AceRead: Enhancing Vision-Language Understanding with a Semantic-Enhanced Querying Mechanism

Anonymous ACL submission

Abstract

Vision-Language models (VLMs) integrate visual and textual features through modality adaptors, demonstrating outstanding performance in image understanding tasks. Among these adaptors, *compression-based* approaches have gained particular prominence, as they can prune visual redundancy, highlight key details, and streamline computational costs. However, existing compression-based adaptors often fail to fully exploit the *deep semantics* of input questions, resulting in static or uninformative compressed features across different questions. In this study, we address this gap by leveraging question semantics to guide the compression of visual features. We propose a Semantic-Enhanced Resampler (SER), integrated into our VLM, **AceRead**, which serves as a conditional information bottleneck, channeling the most question-relevant information to the language model for answer generation. SER integrates semantic tokens with visual tokens and employs learnable queries to produce compressed representations. Additionally, AceRead incorporates an adaptive image encoder, enabling the processing of images with arbitrary sizes while minimizing distortion. Notably, AceRead achieves state-of-the-art performance, with improvements of 16% on TableVQABench and 10% on A-OKVQA, while requiring only 2.75% of the model's parameters to be trained. Our code and model are available at https://anonymous.4open.science/r/AceRead-77BF.

1 Introduction

Vision-Language Models (VLMs) integrate information from both visual and textual modalities, enabling more holistic multimodal understanding and reasoning, which have gained significant attention in fields such as image captioning (Dong et al., 2024), visual question answering (Peng et al., 2023; Ji et al., 2024), and cross-modal retrieval (Li et al., 2022). Typically, VLMs build on powerful pre-trained large language models (LLMs) and vision encoders, requiring only a modality adaptor to seamlessly fuse visual features with textual tokens. Different adaptor designs have been explored to optimize this fusion process, each targeting specific challenges in cross-modal integration. In particular, *compression-based* adaptors have become increasingly popular for their ability to remove redundant information and amplify critical cues in the visual features (Alayrac et al., 2022; Li et al., 2023; Xue et al., 2024; Huang et al., 2024).

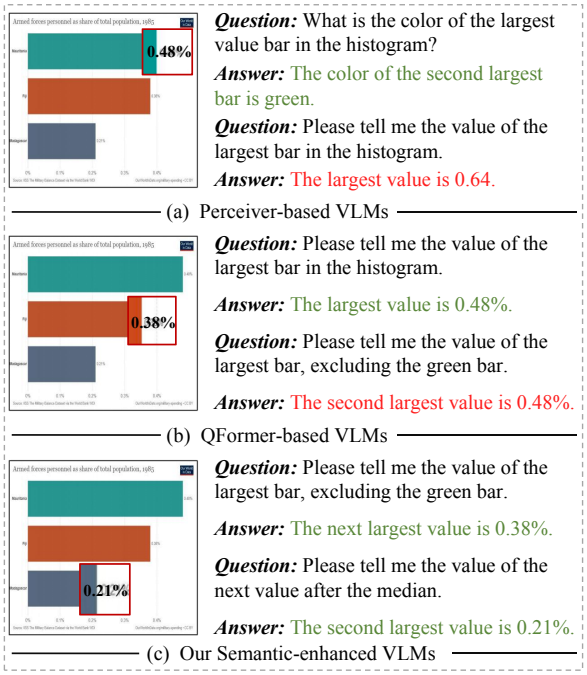


Figure 1: **Comparison of different compression-based VLMs.** (a) Perceiver-based VLMs suffer from information loss due to static compression. (b) QFormer-based VLMs primarily rely on keywords, overlooking the role of semantics. (c) Semantic-enhanced VLMs accurately align visual regions by incorporating question semantics. Correct and wrong answers are colored green and red, respectively.

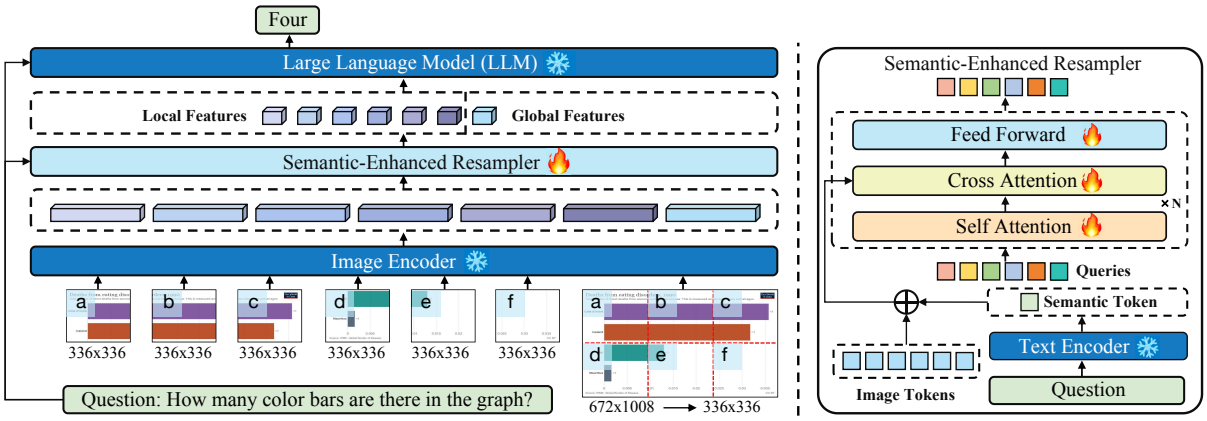


Figure 2: The overall architecture of AceRead. The left part illustrates how AceRead processes images and compresses visual features using the Semantic-Enhanced Resampler (SER). The right part provides a detailed overview of SER, where a frozen text encoder is employed to extract the semantic token from the question.

Despite these advantages, most existing compression-based adaptors focus on either purely visual information or merely textual keywords, often neglecting the deeper semantics embedded in input questions. Consequently, the compressed features often remain static or not sufficiently question-specific when handling different questions regarding the same image. For example, Perceiver-based VLMs (Bai et al., 2023; Ye et al., 2023b; Li et al., 2024b; Zhang et al., 2024) rely solely on visual information, failing to adapt if the question requires fine-grained and detailed context-specific information. As shown in Figure 1 (a), when the model is asked to identify a specific number in the bar chart, it fails due to the loss of fine-grained visual details during feature compression. QFormer-based VLMs (Li et al., 2023; Zhu et al., 2023; Li et al., 2024a) allow more dynamic compression based on shallow textual signals of the question keywords. They thus overlook the deeper semantic cues needed to differentiate changes across user questions and result in limited performance improvement (Xue et al., 2024). In Figure 1 (b), the model overly focuses on the keywords "largest" and "green bar", leading to an incorrect answer.

In this work, we address these limitations by proposing a semantic-enhanced visual token querying mechanism and introducing the Semantic-Enhanced Resampler (SER) to fully incorporate question semantics during visual feature compression. As shown in Figure 2, SER is a lightweight transformer module that first prepends a semantic token (extracted from the question's deep representation) to the visual tokens and then employs a fixed number of learnable queries to produce compressed, question-oriented features. This design allows SER to serve as a conditional information bottleneck, ensuring that the most contextually relevant visual signals are preserved while filtering out irrelevant details.

