Multiple-Question Multiple-Answer Text-VQA

Peng Tang* **Srikar Appalaraju*** **R. Manmatha** **Yusheng Xie** **Vijay Mahadevan**
AWS AI Labs
{tangpen, srikara, manmatha, yushx, vmahad}@amazon.com

Abstract

We present Multiple-Question Multiple-Answer (MQMA), a novel approach to do text-VQA in encoder-decoder transformer models. To the best of our knowledge, almost all previous approaches for text-VQA process a single question and its associated content to predict a single answer. However, in industry applications, users may come up with multiple questions about a single image. In order to answer multiple questions from the same image, each question and content are fed into the model multiple times. In contrast, our proposed MQMA approach takes multiple questions and content as input at the encoder and predicts multiple answers at the decoder in an auto-regressive manner at the same time. We make several novel architectural modifications to standard encoder-decoder transformers to support MQMA. We also propose a novel MQMA denoising pre-training task which is designed to teach the model to align and delineate multiple questions and content with associated answers. MQMA pre-trained model achieves state-of-the-art results on multiple text-VQA datasets, each with strong baselines. Specifically, on OCR-VQA (+2.5%), TextVQA (+1.4%), ST-VQA (+0.6%), DocVQA (+1.1%) absolute improvements over the previous state-of-the-art approaches.

1 Introduction

The task of text-based Visual Question Answering (text-VQA) requires answering questions related to a given image by understanding the text and visual contents in the image. Unlike generic VQA (Antol et al., 2015), where the task is to answer questions mainly using visual information, the text-VQA task involves multiple modalities (*i.e.*, visual, language, and layout) to answer questions (Biten et al., 2022; Hu et al., 2020; Appalaraju et al., 2021; Huang et al., 2022; Kant et al., 2020; Mathew et al., 2021, 2020; Xu et al., 2020; Gao et al., 2024; Xu et al., 2021; Yang et al., 2021; Appalaraju et al., 2024; Tang et al., 2024; Zhuowan et al., 2024). The task needs a model to not only consume multiple modalities (text and image) but also to reason within and across modalities to answer a question (see Figure 1).

In recent years, the text-VQA task has attracted a lot of attention (Biten et al., 2019b; Mathew et al., 2021, 2020; Methani et al., 2020; Mishra et al., 2019; Singh et al., 2019; Tanaka et al., 2021; Li et al., 2022). Almost all text-VQA approaches known to us, consume a single question and associated content to predict a single answer. We call these approaches Single-Question Single-Answer (SQSA) text-VQA, see Figure 2 (a). Typical SQSA approaches (Biten et al., 2022; Hu et al., 2020; Huang et al., 2022; Kant et al., 2020; Powalski et al., 2021; Xu et al., 2021; Yang et al., 2021; Appalaraju et al., 2024) first extract text in a given image using an OCR engine. Then the entire content – image, OCR text and in some cases bounding box information (Biten et al., 2022; Powalski et al., 2021; Appalaraju et al., 2024), along with the text of a single question are fed to a multi-modal transformer model which then predicts an answer.

Industry text-VQA applications often involve multiple questions. For example, a user may ask multiple questions about a single image, or a group of users may ask different questions about the same image (*e.g.*, shipped date, order no., address, *etc.* in Figure 1 (a)). Existing text-VQA models are not well-equipped for answering multiple questions. These models typically process a single question and its associated content to predict a single answer. In order to answer multiple questions from the same image, each question and content are fed into the model multiple times. This is inefficient and can lead to sub-optimal performance (Sec. 5).

MQMA can address the limitations of existing text-VQA models. MQMA takes multiple ques-

*Equal contribution.

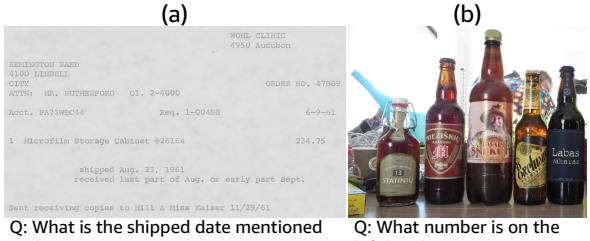

Figure 1: **Examples of text-VQA**. Examples are from (a) DocVQA (Mathew et al., 2021) for document VQA and (b) ST-VQA (Biten et al., 2019b) for scene-text VQA. Answering questions for text-VQA requires multi-modal information, including visual, language, and layout information. Zoom in to see better.

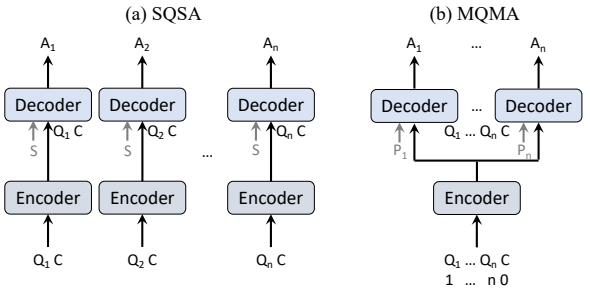

Figure 2: **Single-Question Single-Answer (SQSA) *vs*. Multiple-Question Multiple-Answer (MQMA)**. $Q_i/A_i/P_i$ ($i \in \{1, 2, ..., n\}$): the i-th question/answer/prompt, C: content, S: [START] token for decoder. i ($i \in \{0, 1, 2, ..., n\}$) at the bottom of (b): question index. SQSA and MQMA share the same architecture of encoder and decoder except for the starting token/prompt. The blocks with the same color share the same weights.

tions and content as a single input sequence and predicts multiple answers at the same time. This also opens up a possibility for the model to leverage correlations between multiple questions and content to improve accuracy. Our choice of architecture for MQMA is an encoder-decoder seq-to-seq transformer (Vaswani et al., 2017), see Figure 2 (b). In order to facilitate MQMA in this architecture, we introduce question index embedding at encoder and learnable prompt-based decoding, so that the model learns to align multiple questions and content with the respective predicted answers during auto-regressive decoding (*i.e.*, Q1 $\rightarrow$ A1, Q2 $\rightarrow$ A2 ..., *etc.*). During inference, each answer has its own prompt to associate the corresponding question and content and different answers are decoded separately. At the core of our approach is a novel MQMA unsupervised denosing pre-training task. Unlike the standard denoising language modeling task (Raffel et al., 2020) used in the previous state-of-the-art text-VQA approaches (Biten et al., 2022; Powalski et al., 2021; Appalaraju et al., 2024), our MQMA denoising task pre-trains on unlabeled document data on a proxy VQA task, *i.e.*, a denoising language modeling task formulated as a VQA task, to align the pre-training task and the downstream text-VQA task better. We highlight the contributions of our paper as follows.

- To our knowledge, we are the first to propose MQMA, a novel approach to consume multiple questions and content as a single input sequence and predict multiple answers *at the same time* for text-VQA (see Section 3).
- We also propose an MQMA unsupervised denoising task, a novel way to train a multi-modal encoder-decoder transformer on a denoising language modeling posed as a text-VQA task (see Section 4).
- The MQMA pre-trained model achieves state-of-the-art results on the OCR-VQA, TextVQA, ST-VQA, and DocVQA datasets, each with strong baselines. In particular, +2.5% on OCR-VQA, +1.4% on TextVQA, +0.6% on ST-VQA, and +1.1% on DocVQA (see Section 5).

2 Related Work

Text-VQA has attracted more and more attention recently (Biten et al., 2019b; Kafle et al., 2018; Kahou et al., 2017; Mathew et al., 2022, 2021, 2020; Methani et al., 2020; Mishra et al., 2019; Singh et al., 2019; Tanaka et al., 2021). Focusing on different types of images with texts, several works introduce various text-VQA datasets, including OCR-VQA (Mishra et al., 2019) for book and movie covers, TextVQA (Singh et al., 2019) and ST-VQA (Biten et al., 2019b) for scene-text images, DocVQA (Mathew et al., 2021, 2020) for document images, *etc*. Unlike generic VQA (Antol et al., 2015) which answers questions by reasoning visual contents, text-VQA reasons from both text and visual contents in images to answer questions, which introduces more challenges to the text-VQA task compared with the generic VQA.

