DocAssistant: Integrating Key-region Reading and Step-wise Reasoning for Robust Document Visual Question Answering

Jinxu Zhang, Qiyuan Fan, Yu Zhang†
Harbin Institute of Technology
{jxzhang, qyfan, zhangyu}@ir.hit.edu.cn

Abstract

Understanding multimodal documents is essential for accurately extracting relevant evidence and using it for reasoning. Existing document understanding models struggle to focus on key information and tend to generate answers straightforwardly, ignoring evidence from source documents and lacking interpretability. In this work, we improve the visual encoder to focus on key information relevant to the question and address the shortcomings of existing document visual question-answering datasets to provide the model with the ability to answer questions step-wise, dubbed DocAssistant. Specifically, for the visual side, we propose an effective vision-language adaptation that fuses text into visual encoders without compromising the performance of the original model. For the language side, we use Multimodal Large Language Models (MLLMs) as data generators and checkers to produce high-quality step-wise question-and-answer pairs for document images. Then the generated high-quality data is used to train our enhanced model, specifically designed to solve complex questions that require reasoning or multi-hop question answering. The experimental results demonstrate the effectiveness of the model.

1 Introduction

[1] There are various documents in the real world, which differ from images of real-world scenarios. Document images are often filled with extensive text and graph information, requiring the model to have strong capabilities in document layout understanding, text semantic understanding, and numerical reasoning. In the document visual question answering (DVQA) task, as shown in Figure 1, existing document understanding models (Xu et al., 2020b,a; Davis et al., 2022) tend to generate wrong answers directly based on complex questions, making it harder to determine the source of the answers and their accuracy.

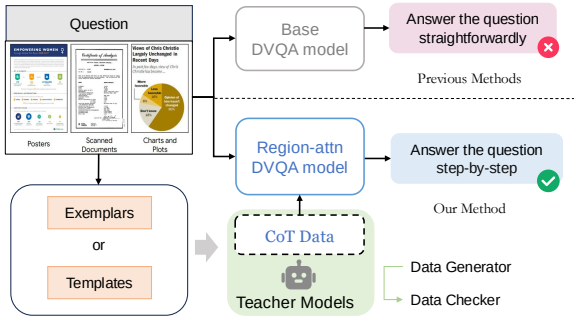


Figure 1: Existing DVQA models tend to mislocate information related to the question and generate a word or phrase directly as an answer. We modify existing vision encoders to focus on regions of the document image associated with the keywords in the question and generate high-quality data with intermediate results using a strong MLLM. These augmented and extended data are then used to enhance a small-scale multi-modal model, achieving an effective step-wise document understanding and reasoning model.

Recently, a number of MLLMs have made breakthroughs in document understanding, such as PaLI-3 (Chen et al., 2023), InternVL (Chen et al., 2024b), LlaVA-UHD (Xu et al., 2024b), and mPLUG-DocOwl (Hu et al., 2024). By improving image resolution and understanding documents from both macro and micro perspectives, these models have greatly enhanced the ability of visual encoders to understand fine-grained information in documents. Most of them have achieved promising performance on document understanding datasets like DocVQA (Mathew et al., 2021), ChartQA (Masry et al., 2022), InfographicVQA (Mathew et al., 2022), etc. However, they can only handle simple questions in documents. When faced with complex layout documents, they are likely to make errors in information identification. Additionally, when dealing with complex questions, they tend to ignore the reasoning steps and directly produce answers without providing the basis for answers.

[1] †Corresponding Author

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 3496–3511
November 4-9, 2025 ©2025 Association for Computational Linguistics

At present, existing multi-modal document datasets are gradually improving, encompassing a range from scanned documents in DocVQA to complex-layout poster documents in InfographicVQA, as well as chart documents involving mathematical logic operations in ChartQA. This progression allows for a more comprehensive assessment of the MLLMs' ability to understand different types of documents. However, most of the questions are relatively simple, focusing primarily on document information recognition, with fewer questions requiring reasoning. Additionally, the answers do not provide contextual information, making it difficult to intuitively confirm their correctness.

In this paper, we aim to develop an effective and general document understanding model. When answering questions, the model is designed to read and think like a human: first, by locating the contextual information related to the keywords in the question, and then deducing the corresponding answer based on this context. On the visual side, we propose a lightweight Mixture-of-Modality Adaptation module, which is inserted into the last L-layers of the visual encoder. This enables the original model to focus on the areas related to the question keywords without affecting its performance, and prevents information recognition errors. On the language side, as no such data currently exists, we enhance the existing DVQA data by treating the answer as the supervisory signal, enabling large-scale MLLMs to perform the reasoning process from question to answer using our designed templates. Since the generated data may contain noise and hallucinations, we have designed a pipeline based on multi-agent collaborative filtering and rule-based techniques to filter out the noisy data. Finally, the filtered data is used to train the improved model on the visual side, enhancing its proficiency in document understanding and reasoning.

To summarize, our primary contributions include:

- We use MLLMs to design a data generator based on templates and few-shots, and a multi-agent-based data filter to augment and extend high-quality, step-wise DVQA data.
- We propose a Mixture-of-Modality adaptation, which aims to improve the visual encoder of the existing document understanding model to focus on areas related to the question keywords and prevent noise information.
- We trained an efficient SLVM, dubbed DocAssistant, with both question-aware document visual context understanding and reasoning.
- We achieved robust results on complex layout document understanding and reasoning datasets, and provided more extensive experiments and analysis to validate the superiority.

2 Related Work

2.1 Visually Rich Document Understanding

The VRDU task is designed to provide a document image and a question, requiring the model to answer the question by understanding the text, images, and layout of the document. Existing document understanding models fall into two categories: OCR-based and end-to-end models that do not rely on external tools. The former includes models like LayoutLMv3 (Huang et al., 2022) and DocFormerv2 (Appalaraju et al., 2024), which use OCR to extract text and corresponding coordinate information and design pre-training tasks based on text, layout, and image information to understand documents. The latter includes simpler architectures such as Donut (Kim et al., 2022) and Pix2Struct (Lee et al., 2023), which use only a visual encoder and a text decoder to improve document understanding through pseudo-OCR tasks or pre-training tasks like image masking. With the rise of MLLMs, some models (Chen et al., 2023; Hu et al., 2024; Chen et al., 2024a) involving DVQA tasks have achieved good results in simple questions, such as document information extraction with simple layouts.

However, the aforementioned models perform suboptimally for documents with complex layouts. To address this issue, we propose a lightweight multimodal adaptation strategy that integrates multimodal information, ensuring the model's original performance while focusing on the visual regions relevant to the question, thereby enhancing the accuracy of its responses.