We incorporate SER into a new VLM called **AceRead**, whose main contributions are:

- AceRead is among the first VLMs that leverage a deeper semantic understanding of questions to guide visual feature compression, transcending naive keyword-based approaches.
- With only 110M trainable parameters (just 2.75% of the total), AceRead outperforms seven other VLMs, achieving state-of-the-art results on TextVQA, A-OKVQA, and TableVQABench, while remaining competitive on other benchmarks.
- AceRead is the first approach to systematically analyze the role of question semantics in compression-based adaptors through the lens of the information bottleneck framework.

2 Related Work

Vision-Language Models Current VLMs can be categorized into three paradigms. The first paradigm maintains the vision model unchanged while focusing on designing the internal architecture of language models, which enables thorough fusion between visual and textual features. For instance, Flamingo (Alayrac et al., 2022) inserts new cross-attention layers between frozen language model layers to facilitate visual-textual information

interaction and integration. Following Flamingo's approach, Llama 3.1 (Dubey et al., 2024) incorporates cross-attention layers every fourth layer within the language model to further enhance the fusion of visual information. The second paradigm maintains the architecture of both vision and language models unchanged, instead focusing on designing modality adaptors. For example, in the LLaVA model series (Liu et al., 2024b,a; Gao et al., 2024; Agrawal et al., 2024), researchers employ MLPs as modality adaptors to directly project visual features into the language space, achieving remarkable results. BLIP-2 (Li et al., 2023) further introduced the Q-Former architecture, which not only bridges the gap between vision and text modalities but also compresses and filters visual features to extract the most valuable information for downstream tasks. The third paradigm adopts an end-to-end approach by directly concatenating visual and textual features. These combined features are then fed into a multimodal language model (MLLM) for answer generation, enabling the language model to possess inherent multimodal understanding capabilities. For example, Florence-2 (Xiao et al., 2024) leverages BART (Lewis, 2019) as its foundation model and follows an encoder-decoder architecture, where end-to-end training equips the model with multimodal comprehension. Chameleon (Team, 2024), on the other hand, employs a decoder-only architecture and introduces interleaved vision-text autoregressive pretraining tasks to achieve modality fusion, allowing the model to effectively integrate visual and linguistic information.

Vision Features Compression Visual feature compression is a widely adopted technique in VLMs, effectively reducing the number of visual tokens to minimize redundancy and enhance computational efficiency. Perceiver (Jaegle et al., 2021) first introduces the use of learnable queries to compress input visual features into a compact latent bottleneck. mPlug (Ye et al., 2023b) improves this approach by concatenating visual tokens with queries before compression, mitigating the issue of fragmented fine-grained image information. Blip-2 (Li et al., 2023) highlights the importance of textual information in visual token compression and introduces Q-Former, which, through a two-stage training process, enables an understanding of shallow semantics. Instructblip (Dai et al., 2023) further enhances this by concatenating question word embeddings with queries, allowing queries to capture question-dependent variations and dynamically compress visual features. MiniMonkey (Huang et al., 2024) extends this idea by integrating textual and visual embeddings and employing learnable queries to compress tokens at the patch level. However, the aforementioned visual feature compression approaches either entirely disregard the semantic information of the question or focus solely on keywords, failing to fully leverage a deep semantic understanding of questions to optimize the visual feature compression process.

xGen-MM (Xue et al., 2024) finds that incorporating keywords into visual feature compression yields minimal performance improvement. Consequently, their model relies exclusively on the Perceiver for visual feature compression. In our study, we further investigate this phenomenon and observe that integrating only shallow semantics from the question indeed results in limited performance improvements. However, when deeper semantic information is incorporated, the model becomes significantly more effective in selecting critical visual features, leading to substantial enhancements in reasoning capability and answer accuracy.

3 Methodology

In this section, we first introduce the information bottleneck theory in VLMs, followed by a detailed explanation of the Semantic-Enhanced Resampler and its conditional information bottleneck optimization objective. Next, we describe the adaptive image encoder in AceRead, which is capable of handling images at arbitrary sizes. Finally, we describe the instruction fine-tuning strategy used during training.

3.1 Information Bottleneck in VLMs

The Information Bottleneck (IB) (Tishby et al. (2000)) aims to learn a compressed representation of inputs, which retains task-relevant content while discarding redundant details. In VLMs, we denote the visual features processed by the image encoder as variable X, the target labels as Y, and the compressed visual features as Z. To obtain the optimal feature representation Z^*, the IB principle can be formalized as the following optimization problem:

$$Z^* = \underset{z \sim p_\theta(z|x)}{\arg\max} \; I(Y;Z) - \beta I(X;Z) \quad (1)$$

where $p_\theta(z|x)$ represents the conditional distribution of Z given X, with θ denoting the model parameters. $I(Y;Z)$ measures the mutual information between Y and Z, quantifying how much

information Z retains for predicting the target Y. A larger $I(Y;Z)$ indicates that Z better preserves task-relevant content. Meanwhile, $I(X;Z)$ quantifies the dependence between X and Z, reflecting the amount of information Z retains from X. A smaller $I(X;Z)$ implies stronger compression, as it reduces redundancy while maintaining essential information for downstream tasks. The hyperparameter $\beta \geq 0$ controls the trade-off between preserving predictive information and compressing irrelevant details.

To optimize the objective function in Equation 1, we adopt the method from VIB (Alemi et al., 2016) to derive the lower bound L_{IB} of the information bottleneck. For brevity, we place the deduction in Appendix A and only present the result:

$$I(Y;Z) - \beta I(X;Z) \geq L_{IB} = \quad (2)$$
$$\mathop{\mathbb{E}}_{\substack{z\sim p_\theta(z|x)\\(x,y)\sim p_\theta(x,y)}} \Big[\log q_\phi(y|z) - \beta KL\Big(p_\theta(z|x)||q_\phi(z)\Big)\Big]$$

By maximizing the lower bound L_{IB}, Z can be optimized. In Equation 2, $q_\phi(y|z)$ represents the language decoder with frozen parameters. The only trainable component is $p_\theta(z|x)$, which learns to extract a compressed yet informative representation.

3.2 Semantic-Enhanced Resampler

As shown in Figure 2, our Semantic-Enhanced Resampler (SER) is a lightweight transformer module composed of three key components: a self-attention block, a cross-attention block, and a feed-forward network. In contrast to the QFormer-based approach, our method introduces two notable innovations: First, we introduce a single semantic token that encapsulates the deep semantic information of the question to guide visual feature compression. Second, directly concatenating semantic and visual tokens facilitates more effective cross-modal fusion, thereby enhancing the model's capacity to retain and utilize information.