The most common text-VQA pipeline first extracts texts and bounding boxes using OCR, and then feed multi-modal inputs (*i.e.*, texts, bounding boxes, and image) into multi-modal models

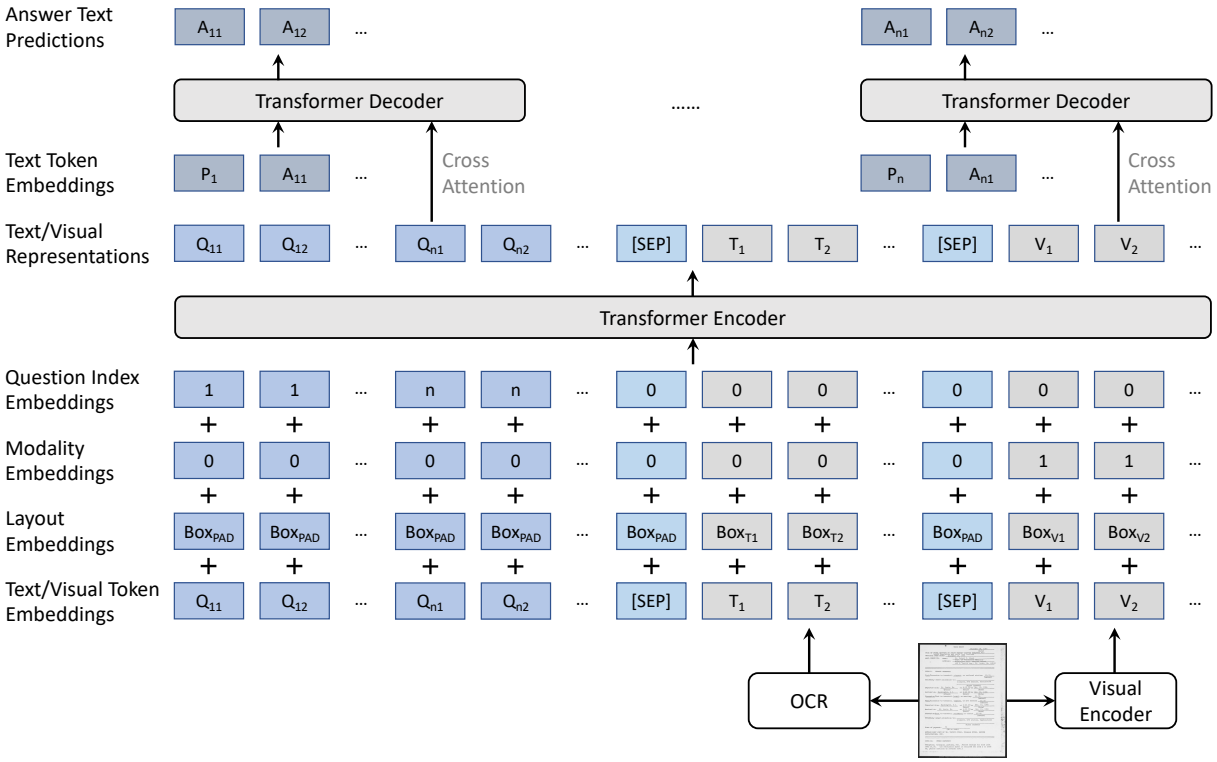

Figure 3: **MQMA Approach:** Encoder-Decoder Transformer model architecture for the proposed MQMA approach. Please note, transformer decoder has shared weights and is to be interpreted as a single decoder.

(*e.g.*, multi-modal transformers) to get predictions (Biten et al., 2022; Gao et al., 2020; Hu et al., 2020; Huang et al., 2022; Kant et al., 2020; Li et al., 2021; Lu et al., 2021; Powalski et al., 2021; Xu et al., 2021; Yang et al., 2021; Appalaraju et al., 2024). Xu et al. (2020) propose LayoutLM based on the encoder only transformer model BERT (Kenton and Toutanova, 2019) by using both language and layout information as inputs. Xu et al. (2021) and Huang et al. (2022) add visual information to the inputs of LayoutLM to improve the accuracy. Hu et al. (2020) and Kant et al. (2020) use multi-modal transformers to fuse information from different modalities and select answers from either a fixed vocabulary or OCR texts by a pointer network (Vinyals et al., 2015). Biten et al. (2022), Powalski et al. (2021), and Appalaraju et al. (2024) propose encoder-decoder transformer based approaches which encode multi-modal information and decode the answer in an auto-regressive manner (Raffel et al., 2020). These approaches do text-VQA in a Single-Question Single-Answer (SQSA) way by answering a single question at a time. Similar to (Biten et al., 2022; Powalski et al., 2021; Appalaraju et al., 2024), our approach is built on top of encoder-decoder transformers. Unlike previous approaches that answer a single question at a time, our approach answers multiple questions at a time using our proposed Multiple-Question Multiple-Answer (MQMA) approach.

Before fine-tuning on text-VQA datasets, previous approaches pre-train their models on unlabeled data using tasks like masked language modeling (Huang et al., 2022; Xu et al., 2021, 2020; Yang et al., 2021), image-text matching (Yang et al., 2021), and the standard denoising (Biten et al., 2022; Powalski et al., 2021; Appalaraju et al., 2024). These pre-training tasks do not align well with the downstream task text-VQA, which may limit the accuracy on the downstream task. In contrast, we propose a new unsupervised pre-training task MQMA denoising which pre-trains the model in a proxy VQA task. The MQMA denoising task aligns the pre-training task with the downstream task and improves the text-VQA accuracy.

3 MQMA Model Architecture

In this section, we discuss in detail the MQMA model architecture. Our choice of architecture for MQMA is an encoder-decoder transformer model (see Figure 3). This architecture is chosen due to its popularity, versatility, and state-of-the-art text-VQA accuracy (Biten et al., 2022; Powalski et al.,

2021; Appalaraju et al., 2024). In addition, using a vocabulary-free generative decoder lends itself as a generic VQA architecture over approaches which are designed for closed-vocabulary VQA (Antol et al., 2015; Wu et al., 2017). The use of decoder elicits additional challenges for MQMA as it is not obvious how the model can auto-regressively generate multiple answers for arbitrary number (>1) of input questions for a content.

Our MQMA model is built on top of the state-of-the-art multi-modal encoder-decoder model DocFormerv2 (Appalaraju et al., 2024) which is termed as the Single-Question Single-Answer (SQSA) baseline in the experiment section 5. The input questions and content - image, OCR text, layout information are vectorized and fed into the transformer encoder. So the model can process multiple modalities at the same time. See Section 3.1 for more details. The transformer encoder processes these inputs with a series of self-attention layers, feed-forward layers, and layer normalization layers to get transformer encoder representations. This representation is then fed into the transformer decoder, consisting of a series of self-attention layers, cross-attention layers, feed-forward layers, and layer normalization layers, decoding answers as predictions in an auto-regressive manner.

In order to support MQMA functionality, the model needs to be made aware of that the input has multiple questions and that at the decoder, the model needs to appropriately align each question with the predicted answer. To facilitate this behavior, we introduce two key changes to the above described SQSA multi-modal encoder-decoder transformer architecture: **a) Question distinguishing multi-modal encoder** - in order to distinguish different questions and content in the inputs, we introduce a question index embedding layer which uses different embeddings for different questions and content, where the embedding of index i is used for the i-th question and the embedding of index 0 is used for content (see Section 3.1). **b) Learnable prompt at the decoder** - Traditionally, a decoder is trained to auto-regressively predict a token beginning with a fixed `[START]` token (Raffel et al., 2020; Vaswani et al., 2017). Instead, in our approach, we introduce n learnable prompts corresponding to the n questions we fed into the model at the encoder. The decoder auto-regressively predicts n answers beginning with these learnt prompts instead of the `[START]` token. Each question uses a separate prompt to decode the corresponding answer (see Section 3.2).

3.1 Multi-modal Encoder Inputs

Both visual, language, and layout information are important to answer questions for text-VQA. Following common practice (Appalaraju et al., 2024; Biten et al., 2022; Hu et al., 2020; Huang et al., 2022; Kant et al., 2020; Powalski et al., 2021; Xu et al., 2021; Yang et al., 2021), a given input image is first processed by an OCR engine to extract text $\{\mathrm{T}_i\}$ and bounding boxes $\{\mathrm{Box}_{\mathrm{T}i}\}$ ($i \in \{1, 2, 3, ...\}$. The OCR text, OCR bounding boxes, question text ($\mathrm{Q}_{ij}, i \in \{1, 2, ..., n\}, j \in \{1, 2, ...\}$, where n corresponds to the number of questions we want to answer at a time), and the image itself are fed into different embedding layers to get different embeddings for different modalities. Notice that here we use text from all n questions as inputs instead of a single question in previous SQSA approaches (Appalaraju et al., 2024; Biten et al., 2022; Hu et al., 2020; Huang et al., 2022; Kant et al., 2020; Powalski et al., 2021; Xu et al., 2021; Yang et al., 2021). See Figure 3.

