Towards Video Text Visual Question Answering: Benchmark and Baseline

Minyi Zhao[1]*, **Bingjia Li**[1]*, **Jie Wang**[2], **Wanqing Li**[2], **Wenjing Zhou**[2], **Lan Zhang**[2]
Shijie Xuyang[1], **Zhihang Yu**[1], **Xinkun Yu**[1], **Guangze Li**[1], **Aobotao Dai**[1], **Shuigeng Zhou**[1]†
[1]Shanghai Key Lab of Intelligent Information Processing, and School of Computer Science, Fudan University, Shanghai 200438, China
[2]ByteDance, China
{zhaomy20, bjli20, abdai20, sgzhou}@fudan.edu.cn
{wangjie.bernard, liwanqing.0415, zhouwenjing.233, zhanglan.11}@bytedance.com
{shijiexuyang, gzlifd}@gmail.com {zhyu21, xkyu21}@m.fudan.edu.cn

Abstract

There are already some *text-based visual question answering* (TextVQA) benchmarks for developing machine's ability to answer questions based on texts in images in recent years. However, models developed on these benchmarks cannot work effectively in many real-life scenarios (e.g. traffic monitoring, shopping ads and e-learning videos) where temporal reasoning ability is required. To this end, we propose a new task named ***Vi**deo **Te**xt **V**isual **Q**uestion **A**nswering* (ViteVQA in short) that aims at answering questions by reasoning texts and visual information spatiotemporally in a given video. In particular, on the one hand, we build the first ViteVQA benchmark dataset named M4-ViteVQA — the abbreviation of **M**ulti-category **M**ulti-frame **M**ulti-resolution **M**ulti-modal benchmark for **ViteVQA**, which contains 7,620 video clips of 9 categories (i.e., *shopping*, *traveling*, *driving*, *vlog*, *sport*, *advertisement*, *movie*, *game* and *talking*) and 3 kinds of resolutions (i.e., 720p, 1080p and 1176×664), and 25,123 question-answer pairs. On the other hand, we develop a baseline method named T5-ViteVQA for the ViteVQA task. T5-ViteVQA consists of five transformers. It first extracts optical character recognition (OCR) tokens, question features, and video representations via two OCR transformers, one language transformer and one video-language transformer, respectively. Then, a multimodal fusion transformer and an answer generation module are applied to fusing multimodal information and generating the final prediction. Extensive experiments on M4-ViteVQA demonstrate the superiority of T5-ViteVQA to the existing approaches of TextVQA and VQA tasks. The ViteVQA benchmark is available at https://github.com/bytedance/VTVQA.

1 Introduction

Several datasets [1, 2, 3, 4, 5, 6, 7] have been built to facilitate the development of *visual question answering* (VQA) [1] methods and systems, but none of them consider the high-level scene text information that is ubiquitous in real-life scenarios and urgently needed for visually-impaired users [8]. Thus, researchers later proposed some *text-based VQA* (TextVQA) benchmarks [8, 9] to promote the research on jointly understanding both visual information and scene texts in images.

Though existing TextVQA benchmarks have greatly advanced TextVQA techniques, all these benchmarks focus on single well-photographed images, which makes developed TextVQA models unable

*Equal contribution. This work is done while authors are interns in ByteDance.
†Corresponding author.

Preprint. Under review.

Figure 1: Some examples from our M4-ViteVQA dataset. 'A' indicates the answers returned by TextVQA models and wrong answers are colored in red. Leveraging temporal, textual, and visual information in the video is the only way to correctly answer the questions.

to answer questions related to or depending on consecutive video frames or events. This is a critical weakness of existing TextVQA methods, which has already realized by some existing work [10], and seriously limits their applications. To make it simpler, let us see the examples in Fig. 1. The 1st case in Fig. 1 is querying the highest fruit price. TextVQA models will get three answers, *i.e.,* $4.90, $3.20, and $3.50 from three single frames, while the correct answer is $4.90. For the 2nd case in Fig 1, considering the scene texts in the video usually suffer from low quality due to motion blur and low resolution, which makes the TextVQA models prone to wrong answers because of lacking global understanding of the texts. As for the 3rd case, to get the correct answer the models must be able to infer temporarily. Unfortunately, such mechanism is unavailable in existing TextVQA models.