The image is processed by a visual encoder, generating visual feature tokens $\mathbf{V} \in \mathbb{R}^{N_v \times d_v}$, where N_v represents the number of visual tokens and d_v denotes the visual feature dimension. Simultaneously, the question is encoded by a frozen text encoder, yielding text features $\mathbf{T} \in \mathbb{R}^{N_t \times d_t}$, with N_t representing the question token length and d_t the text feature dimension. By applying mean pooling to the text features, a semantic token $\mathbf{S} \in \mathbb{R}^{1 \times d_t}$ is obtained. The semantic token $\mathbf{S}$ and visual token $\mathbf{V}$ are concatenated as keys and values in the cross-attention block, while a set of learnable queries $\mathbf{Q} \in \mathbb{R}^{N_q \times d_q}$ is selected as the query, where $N_q \ll N_v$ controls the capacity of information bottleneck. After performing the feature aggregation computation in the cross-attention block, compressed tokens $\mathbf{Z} \in \mathbb{R}^{N_q \times d}$ are ultimately obtained. Since the text and visual encoders both originate from SigLIP (Zhai et al., 2023), the dimensions satisfy $d_v = d_t = d_q = d$.

SER serves as the information bottleneck in AceRead, selecting the most question-relevant visual tokens to the language model for answer generation. Compared to Equation 2, the key innovation lies in reformulating the compression model from $p_\theta(z|x)$ to $p_\theta(z|x \oplus s)$, where the semantic token s is incorporated as a conditional input. Through extensive fine-tuning with instruction signals from diverse VQA datasets, p_θ is optimized to maximize L_{IB} , thereby realizing what we refer to as a "conditional information bottleneck" mechanism.

Algorithm 1 Adaptive Image Encoding Algorithm

input: Image I, Grids G, Patch size (H_v, H_w)
output: Visual features V

```
Initialize optimal grids g* and IoU score s;
for each grid g ∈ G do
    Compute IoU_r and IoU_s as in Equation3
    s1 ← IoU_r + IoU_s
    if s1 ≥ s then
        s ← s1
        g* ← g
    end if
end for
P ← Crop image I with g*
I' ← Resize image I to (H_v, H_w)
I ← Concatenate P and I'
V ← Encode I using visual encoder f_v
return V
```

3.3 Adaptive Image Encoder

In AceRead, we implement an adaptive image encoder that prevents content distortion typically caused by fixed-size scaling. Following UReader(Ye et al., 2023a), we employ a Shape-Adaptive Cropping Module that preprocesses images according to their aspect ratios, generating uniform-sized patches. Each patch undergoes independent encoding, followed by feature concatenation to form a complete image representation.

We predefine diverse grids $G = \{(n_h \times n_w) | n_h \cdot n_w \leq N_c, n_h \in \mathbb{N}, n_w \in \mathbb{N}\}$, where n_h and n_w

denote the number of rows and columns of grid $g \in G$, respectively, and N_c represents the maximum number of image patches after cropping. These predefined grids ensure flexibility in adapting to various image layouts. For an input image $I \in \mathbb{R}^{3\times H\times W}$, we select the optimal grid g^* based on two criteria: (1) g^* should preserve the resolution of the image as much as possible, and (2) the aspect ratio of g^* should align with that of the image. To measure the similarity between image I and grid g in terms of pixel coverage and aspect ratio alignment, we compute Intersection over Union (IoU) at both pixel and grid levels.

$$
\begin{aligned}
IoU_r(I,g) &= IoU((H,W),(n_hH_v, n_wW_v)) \\
IoU_s(I,g) &= IoU((\frac{n_wH}{W}, n_w),(n_h,n_w)) \quad (3)
\end{aligned}
$$

Where H_v and W_v denote the height and width of each patch, respectively, which also serve as the input dimensions for the visual encoder. The optimal grid g can be obtained by maximizing the following matching score:

$$
g^* = \arg\max_{g\in G} \; IoU_r(I,g) + IoU_s(I,g) \quad (4)
$$

After selecting the optimal grid $g^* = (n_h \times n_w)$, we resize image I to (n_hH_v, n_wW_v) and partition it into $n_h \cdot n_w$ local patches. Meanwhile, to preserve the global structure of the input image, we additionally scale I to (H_v, W_v) as a global patch, which is then processed alongside the local patches by the visual encoder. The resulting visual features $V \in \mathbb{R}^{N_v\times d_v}$ have feature numbers $N_v = (n_h \cdot n_w + 1) \times N_p$, where N_p represents the number of visual features generated per patch by the visual encoder. Algorithm 1 outlines the complete processing workflow.

3.4 Instruction Tuning

Instruction tuning is a supervised fine-tuning (SFT) approach that further trains a pre-trained model on (instruction, output) pairs to improve the model's ability to follow human instructions. This method bridges the gap between large language models (LLMs) that are trained purely on next-token prediction and the expectation for them to adhere to user instructions. It also enhances performance on unseen tasks (Feng et al., 2023). To enhance the model's task generalization capabilities, we reformat all training datasets into an instruction-tuning paradigm: "<|user|>:{Question} <|assistant|>:{<image>answer}". The special tokens '<|user|>' and '<|assistant|>' indicate the user and assistant dialogue segments respectively, while '<image>' functions as a placeholder for image position, which is consistently placed before the question in our experiments. Additionally, we implement response control by prefixing '<|system|>' instructions to '<user>' queries, allowing explicit specification of both response style and level of comprehensiveness. Instruction-tuning dataset examples are provided in the Appendix B.

Dataset	# Imgs.	# Anns.	Modality
TextVQA	28K	40K	Scenes
A-OKVQA	19K	19K	Scenes
ChartVQA	21K	33K	Charts
InfoVQA	6K	30K	Charts
TableVQABench	1.5K	1.5K	Charts
DocVQA	13K	50K	Documents

Table 1: Comparison of different VQA datasets. 'Scenes' refers to images from natural scenes, 'Charts' denotes statistical chart images, and 'Documents' represents images of document pages.

4 Experiments

4.1 Implementation Details

AceRead builds upon the recently proposed xGen-MM (Xue et al., 2024), which is a powerful Perceiver-based VLM. To incorporate semantics, we replace its visual compression model with our semantic-enhanced resampler. The dimensions of hidden states d_v, d_t and d_q are 1152, and the number of learnable queries of SER N_q is set to 128. For the Adaptive Image Encoder, we set the maximum number of grids N_c to 9, covering 23 distinct aspect ratio grids. The size of each patch $H_v \times W_v$ is set to $= 384$ to match the pre-trained resolution of the visual encoder. During instruction tuning, we set the maximum input sequence length to 4096 and batch size to 16. The learning rate follows a linear warm-up strategy, increasing to $2e^{-4}$ during the initial 10% of training steps, followed by cosine decay to 0. Our experiments are conducted on six datasets: ChartVQA (Masry et al., 2022), TableVQABench (Kim et al., 2024), InfoVQA (Mathew et al., 2022), TextVQA (Singh et al., 2019), and A-OKVQA (Schwenk et al., 2022), DocVQA (Mathew et al., 2021), with

Table 2: Main results on six visual question answering benchmarks. AceRead achieves state-of-the-art performance on TableVQABench, TextVQA, and A-OKVQA. Underline indicates the second-highest performance.