Text Embedding. We compute text embeddings for question text and OCR results. For text, we first use the Sentence-piece tokenizer (Wu et al., 2016) to tokenize the text, and we then use a learnable text token embedding layer to get the text token embeddings. In particular, we add a `[SEP]` token between question text tokens and OCR text tokens and append a `[SEP]` token after OCR text tokens. Apart from text token embeddings, we compute layout embeddings of text by using learnable layout embedding layers to map the coordinates $(x_1, y_1, x_2, y_2, w, h)$ of text bounding boxes into layout embeddings, where all coordinates are normalized to [0, 1000]. For question text tokens and `[SEP]`, we use a pseudo box $\texttt{[BOX]}_{\mathrm{PAD}}$ which represents the box $(0, 0, 1000, 1000, 1000, 1000)$ (Appalaraju et al., 2021, 2024; Biten et al., 2022). We also use a learnable modality embedding layer to distinguish text modality and visual modality, where the modality embeddings of 0 are used for the text modality. In addition, we use a learnable question index embedding layer to distinguish different questions and content, where the question index embeddings of i and 0 are used for the i-th question and content respectively. The final text embeddings are the sum of text token, layout, modality and question index embeddings.

Visual Embedding. We compute visual embeddings for the image itself. Given an input im-

age, first we resize the image to height 500 and width 384. Then we split the image into 192 non-overlapped patches with size 32×32. Next we map the patches to embeddings by a linear layer with Layer Normalization (Ba et al., 2016) and get 192 embeddings with dimension d_{emb} which depends on the model size (*e.g.*, 512 for the small size model and 768 for the base size model). After that, we use a linear layer to map the embeddings to the final visual token embeddings $\{\text{V}_i\}_{i=1}^{128}$, $\text{V}_i \in \mathbb{R}^{d_{\text{emb}}}$, which means the final sequence length of the visual embeddings is 128. To compute layout embeddings of the visual part, we first use some learnable layout embedding layers to map the location of the image patches into 192 layout embeddings, and we then use a linear layer to map these 192 layout embeddings into the final 128 layout embeddings. Similar to text embeddings, the final visual embeddings are the sum of visual token embeddings, layout embeddings, modality embeddings, and question index embeddings, where the modality embeddings of 1 and the question index embeddings of 0 are used for visual embeddings.

3.2 Prompt-Based Decoder

In SQSA, it is straightforward to follow the standard decoding steps to do auto-regressive answer prediction beginning with the `[START]` token (Powalski et al., 2021; Vaswani et al., 2017). For MQMA, the most naive way to get multiple answers is to decode the concatenation of multiple answers. More precisely, suppose the answer sequence length is L, to answer n questions, the time complexities of the self-attention layers in decoder of SQSA and MQMA are $n \times O(L^2)$ and $O((n \times L)^2) = n^2 \times O(L^2)$ respectively. Particularly, SQSA can decode n answers in parallel which can benefit from the parallel GPU computations, whereas MQMA has to decode n answers sequentially. All these facts show that decoding the concatenation of multiple answers for MQMA might not be a good choice.

To address the issues mentioned above and enable parallel answer decoding for multiple-answers, we propose a prompt-based approach for the MQMA decoder. More precisely, we use n learnable prompts $\{\text{P}_i\}_{i=1}^{n}$ to decode n answers in parallel. Instead of beginning with the `[START]` token, the decoder begins with the i-th prompt P_i to decode the answer A_i for the i-th question in an auto-regressive manner. These prompts are learnt to associate the corresponding questions and content.

Ip / Target	Standard denoising	MQMA denoising
Original text	Thank you ~~for inviting~~ me to your party ~~last~~ week ...	
Input text	Thank you [MASK$_1$] me to your party [MASK$_2$] week ...	Q_1 Q_2 ... Q_n [SEP] Thank you [MASK$_1$] me to your party [MASK$_2$] week ...
Target	[MASK$_1$] for inviting [MASK$_2$] last ...	A_1 A_2 ... A_n

Table 1: **Pre-training tasks**: Standard *vs*. MQMA denoising.

Compared with SQSA, the prompt-based MQMA decoder has almost the same decoder latency as SQSA because the decoding processes of SQSA and MQMA are the same except for which token the decoder begins with. See Appendix A for analyses on different MQMA approaches and why our approach is most optimal for big-oh complexity.

4 MQMA Unsupervised Pre-training

It is well established that pre-training followed by task specific fine-tuning almost always leads to superior performance when compared with models trained with just supervised fine-tuning (Appalaraju et al., 2021, 2024; Biten et al., 2022; Kenton and Toutanova, 2019; He et al., 2019; Chen et al., 2022; Ho et al., 2022; Brown et al., 2020). Ability to train on vast amounts of unsupervised data has a key role to play in the success of this training strategy. In language domain, a number of pre-training strategies inspired by cloze task (Taylor, 1953) have been designed, *e.g.*, masked language modeling (Kenton and Toutanova, 2019). More recently, a denoising language modeling pre-training task was proposed in the T5 model (Raffel et al., 2020) and this pre-training task has been successfully used in previous text-VQA models like DocFormerv2 (Appalaraju et al., 2024) and LaTr (Biten et al., 2022). The denoising language modeling task is unsupervised. The task masks spans of original text and the objective is to reconstruct the masked text during training (see "Standard denoising" in Table 1).

However, this standard denoising task is not well coordinated with our downstream task of text-VQA (we show in experiments, see Table 8). In order to leverage unsupervised pre-training, we propose a novel MQMA denoising language modeling task as a proxy VQA task. We show that this pre-training not only helps the MQMA setting but also helps in general when the downstream task is text-VQA (see Table 8). More precisely, we modify the standard denoising pre-training task to an MQMA text-VQA task by asking and answering questions on `[MASK]` tokens, see "MQMA denoising" Table 1.

We design `which` and `what` style questions, *i.e.*,
1) *Which text tokens are masked by* `[MASK`$_i$`]` *after "xxx"?*,
2) *What are the masked text tokens of* `[MASK`$_i$`]` *after "xxx"?*
Where `[MASK`$_i$`]` corresponds to the i-th mask and *"xxx"* corresponds to the text before `[MASK`$_i$`]`. The answer to the question above is the original text of `[MASK`$_i$`]`. An example question-answer pair for `[MASK`$_i$`]` is
Q: Which text tokens are masked by `[MASK`$_1$`]` *after "Thank you"? - A: for inviting*

We experimentally show that this novel pre-training task is better aligned with the downstream text-VQA task and benefits the model for text-VQA even if the MQMA setting is not desired. We also tried "before" style question formulation and found it to be not as beneficial when compared with the "after" style. So in experiments we stick to the "after" style questions only. There could be other ways to formulate the questions to get more benefits.

5 Experiments

5.1 Experimental Setup

Datasets and Evaluation Metrics. For unsupervised per-training, we use 1M, 64M, and 64M unlabeled document images from the Industrial Document Library (IDL)[1] dataset for small, base, and large size models, respectively, following (Biten et al., 2022; Appalaraju et al., 2024). For text-VQA, we use OCR-VQA (Mishra et al., 2019) for book/movie cover VQA, TextVQA (Singh et al., 2019) and ST-VQA (Biten et al., 2019b) for scene-text VQA, and DocVQA (Mathew et al., 2021, 2020) for document VQA. See Appendix B for more stats on these datasets. For evaluation, we use Average Normalized Levenshtein Similarity (ANLS) (Biten et al., 2019a) which measures the similarity between predicted and ground truth answers for DocVQA and ST-VQA and the standard VQA accuracy (Antol et al., 2015) for other datasets, following the standard evaluation protocol (Appalaraju et al., 2024; Biten et al., 2019b; Mathew et al., 2021; Mishra et al., 2019; Singh et al., 2019). Higher the better.

Implementation Details. Please see Appendix C for implementation details.

[1] https://www.industrydocuments.ucsf.edu/

Approach	Val Accuracy (%)	Test Accuracy (%)
M4C (Hu et al., 2020)	63.5	63.9
LaAP (Han et al., 2020)	63.8	64.1
LaTr$_{base}$ (Biten et al., 2022)	67.5	67.9
GIT (Wang et al., 2022a)	67.8	68.1
SQSA$_{base}$ (Appalaraju et al., 2024)	69.7	70.3
SQSA$_{large}$ (Appalaraju et al., 2024)	71.1	<u>71.5</u>
MQMA$_{base}$ (ours)	71.9	72.4
MQMA$_{large}$ (ours)	**73.6**	**74.0** (+2.5)

Table 2: **Comparison on OCR-VQA**: We answer 5 questions at a time for MQMA. +2.5% is absolute improvement from the previous state of the art (Appalaraju et al., 2024) in that class. **Bold** indicates best and <u>underline</u> indicates the previous state of the art.