2.2 Reasoning step-by-step

For LLMs, chain-of-thought prompting has been found to be a simple and effective method to improve reasoning performance, often used to tackle complex tasks. For example, SymbCoT (Xu et al., 2024a) uses LLMs to implement a system with translation, planning, solving, and verification, following a logical reasoning framework to maximize LLMs' ability to stimulate the chain of thought.

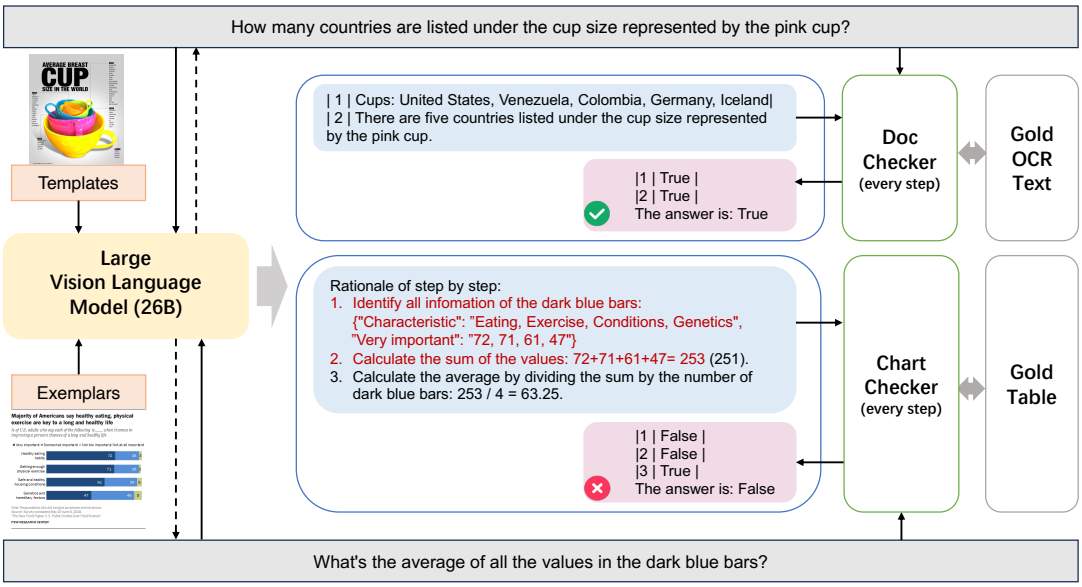


Figure 2: Data generator based on multi-agent interaction. Dotted arrows indicate the extended data including question generation. The data checker uses OCR text from ordinary documents and tabular information equivalent to charts. First, it checks for identification errors in the generated data. Second, it checks for errors in the intermediate steps of reasoning. If any errors are found, the data is considered unqualified.

In the multimodal field, recent works such as VisProg (Gupta and Kembhavi, 2023) and ViperGPT (Surís et al., 2023) utilize LLMs as planners to compose domain-specific models (Lu et al., 2024; Yang et al., 2023; Zeng et al., 2022) to solve complex tasks. There are also some methods (Zhang et al.; Li et al., 2024; Wang et al., 2024a) using augmented data that have emerged to improve model performance, but these methods rely on data with special annotations, and their data generation methods are limited to a single field, which is not widely applicable. Moreover, since they use small models trained in a specific field, requiring large-scale training data. Furthermore, the improvement of complex questions is also limited.

In the field of document VQA, there is a lack of data and multimodal models capable of reasoning about document images. We employ a comprehensive approach that includes designing templates and exemplars, using tools, etc. This approach involves teaching a smaller model by learning from a teacher model to fill data gaps.

3 Method

Given a document image and a question, the goal is to enable the model to process and reason similarly to a human. First, we design a data generator and a data checker to create high-quality data. Next, we enhance the vision encoder by introducing a Mixture-of-Modality adaptation mechanism, which directs the model to focus on the visual regions relevant to the question when processing the document image. Finally, these generated data are then utilized to train the multimodal DVQA model we have developed.

3.1 Data Construction

Data Generator. The overview of our MLLM-based data generator and checker is shown in Figure 2. We leverage InternVL2-26B (Chen et al., 2024a) as the MLLM for data generation and the LLaMA3-70B (AI@Meta, 2024) for data check. Data generation includes creating data based on the existing training set and generating triplets (question, rationale, answer) through our designed templates, which can be found in the Appendix. During the generation process based on the existing training set, we generate a raw related rationale R' using the template P, image I, question Q, and answer A as inputs:

$$R' = F_g(P, I, Q, A) \quad (1)$$

During the triplet generation process, we design a question generation template, as shown in Table 10 and Table 11 of the Appendix. In particular, we use templates for extractive or abstractive documents and use few-shot learning for chart documents:

$$(Q, R', A) = F_g(P, I) \quad (2)$$

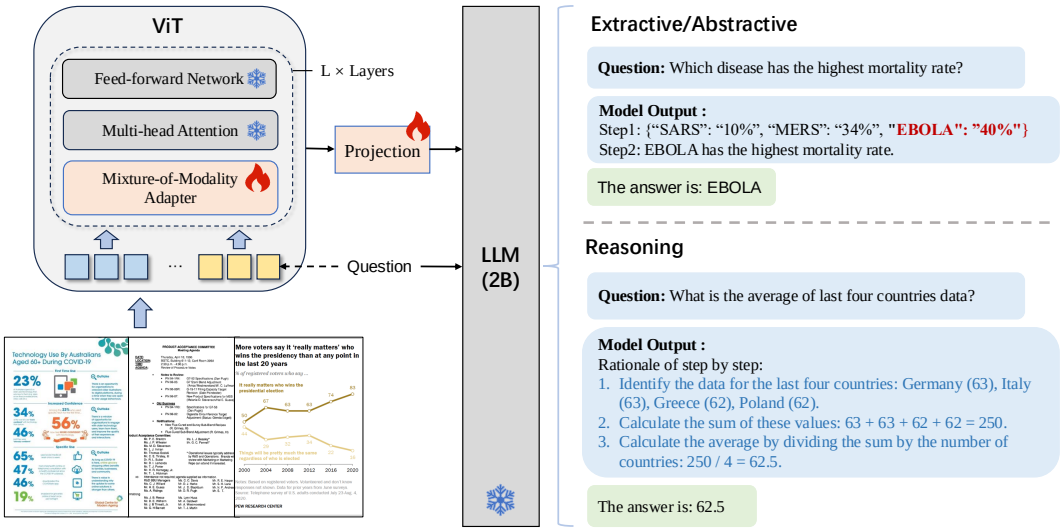


Figure 3: Model overview. The document image concerned by key regions, projected by projection layers concatenated with a question, is fed into the LLM for step-wise generation. Answer generation for the extractive and abstractive types consists of two steps: the first step generates the context relevant to the keywords in the question, and the second step generates the corresponding answer based on the context. For the reasoning type of answer generation, the steps depend on the question type and vary with the complexity of the question, using exemplars as the prompt.