Model	Train Param	Chart VQA	TabelVQA Bench	InfoVQA	TextVQA	A-OKVQA	DocVQA
TinyChart-3B	9M	**72.0**	15.3	15.4	19.0	16.6	21.9
Monkey-2B	12M	55.9	37.3	**50.8**	77.3	50.0	**78.0**
LLaVA-7B	20M	10.2	14.6	17.6	62.4	52.8	20.0
Blip2-3B	105M	7.00	5.40	12.2	35.8	41.0	6.80
xGenMM-4B	110M	50.1	18.9	27.2	81.6	59.7	55.4
MiniGPT4-7B	114M	4.40	3.60	11.2	16.4	24.4	6.30
Instructblip-7B	186M	9.60	7.40	18.0	43.2	59.4	9.20
AceRead-4B	110M	51.8	**53.4**	32.5	**82.6**	**68.9**	37.6

dataset statistics detailed in Table 1. These datasets cover a diverse range of visual question-answering tasks, spanning natural scenes, charts, and document images. More details are provided in Appendix C. To address dataset imbalance, we employ downsampling for larger datasets and upsampling for smaller ones, ultimately resulting in 122,959 instruction-tuning pairs. All experiments are conducted on a cluster of 8 NVIDIA GeForce RTX 4090 GPUs for 10 epochs, with a total training time of 24 hours.

4.2 Evaluation

Since the answers generated by VLMs can be phrased differently while conveying the same meaning, traditional keyword-based evaluation methods (Papineni et al., 2002; Levenshtein, 1966) might underestimate their performance. Therefore, we adopt the method proposed by VLMEvalKit (Duan et al., 2024), which employs a large language model to assess the correctness of the generated answers. In our evaluation, we use GPT-4o (Achiam et al., 2023) as the evaluator to determine whether the generated answers are consistent with the reference answers. We provide explicit judgment criteria and examples in the prompt to ensure that the model can accurately recognize the alignment. Ultimately, we assess the performance by calculating the accuracy of the generated answers. For the complete evaluation prompt, please refer to Appendix D.

4.3 Main Results

Table 2 shows a comprehensive comparison of AceRead against seven similarly-sized VLMs across six VQA benchmarks. AceRead achieves state-of-the-art performance on TableVQABench, TextVQA, and A-OKVQA, with a notable improvement of 16 percentage points in accuracy on TableVQABench and 10 percentage points on A-OKVQA. In terms of computational efficiency, AceRead has only 110M trainable parameters, accounting for 2.75% of the total model size. Compared to QFormer-based VLMs such as Blip2 and Instructblip, AceRead demonstrates higher compute efficiency. However, when compared to models that employ MLP-based cross-modal alignments, such as TinyChart, Monkey, and LLaVA, AceRead exhibits lower parameter efficiency.

Although AceRead performs well on scene and chart images, achieving precise visual feature compression enhanced by semantics, it underperforms on text-intensive tasks like DocVQA and InfoVQA compared to TinyChart and Monkey. We attribute this limitation to the choice of image encoder, which is not specifically optimized for processing text-intensive visual inputs. In contrast, TinyChart and Monkey leverage vision encoders designed for document understanding, enabling them to achieve superior performance on these tasks.

4.4 Ablation Study

To validate the effectiveness of the semantic-enhanced visual token querying mechanism and examine how SER performance varies with hyper-parameter changes, we conduct extensive ablation studies as shown in Table 3.

Removing Semantic-Enhanced Resampler. Removing SER and utilizing only visual features for compression (r1 vs r10) leads to a performance decrease. Comparable performance to the semantic-enhanced version is only achieved when increasing the number of learnable queries N_q to 196 (r2 vs r10). This indicates that without semantic guidance, more compressed image tokens

Table 3: Ablation study about semantic-enhanced querying mechanism and different settings about hyperparameters. '# N_q' represents the number of learnable queries, and '# N_c' represents the number of maximum number of image patches after cropping. 'Sent.' is an abbreviation for 'Sentence', indicating the use of features processed by the text encoder instead of word embeddings. 'Qry' is an abbreviation for 'Query', referring to the concatenation of semantic tokens with learnable queries.

	SER	Embedding		Pooling		Concat2		# N_q	# N_c	TextVQA	A-OKVQA	ChartVQA
		Word	Sent.	[CLS]	Mean	Qry	Img					
r1			✓		✓		✓	128	9	75.4	59.1	47.2
r2			✓		✓		✓	196	9	82.3	68.4	51.2
r3	✓	✓			✓		✓	128	9	79.8	67.1	48.8
r4	✓		✓	✓			✓	128	9	82.3	68.6	51.4
r5	✓		✓		✓	✓		128	9	81.0	67.6	51.6
r6	✓		✓		✓		✓	128	1	42.4	31.1	22.5
r7	✓		✓		✓		✓	128	3	51.2	51.3	42.1
r8	✓		✓		✓		✓	128	5	62.3	61.7	50.2
r9	✓		✓		✓		✓	128	12	**82.9**	68.5	51.7
r10	✓		✓		✓		✓	128	9	82.6	**68.9**	**51.8**

are needed to retain question-relevant information. This is also illustrated in Figure 3, where the light blue bars represent performance with SER, and the dark blue bars represent performance without it. As shown, using SER leads to improved results across all six datasets, confirming the necessity of semantic-enhanced resampling.

Using Word Embeddings. In AceRead (r10), semantic tokens are derived from the features processed by a text encoder which captures an overall understanding of the question. The replacement of semantic tokens with word embeddings leads to a performance decrease (r3 vs r10), with a notable drop of 3 percentage points on ChartVQA, indicating that shallow semantics are insufficient to guide visual feature compression effectively.

Pooling with [CLS] Token. In addition to using mean pooling to aggregate semantic tokens, we also leverage the question's [CLS] token as a global semantic representation to guide visual feature compression. However, the model's performance (r4 vs r10) exhibits no improvement, suggesting that the choice between the [CLS] token and mean pooling has minimal impact on feature compression, as both methods retain similar high-level semantics.

Concatenating to Queries. We follow Instructblip's approach by concatenating semantic tokens with learnable queries before compressing visual features. However, we observe a slight performance degradation with this setup (r5 vs r10). A possible reason is that directly concatenating semantic tokens with queries may disrupt the querying mechanism, making it less effective in extracting critical visual features.