Approach	Val Accuracy (%)	Test Accuracy (%)
LaAP (Han et al., 2020)	41.0	41.4
SA-M4C (Kant et al., 2020)	45.4	44.6
SMA (Gao et al., 2021)	44.5	45.5
M4C (Hu et al., 2020)	47.8	-
LOGOS (Lu et al., 2021)	51.5	51.1
TAP + TAG (Wang et al., 2022b)	53.6	53.7
TAP (Yang et al., 2021)	54.7	54.0
PreSTU (Kil et al., 2022)	56.7	56.3
GIT† (Wang et al., 2022a)	59.9	59.8
LaTr$^{\dagger}_{base}$ (Biten et al., 2022)	59.5	59.6
LaTr$^{\dagger}_{large}$ (Biten et al., 2022)	61.1	61.6
SQSA$^{\dagger}_{base}$ (Appalaraju et al., 2024)	61.6	60.0
SQSA$^{\dagger}_{large}$ (Appalaraju et al., 2024)	65.6	<u>64.0</u>
MQMA$^{\dagger}_{base}$ (ours)	63.1	62.3
MQMA$^{\dagger}_{large}$ (ours)	**66.6**	**65.4** (+1.4)

Table 3: **Comparison on TextVQA**: We answer 2 questions at a time for MQMA. † indicates using the combination of the ST-VQA and TextVQA training sets to train the model.

5.2 Comparisons with State of the Art

Results on OCR-VQA. Table 2 shows results of different approaches on the OCR-VQA (Mishra et al., 2019) dataset. Here we train our model on the training set. We answer 5 questions at a time for MQMA (*i.e.*, $n = 5$) because the accuracy of using different numbers of questions is similar on OCR-VQA (see Table 10 in Appendix). On OCR-VQA, there could be potential information leak from the questions "Is this book related to xxx?" to the answer of the questions "What type of book is this?" / "What is the genre of this book?" if we ask these questions together. To avoid such information leak, we keep these two sets of questions separate and answer them separately. See Appendix F for more detailed analyses. On the OCR-VQA testing set, our MQMA approach obtains accuracy 74.0% which is 2.5% higher than 71.5% of the previous state-of-the-art SQSA approach (Appalaraju et al., 2024) using the large size model.

Results on TextVQA and ST-VQA. Following previous approaches (Biten et al., 2022; Appalaraju et al., 2024), we train our models on the combina-

Approach	Val ANLS (%)	Test ANLS (%)
M4C (Hu et al., 2020)	47.2	46.2
LaAP (Han et al., 2020)	49.7	48.5
SA-M4C (Kant et al., 2020)	51.2	50.4
LOGOS (Lu et al., 2021)	58.1	57.9
TAP (Yang et al., 2021)	59.8	59.7
TAP + TAG (Wang et al., 2022b)	62.0	60.2
PreSTU (Kil et al., 2022)	-	65.5
LaTr$^{\dagger}_{\text{base}}$ (Biten et al., 2022)	68.3	68.4
LaTr$^{\dagger}_{\text{large}}$ (Biten et al., 2022)	70.2	69.6
GIT† (Wang et al., 2022a)	69.1	69.6
SQSA$^{\dagger}_{\text{base}}$ (Appalaraju et al., 2024)	70.1	68.4
SQSA$^{\dagger}_{\text{large}}$ (Appalaraju et al., 2024)	72.9	<u>71.8</u>
MQMA$^{\dagger}_{\text{base}}$ (ours)	70.6	70.0
MQMA$^{\dagger}_{\text{large}}$ (ours)	**73.9**	**72.4** (+0.6)

Table 4: **Comparison on ST-VQA**: We answer 2 questions at a time for MQMA. † indicates using the combination of the ST-VQA and TextVQA training sets to train the model.

tion of TextVQA (Singh et al., 2019) and ST-VQA (Biten et al., 2019b) training sets. We answer 2 questions at a time for MQMA (*i.e.*, $n = 2$) because most images in TextVQA and ST-VQA only have 1 or 2 questions. From the results shown in Table 3 and Table 4, our MQMA approach consistently gives the best accuracy on both datasets under different settings. In particular, Table 3 shows that our MQMA approach obtains accuracy 65.4% on the TextVQA testing set, which is 1.4% higher than the previous state-of-the-art SQSA approach (Appalaraju et al., 2024). In addition, on the ST-VQA testing set, our MQMA approach improves ANLS from 71.8% to 72.4% compared with the state-of-the-art SQSA approach (Appalaraju et al., 2024), see Table 4.

Results on DocVQA. Here we compare our approach with the previous state of the art on the DocVQA dataset (Mathew et al., 2021). We train our model on the combination of training and validation set and show results on the testing set (by submitting to leaderboard). We answer 2 questions at a time for MQMA (*i.e.*, $n = 2$) because $n = 2$ gives the best accuracy on DocVQA (see Figure 5 in Appendix). As shown in Table 5, our approach obtains ANLS 88.3% on the DocVQA testing set, 1.1% higher than 87.2% of the previous state-of-the-art SQSA approach (Appalaraju et al., 2024).

See Appendix D for ablation studies on different components of our approach, including the MQMA architecture, the training data augmentation strategy, the unsupervised pre-training task, the question order, and the number of questions.

Approach	Test ANLS (%)
LayoutLMv2$_{\text{base}}$ (Xu et al., 2021)	78.1
LayoutLMv2$_{\text{large}}$ (Xu et al., 2021)	85.3
LayoutLMv3$_{\text{base}}$ (Huang et al., 2022)	78.8
LayoutLMv3$_{\text{large}}$ (Huang et al., 2022)	83.4
StructuralLM$_{\text{large}}$ (Li et al., 2021)	83.9
UDOP$_{\text{large}}$ (Tang et al., 2023)	84.7
ERNIE-Layout$_{\text{large}}$ (Peng et al., 2022)	84.9
TILT$^{\dagger}_{\text{base}}$ (Powalski et al., 2021)	83.9
TILT$^{\dagger}_{\text{large}}$ (Powalski et al., 2021)	87.1
SQSA$_{\text{base}}$ (Appalaraju et al., 2024)	83.4
SQSA$_{\text{large}}$ (Appalaraju et al., 2024)	<u>87.2</u>
ERNIE-Layout$_{\text{ens}}$ (Peng et al., 2022)	88.4
GPT4	88.4
MQMA$_{\text{base}}$ (ours)	**84.8**
MQMA$_{\text{large}}$ (ours)	**88.3** (+1.1)

Table 5: **Comparison on DocVQA**: We answer 2 questions at a time for MQMA. † indicates using more QA datasets instead of only DocVQA to train the model. ERNIE-Layout$_{\text{ens}}$ is the ensemble of 30 models and GPT4 has billions of parameters, both of which are much bigger than MQMA$_{\text{large}}$ using a single model with 750M parameters.

6 Conclusion

In this paper, we propose a Multiple-Question Multiple-Answer (MQMA) text-VQA approach. Unlike previous approaches that process a single question each time, MQMA can answer multiple questions at a time. In addition, we propose an MQMA denoising task for unsupervised pre-training. The MQMA denoising task aligns the pre-training task with the downstream text-VQA task to improve accuracy. Experimental results show that the proposed approach improves accuracy on a variety of challenging text-VQA datasets compared with the previous state of the art.

References

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433.

Srikar Appalaraju, Bhavan Jasani, Bhargava Urala Kota, Yusheng Xie, and R Manmatha. 2021. Docformer: End-to-end transformer for document understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 993–1003.

Srikar Appalaraju, Peng Tang, Qi Dong, Nishant Sankaran, Yichu Zhou, and R. Manmatha. 2024. Docformerv2: Local features for document under-

standing. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(2):709–718.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Ali Furkan Biten, Ron Litman, Yusheng Xie, Srikar Appalaraju, and R Manmatha. 2022. Latr: Layout-aware transformer for scene-text vqa. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16548–16558.

Ali Furkan Biten, Ruben Tito, Andres Mafla, Lluis Gomez, Marçal Rusinol, Minesh Mathew, CV Jawahar, Ernest Valveny, and Dimosthenis Karatzas. 2019a. Icdar 2019 competition on scene text visual question answering. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1563–1570.

Ali Furkan Biten, Ruben Tito, Andres Mafla, Lluis Gomez, Marçal Rusinol, Ernest Valveny, CV Jawahar, and Dimosthenis Karatzas. 2019b. Scene text visual question answering. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4291–4301.

Fedor Borisyuk, Albert Gordo, and Viswanath Sivakumar. 2018. Rosetta: Large scale system for text detection and recognition in images. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 71–79.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, T. J. Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *ArXiv*, abs/2005.14165.

