{i=1}^{|T|} \frac{1}{\sum_{j=1}^{|T|} \text{Sim}(t_i, t_j)}$$

$$\text{card}(R \cap G) = \text{card}(R) + \text{card}(G) - \text{card}(R \cup G)$$

$$\text{S-H Recall} = \frac{\text{card}(R \cap G)}{\text{card}(R)}$$

Where R and G represent the set of section titles in the reference and generated survey, respectively, and card$(\cdot)$ refers to the cardinality (or total number of titles). This score encourages the alignment of sections titles while punishing the generation of redundant titles within the same survey.

E Ablation Study Details

E.1 Human Evaluation Design

To perform human evaluation, we create a simple rating application using Streamlit[9] library. In this application, we provide 100 survey files with both generated text and human reference text. The raters can upload the file to the application and read the content as illustrated in Figure 6. After they read the files, they can rate the generated text on three designated aspects: *Relatedness*, *Readability*, and *Citation Capture* as illustrated in Figure 7. The questions for each aspect are described as follows:

How well the generated text match the reference text?

1. **No relation at all**: The generated text is completely unrelated to the reference.
2. **Minimal relation**: The generated text shares only a vague or indirect connection with the reference.
3. **Somewhat related**: The generated text covers similar themes or general topics but differs significantly in content or focus.
4. **Sufficiently related**: The generated text closely follows the reference text in meaning and key content. Captures most of the important ideas
5. **Extremely related**: The generated text is semantically equivalent to the reference text. Captures all key ideas accurately.

How well is the generated text in terms of fluency and coherence?

1. **Not fluent or coherent at all**: The text is grammatically incorrect, fragmented, or unreadable.
2. **Minimal fluency and coherence**: Some parts are readable, but major grammar issues exist.
3. **Moderately fluent and coherent**: Sentences are generally well-formed but may contain minor grammar or structure issues. Understandable, but lacks smooth flow.
4. **Mostly fluent and coherent**: The text is clear and mostly free of grammar or syntax errors. Slight awkwardness may exist but does not hinder comprehension.
5. **Perfectly fluent and coherent**: The text reads naturally, like it was written by a native speaker or professional writer. Fully grammatical, cohesive, and well-structured.

[9] https://streamlit.io/

How well the generated text include citations?

1. **No citation at all**: The generated text includes no citations.
2. **Minimal citation**: Contains only 1 or 2 citations, or the citations appear incomplete or misplaced.
3. **Some citations present**: Several key points are supported by citations, but others are missing or inconsistently cited. Citation formatting may be inconsistent but recognizable.
4. **Well-cited**: Most major claims and sections include relevant citations with only minor omissions.
5. **Fully and properly cited**: All claims that require support are backed by appropriate citations.

Annotator payment Each rater receives 1.5 USD for each survey rated.

E.2 Model cost-effectiveness

To evaluate the cost-effectiveness of different models, we compute the *Effectiveness* score E, which quantifies how much performance a model achieves per unit of training time. It is calculated as the ratio between the overall *Performance* score P and the training time T (in hours). The performance score P is defined as the average of M evaluation metrics, such as ROUGE, BERTScore, and S-H Recall. To calculate P correctly, all metric scores must be normalized to the range $[0, 1]$, where 1 is the highest possible score.

$$P = \frac{1}{M} \sum_{i=1}^{M} \text{score}_i \tag{3}$$

$$E = \frac{P}{T} \tag{4}$$

Where score_i denotes the i^{th} metric score, M is the total number of metrics, and T is the training time in hours.

Our results in Table 10 show that while larger models such as Qwen-72B and DeepSeek-Distilled-70B achieve relatively high performance scores (P), their long training times result in low effectiveness. In contrast, LLaMa-1B yields the highest effectiveness score of 2.253, making it the most efficient model in terms of performance per unit training time. Qwen-1.5B also shows a strong balance with the second-highest effectiveness. Interestingly, although Mistral-7B achieves the best performance score (0.665), its higher training cost reduces its effectiveness to 0.189. This analysis highlights that smaller models like LLaMa-1B can offer strong trade-offs between quality and efficiency, which is essential for resource-constrained settings.

E.3 DPO Implementation

To enable DPO (Rafailov et al., 2023) training, we construct a preference dataset derived from our original GenSurvey data. For each training instance, we create a pair of responses: a *chosen* output that represents a high-quality, human-aligned survey generation, and a *rejected* output that represents a lower-quality or less-preferred alternative. The *instruction* field contains the original generation prompt, and the *input* field optionally includes the set of abstracts as supporting context. The *chosen* response is typically either the original human-written survey or a model output refined through human preference. The *rejected* response is generated using zero-shot prompting on the Qwen2 1.5B model which produces the worst generation.