Data Checker. During data inspection, we use tools to extract text information from documents, specifically using OCR tools for general documents and DePlot (Liu et al., 2022) to convert charts into tables, using these contexts as the ground labels (C) of the checker. By designing error detection templates shown in Table 13 in the Appendix, the LLM can determine whether the corresponding logical step is correct based on the question and the gold text of the image. This includes (1) the extraction of information step from the document itself and (2) the reasoning steps based on the extracted information. If any judgment errors are made, it will return False and delete the piece of data:

$$R = F_c(P, C, A) \tag{3}$$

The data generation pipeline we designed can be easily applied to other document image datasets, demonstrating strong universality.

3.2 Model Architecture

The architecture of our trained model is shown in Figure 3. We leverage InternVL2-Chat-2B as the backbone for DVQA. Existing vision encoders are unable to fully understand the information-dense document images and, as a result, are prone to misidentifying information when answering questions, resulting in incorrect reasoning for the final answer. Therefore, focusing on the area relevant to the question is necessary to prevent interference from other noisy information.

Mixture-of-Modality Adaptation. As shown in Figure 3, we use plug-and-play, lightweight multimodality adapters that do not affect the backbone. Inspired by LLaMA-AdapterV2 (Gao et al., 2023), we use a late fusion strategy that allows it to interact at a higher level by adjusting visual features at different levels. Enables it to dynamically adjust attention based on the current query and historical context, allowing the model to dynamically adapt to different query requirements while maintaining the original visual coding capabilities. These modules can be the common adapters (Houlsby et al., 2019; Luo et al., 2023). For the multimodal input features $\mathbf{Z} \in \mathbb{R}^{l \times d}$, the Mixture-of-Modality Adapter can be defined as:

$$Z' = Z + s \cdot f(Z). \tag{4}$$

Here, f refers to the RepAdapter (Luo et al., 2023), and s is the scale factor. To further reduce parameter costs, the downsampling projection of the two adapters is shared. Based on the Mixture-of-Modality Adapter, the training objective is to freeze the vision encoder and LLM, fine-tuning only the inserted adapters and the projection connector. In this case, the entire multimodal language model can be jointly optimized in an end-to-end manner. Specifically, the end-to-end optimization

objective can be formulated as:

$$\arg\min \mathcal{L}(f_\phi(Z), R; \theta_a, \text{projection}). \quad (5)$$

Here, R and $\mathcal{L}(\cdot)$ denote the ground truth and the objective loss function, respectively. f_ϕ is the LLM, and θ_a represents the adaptation parameters.

4 Experiments

4.1 Experimental Setup

Dataset. We run experiments on three document VQA datasets: DocVQA (Mathew et al., 2021), InfographicVQA (Mathew et al., 2022), and ChartQA (Masry et al., 2022). DocVQA images come from scanned documents, including letters, tables, articles, etc. Most of the questions are relatively simple and involve text extraction directly. InfographicVQA images are taken from posters, where the layout is complex. Most of the questions involve text extraction, and some require simple reasoning. The charts in the ChartQA dataset include bar charts, line charts, and pie charts, all of which are synthetic data or human-marked data. Most of the synthetic data is simple chart information identification, while the human-marked data contains complex mathematical logic operations.

Data extension. Table 1 shows the generated QA statistics. The sum includes data based on the original training set and data generated from the question generation template extension. The template can be found in the Appendix (Table 9,10,11). For DocVQA, 3 new questions were generated for each image, while InfographicVQA had 4 new questions per image. Given the small amount of original data in ChartQA Human, since there is less raw data in ChartQA Human, we collected the related chart images of Chart-to-text (Kantharaj et al., 2022) and the corresponding gold table to generate more chart reasoning data. Filtered indicates the data filtered through the LLM and rules.

In addition, we generated questions based on the types of questions in each dataset, such as Count, Spatial, and Reasoning data for DocVQA. For more details, see the Appendix (Table 12 and Table 14).

Compared Methods. We compare existing document understanding models across various categories: (1) plain text models represented by T5, (2) the LayoutLM series represented by LayoutLMv3, and DocFormerv2, which has the best performance among the T+L+V models, (3) the first OCR-free model Donut for understanding documents with image encoders, and Pix2Struct for performing best in end-to-end small-scale document understanding models, and (4) multimodal document understanding models combined with LLMs. Most previous models adopted a fixed resolution of 224x224, which works effectively for real-world scene images as the models can still recognize object contours at this resolution. However, for dense fine-grained document images, such a low resolution is likely to lead to information recognition errors. To improve effectiveness in applications, recent models are gradually eliminating the use of external tools such as OCR and improving the resolution of document images for corrective recognition.

Dataset	Images	Generated	Filtered
DocVQA	10194	39459+30582	58324
InfoVQA	4406	23945+17624	36832
ChartQA	18317+44096	7398+80831	67649

Table 1: Statistics of generated data.

Implementation Details. All of our experiments were performed on 2 80G A100 GPUs. During the training process, we set 2 epochs with a batch size of 8 and a learning rate of 4e-5. Specifically, dynamically resizing the image, adjusting the resolution to 448x448, and setting the maximum patch size to 12 are crucial for understanding document-type images. All experiments were conducted on a 2B model, with training focused on the adapter layers (N-2), the projection module, and the language model incorporating LoRA.

4.2 Main Results

Comparison with Existing Models. Existing models tend to provide single words or phrases as answers for the DVQA task. On the one hand, the complex layout of document images, prone to locating the wrong image regions. On the other hand, for complex questions requiring reasoning, ignoring intermediate reasoning steps is more likely to lead to errors. Through our reconstruction of the dataset and efficient training on the 2B multimodal model, we have achieved impressive results, as shown in Table 2. Using only a small number of training parameters, it outperforms many larger document understanding models and demonstrates greater reasoning power than other existing models.