Xi Chen, Xiao Wang, Soravit Changpinyo, A. J. Piergiovanni, Piotr Padlewski, Daniel M. Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, Alexander Kolesnikov, Joan Puigcerver, Nan Ding, Keran Rong, Hassan Akbari, Gaurav Mishra, Linting Xue, Ashish V. Thapliyal, James Bradbury, Weicheng Kuo, Mojtaba Seyedhosseini, Chao Jia, Burcu Karagol Ayan, Carlos Riquelme, Andreas Steiner, Anelia Angelova, Xiaohua Zhai, Neil Houlsby, and Radu Soricut. 2022. Pali: A jointly-scaled multilingual language-image model. *ArXiv*, abs/2209.06794.

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard H. Hovy. 2021. A survey of data augmentation approaches for nlp. *ArXiv*, abs/2105.03075.

Chenyu Gao, Qi Zhu, Peng Wang, Hui Li, Yuliang Liu, Anton Van den Hengel, and Qi Wu. 2021. Structured multimodal attentions for textvqa. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(12):9603–9614.

Difei Gao, Ke Li, Ruiping Wang, Shiguang Shan, and Xilin Chen. 2020. Multi-modal graph neural network for joint reasoning on vision and scene text. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12746–12756.

Yuan Gao, Kunyu Shi, Pengkai Zhu, Edouard Belval, Oren Nuriel, Srikar Appalaraju, Shabnam Ghadar, Vijay Mahadevan, Zhuowen Tu, and Stefano Soatto. 2024. Enhancing vision-language pre-training with rich supervisions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

Wei Han, Hantao Huang, and Tao Han. 2020. Finding the evidence: Localization-aware answer prediction for text visual question answering. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3118–3131.

Xiaoshuai Hao, Yi Zhu, Srikar Appalaraju, Aston Zhang, Wanqian Zhang, Boyang Li, and Mu Li. 2023. Mixgen: A new multi-modal data augmentation. In *IEEE WACV 2023 - Pre train Workshop*, volume abs/2206.08358.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2019. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*.

Chih-Hui Ho, Srikar Appalaraju, Bhavan Jasani, R Manmatha, and Nuno Vasconcelos. 2022. Yoro-lightweight end to end visual grounding. In *European Conference on Computer Vision - ECCV CAMP Workshop*.

Ronghang Hu, Amanpreet Singh, Trevor Darrell, and Marcus Rohrbach. 2020. Iterative answer prediction with pointer-augmented multimodal transformers for textvqa. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9992–10002.

Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. 2022. Layoutlmv3: Pre-training for document ai with unified text and image masking. *arXiv preprint arXiv:2204.08387*.

Kushal Kafle, Brian Price, Scott Cohen, and Christopher Kanan. 2018. Dvqa: Understanding data visualizations via question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5648–5656.

Samira Ebrahimi Kahou, Vincent Michalski, Adam Atkinson, Ákos Kádár, Adam Trischler, and Yoshua Bengio. 2017. Figureqa: An annotated figure dataset for visual reasoning. *arXiv preprint arXiv:1710.07300*.

Yash Kant, Dhruv Batra, Peter Anderson, Alexander Schwing, Devi Parikh, Jiasen Lu, and Harsh Agrawal. 2020. Spatially aware multimodal transformers for textvqa. In *European Conference on Computer Vision*, pages 715–732.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Jihyung Kil, Soravit Changpinyo, Xi Chen, Hexiang Hu, Sebastian Goodman, Wei-Lun Chao, and Radu Soricut. 2022. Prestu: Pre-training for scene-text understanding. *arXiv preprint arXiv:2209.05534*.

Chenge Li, István Fehérvári, Xiaonan Zhao, Ives Macedo, and Srikar Appalaraju. 2022. Seetek: Very large-scale open-set logo recognition with text-aware metric learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2544–2553.

Chenliang Li, Bin Bi, Ming Yan, Wei Wang, Songfang Huang, Fei Huang, and Luo Si. 2021. Structurallm: Structural pre-training for form understanding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6309–6318.

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Xiaopeng Lu, Zhen Fan, Yansen Wang, Jean Oh, and Carolyn P Rosé. 2021. Localize, group, and select: Boosting text-vqa by scene text modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2631–2639.

Edward Ma. 2019. Nlp augmentation. https://github.com/makcedward/nlpaug.

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.

Minesh Mathew, Ruben Tito, Dimosthenis Karatzas, R Manmatha, and CV Jawahar. 2020. Document visual question answering challenge 2020. *arXiv preprint arXiv:2008.08899*.

Nitesh Methani, Pritha Ganguly, Mitesh M Khapra, and Pratyush Kumar. 2020. Plotqa: Reasoning over scientific plots. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1527–1536.

Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. 2019. Ocr-vqa: Visual question answering by reading text in images. In *2019 international conference on document analysis and recognition (ICDAR)*, pages 947–952. IEEE.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Qiming Peng, Yinxu Pan, Wenjin Wang, Bin Luo, Zhenyu Zhang, Zhengjie Huang, Yuhui Cao, Weichong Yin, Yongfeng Chen, Yin Zhang, Shikun Feng, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. 2022. ERNIE-layout: Layout knowledge enhanced pre-training for visually-rich document understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3744–3756, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Rafał Powalski, Łukasz Borchmann, Dawid Jurkiewicz, Tomasz Dwojak, Michał Pietruszka, and Gabriela Pałka. 2021. Going full-tilt boogie on document understanding with text-image-layout transformer. In *International Conference on Document Analysis and Recognition*, pages 732–747.

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*, 21:1–67.

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.

Ryota Tanaka, Kyosuke Nishida, and Sen Yoshida. 2021. Visualmrc: Machine reading comprehension on document images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 13878–13888.

Peng Tang, Pengkai Zhu, Tian Li, Srikar Appalaraju, Vijay Mahadevan, and R Manmatha. 2024. Deed: Dynamic early exit on decoder for accelerating encoder-decoder transformer models. *NAACL Findings*.

Zineng Tang, Ziyi Yang, Guoxin Wang, Yuwei Fang, Yang Liu, Chenguang Zhu, Michael Zeng, Cha Zhang, and Mohit Bansal. 2023. Unifying vision, text, and layout for universal document processing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19254–19264.

Wilson L. Taylor. 1953. "cloze procedure": A new tool for measuring readability. *Journalism & Mass Communication Quarterly*, 30:415 – 433.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. *Advances in neural information processing systems*, 28.

Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. 2022a. Git: A generative image-to-text transformer for vision and language. *arXiv preprint arXiv:2205.14100*.

Jun Wang, Mingfei Gao, Yuqian Hu, Ramprasaath R Selvaraju, Chetan Ramaiah, Ran Xu, Joseph F JaJa, and Larry S Davis. 2022b. Tag: Boosting text-vqa via text-aware visual question-answer generation. *arXiv preprint arXiv:2208.01813*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

Qi Wu, Damien Teney, Peng Wang, Chunhua Shen, Anthony Dick, and Anton Van Den Hengel. 2017. Visual question answering: A survey of methods and datasets. *Computer Vision and Image Understanding*, 163:21–40.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, et al. 2021. Layoutlmv2: Multi-modal pre-training for visually-rich document understanding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2579–2591.

Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. 2020. 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, Yijuan Lu, Jianfeng Wang, Xi Yin, Dinei Florencio, Lijuan Wang, Cha Zhang, Lei Zhang, and Jiebo Luo. 2021. Tap: Text-aware pre-training for text-vqa and text-caption. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8751–8761.

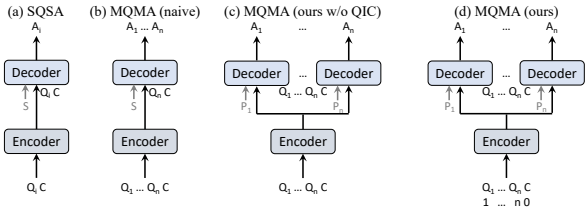

Figure 4: **Architecture Comparisons among SQSA and Different MQMA approaches:** SQSA: the SQSA baseline, MQMA (naive): the naive MQMA approach that concatenates answers of multiple questions to form a single long output sequence, MQMA (ours w/o QIE): our MQMA approach w/o question index embeddings, MQMA (ours): our MQMA approach, Q_i/A_i/P_i ($i \in \{1, 2, ..., n\}$): the i-th question/answer/prompt, C: content, S: [START] token for decoder. i ($i \in \{0, 1, 2, ..., n\}$) at the bottom of (d): question index.

Li Zhuowan, Jasani Bhavan, Tang Peng, and Ghadar Shabnam. 2024. Synthesize step-by-step: Tools, templates and llms as data generators for reasoning-based chart vqa. *arXiv preprint arXiv:2403.16385*.