Each entry in the dataset thus follows the format:

```
{
  "instruction": "<prompt>",
  "input": "<abstracts>",
  "chosen": "<preferred survey text>",
 "rejected": "<less preferred survey text>"
}
```

This format aligns with the standard input required by DPO training frameworks, enabling the model to learn direct preferences without explicit reward modeling. Following best practices from prior work (Rafailov et al., 2023; Zheng et al., 2023), our preference pairs emphasize meaningful semantic differences in relevance, fluency, and structure to ensure effective preference-based learning. In our experiments, we only implement DPO on LLaMa3 1B, Mistral 7B and DeepSeek 70B and in two settings: (i) directly training models with DPO from their pre-trained checkpoints, and (ii) further training LoRA-fine-tuned models with DPO. For both cases, we set `pref_beta` to 0.1 and use the `sigmoid` preference loss function.

E.4 Variants of LoRA Implementation

Quantized Low-Rank Adaptation (QLoRA) (Dettmers et al., 2023) enhances the efficiency of model fine-tuning by combining low-rank

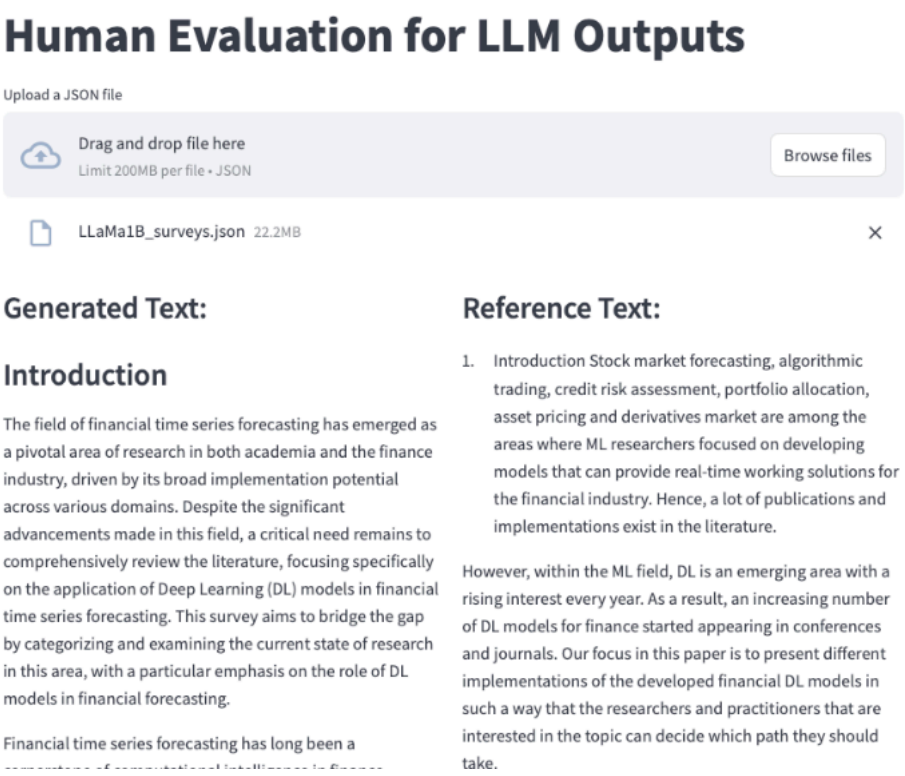

Figure 6: GUI displaying generated and human reference for evaluation.

adaptation with 4-bit quantization of the base model weights. In this approach, the pre-trained model remains frozen and quantized, while gradients are propagated only through lightweight low-rank adapter layers. In our experiments, we set `quantization_bit` to 4 and `quantization_method` to *bitsandbytes*.

Directional and Magnitude Decomposed LoRA (DoRA) (Mao et al., 2024) reformulates LoRA by separating the adaptation process into two orthogonal components: weight direction and magnitude. Unlike standard LoRA, which applies low-rank updates to the full weight matrix, DoRA modifies only the directional component while keeping the original magnitude fixed. This decoupling enables for more targeted updates, which improves the generalization of the model and the stability of the training.

Principal Singular Value and Vector Adaptation (PiSSA) (Meng et al., 2024) improves LoRA by initializing its low-rank adapters based on the principal components of the pre-trained weights. Specifically, PiSSA updates only the most informative subspaces, determined by singular value decomposition, which better approximates full fine-tuning behavior with fewer parameters. This method achieves performance superior to that of standard LoRA on several benchmarks. We set `pissa_iter` to 16 in our experiments.