In terms of details, due to the large amount of text information contained in the document, it is difficult for the visual encoder to fully comprehend all the fine-grained semantic content. The Mix-

Method	Modality	Params	DocVQA	InfographicVQA	ChartQA
			ANLS	ANLS	Relax Accuracy
T5 (Raffel et al., 2020)	T	223M	70.4	36.7	59.8
LayoutLMv3 (Huang et al., 2022)	T+V+L	368M	83.4	45.1	-
DocFormerv2 (Appalaraju et al., 2024)	T+V+L	750M	87.8	48.8	-
Donut (Kim et al., 2022)	V	201M	67.5	11.5	41.8
Pix2Struct (Lee et al., 2023)	V	1.3B	76.6	40.0	58.6
mPLUG-DocOwl2 (Hu et al., 2024)	V	8B	80.7	46.4	70.0
SMoLA-PaLI-3 (Wu et al., 2024)	V	5B	84.5	52.4	68.9
ScreenAI (Baechler et al., 2024)	V	5B	89.9	65.9	76.7
Qwen2-VL (Wang et al., 2024b)	V	2B	**90.1**	65.5	73.5
InternVL2 (Baseline) (Chen et al., 2024b)	V	2B	86.9	58.9	76.2
DocAssistant (Ours)	V	2B	88.5	61.5	78.9
DocAssistant†(Ours†)	V	2B	89.8	**66.7**	**81.4**

Table 2: Comparison with models of different scales and different modal models. T, L, and V represent text, layout, and visual information, respectively. †represents the result of training using the data we constructed. The evaluation method of DocVQA and InfoVQA is ANLS, while the evaluation method of ChartQA is Relax Accuracy.

Model	Strategy	DocVQA	InfoVQA	ChartQA	
				hum.	aug.
Base	ZS	80.2	56.3	53.6	75.8
	FS	79.5	55.1	55.9	78.7
	SFT	84.8	59.8	63.2	83.3
Ours	ZS	81.1	57.7	55.8	80.4
	FS	80.5	56.5	59.2	81.8
	SFT	85.6	62.4	64.9	85.2

Table 3: The comparison results of DocAssistant under different strategies, and the evaluation method is Accuracy, where ZS stands for Zero-shot, FS represents Few-shot, and SFT indicates a step-wise result trained not with a special prompt but with the generated data.

ture of Modality module we designed effectively alleviates this limitation. As shown in the experimental results, except for Qwen2-VL, DocAssistant outperforms all other models. Furthermore, for the Baseline model (InternVL2), the addition of the Mixture-of-Modality Adaptation leads to improvements of 1.6%, 2.6%, and 2.7% in the three datasets. It demonstrates significantly improved performance and exhibits strong generalization.

Additionally, based on the improvements made to the above model, after training through the step-wise process we constructed, DocAssistant's performance on DocVQA is slightly inferior to Qwen2-VL. This may be due to differences in the performance of the respective visual encoders of InternVL and QwenVL. However, existing models perform poorly on documents with complex layouts and those requiring reasoning, such as InfoVQA and ChartQA. Furthermore, after fine-tuning with our step-wise constructed data, the performance improved by 1.3%, 5.2%, and 2.5% respectively across relevant metrics. Through these enhancements, we have narrowed the gap in document understanding and reasoning capabilities.

Comparison of Different Strategies. In the strategy experiments presented in Table 3, the zero-shot setting involved changing the original prompt to "Answer the question step by step." According to the experimental results, the model performs better on complex layout documents and complex reasoning questions when answering step by step. For extracted answers to simple questions like DocVQA, the results are similar regardless of the approach. In the few-shot setup, we conducted 3-shot experiments and found that the instruction compliance of the 2B model was poor. Its answers did not follow the example format, especially in DocVQA and InfoVQA, which do not require reasoning and typically have concise answers. Although the model's responses on ChartQA did not follow the example format, it still provided step-by-step answers, leading to improved performance. In contrast, the other two datasets with extracted answers performed worse. This may be because small-scale multimodal models lack sufficient chain-of-thought ability and have poor capacity to follow long instructions. In addition, the results show that the adapter module we designed does not destroy the performance of the original model. Since the generation

styles of different strategies are different, accuracy is used as the evaluation metric.

Generating the answer directly can better fit the answer format, but they lack contextual information, are less interpretable, and are more difficult to judge the correctness. Zero-shot and few-shot are unstable and heavily influenced by instruction settings. For documents with extractive or abstractive answers, most answers still do not provide source information. These are the original intentions behind our approach to building step-by-step datasets and training the model. By comparing different strategies and methods, we conclude that only by using our synthetic data can we achieve the answer logic that meets our requirements for SVLMs. Not only is the answer accuracy higher, but the source information of the answer can also be provided, helping users confirm the reliability of the answer.

4.3 Ablations and Analysis

The effect of the Mixture-of-Modality adaptation is reflected in Table 2. The impact of extended data on model performance will be analyzed below.

	DocVQA	InfoVQA	ChartQA
w adaption	88.5	61.5	78.9
+ data_expansion	89.0	64.9	80.8
+ data_filtration	89.8	66.7	81.4

Table 4: Before and after using extended data and before and after using filtered training data. The evaluation method of DocVQA and InfoVQA is ANLS, while the evaluation method of ChartQA is Relax Accuracy.

The Effect of Data Expansion. The modification of the original training set data, combined with our extended data, is sufficient for the model to learn the required response style. Experimental results in Table 4 show a performance gain after extension. This also provides a data generation tool for other smaller models, enabling them to further improve performance by generating more data.

The Effect of Checker. We fine-tuned the model using both pre-filtered and filtered data, and the results in Table 4 showed improved performance on all three datasets. Furthermore, the data we generated is categorized into two distinct components. The first component involves the supplementation of rationales based on the existing training set data. Given that this portion utilizes answers as supervisory signals, it effectively mitigates model hallucinations. The second component consists of question-answer pairs produced by MLLM in accordance with our predefined rules. Due to the absence of supervisory signals in this part, we developed an LLM-based filter that leverages Gold OCR Text as supervisory signals. By meticulously reviewing each step, this mechanism successfully suppresses low-quality data arising from model hallucinations. Consequently, the data we generated significantly reduces model hallucinations. Since the majority of the filtered low-quality data comprises entirely MLLM-generated question-answer pairs, which account for only 16.7%, 11.4%, and 23.3% of the total data respectively, their influence on the model remains limited.

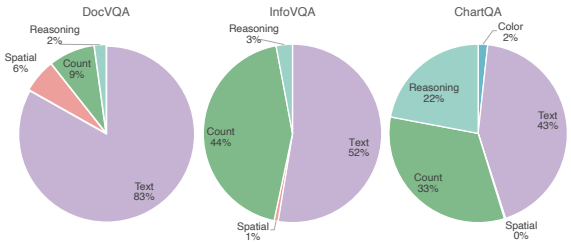


Figure 4: Analysis of different types of questions in three test sets.