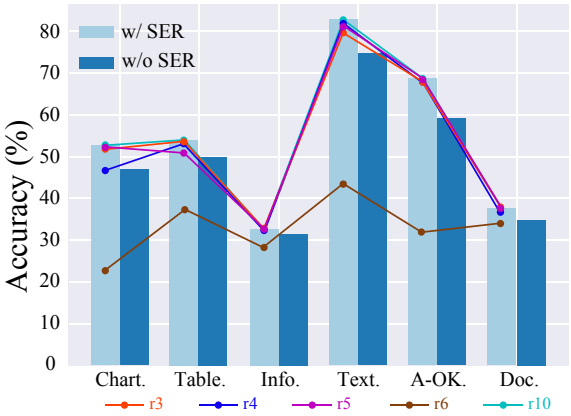


Figure 3: Visualization of AceRead under different settings across six VQA benchmarks. The bar chart compares performance w/ and w/o SER for visual feature compression, showing that incorporating SER leads to overall performance improvements. The line chart illustrates the impact of different hyperparameter settings on performance, with r10 achieving the best results.

Varying Number of Adaptive Grids. We investigate the impact of varying N_c, which controls the number of adaptive grids, on model performance. When adaptive image encoding is disabled ($N_c = 1$) and all images are uniformly resized to the base encoding size, model performance drops significantly (r6 vs r10). This result suggests that fixed resizing leads to a loss of image details, negatively affecting the model's ability to capture visual information. As N_c increases (r7, r8), the adaptive image encoder progressively expands its coverage across different image sizes and aspect ratios, en-

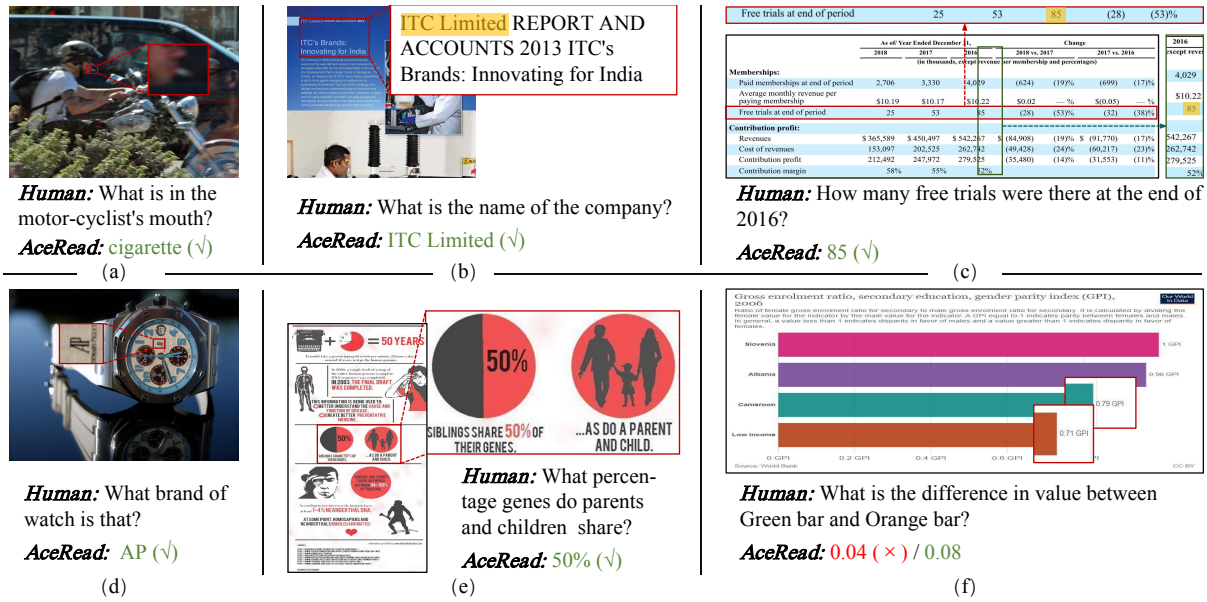


Figure 4: Qualitative results of AceRead across six VQA benchmarks. (a) Scene image from A-OKVQA containing subtle visual details. (b) Document page from DocVQA with dense textual information. (c) Table image from TableVQABench with a structured row-column format. (d) Scene image from TextVQA featuring scene text. (e) Complex chart from InfoVQA requiring an understanding of relationships between different regions. (f) Bar chart from ChartVQA involving arithmetic reasoning.

hancing its ability to encode fine-grained details. This leads to a steady improvement in model performance. However, when N_c reaches 12 (r9), further increasing it does not yield noticeable performance improvement. Since increasing N_c results in a greater number of image tokens, which increases computational overhead, we determine $N_c = 9$ to be the optimal grid size in our experiments.

4.5 Qualitative Results

Figure 4 presents some qualitative results by AceRead across six VQA datasets. For scene images, AceRead demonstrates not only precise detection of subtle visual details (case a) but also robust scene text recognition capabilities (case d). When processing tabular data, the model exhibits strong structural comprehension, effectively locating answers based on row and column relationships (case c). For information-rich visualizations in InfoVQA (case e), AceRead successfully infers the 50% gene proportion from the textual cue "... AS DO A PARENT AND CHILD," indicating its ability to select semantically relevant visual features. The model also shows sophisticated reasoning in complex scenarios where multiple pieces of information need to be synthesized to arrive at the correct answer. In text-intensive tasks (case b), the model demonstrates semantic understanding by correctly mapping the relationship between "name of company" and "brand" to identify "ITC" as the company name. Although AceRead exhibits exceptional comprehension capabilities across diverse VQA tasks, it still faces limitations inherent to the language model's capabilities, particularly in arithmetic reasoning (case f). The model struggles to compute accurate results when questions involve basic arithmetic operations such as addition and subtraction. More qualitative results can be found in Appendix E.

5 Conclusion

We introduce AceRead, a vision-language model that enhances visual feature compression by integrating question semantics. Additionally, AceRead features an adaptive image encoder, enabling it to process images of arbitrary sizes with minimal distortion. Our Semantic-Enhanced Resampler (SER) acts as a conditional information bottleneck, preserving question-relevant details while reducing redundancy. We also conduct extensive ablation studies to confirm its effectiveness in improving visual compression. AceRead achieves state-of-the-art performance across multiple VQA benchmarks while requiring only 2.75% of the model's parameters to be trained. We hope that our findings on semantic-enhanced visual compression will inspire further research in developing more efficient and interpretable multimodal models.

Limitations

Our experiments demonstrate that AceRead effectively compresses visual features while preserving question-relevant details. However, it struggles with text-intensive tasks (e.g., DocVQA and InfoVQA) due to the limitations of its vision encoder, which is not specifically optimized for dense text recognition. Additionally, AceRead has difficulty handling numerical reasoning, as its language model-based decoder lacks strong arithmetic computation abilities, leading to errors in math-related questions. Furthermore, AceRead does not support multi-image question answering, as it processes only a single image at a time, limiting its ability to reason over multiple related visual inputs. In the future, we aim to enhance AceRead's text understanding by integrating more specialized vision encoders and improving its numerical reasoning with external tools or enhanced training strategies. Moreover, extending AceRead to handle multi-image tasks remains an important direction for future research.

References

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

Pravesh Agrawal, Szymon Antoniak, Emma Bou Hanna, Baptiste Bout, Devendra Chaplot, Jessica Chudnovsky, Diogo Costa, Baudouin De Monicault, Saurabh Garg, Theophile Gervet, et al. 2024. Pixtral 12b. *arXiv preprint arXiv:2410.07073*.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. 2022. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736.

Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. 2016. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*, 1(2):3.

Wenliang Dai, Junnan Li, D Li, AMH Tiong, J Zhao, W Wang, B Li, P Fung, and S Hoi. 2023. Instructblip: Towards general-purpose vision-language models with instruction tuning. arxiv 2023. *arXiv preprint arXiv:2305.06500*, 2.

Hongyuan Dong, Jiawen Li, Bohong Wu, Jiacong Wang, Yuan Zhang, and Haoyuan Guo. 2024. Benchmarking and improving detail image caption. *arXiv preprint arXiv:2405.19092*.

Haodong Duan, Junming Yang, Yuxuan Qiao, Xinyu Fang, Lin Chen, Yuan Liu, Xiaoyi Dong, Yuhang Zang, Pan Zhang, Jiaqi Wang, et al. 2024. Vlmevalkit: An open-source toolkit for evaluating large multi-modality models. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 11198–11201.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Hao Feng, Zijian Wang, Jingqun Tang, Jinghui Lu, Wengang Zhou, Houqiang Li, and Can Huang. 2023. Unidoc: A universal large multimodal model for simultaneous text detection, recognition, spotting and understanding. *arXiv preprint arXiv:2308.11592*.

Zhangwei Gao, Zhe Chen, Erfei Cui, Yiming Ren, Weiyun Wang, Jinguo Zhu, Hao Tian, Shenglong Ye, Junjun He, Xizhou Zhu, et al. 2024. Mini-internvl: a flexible-transfer pocket multi-modal model with 5% parameters and 90% performance. *Visual Intelligence*, 2(1):1–17.

Mingxin Huang, Yuliang Liu, Dingkang Liang, Lianwen Jin, and Xiang Bai. 2024. Mini-monkey: Alleviating the semantic sawtooth effect for lightweight mllms via complementary image pyramid. *arXiv preprint arXiv:2408.02034*.

Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. 2021. Perceiver: General perception with iterative attention. In *International conference on machine learning*, pages 4651–4664. PMLR.

Huawei Ji, Cheng Deng, Bo Xue, Zhouyang Jin, Jiaxin Ding, Xiaoying Gan, Luoyi Fu, Xinbing Wang, and Chenghu Zhou. 2024. Aceparse: A comprehensive dataset with diverse structured texts for academic literature parsing. *arXiv preprint arXiv:2409.10016*.

Yoonsik Kim, Moonbin Yim, and Ka Yeon Song. 2024. Tablevqa-bench: A visual question answering benchmark on multiple table domains. *arXiv preprint arXiv:2404.19205*.

VI Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Proceedings of the Soviet physics doklady*.

Mike Lewis. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Dongxu Li, Junnan Li, and Steven Hoi. 2024a. Blip-diffusion: Pre-trained subject representation for controllable text-to-image generation and editing. *Advances in Neural Information Processing Systems*, 36.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pages 12888–12900. PMLR.

Zhang Li, Biao Yang, Qiang Liu, Zhiyin Ma, Shuo Zhang, Jingxu Yang, Yabo Sun, Yuliang Liu, and Xiang Bai. 2024b. Monkey: Image resolution and text label are important things for large multi-modal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26763–26773.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024a. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2024b. Visual instruction tuning. *Advances in neural information processing systems*, 36.

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.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. 2023. Kosmos-2: Grounding multimodal large language models to the world. *arXiv preprint arXiv:2306.14824*.

Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. 2022. A-okvqa: A benchmark for visual question answering using world knowledge. In *European conference on computer vision*, pages 146–162. Springer.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. 2019. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8317–8326.

Chameleon Team. 2024. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*.

Naftali Tishby, Fernando C Pereira, and William Bialek. 2000. The information bottleneck method. *arXiv preprint physics/0004057*.

Bin Xiao, Haiping Wu, Weijian Xu, Xiyang Dai, Houdong Hu, Yumao Lu, Michael Zeng, Ce Liu, and Lu Yuan. 2024. Florence-2: Advancing a unified representation for a variety of vision tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4818–4829.

Le Xue, Manli Shu, Anas Awadalla, Jun Wang, An Yan, Senthil Purushwalkam, Honglu Zhou, Viraj Prabhu, Yutong Dai, Michael S Ryoo, et al. 2024. xgen-mm (blip-3): A family of open large multimodal models. *arXiv preprint arXiv:2408.08872*.

Jiabo Ye, Anwen Hu, Haiyang Xu, Qinghao Ye, Ming Yan, Guohai Xu, Chenliang Li, Junfeng Tian, Qi Qian, Ji Zhang, et al. 2023a. Ureader: Universal ocr-free visually-situated language understanding with multimodal large language model. *arXiv preprint arXiv:2310.05126*.

Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, et al. 2023b. mplug-owl: Modularization empowers large language models with multimodality. *arXiv preprint arXiv:2304.14178*.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. 2023. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11975–11986.

Liang Zhang, Anwen Hu, Haiyang Xu, Ming Yan, Yichen Xu, Qin Jin, Ji Zhang, and Fei Huang. 2024. Tinychart: Efficient chart understanding with visual token merging and program-of-thoughts learning. *arXiv preprint arXiv:2404.16635*.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*.

A Lower Bound of Information Bottleneck

Define the relevant probability distributions as follows: let $p_\theta(x, y)$ be the joint distribution of the data, $q_\phi(z|x)$ be the encoder's conditional distribution of the latent variable z, and $p_\theta(y|z)$ be the decoder's conditional distribution of the output y. We first derive the variational lower bound of $I(Y; Z)$:

$$
\begin{aligned}
I(Y, Z) =& \mathbb{E}_{p_\theta(y,z)} \left[\log \frac{p_\theta(y, z)}{p_\theta(y)p_\theta(z)} \right] \\
=& \mathbb{E}_{p_\theta(x,y)} \left[\mathbb{E}_{q_\phi(z|x)} \left[\log p_\theta(y|z) \right] \right] \\
& - \mathbb{E}_{p_\theta(x)} \left[\mathbb{E}_{q_{\phi(z|x)}} \left[\log \frac{p_\theta(z)}{q_\phi(z|x)} \right] \right] \\
\geq& \mathbb{E}_{p_\theta(x,y)} \left[\mathbb{E}_{q_\phi(z|x)} \left[\log q_\phi(y|z) \right] \right] \\
& - \mathbb{E}_{p_\theta(x)} \left[\mathbb{E}_{q_{\phi(z|x)}} \left[\log \frac{p_\theta(z)}{q_\phi(z|x)} \right] \right]
\end{aligned}
$$

Here, we apply Jensen's inequality:

$$
\begin{aligned}
& \mathbb{E}_{p_\theta(x,y)} \left[\mathbb{E}_{q_\phi(z|x)} \left[\log p_\theta(y|z) \right] \right] \\
& \quad \geq \mathbb{E}_{p_\theta(x,y)} \left[\mathbb{E}_{q_\phi(z|x)} \left[\log q_\phi(y|z) \right] \right]
\end{aligned}
$$