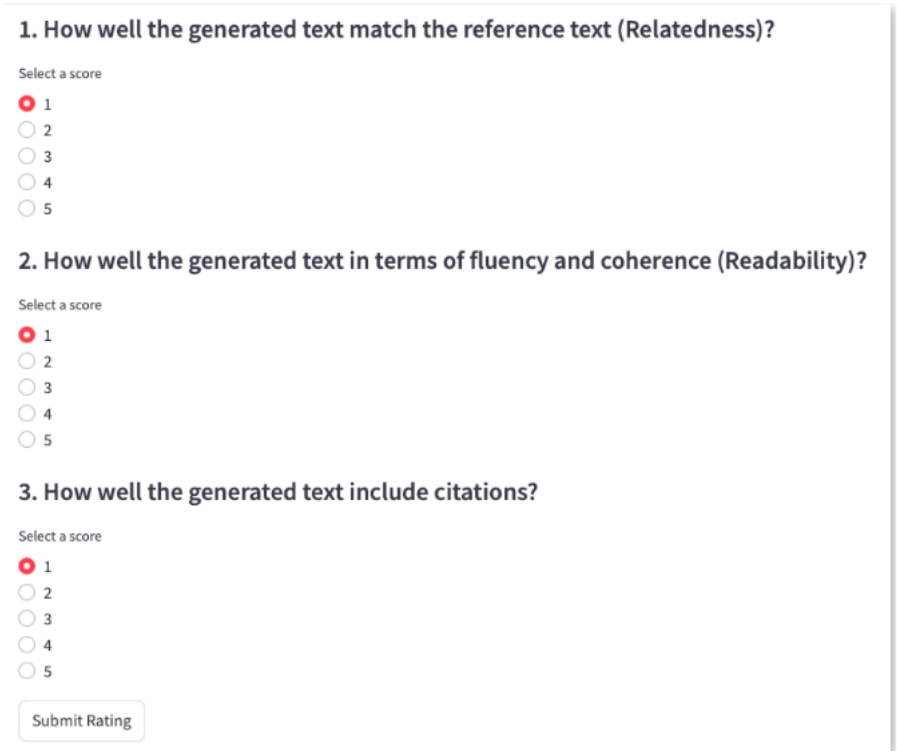

Figure 7: GUI displaying rating for each aspect.

Model	**Training time (T) (in hour)** ↓	**Performance score (P) = avg. of scores** ↑	**Effectiveness (E) = P/T (P per hour)** ↑
DeepSeek R1 Distilled 70B	18.000	0.574	0.032
LLaMa3 1B	0.267	0.601	**2.253**
LLaMa3 3B	1.000	0.632	0.642
LLaMa3 8B	3.717	<u>0.644</u>	0.173
Mistral 7B	3.517	**0.665**	0.189
Mixtral 8x7B	5.900	0.557	0.094
Qwen2 1.5B	0.733	0.502	<u>0.684</u>
Qwen2 3B	2.033	0.494	0.243
Qwen2 7B	3.367	0.620	0.184
Qwen2 72B	23.067	0.624	0.027

Table 10: Cost-effectiveness analysis of various models.