Analysis of Different Question Types. Figure 4 shows the proportion of question types of the three datasets. The text_extractive type has the largest proportion in all datasets, especially in DocVQA. InfoVQA and ChartQA have a higher proportion of Count and Reasoning data, and these types of data can highlight more of DocAssistant's advantages.

Dataset	Method	COL	TXT	SPA	COU	REA
DocVQA	Base	-	86.2	87.0	88.2	58.7
	Ours	-	89.5	88.7	88.3	61.5
	Ours†	-	90.4	87.5	88.8	68.3
InfoVQA	Base	63.2	60.8	57.2	56.9	55.4
	Ours	62.5	61.3	56.8	59.4	57.2
	Ours†	66.7	59.5	57.6	66.9	79.8
ChartQA	Base	77.1	91.1	75.0	67.5	62.7
	Ours	72.9	90.2	75.0	69.4	65.1
	Ours†	70.8	92.6	100.0	77.5	75.3

Table 5: Comparison of different types of question. The evaluation also uses ANLS and Relax Accuracy.

Table 5 shows the performance of five types of questions in three different types of datasets under different settings. The experimental results show that the Count and Reasoning types are effectively improved after inserting the Mixture-of-Modality

Figure 5: Output comparison of DocAssistant and other models on three datasets, with red font representing rationales relevant to the question. The heat map in the figure shows the regions associated with the question.

adapter, while for other types of questions, different datasets exhibit their own advantages. For instance, DocAssistant performs well on DocVQA and InfoVQA's Text_extraction data, but its performance deteriorates with chart data. This may be attributed to misidentification in the chart and weak correlation between the local areas of the chart data. For the Color-type question, performance depends on the original capability of the visual encoder.

Furthermore, after training with our extended data, the performance of the Text_extraction, Count, and Reasoning types has improved, with the latter two showing particularly significant gains. This suggests that the step-wise reasoning process has a substantial positive impact on the model's ability to handle reasoning tasks.

Qualitative Analysis. Figure 5 shows a comparison between cases generated by DocAssistant and those generated by the base model. We output the attention scores from the last layer of the visual encoder applied to the document image. It is intuitively evident that the Mixture-of-Modality adapter effectively integrates text and visual features, enabling the visual encoder to focus more on key information. As a result, the interference from irrelevant noise is minimized, improving the accuracy of information extraction by the model. On the language side, the answers provided by the base model in different documents are single words or phrases that tend to be incorrect when faced with questions such as chart reasoning and document information extraction with complex layouts, and they do not contain contextual information. The first two examples are questions and answers about complex layout documents, where DocAssistant locates the right visual region and provides detailed context. The third example is a chart reasoning question, where DocAssistant also gives specific reasoning steps, providing stronger interpretation and higher accuracy, and allowing users to understand how the model reached the answer and easily identify any mistakes. Most importantly, model performance can be further improved through extensive training with step-wise data.

5 Conclusion

This paper introduces DocAssistant, designed to perform document understanding and reasoning. Specifically, we first propose a Mixture of Modality adaptation that can be inserted into the vision encoder layers to make the model pay more attention to the region related to the question keywords. Then we transform and expand the data of the existing document training set to include the intermediate analysis process in answering questions. The results surpass larger-scale models. Furthermore, we conducted extensive experiments, comparing different strategies of the model, performance before and after data expansion and filtering, and various types of questions to validate our work.

6 Limitations

Although our proposed model, DocAssistant, has demonstrated promising results in experiments, it still has certain limitations. First, the generalizability of the constructed dataset remains limited. While the model performs well within the data scope we created, its effectiveness in other domains remains uncertain, potentially requiring dataset reconstruction. Second, although we have enhanced the visual encoder, its performance is constrained by the limitations of the original backbone, which still struggles to fully address complex problems involving cross-region and multi-structured data. Moreover, the model is primarily effective in single-page document understanding and cannot efficiently handle multi-page documents. These limitations highlight key directions for future research.

7 Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments. This work was supported by the National Natural Science Foundation of China (No.62476066).

References

AI@Meta. 2024. Llama 3 model card.

Srikar Appalaraju, Peng Tang, Qi Dong, Nishant Sankaran, Yichu Zhou, and R Manmatha. 2024. Docformerv2: Local features for document understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 709–718.

Gilles Baechler, Srinivas Sunkara, Maria Wang, Fedir Zubach, Hassan Mansoor, Vincent Etter, Victor Cărbune, Jason Lin, Jindong Chen, and Abhanshu Sharma. 2024. Screenai: A vision-language model for ui and infographics understanding. *arXiv preprint arXiv:2402.04615*.

Xi Chen, Xiao Wang, Lucas Beyer, Alexander Kolesnikov, Jialin Wu, Paul Voigtlaender, Basil Mustafa, Sebastian Goodman, Ibrahim Alabdulmohsin, Piotr Padlewski, et al. 2023. Pali-3 vision language models: Smaller, faster, stronger. *arXiv preprint arXiv:2310.09199*.

Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, et al. 2024a. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *arXiv preprint arXiv:2404.16821*.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. 2024b. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24185–24198.

Brian Davis, Bryan Morse, Brian Price, Chris Tensmeyer, Curtis Wigington, and Vlad Morariu. 2022. End-to-end document recognition and understanding with dessurt. In *European Conference on Computer Vision*, pages 280–296. Springer.

Peng Gao, Jiaming Han, Renrui Zhang, Ziyi Lin, Shijie Geng, Aojun Zhou, Wei Zhang, Pan Lu, Conghui He, Xiangyu Yue, et al. 2023. Llama-adapter v2: Parameter-efficient visual instruction model. *arXiv preprint arXiv:2304.15010*.

Tanmay Gupta and Aniruddha Kembhavi. 2023. Visual programming: Compositional visual reasoning without training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14953–14962.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pages 2790–2799. PMLR.

Anwen Hu, Haiyang Xu, Jiabo Ye, Ming Yan, Liang Zhang, Bo Zhang, Chen Li, Ji Zhang, Qin Jin, Fei Huang, et al. 2024. mplug-docowl 1.5: Unified structure learning for ocr-free document understanding. *arXiv preprint arXiv:2403.12895*.

Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. 2022. Layoutlmv3: Pre-training for document ai with unified text and image masking. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 4083–4091.