To solve the aforementioned problems, in this paper we propose a novel task named ***Vi**deo **Te**xt **V**isual **Q**uestion **A**nswering* (ViteVQA in short), which aims at answering questions by jointly reasoning textual and visual information in a given video. For better understanding, let us go back to Fig. 1. To accurately answer the given questions, the models are required to leverage not only the semantics of texts (in the 1st case, $4.90 is the highest price), visual information (in the 2nd case, the models should recognize the yellow sign) and the spatial relationships among texts and objects (in the 3rd case, the models should know who kills who), but also the temporal relationships among different frames or events (in the 2nd and 3rd cases, the correct answer can be only obtained from the last frame). As ViteVQA is an extension to the TextVQA task, it is more general and has wider applications. Meanwhile, ViteVQA is also a more challenging task as it must jointly exploit both textual and visual information as well as temporal logic among video frames or events.

To support ViteVQA research, on the one hand, we build the first ViteVQA benchmark dataset, which is named **M**ulti-category **M**ulti-frame **M**ulti-resolution **M**ulti-modal benchmark for ViteVQA (M4-ViteVQA in short). M4-ViteVQA consists of 7,620 video clips of nine categories (i.e., *shopping*, *traveling*, *driving*, *vlog*, *sport*, *advertisement*, *movie*, *game* and *talking*) and three kinds of resolutions (i.e., 720p, 1080p and 1176×664), and 25,123 question-answer pairs (QA pairs). On the other hand, we develop a baseline method for the task, which is a novel model called T5-ViteVQA, as it consists of five transformers to conduct both textual and visual understanding as well as temporal reasoning over three modalities: texts from the video, a given question and a video. Specifically, T5-ViteVQA first extracts optical character recognition (OCR) tokens in the form of temporal representation,

question features, and video features via two OCR transformers, one language transformer and one video-language transformer, respectively. Then, a multimodal fusion transformer is employed to fuse and enhance these features. Finally, an answer generation module is applied to inferring the answer from the OCR tokens and a given vocabulary.

Contributions of this paper are as follows: 1) We propose a novel task of *video text visual question answering* (ViteVQA), which is an extension to the TextVQA task and has broader applications. 2) To support ViteVQA research, we build the first high-diversity benchmark dataset M4-ViteVQA. 3) We develop a baseline method T5-ViteVQA for ViteVQA, which consists of five transformers to conduct joint reasoning over three modal inputs. 4) We conduct extensive experiments on M4-ViteVQA, which show that T5-ViteVQA outperforms the existing methods of TextVQA and VQA tasks.

2 Related Work

2.1 Text-based VQA

Text-based visual question answering (TextVQA) [8] is gaining popularity to answer text-related questions by reading and understanding scene texts in images. As a pioneering work, Singh *et al.* [8] proposed the first dataset TextVQA along with a new framework LoRRA that extends the VQA model Pythia [11] with an OCR attention branch. Following that, several other datasets were introduced. Biten *et al.* [9] built the dataset ST-VQA of daily natural scenes where questions can only be answered with texts in images. OCR-VQA [12] utilizes an existing dataset [13] with cover images of books and contains around 200K QA pairs. DocVQA [14] focuses on the understanding of texts in documents. STE-VQA [15] is the first bilingual dataset containing both English and Chinese question-answer pairs. It also provides a bounding box for each question to indicate the area that contains the answer. Nevertheless, all these datasets focus on single static images while many real-world scenarios provide consecutive videos. Models trained on these TextVQA datasets cannot work well in the video scenarios where temporal reasoning ability is required.