Next, we derive $I(X; Z)$:

$$
\begin{aligned}
I(X; Z) &= \mathbb{E}_{p_\theta(x,z)} \left[\log \frac{p_\theta(x, z)}{p_\theta(x)p_\theta(z)} \right] \\
&= \mathbb{E}_{p_\theta(x)} \left[\mathbb{E}_{q_\phi(z|x)} \left[\log \frac{q_\phi(z|x)}{p_\theta(z)} \right] \right]
\end{aligned}
$$

Thus, the lower bound of the objective in the Information Bottleneck is:

$$
\begin{aligned}
& I(Y; Z) - \beta I(X; Z) \geq L_{IB} = \\
& \mathbb{E}_{p_\theta(x,y)} \left[\mathbb{E}_{q_\phi(z|x)} \left[\log q_\phi(y|z) \right] \right] \\
& - \mathbb{E}_{p_\theta(x)} \left[\mathbb{E}_{q_{\phi(z|x)}} \left[\log \frac{p_\theta(z)}{q_\phi(z|x)} \right] \right] \\
& - \beta \mathbb{E}_{p_\theta(x)} \left[\mathbb{E}_{q_\phi(z|x)} \left[\log \frac{q_\phi(z|x)}{p_\theta(z)} \right] \right] = \\
& \mathop{\mathbb{E}}_{\substack{z \sim p_\theta(z|x) \\ (x,y) \sim p_\theta(x,y)}} \left[\log q_\phi(y|z) - \beta KL\Big(p_\theta(z|x) || q_\phi(z)\Big) \right]
\end{aligned}
$$

B Instruction Templates

In our framework, we utilize the special token '<|system|>' to explicitly instruct the Vision-Language Model (VLM) to generate a short and concise response. The token '<|user|>' is employed to indicate the question input, with all associated images being prepended to the textual question using the special token '<|image|>' as a designated placeholder. This ensures a structured and consistent input format for the model. The model's response is prefixed with the token '<|assistant|>', maintaining a clear separation between different dialogue components. Furthermore, each segment of the input and output sequence is properly terminated using the '<|end|>' token to delineate boundaries and prevent ambiguity. All experiments in this study adhere to this standardized instruction template, as illustrated in Figure 5, ensuring consistency across different evaluation settings.

Instruction Template for AceRead

<|system|>\nA chat between a curious user and an artificial intelligence assistant. The assistant gives concise answers to the user's questions.**<|end|>**

<|user|>\n**<image>**\nWhat is the second column name?**<|end|>**

<|assistant|>\n2-9 Sept. 1972**<|end|>**

Figure 5: The standardized instruction template used for training AceRead. This template defines the structured input format, where the '<|system|>' token provides high-level directives, the '<|user|>' token introduces the question along with associated images marked by '<|image|>', and the '<|assistant|>' token signals the model's response. Each segment is clearly delineated and terminated with the '<|end|>' token, ensuring consistency in the training process.

C Datasets

In our study, we perform experiments on six diverse VQA datasets, each posing unique challenges and providing a comprehensive evaluation of VLM capabilities.

- **TextVQA** (Singh et al., 2019) addresses the significant challenge of reading and reasoning about text within images, comprising 45,336 questions across 28,408 images, highlighting the gap between human and machine performance in text-based reasoning.

- **A-OKVQA** (Schwenk et al., 2022) consists of 19K questions requiring commonsense and world knowledge. Unlike traditional VQA datasets, A-OKVQA's questions demand more complex reasoning beyond sim-

ple knowledge retrieval, testing models' real-world reasoning capabilities.

- **ChartVQA** (Masry et al., 2022) includes 21K generated questions based on chart summaries. It focuses on logical and arithmetic reasoning over chart data, pushing models to handle both visual features and data tables.

- **InfoVQA** (Mathew et al., 2022) through its collection of infographic images, emphasizes reasoning over document layout, textual content, and graphical elements, challenging models to perform elementary reasoning and arithmetic tasks.

- **TableVQABench** (Kim et al., 2024) is a benchmark for VQA on tables, with 1,500 QA pairs. It focuses on evaluating the ability to reason over both textual and visual information presented in tabular format.

- **DocVQA**(Mathew et al., 2021) contains 50,000 questions on 12,000+ document images, testing models' ability to understand document structure and answer questions that require detailed reading comprehension. It reveals performance gaps between machine and human understanding of document layouts.

D Evaluation Prompts

Figure 9 illustrates the instruction template used for evaluating the correctness of the generated answers. In our evaluation framework, GPT-4 is prompted with a predefined set of criteria to rigorously determine whether a model-generated response aligns with the ground truth answers. Specifically, GPT-4 is instructed to consider an answer correct if it explicitly contains the ground truth, conveys the same meaning using different wording, or is ambiguous to the extent that the true answer cannot be easily inferred. Conversely, an answer is deemed incorrect if it is entirely unrelated to the question or significantly deviates in meaning from the ground truth, such as confusing a "date" with "a dinner party." To ensure objective and consistent evaluation, GPT-4 is strictly required to return only "yes" or "no" as its response, without any additional explanations or justifications. Examples of correct and incorrect classifications are provided in the instructions to guide the model in making precise judgments.

E More Qualitative Results

E.1 Understanding of Natural Images

We selected natural images from A-OKVQA and TextVQA to evaluate AceRead's comprehension, as shown in Figure 6. The two upper subfigures assess image understanding in cases where the answer is not explicitly present, requiring inference from contextual cues. The lower subfigures contain images with directly embedded answers, such as scene text. Here, AceRead must correctly recognize and interpret the text before generating the answer. These examples highlight the model's ability to handle both implicit reasoning and explicit text extraction tasks.

E.2 Understanding of Charts

Figure 7 showcases AceRead's performance on chart-type images, where it leverages textual and graphical cues to locate relevant information. The model successfully identifies proportions in pie charts, trends in line charts, and numerical values in bar charts. However, it struggles in some cases, such as the third subfigure, where it misidentifies the fourth item as the third, likely due to image fragmentation. These errors highlight the need for improved spatial reasoning and layout robustness.

E.3 Understanding of Document Pages

AceRead's performance on document pages is comparatively weaker than on natural images and charts, as shown in Figure 8. Unlike natural images, which contain well-defined visual elements, or charts, which have structured layouts, document pages often introduce noise, distortions, and varied text formats that pose significant challenges. Interestingly, AceRead demonstrates an unexpected ability to correctly recognize handwritten text, as shown in the first subfigure. However, its performance deteriorates when faced with more complex document layouts, such as forms or low-resolution scanned pages. In these cases, noise and artifacts in the document images likely interfere with AceRead's text extraction process, leading to errors in numerical recognition and semantic interpretation. These findings indicate that while AceRead exhibits promising capabilities in understanding structured and semi-structured text, further refinements in layout-aware processing and noise robustness are needed to enhance its performance on document-based tasks.