Shankar Kantharaj, Rixie Tiffany Leong, Xiang Lin, Ahmed Masry, Megh Thakkar, Enamul Hoque, and Shafiq Joty. 2022. Chart-to-text: A large-scale benchmark for chart summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4005–4023, Dublin, Ireland. Association for Computational Linguistics.

Geewook Kim, Teakgyu Hong, Moonbin Yim, JeongYeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. 2022. Ocr-free document understanding transformer. In *European Conference on Computer Vision*, pages 498–517. Springer.

Kenton Lee, Mandar Joshi, Iulia Raluca Turc, Hexiang Hu, Fangyu Liu, Julian Martin Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. 2023. Pix2struct: Screenshot parsing as pretraining for visual language understanding. In *International Conference on Machine Learning*, pages 18893–18912. PMLR.

Zhuowan Li, Bhavan Jasani, Peng Tang, and Shabnam Ghadar. 2024. Synthesize step-by-step: Tools templates and llms as data generators for reasoning-based chart vqa. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13613–13623.

Fangyu Liu, Julian Martin Eisenschlos, Francesco Piccinno, Syrine Krichene, Chenxi Pang, Kenton Lee, Mandar Joshi, Wenhu Chen, Nigel Collier, and Yasemin Altun. 2022. Deplot: One-shot visual language reasoning by plot-to-table translation. *arXiv preprint arXiv:2212.10505*.

Pan Lu, Baolin Peng, Hao Cheng, Michel Galley, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, and Jianfeng Gao. 2024. Chameleon: Plug-and-play compositional reasoning with large language models. *Advances in Neural Information Processing Systems*, 36.

Gen Luo, Minglang Huang, Yiyi Zhou, Xiaoshuai Sun, Guannan Jiang, Zhiyu Wang, and Rongrong Ji. 2023. Towards efficient visual adaption via structural re-parameterization. *arXiv preprint arXiv:2302.08106*.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 2022. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. *arXiv preprint arXiv:2203.10244*.

Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and CV Jawahar. 2022. Infographicvqa. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1697–1706.

Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. 2021. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2200–2209.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Dídac Surís, Sachit Menon, and Carl Vondrick. 2023. Vipergpt: Visual inference via python execution for reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11888–11898.

Lei Wang, Yi Hu, Jiabang He, Xing Xu, Ning Liu, Hui Liu, and Heng Tao Shen. 2024a. T-sciq: Teaching multimodal chain-of-thought reasoning via large language model signals for science question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19162–19170.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. 2024b. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Jialin Wu, Xia Hu, Yaqing Wang, Bo Pang, and Radu Soricut. 2024. Omni-smola: Boosting generalist multimodal models with soft mixture of low-rank experts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14205–14215.

Jundong Xu, Hao Fei, Liangming Pan, Qian Liu, Mong-Li Lee, and Wynne Hsu. 2024a. Faithful logical reasoning via symbolic chain-of-thought. *arXiv preprint arXiv:2405.18357*.

Ruyi Xu, Yuan Yao, Zonghao Guo, Junbo Cui, Zanlin Ni, Chunjiang Ge, Tat-Seng Chua, Zhiyuan Liu, Maosong Sun, and Gao Huang. 2024b. Llava-uhd: an lmm perceiving any aspect ratio and high-resolution images. *arXiv preprint arXiv:2403.11703*.

Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, et al. 2020a. Layoutlmv2: Multi-modal pre-training for visually-rich document understanding. *arXiv preprint arXiv:2012.14740*.

Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. 2020b. Layoutlm: Pre-training of text and layout for document image understanding. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1192–1200.

Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Ehsan Azarnasab, Faisal Ahmed, Zicheng Liu, Ce Liu, Michael Zeng, and Lijuan Wang. 2023. Mm-react: Prompting chatgpt for multimodal reasoning and action. *arXiv preprint arXiv:2303.11381*.

Andy Zeng, Maria Attarian, Brian Ichter, Krzysztof Choromanski, Adrian Wong, Stefan Welker, Federico Tombari, Aveek Purohit, Michael Ryoo, Vikas Sindhwani, et al. 2022. Socratic models: Composing zero-shot multimodal reasoning with language. *arXiv preprint arXiv:2204.00598*.

Zhuosheng Zhang, Aston Zhang, Mu Li, George Karypis, Alex Smola, et al. Multimodal chain-of-thought reasoning in language models. *Transactions on Machine Learning Research*.

A Generated Data Analysis

We classify the generated questions into five categories to clarify the model's performance on different types of questions. Among these, since DocVQA is a scanned document dataset, we did not generate questions involving color information. Additionally, since the training sets of DocVQA and InfoVQA contain a large number of text extraction questions, we limited the types of questions generated by DocVQA to spatial, count, and reasoning when generating incremental questions. InfoVQA limits the types to color, spatial, count, and reasoning in the process of generating incremental questions. For details, as shown in Table 6.

Dataset	Color	Spatial	Text	Count	Reasoning
DocVQA	0	3779	39769	9386	5390
InfoVQA	2019	4192	15820	10775	4026
ChartQA	5094	16	12254	23596	26689

Table 6: Statistics of different question types of generated data.

B The Effect of Data Scale

During the initial experiments, we conducted experiments on 7B, 4B, and 2B models respectively using data of the same scale, and found that the 7B and 4B models had poor instruction-following ability on our generated data, while the 2B model better fit the data scale. To further explore the data scale required by DocAssistant, as shown in Table 7, we divided the generated data into equal thirds and found that the model achieved better results on the three datasets when using between two-thirds and all of the data.

	DocVQA	InfoVQA	ChartQA
1/3 of data	88.0	64.8	77.2
2/3 of data	89.4	65.7	79.8
All of data	89.8	66.7	81.4

Table 7: The performance of DocAssistant on different scale training data. The evaluation method of DocVQA and InfoVQA is ANLS, while the evaluation of ChartQA is Relax Accuracy.

C Further Analysis of ChartQA

ChartQA is divided into synthetic data (Augmented) and human-generated data (Human). The questions in Augmented are relatively simple, mostly involving the direct extraction of chart information without complex calculations. Human questions are more complex and include multi-step or nested operations, which better test the model's reasoning ability. In Table 8, it is evident that Zero-shot, Few-shot, and Finetune methods perform better than straightforward methods, with the Finetune method we trained being far superior for counting and reasoning questions. For Text_extractive type data, performance is more unstable, and extracting information from complex charts, including color information, remains a significant challenge. Given the small amount of data in Color and Spatial categories, these results are not definitive but indicate that after fine-tuning, the model's understanding of color information has declined, which is closely related to the performance of the model's visual encoder.