A Time Complexity and Latency of SQSA and Different MQMA Approaches

We do detailed time complexity and latency analyses of SQSA and different MQMA approaches here. See Figure 4 for the architectures of SQSA and different MQMA approaches. Suppose we have n questions, the sequence length of each question is L_{Q}, the sequence length of content is L_{C}, and the sequence length of each answer is L_{A}. Without loss of generality, $L_{\mathrm{Q}} << L_{\mathrm{C}}$.

For SQSA, to answer each question, the time complexity of each self-attention layer in the encoder is $O\left((L_{\mathrm{Q}} + L_{\mathrm{C}})^2\right) \approx O\left(L_{\mathrm{C}}^2\right)$. The time complexity of each self-attention layer and cross-attention layer in the decoder is $O\left(L_{\mathrm{A}}^2 + L_{\mathrm{A}} * (L_{\mathrm{Q}} + L_{\mathrm{C}})\right) \approx O\left(L_{\mathrm{A}}^2 + L_{\mathrm{A}} * L_{\mathrm{C}}\right)$, where L_{A}^2 is from the self-attention layer and $L_{\mathrm{A}} * L_{\mathrm{C}}$ is from the cross-attention layer. So the encoder and decoder time complexities of answering n questions are $n * O\left(L_{\mathrm{C}}^2\right)$ and $n * O\left(L_{\mathrm{A}}^2 + L_{\mathrm{A}} * L_{\mathrm{C}}\right)$ respectively.

For MQMA (naive), we answer n questions at a time. The time complexity of each self-attention layer in the encoder to answer n questions is $O\left((n * L_{\mathrm{Q}} + L_{\mathrm{C}})^2\right) \approx O\left(L_{\mathrm{C}}^2\right)$ ($n * L_{\mathrm{Q}} << L_{\mathrm{C}}$) which is $\frac{1}{n}$ of the encoder time complexity of SQSA. The time complexity of each self-attention layer and cross-attention layer in the decoder to answer n questions is

	SQSA	MQMA (naive)	MQMA (ours w/o QIE)	MQMA (ours)
Encoder Time Complexity	$n * O\left(L_{\mathrm{C}}^2\right)$	$O\left(L_{\mathrm{C}}^2\right)$	$O\left(L_{\mathrm{C}}^2\right)$	$O\left(L_{\mathrm{C}}^2\right)$
Encoder Latency (ms/image)	19.7	**11.5**	**11.5**	**11.5**
Decoder Time Complexity	$n * O\left(L_{\mathrm{A}}^2 + L_{\mathrm{A}} * L_{\mathrm{C}}\right)$	$n * O\left(n * L_{\mathrm{A}}^2 + L_{\mathrm{A}} * L_{\mathrm{C}}\right)$	$n * O\left(L_{\mathrm{A}}^2 + L_{\mathrm{A}} * L_{\mathrm{C}}\right)$	$n * O\left(L_{\mathrm{A}}^2 + L_{\mathrm{A}} * L_{\mathrm{C}}\right)$
Decoder Latency (ms/image)	**68.9**	77.6	**68.9**	**68.9**

Table 6: **Time Complexity and Latency Comparisons among SQSA and Different MQMA Approaches:** SQSA: the SQSA baseline, MQMA (naive): the naive MQMA approach that concatenates answers of multiple questions to form a single long output sequence, MQMA (ours w/o QIE): our MQMA approach w/o question index embeddings, MQMA (ours): our MQMA approach, n: the number of questions, L_{C}: the sequence length of content, L_{A}: the sequence length of answer. The latency numbers here are from $\mathrm{MQMA}_{\mathrm{small}}$ on DocVQA (Mathew et al., 2021).

Dataset	Train Set	Val Set	Test Set
OCR-VQA (Mishra et al., 2019)	166K/801.7K	20.7K/100K	20.8K/100.4K
TextVQA (Singh et al., 2019)	21.9K/34.6K	3.2K/5K	3.3K/5.7K
ST-VQA (Biten et al., 2019b)	17K/23.4K	1.9K/2.6K	3K/4.1K
DocVQA (Mathew et al., 2020, 2021)	10.2K/39.5K	1.3K/5.3K	1.3K/5.2K

Table 7: **Dataset Stats:** The number of images/questions of different text-VQA datasets.

$O\left((n * L_{\mathrm{A}})^2 + (n * L_{\mathrm{A}}) * (L_{\mathrm{Q}} + L_{\mathrm{C}})\right) \approx n * O\left(n * L_{\mathrm{A}}^2 + L_{\mathrm{A}} * L_{\mathrm{C}}\right)$ which is higher than the decoder time complexity $n * O\left(L_{\mathrm{A}}^2 + L_{\mathrm{A}} * L_{\mathrm{C}}\right)$ of SQSA.

For MQMA (ours w/o QIE) and MQMA (ours), we answer n questions at a time. The time complexity of each self-attention layer in the encoder to answer n questions is the same as MQMA (naive) because the input sequence length of different MQMA approaches is the same. The time complexity of each self-attention layer and cross-attention layer in the decoder to answer n questions is the same as SQSA because we decode n answers separately as in SQSA.

We summarize the time complexities of different approaches and report latency in Table 6. Our MQMA approaches give lower encoder time complexity and latency than SQSA. In addition, the decoder time complexity and latency of MQMA (ours w/o QIE) and MQMA (ours) are the same as that of SQSA and are lower than that of MQMA (naive). So MQMA (ours w/o QIE) and MQMA (ours) give the lowest overall time complexity and latency among all these approaches.

B Datasets

As stated in the main paper, we use OCR-VQA (Mishra et al., 2019) for book/movie cover VQA, TextVQA (Singh et al., 2019) and ST-VQA (Biten et al., 2019b) for scene-text VQA, and DocVQA (Mathew et al., 2021, 2020) for document VQA. See Table 7 for details of these text-VQA datasets. As we can see, there are on average $\sim$ 5 questions/image on OCR-VQA, 1 or 2 questions/image on TextVQA and ST-VQA, and on average $\sim$ 4 questions/image on DocVQA.

C Implementation Details

Pre-training. We use small, base, and large size models which are termed as $\mathrm{MQMA}_{\mathrm{small}}$, $\mathrm{MQMA}_{\mathrm{base}}$, and $\mathrm{MQMA}_{\mathrm{large}}$, respectively. Our model is first initialized from the T5 pre-trained weights (Raffel et al., 2020), then pre-trained on the unlabeled document data following DocFormerv2 (Appalaraju et al., 2024) - we call this model as SQSA basline in our experiments. SQSA is next pre-trained on the same unlabeled document data using the MQMA denoising task descried in Sec. 4 of the main paper. In both, we pre-train for 50/3/3 epochs on 1M/64M/64M IDL data for the small/base/large size model. We also do not do any text augmentation (Ma, 2019; Feng et al., 2021) or multi-modal augmentation (Hao et al., 2023). We simply normalize the images to unit mean and variance for training stability. The maximum input sequence length of the text token embeddings is set to 512. The input sequence length of the visual token embeddings is set to 128. The learnable prompt P_i is first initialized by the embeddings of "answer of question i:".

Fine-Tuning. For text-VQA fine-tuning, we train our models for 8 epochs on OCR-VQA and for 50 epochs on other datasets. The learning rate is set to 0.0001 and the AdamW (Loshchilov and Hutter, 2018) optimizer is used to train our models. Our training batch size is set to 128. The maximum input sequence length of the text token embeddings is set to 2048 for small and base size models and 1024 for large size model. The input sequence length of the visual token embeddings is set to 128.

MQMA Dynamic Data Augmentation. During pre-training and fine-tuning, we use an MQMA specific dynamic data augmentation strategy. Specif-

ically, during unsupervised pre-training, we randomly sample 5 masks at a time with uniform-random order and create 5 questions (as shown in Section 4). During downstream fine-tuning, suppose we want to answer n questions at a time, we randomly sample $n', n' \in \{1, 2, ..., n\}$ question-answer pairs and randomly order the n' question-answer pairs. These randomly sampled and ordered n' question-answer pairs are used during fine-tuning. So if there are m questions for an image, there will be $m^n + m^{n-1} + ... + 1$ random combinations during fine-tuning. We do this to prevent any memorization and learn spurious corelations by the model. During inference, we fix the order of questions and feed every n questions into the model (if the remaining number of questions is smaller than n we simply feed all the remaining questions into the model).

Other Details. Following (Biten et al., 2022; Powalski et al., 2021), we use Amazon Textract[2], Amazon Text-in-Image[3], and Rosetta (Borisyuk et al., 2018) to extract OCR results for document images (*i.e.*, IDL and DocVQA images), non-document images (except for OCR-VQA images), and OCR-VQA images, respectively. Our implementations are based on the PyTorch (Paszke et al., 2019) deep learning framework and the HuggingFace (Wolf et al., 2020) library. All experiments are ran on eight NVIDIA A100 GPUs with `cuda11.x`.