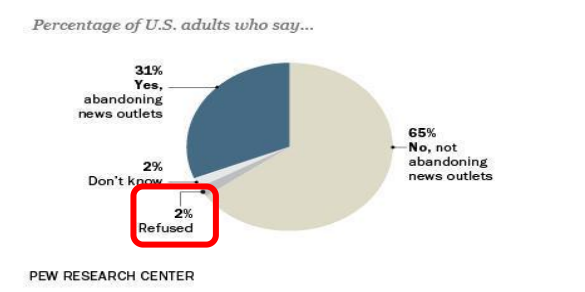
Human: What country do these planes belong to? ***AceRead***: United States

Human: In what nation is this scene located? ***AceRead***: France

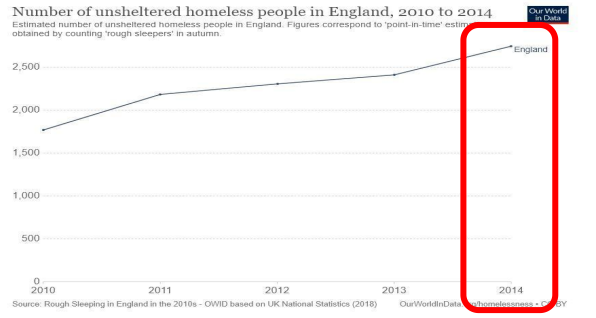
Human: What number is on the player's jersey? ***AceRead***: 22

Human: what kind of beer is this? ***AceRead***: Ale

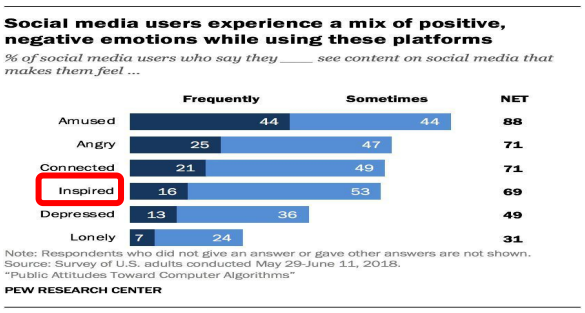
Figure 6: Text Reading and Visual Question Answering performance of AceRead on natural images from A-OKVQA and TextVQA. Correct answers are colored in green.



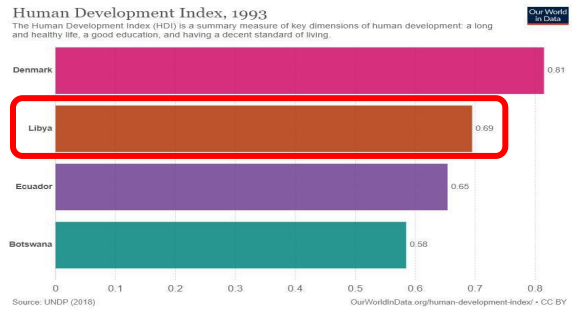
Human: What's the percentage of U.S adults who refused? ***AceRead***: 2



Human: When does the line reach the peak? ***AceRead***: 2014

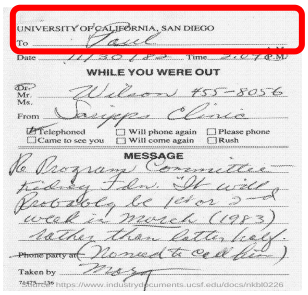


Human: What was the 4th most popular emotion? ***AceRead***: Connected / Inspired

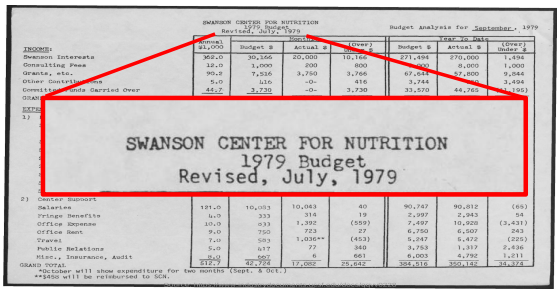


Human: Which country is represented by brown color bar? ***AceRead***: Libya

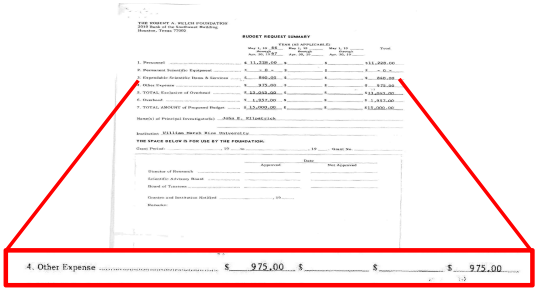
Figure 7: Text Reading and Visual Question Answering performance of AceRead on charts from ChartVQA and TableVQABench. Correct and wrong answers are colored green and red, respectively.



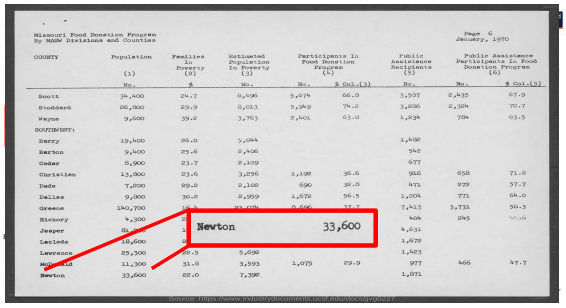
Human: To whom is the document sent? ***AceRead***: The document is sent to Paul.



Human: In which month was the budget 'revised' ? ***AceRead***: 2014



Human: According to budget request summary what is total amount of other expenses? ***AceRead***: $750.00/ $975.00



Human: What is the population in the 'Newton' county? ***AceRead***: 33,870 / 33,600

Figure 8: Text Reading and Visual Question Answering performance of AceRead on document pages from DocVQA and InfoVQA. Correct and wrong answers are colored green and red, respectively.

Instructions to GPT-4o

You are an assistant to help me determine whether the answers generated by my model are correct. I will provide you with the true answer(s) to a question (which may be one or multiple answers; if there are multiple, any one of them is acceptable), and I will also give you the answer generated by my model.You need to judge whether the answer generated by my model is correct.

The following cases are considered correct:
1. The generated answer contains the true answer.
2. The generated answer refers to the same thing as the true answer, but with different wording.
3. It is difficult for me to guess the true answer from the generated answer.

The following cases are considered incorrect:
1. The generated answer is completely unrelated to the question.
2. The generated answer is too far off from the true answer in meaning, such as “date” and “A dinner party.”

Return “yes” if correct, and “no” if incorrect.
Remember, only return “yes” or “no”, do not return anything else.

Example #1:
True answer: New York
Generated answer by model: The state where the license plate was issued is not specified in the image.
Answer: no

Example #2:
True answer: ['2.9%', '2.9']
Generated answer by model: Banks contribute 2.9% of the UK's debt.
Answer: yes

True answer: {real_answer}
Generated answer by model: {generated_answer}
Answer:

Figure 9: Instructions for evaluating whether the generated answer is correct.