D Error Analysis

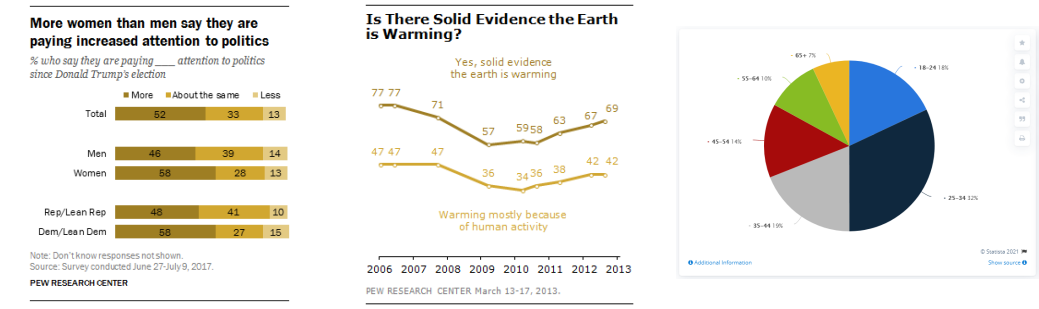
Although DocAssistant has achieved the most advanced results on the three document datasets, it still has shortcomings. After our analysis, we identified four types of errors.

- One type of error is information recognition. For example, the pie chart in Figure 6 (c) fails to identify the percentage corresponding to "18-24". The percentage corresponding to "24-35" is identified incorrectly (32% is identified as 35%), and the 7% corresponding to "65+" is identified as 65.7%.
- The second type of error is the inference error of the intermediate step, as shown in Figure 6 (b), which occurs when addition calculations are performed with accurate information identification.
- All two types of errors may occur simultaneously. As shown in Figure 6 (a), no information related to the keywords of the problem is obtained, the information in the chart is identified incorrectly, and there are calculation errors in the intermediate steps.

The occurrence of the above errors is related to the performance of the model's three main modules. Identification errors are mainly due to deficiencies in the visual encoder's fine-grained understanding of document information. Errors in the intermediate inference step show that the language model is insufficient for inferring and calculating complex information.

Dataset	Strategies	Color	Text	Spatial	Count	Reasoning
Augmented	Straightforward	-	71.0	100.0	65.7	54.4
	Zero-shot	-	76.1	100.0	66.3	60.9
	Few-shot	-	80.4	0.0	72.9	65.2
	Finetune	-	83.8	100.0	85.6	88.4
Human	Straightforward	72.4	81.1	50.0	39.3	47.1
	Zero-shot	67.1	79.7	75.0	42.5	49.3
	Few-shot	58.6	78.0	25.0	48.0	57.3
	Finetune	62.8	76.7	100.0	65.12	64.6

Table 8: ChartQA experimental results on two datasets. The evaluation method is Accuracy.



Question: What is the total value of the More bar?
Gold Answer: 52

DocAssistant:
I step I output I
I 1 I Identify the values of the More bar for each gender: Men (46), Women (58), Rep/Lean Rep (48), Dem/Lean Dem (88). I
I 2 I Calculate the sum of the values: 46 + 58 + 48 + 88 = 235. I
I 3 I The final calculation result is obtained: 235.I
The answer is: 235

(a)

Question: What is the sum of the three same values in the lower graph?
Gold Answer: 141

DocAssistant:
I step I output I
I 1 I Identify the three same values in the lower graph: 47, 47, and 47. I
I 2 I Calculate the sum of these three values: 47 + 47 + 47 = 168. I
I 3 I The final calculation result is obtained: 168. I
The answer is: 168

(b)

Question: Which age group has the maximum number of people using internet?
Gold Answer: 25-34

I step I output I
I 1 I Identify the age groups and their corresponding percentages in the image: 18-24 (65.7%) and 25-34 (35%). I
I 2 I Compare the percentages to find the age group with the highest percentage: 18-24 has the highest percentage at 65.7%. I
I 3 I The final answer is obtained: 18-24.
The answer is: 18-24"

(c)

Figure 6: DocAssistant error examples on ChartQA.

E Templates

Templates for Generated QA. Table 9 shows the instructions for the DocVQA and InfoVQA training sets to generate evidence corresponding to the question using MLLM based on the existing Q&A pairs. Table 10 shows the generation of specified types of questions, corresponding evidence, and answers based on the training set images in DocVQA and InfoVQA. Table 11 shows the few-shot template designed for ChartQA. The design of the few-shot template enables it to flexibly adapt to various questions due to the variety of intermediate steps. Additionally, this template can generate corresponding intermediate reasoning steps and questions for the existing question and answer pairs simultaneously. Table 12 is a list of instructions that constrain questions generated from different datasets.

Template for the data checker. The data checker we designed in Table 2 operates in two aspects: First, it detects whether there is an information detection error based on the question generation. This detection uses external tools to extract the text information of the image. Second, it checks whether the intermediate reasoning step is correct, such as identifying calculation errors. If an error is detected, the data is discarded. Thus, if either of these two errors is present, the data is discarded to ensure the quality of the generated data.

Template for question classification. To further analyze which question types DocAssistant has improved, we used a multimodal large language model to classify questions, and the template is shown in Table 14. Since existing datasets do not categorize each question specifically, we subjectively designed five categories to cover almost all the questions in the DVQA datasets.

Instruction:

Given an image and a question in the following, what is the answer to the question? Please complete the task in two steps:
1. In the first step, extract the relevant contexts related to the keywords in the question from the provided image.
Store these in the variable "{evidence}". If there are multiple contexts, separate them using the "#" symbol.
2. In the second step, predict the answer based on the {evidence} and store it in the variable "{answer}".

Please organize the results in the following table:
step	output
1	{evidence}
2	{answer}

The example format of response:
Response
step	output
1	{"SARS": "10%", "MERS": "34%", "EBOLA": "50%+"}
2	EBOLA has the highest mortality rate.

Follow the format of the instruction above, Generate the corresponding response based on the question and answer. ###Question
question
###Gold_answer
answer

Table 9: DVQA rationale generation template.

Instruction:

Given an image in the following, generate a question and the corresponding answer. Please complete the task in three steps:
1. In the first step, Generate a question, the question should ... Store the question in the Variable {question}.
1. In the second step, extract the relevant contexts related to the keywords in the question from the provided image.
Store these in the variable "{evidence}". If there are multiple contexts, separate them using the "#" symbol.
2. In the third step, predict the answer based on the {evidence} and store it in the variable "{answer}".