D Ablation Studies on DocVQA

We conduct several ablations on the DocVQA validation set to analyze the influence of different components of our approach, including the MQMA architecture, the training data augmentation strategy, the unsupervised pre-training task, the question order, and the number of questions. If not specified, all experiments here are based on $\text{MQMA}_{\text{small}}$.

The Influence of the MQMA Architecture. As we discussed in Section 3.2, apart from the prompt-based decoder, we can also use a naive approach that concatenates the answers of multiple questions to form a single long output sequence. In addition, we also remove the question index embeddings to check the influence of the question index embeddings. Here we compare these three different MQMA architectures. We do 2 questions 2 answers document VQA (*i.e.*, $n = 2$). As shown in Table 8, our approach obtains higher ANLS than the naive approach. In addition, our approach has lower latency than the naive approach, see Table 6 in Appendix. Adding question index embeddings also contributes to higher ANLS because the question index embeddings help the model distinguish different questions and content.

Approach	Data Aug.	# Questions	ANLS
$\text{SQSA}_{\text{small}}$	-	1	73.0
$\text{MQMA}_{\text{small}}$ (naive)	Static	2	68.6
$\text{MQMA}_{\text{small}}$ (naive)	Dynamic	2	72.3
$\text{MQMA}_{\text{small}}$ (ours w/o QIE)	Dynamic	2	72.7
$\text{MQMA}_{\text{small}}$ (ours)	Dynamic	2	72.9
$\text{MQMA}_{\text{small}}$ (ours) + MQMA denoising	Dynamic	2	**74.3**
$\text{MQMA}_{\text{small}}$ (ours) + MQMA denoising + FDPF	Dynamic	2	74.1

Table 8: **MQMA Ablations:** Results of different MQMA architectures, training data augmentation strategies, and pre-training tasks on the DocVQA validation set. "$\text{MQMA}_{\text{small}}$ (naive)" means the naive approach that concatenates answers of multiple questions to form a single long output sequence. "$\text{MQMA}_{\text{small}}$ (ours w/o QIE)" means our approach w/o question index embeddings. "$\text{MQMA}_{\text{small}}$ (ours)" means our approach. "$\text{MQMA}_{\text{small}}$ (ours) + MQMA denoising" means using MQMA denoising during pre-training (otherwise using standard denoising). "$\text{MQMA}_{\text{small}}$ (ours) + MQMA denoising + FDPF" is the same as "$\text{MQMA}_{\text{small}}$ (ours) + MQMA denoising" except for freezing decoder prompts during fine-tuning. "Static" means that we do static data generation by fixing question-answer pair combinations during training. "Dynamic" means that we do dynamic data generation by randomly sampling and ordering question-answer pairs during training.

MQMA Training Data Augmentation Strategy. As mentioned in Section C. we use a dynamic training data augmentation strategy by randomly sampling and ordering question-answer pairs. Here we compare the dynamic training data augmentation strategy with the static training data generation approach which fixes question-answer pair combinations during training. From Table 8, we can see that using the dynamic approach obtains 3.7% higher ANLS than the static approach.

The Influence of the Unsupervised Pre-training Task. Here we study the influence of different unsupervised pre-training tasks. From Table 8, we can see that adding the MQMA denoising pre-training task improves ANLS by 1.4% when $n = 2$. With the new pre-training task, our MQMA approach obtains 1.3% higher ANLS compared with SQSA. In addition, from Figure 5, we can see when pre-trained with the MQMA denoising task, even $n = 1$ contributes to higher ANLS than the SQSA baseline with the standard denoising task. These re-

[2] https://aws.amazon.com/textract/

[3] https://docs.aws.amazon.com/rekognition/latest/dg/text-detecting-text-procedure.html

Approach	# Questions	ANLS (%)	ANLS of Q1 (%)	ANLS of Q2 (%)
$\text{MQMA}_{\text{small}}$	2	74.3	75.3	73.6
$\text{MQMA}_{\text{small}}$ (reversed order)	2	74.2	73.4	75.2

Table 9: **MQMA Ablations:** Results of different question orders on the DocVQA validation set. The Q1/Q2 for $\text{MQMA}_{\text{small}}$ corresponds to Q2/Q1 for $\text{MQMA}_{\text{small}}$ (reversed order).

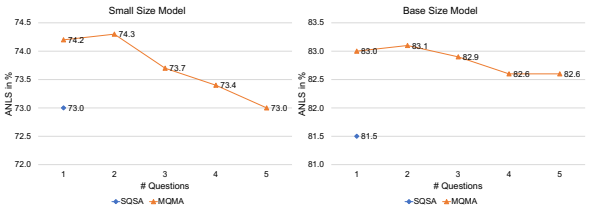

Figure 5: **MQMA Ablations**: Results of different numbers of questions on the DocVQA validation set using the small size and base size models. We use the standard denoising task and the MQMA denoising task for SQSA and MQMA pre-training respectively.

sults confirm that MQMA denoising is beneficial for text-VQA even if $n = 1$. Also, even freezing the decoder prompts during fine-tuning obtains an ANLS of 74.1% (*vs*. 74.3%), which confirms that our pre-training task can learn good decoder prompts to associate the corresponding questions and content even without fine-tuning learnable decoder prompts.

The Influence of the Question Order. In our approach, questions are concatenated with fixed order during inference. Here we study the influence of the question order. From Table 9, we can see our approach is robust to the order of the questions. This is because our model is trained with dynamic data augmentation which randomly samples and orders questions during training.

The Influence of the Number of Questions. We discuss the results of different numbers of questions we answer at a time (*i.e*., different n). As we can see from Figure 5, our MQMA obtains higher accuracy than SQSA for $n = 1$ to 5. Answering 2 questions at a time gives the best accuracy on DocVQA, so we use $n = 2$ in Section 5.2. See Appendix E for the influence of the number of questions on other datasets.

E The Influence of the Number of Questions on Other Datasets

In our main paper, we only show MQMA results of answering 5 questions at a time on OCR-VQA and results of answering 2 questions at a time on TextVQA and ST-VQA. Here we should the influence of the number of questions on OCR-VQA, TextVQA, and ST-VQA datasets. Without loss of generality, we use the base size model and train/test our MQMA approach on the training/validation set.

Approach	# Questions	Accuracy (%)
$\text{SQSA}_{\text{base}}$	1	69.7
$\text{MQMA}_{\text{base}}$	1	70.3
$\text{MQMA}_{\text{base}}$	2	71.7
$\text{MQMA}_{\text{base}}$	3	71.9
$\text{MQMA}_{\text{base}}$	4	71.9
$\text{MQMA}_{\text{base}}$	5	**71.9**

Table 10: **MQMA Ablations:** The influence of the number of questions we answer at a time for MQMA on the OCR-VQA (Mishra et al., 2019) validation set.

Approach	# Questions	TextVQA Accuracy (%)	ST-VQA ANLS (%)
$\text{SQSA}_{\text{base}}$	1	60.4	68.0
$\text{MQMA}_{\text{base}}$	1	61.7	68.7
$\text{MQMA}_{\text{base}}$	2	**61.9**	**69.2**

Table 11: **MQMA Ablations:** The influence of the number of questions we answer at a time for MQMA on the TextVQA (Singh et al., 2019) and ST-VQA (Biten et al., 2019b) validation set.

OCR-VQA. Table 10 shows results of answering different numbers of questions at a time for MQMA on the OCR-VQA (Mishra et al., 2019) validation set. Images in OCR-VQA have on average ~ 5 questions/image, so we compare results of answering $n = 1$ to $n = 5$ questions at a time. As we can see, answering different numbers of questions at a time (when $n > 1$) gives very similar accuracy on the OCR-VQA validation set. Answering $n = 5$ questions at a time gives the highest accuracy on the OCR-VQA validation set, so we only report results of $n = 5$ in our main paper. Answering $n > 1$ questions at a time gives much higher accuracy than answering $n = 1$ question at a time. This is because the questions in the OCR-VQA dataset have correlations. Our MQMA approach can leverage correlations between multiple questions and content to improve accuracy. Even answering $n = 1$ question at a time for MQMA gives higher accuracy than SQSA, because our MQMA denoising pre-training task aligns the pre-training task and downstream text-VQA task.