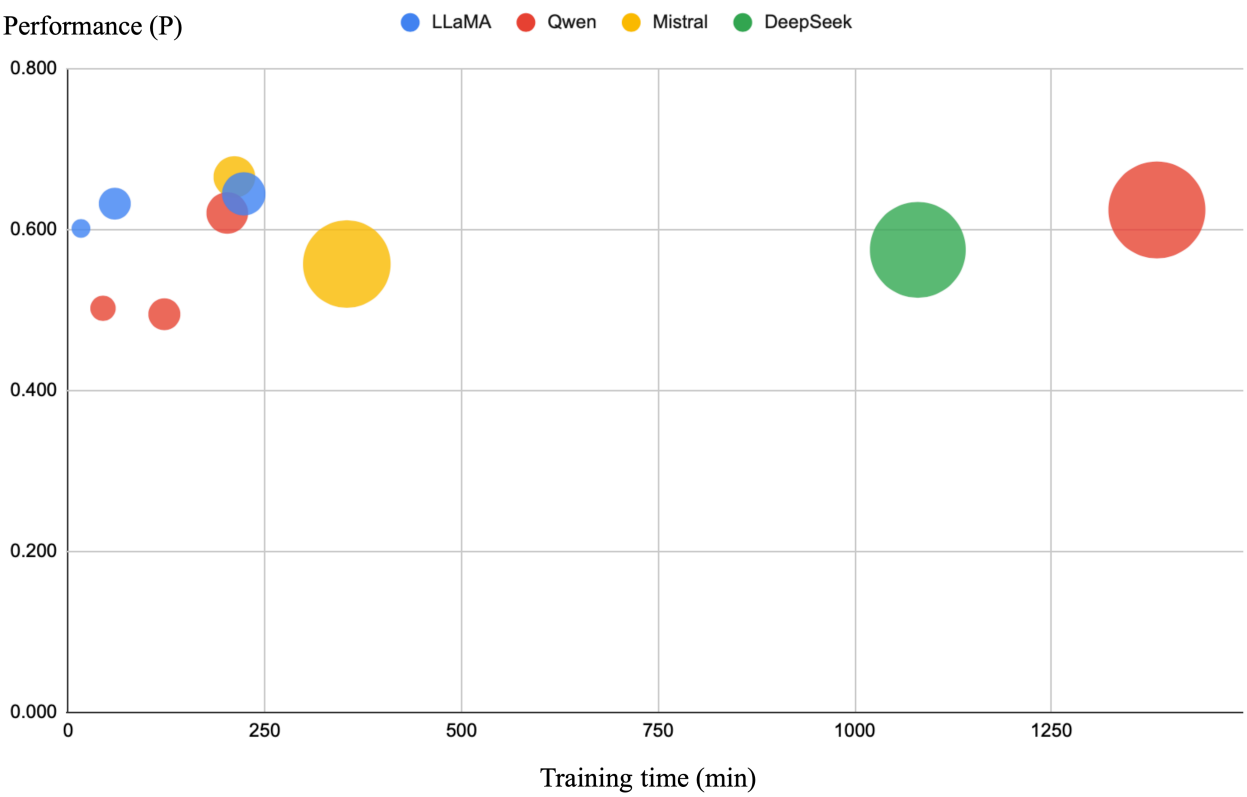

Figure 8: Performance of models compared to their training time. The sizes of the bubble indicates their numbers of parameters.

F Prompt Template

Prompt for generating section titles You are tasked with generating section titles for a survey paper based on the following abstracts and topics.
Instructions:
1. Read through all the abstracts provided.
2. Provide a list of relevant section titles that could be used in the body part of the survey paper.
3. Remember to exclude the Abstract, Introduction and Conclusion sections.
4. Each section title should be a short, descriptive phrase (2–5 words).
5. Use title case for section titles (capitalize the first letter of each major word).
6. Do not use numbering for the titles.
7. The output should be a list of 2 to 5 section titles excluding any content.
8. Provide the section titles in the form of a string separated using commas.
Topics: *{Annotated topics}*
Abstracts: *{List of abstracts}*
Now, generate relevant section titles for the body part of the survey paper.

Prompt for generating Introduction section You are tasked with writing the introduction section of a survey paper using the following abstracts and topics..
Instructions:
1. Read through all the abstracts provided.
2. Synthesize the information from the abstracts to create the Introduction section, ensuring it reflects the context to the topic, the problem, and the purpose of the survey.
3. Ensure the section flows logically and cohesively, even if the number of abstracts varies.
4. Paraphrase and perspective shift the text to avoid direct copying from the abstracts.
5. Use placeholder citations to refer to specific abstracts where relevant.
6. The output can have many paragraph but it should only have one section.

Topics: *{Annotated topics}*
Abstracts: *{List of abstracts}*
Now, generate the Introduction section based on the provided instructions.

Prompt for generating Conclusion section You are tasked with writing the conclusion section of a survey paper using the following abstracts and topics.
Instructions:
1. Read through all the abstracts provided.
2. Synthesize the key insights and findings presented in the abstract to generate conclusion section for the survey paper.
3. The conclusion should recap the main themes, challenges, and contributions from the abstracts.
4. Be concise, while reflecting the overall content of the survey paper.
5. Paraphrase and perspective shift the text to avoid direct copying from the abstracts.
6. The output should only have one paragraph.
Topics: *{Annotated topics}*
Abstracts: *{List of abstracts}*
Now, generate the Conclusion section based on the provided instructions.

Prompt for generating other sections You are tasked with writing the *{Section_titles}* section of a survey paper using the following abstracts and topics..
Instructions:
1. Read through all the abstracts provided.
2. Synthesize the information related to the section theme, ensuring the content reflects relevant topics discussed in the abstracts.
3. Maintain logical flow and coherence within the section.
4. Paraphrase and perspective shift the content from the abstracts.
5. Use placeholder citations where relevant.
Topics: *{Annotated topics}*
Abstracts: *{List of abstracts}*
Now, generate the *{Section_titles}* section based on the provided instructions.

Prompt for generating survey in Section 6.5 You are tasked with writing survey paper using the following abstracts and topics.
Instructions:
1. Read through all the abstracts provided.
2. Synthesize the related information, ensuring the content reflects relevant topics discussed in the abstracts.
3. Maintain logical flow and coherence within the paper.
4. Paraphrase and perspective shift the content from the abstracts.
5. Use placeholder citations where relevant.
Topics: *{Annotated topics}*
Abstracts: *{List of abstracts}*
Now, generate the survey paper based on the provided instructions.