Please organize the results in the following table:
step	output
1	{question}
2	{evidence}
2	{answer}

The example format of response:
Response
step	output
1	Which disease has the highest mortality rate?
2	{"SARS": "10%", "MERS": "34%", "EBOLA": "50%+"}
3	EBOLA has the highest mortality rate.
The answer is: EBOLA.

Follow the format of the instruction above, Generate the corresponding response based on the image.

Table 10: DVQA question generation template.

###Instruction
The following are given a chart image and five examples to complete the task of generating chart question and answer data.

Example1:
Generate a question and the corresponding answer step by step based on the image:

Question: What is the difference in value between Lamb and Corn?
Answer:
Step1. The values of relevant indicators in the question are identified: The value of Lamb is 103.7 and the value of Corn is 103.13,
Step2. Perform the calculation of the difference in value between Lamb and Corn: 103.7-103.13=0.57,
Step3. The final calculation result is obtained: 0.57.

Example2:
Generate a question and the corresponding answer step by step based on the image:

Question: In which year is the difference between the green and blue graphs lowest?
Answer:
Step1. Identify the years of the chart: 2017, 2018, 2019.,
Step2. Compare the values of the green and blue graphs for each year: in 2017, the green graph is at 65 and the blue graph is at 56.
In 2018, the green graph is at 70 and the blue graph is at 41. In 2019, the green graph is at 69 and the blue graph is at 50.,
Step3. Calculate the difference between the green and blue graphs for each year: In 2017: 65-56=9, in 2018: 70-41=29, in 2019: 69-50=19,
Step4. Sort all the differences": "In 2017: 9, in 2018: 19, in 2019: 29,
Step5. Perform the calculations required in the question: in 2017, the difference between green and blue graphs is the lowest,
Step6. The final calculation result is obtained: 2017.

Example3:
Generate a question and the corresponding answer step by step based on the image:

Question: What's the average of all the values in the green bars?
Answer:
Step1. Identify all information of the blue bar: {"Characteristic": "US, EU, China", "More": "29, 19, 17"},
Step2. Perform the calculation of the average of all the values in the green bars: 29 + 19 + 17 = 21.6,
Step3. The final calculation result is obtained: 21.6.

Example4:
Generate a question and the corresponding answer step by step based on the image:

Question: Which country has the third highest rate of cases in Europe?
Answer:
Step1. Identify all values of countries in Europe: {"Montenegro":16111.01, "Czechia": 15 587.77, "Sweden": 10546.7, "Slovenia": 12276, "Slovakia":14259.69},
Step2. Sort all values: {"Montenegro":16111.01, "Czechia": 15 587.77, "Slovakia":14259.69, "Slovenia": 12276, "Sweden": 10546.7},
Step3. The third highest rate of cases in Europe is obtained: Slovakia.

Example5:
Generate a question and the corresponding answer step by step based on the image:

question: What is the sum of all the blue bar?
Answer:
Step1. Identify all information of the blue bar: {"Characteristic": "Number of gamers in millions", "2012": 8.12, "2013": 9.04, "2014": 9.97},
Step2. Calculate the sum of all values: 8.12+9.04+9.97=27.13,
Step3. The final calculation result is obtained: 27.13.

Follow the format of the example above, generate a question and the corresponding answer step by step based on the image.

Table 11: Chart question generation or rationale generation from the few-shot template.

DocVQA & InfoVQA & ChartQA
1. The question should require spatial understanding of the image.
2. The question should require counting.
3. The question should require reasoning of the image.
InfoVQA & ChartQA
1. The question should require color understanding of the image.
2. The question should require counting of colors.
3. The question should require counting and color understanding.
ChartQA
1. The question should require math reasoning about min.
2. The question should require math reasoning about average.
3. The question should require math reasoning about the difference between max and min.
4. The question should require math reasoning about difference.
5. The question should require math reasoning about comparison.
6. The question should require math reasoning about average and max.
7. The question should require math reasoning about sum.
8. The question should require math reasoning about max.
9. The question should require math reasoning about average and min.
10. The question should require math reasoning about ratio.
11. The question should require color understanding and math reasoning to compute the difference.
12. The question should require color understanding and math reasoning about comparison.
13. The question should require spatial understanding and math reasoning to compute difference.
14. The question should require spatial understanding and math reasoning about average.

Table 12: Constraints on question generation for different datasets.

Below is an instruction that describes an evidence error detection task in the general document domain, paired with an image.
Generate an appropriate response to the given instruction.

Instruction:
Given an image, question-answer pair, and corresponding evidence, assess whether the evidence is faithful to the images and corresponding text information and whether it accurately contains the context information of the question-answer pair in the image and table. Please complete the task in three steps:
1. In the first step, assess whether each step of evidence is consistent with the information in the image and the table.
If consistent, store "True" in the variable {is_faithful}; otherwise, store "False".
2. In the second step, assess whether each step of evidence contains the context information of the question-answer pair in the image.
If it does, store "True" in the variable {is_include}; otherwise, store "False".
3. If {is_faithful} is True and {is_include} is True, store "True" in the variable {result}; otherwise, store "False" in the variable {result}.
Please organize the results in the following table:
step	output
1	{is_faithful}
2	{is_include}
Finally, present the predicted answer in the format: "The answer is: {result}"

###Follow the example:

Question_answer pairs
LIver is a source of how many of the vitamins shown here? answer: 6
Evidences
step	output
1	40% of visitors to VIC went to Melbourne.
2	60%
The answer is: 60%

Response
step	output
1	False
2	True
The answer is: False

Follow the format of the instruction above, Generate the corresponding response based on the image, evidence, and question-answer pairs:
###Question_answer pairs
question & answer
###Evidence
rationale

Table 13: Error detection template for generated data.

Given a dataset consisting of DocVQA/InfographicsVQA/ChartQA and corresponding questions.
The task is to classify each question into one of the following five types based on the image and the type of information required to answer the question.

Here are the definitions for each type:

Color: Questions that require an understanding of colors.
Spatial: Questions that involve spatial relationships or positions (e.g., "next to," "above," "below", "left", "right").
Text_extractive: Questions that require extracting specific text information from the document image.
Count: Questions that involve counting elements or objects in the document image.
Reasoning: Questions that require logical reasoning, inference, or combining multiple pieces of information.

For each question, analyze the content and determine the appropriate type.
Now, classify the following questions from the DocVQA/InfographicVQA/ChartQA dataset:
###Question
question

Table 14: Question classification template based on MLLM.