TextVQA and ST-VQA. Table 11 show results of answering different numbers of questions at a time for MQMA on the TextVQA (Singh et al., 2019) and ST-VQA (Biten et al., 2019b) validation set. Here our model is trained on the TextVQA training

set only when evaluating on the TextVQA validation set, and is trained on the ST-VQA training set only when evaluating on the ST-VQA validation set. Images in TextVQA and ST-VQA have only 1 or 2 questions/image, so we compare results of answering $n = 1$ and $n = 2$ questions at a time. From the results, we can see answering $n = 2$ questions at a time gives slightly higher numbers than answering $n = 1$ question at a time on TextVQA and ST-VQA, so we only report results of $n = 2$ in our main paper. Similar to the results on other datasets, even answering $n = 1$ question at a time for MQMA gives higher accuracy than SQSA thanks to the MQMA denoising pre-training task.

F Information Leak Analyses on OCR-VQA

In our initial experiments on OCR-VQA, we get accuracy 77.5% using the MQMA base size model (*vs*. 69.9% of the SQSA base size model) on the validation set when we answer 5 questions at a time. To verify where such big accuracy improvements are from, we conduct detailed analyses on the OCR-VQA dataset.

Unlike other datasets in which questions of the same image are not strongly correlated, there are correlations among different questions in the OCR-VQA dataset. For most images in OCR-VQA, the five questions below are asked

Q1: Who wrote this book? / Who is the author of this book?
Q2: What is the title of this book?
Q3: What type of book is this? / What is the genre of this book?
Q4: Is this book related to xxx? / Is this a xxx book?
Q5: Is this book related to xxx? / Is this a xxx book?

For Q4 and Q5, one of them has answer "yes" and one of them has answer "no". We can see there are correlations among different questions. For example, the title (for Q2) and the type/genre (for Q3) are correlated to each other. Our MQMA approach can leverage this correlation to improve accuracy.

However, there could be potential information leak from the questions of Q4 and Q5 to the answer of Q3, see the example below.

Q3: What is the genre of this book? - A: religion & spirituality
Q4: Is this book related to religion & spirituality? - A: yes
Q5: Is this book related to computers & technology? - A: no

As we can see, the question of Q4 contains the answer of Q3. In addition, if we evaluate the accuracy of Q3 only and other questions, MQMA gives accuracy 94.0% for Q3 only and 73.2% for other questions, whereas SQSA gives accuracy for 67.0% for Q3 only and 70.7% for other questions. These results show that the MQMA might take information from Q4 or Q5 to answer Q3, *i.e*., there might be information leak.

To further analyze the information leak issue, we conduct experiments under three settings as follows. Here we use the MQMA model trained with $n = 5$ for the experiments and we do not add any constraints during training.

Setting 1: Answer Q1, Q2, Q4, Q5 together and answer Q3 alone.
Setting 2. Answer Q1, Q2, Q3 together and answer Q4, Q5 together.
Setting 3. Answer Q1, Q2, Q3 together, answer Q4 alone, and answer Q5 alone.

Both of these settings give accuracy 71.5%, which further confirms answering Q3, Q4, and Q5 together would result in information leak from the questions of Q4 and Q5 to the answer of Q3. In addition, answering Q4 and Q5 together or alone (Setting 2 and Setting 3) gives the same accuracy, which shows our MQMA approach does not take dataset-specific prior knowledge that there will be one "yes" answer and one "no" answer for Q4 and Q5. This is because during training, we do random sampling and ordering, so the training samples could have different numbers of "yes" answers and different numbers of "no" answers.

To avoid such information leak, we check the whole dataset and make sure all questions that could result in information leak will not be answered together during both training and testing, *e.g*., for the five questions discussed before, we always ensure that Q1, Q2, and Q3 can only be answered together with each other, and Q4 and Q5 can only be answered together with each other. After doing this, we get accuracy 71.9% on the OCR-VQA validation set if we answer $n = 5$ questions at a time.

G Qualitative Results

We show qualitative results in Figure 6. As we can see, our MQMA approach shows better multi-

modal understanding ability than SQSA. There are some failure cases from both MQMA and SQSA. The errors are from multiple aspects, like OCR error and hard images/questions. For example, for the top right example in Figure 6, the ground truth is "6.7" but both MQMA and SQSA give answer "607". The reason of this wrong prediction is from the OCR error - OCR mis-recognizes the word "6.7" as "607" and it is hard for models to fix this OCR error. For the example at the last column of row 3 in Figure 6, both MQMA and SQSA gives wrong counts for the number of letters in the word "police". Counting is a difficult problem for text-VQA models. Actually, MQMA gives a reasonable prediction "7", because from the appearance of the word in the image it looks like there are "7" letters. There are some cases that even human has difficulty in answering the question - for the bottom right example, it is hard to answer the time because there is no clear information about which part corresponds to 12 o'clock.

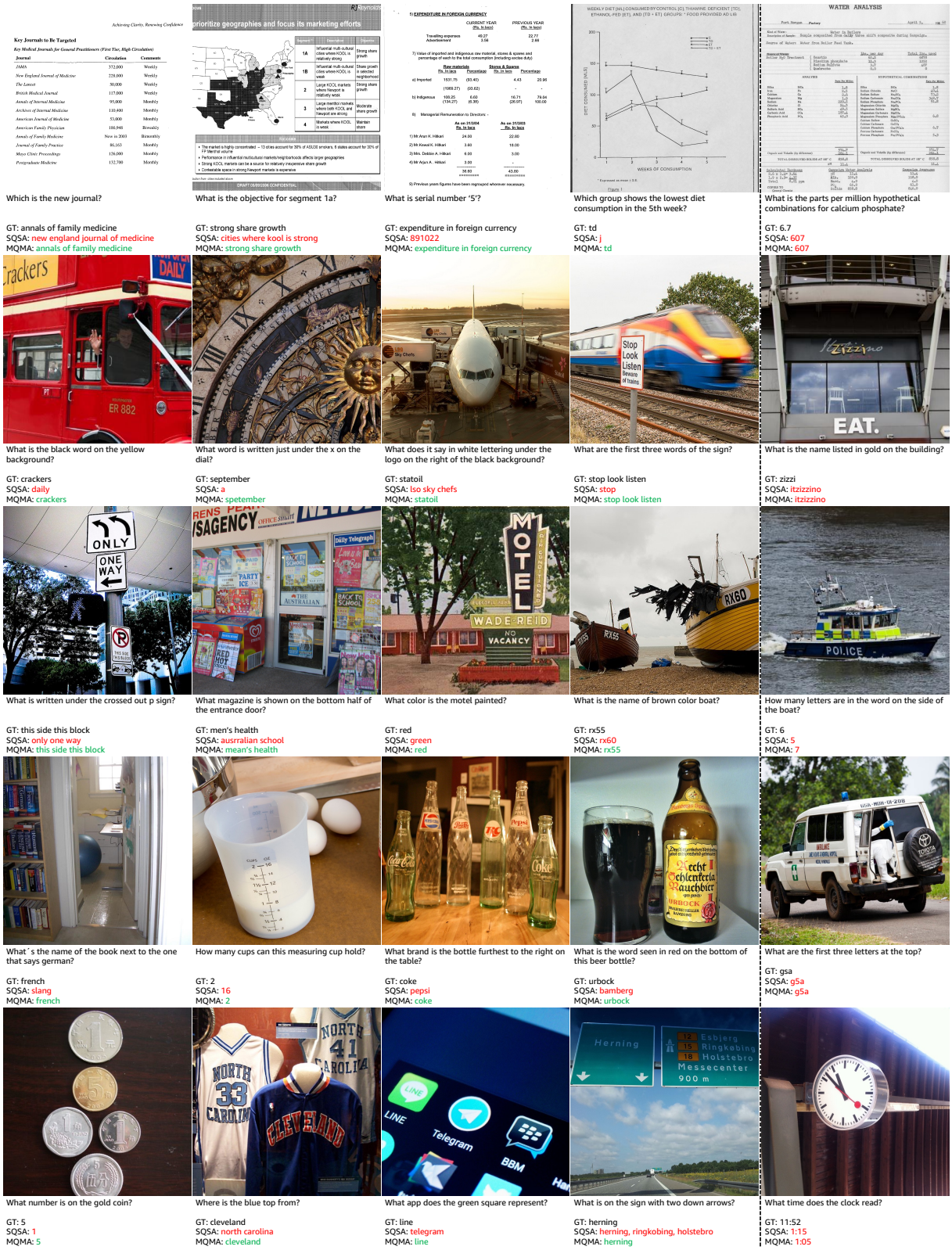

Figure 6: **Qualitative Comparisons between MQMA and SQSA:** The first four columns show examples that MQMA gives correct answers but SQSA gives wrong answers. The last column shows examples that both MQMA and SQSA give wrong answers. MQMA shows better multi-modal understanding ability than SQSA. Zoom in to see better.