2.2 Video Quesion Answering

As an extension of VQA, *video question answering* (VideoQA) aims to answer questions about *video content*, requiring models to have spatiotemporal reasoning ability. There are already a number of datasets [16, 17, 18, 19, 20, 21, 22] for this task, which contain video clips of different scenes, while all the questions focus on the *visual content* of the videos. There are some other VideoQA datasets [23, 24, 25, 26, 27], most of which are about movies or TV series and provide additional texts like subtitles to help understand the videos. However, all these texts are explicitly provided in textual format, and questions in these datasets still focus on *visual content* while the texts just play an auxiliary role. Differently, ViteVQA considers all texts appearing in the scenes of videos, which can be single words or phrases in any possible fonts or orientations, thus cannot be recognized and understood easily. Furthermore, ViteVQA pays more attention to the interaction between texts and visual information in the videos.

2.3 Feature Representations in TextVQA and VideoQA Models

Most existing methods [28, 29, 30, 31, 32] for TextVQA utilize an OCR system [33] to detect and recognize texts in images and an object detector [34] to extract object region proposals. Then, the two modalities along with the question are reasoned jointly via a multimodal fusion transformer. Apparently, such a paradigm does not exploit temporal information in videos and our M4-ViteVQA dataset. While in VideoQA [35, 36, 37, 38, 39, 40], video features are obtained by sampling certain frames or extracted directly via a 3-D backbone [41, 42]. Some works [43, 44, 45, 46, 40, 47] use pre-training to obtain more comprehensive feature representations. In this paper, we extend all the representations of different modalities to the video level.

2.4 Video OCR Systems

Recently, the detection, tracking and recognition of texts in videos [48] have made great progress thanks to several video text spotting benchmarks [49, 50, 51, 52, 53] and models [54, 55, 56, 57, 58, 59, 60]. However, as mentioned in [53], there still remains a blank for the downstream application of

texts in videos. This work is the first to propose and address the ViteVQA task, which is expected to broaden the research of TextVQA to videos and promote the understanding of texts in videos.

3 Benchmark

3.1 Data Collection and Annotation

Data collection. To obtain abundant videos with various text types, we first selected nine different text-rich scenarios, which correspond to nine video categories: *shopping*, *traveling*, *driving*, *vlog*, *sport*, *advertisement*, *movie*, *game*, and *talking*. Then, 6 workers were employed to search qualified videos with texts from YouTube [3]. For the driving category, we collected additional videos from [51] to enrich the dataset. To avoid copyright violation, workers were required to try their best to download only videos that are available on YouTube with a Creative Commons CC-BY (v3.0) License. After collecting 1,150 raw videos, we further cropped these videos into shorter clips to discard frames without texts and shorten the lengths of videos. Then, we masked all private information in the videos, such as faces. Finally, it took about 30 days for the 6 workers to collect 8,511 video clips with texts.

Table 1: The numbers of videos, frames and questions in each category of M4-ViteVQA.

Category	#Videos	#Frames	#Questions
shopping	847	155,275	3,892
traveling	1,154	219,880	4,291
driving	1,316	148,040	3,272
vlog	947	168,715	2,897
sport	665	133,979	2,072
advertisement	623	113,108	1,264
movie	719	103,429	1,449
game	709	155,645	3,672
talking	640	119,321	2,314
Total	7,620	1,317,392	25,123

Data annotation. In this stage, 11 native English-speaking workers were employed by crowdsourcing to label question-answer (QA) pairs. Similar to the annotation phase in [9], the process of designing QA pairs consists of two steps. In the first step, workers were required to come up with closed-ended questions that can be unambiguously answered by reasoning the texts in the corresponding video. The workers were asked to design three to seven QA pairs for each video. As a question may have different answers, so the workers can list multiple answers for each question, usually a complete answer plus a simplified one. Additionally, the workers were asked to attach each question two extra labels. The first label is from {"easy", "hard"} to indicate the difficulty of question answering. An "easy" question can be answered by just reading the texts from one single frame (*e.g.* the 3rd frame of the 2nd case in Fig. 1), while a "hard" question can be answered only by leveraging two or more frames (*e.g.* the 2nd and 3rd frames of the 3rd case in Fig. 1). The second label is from {"text", "vision", "knowledge"} to indicate what kind of information is required to answer the question. Concretely, a "text" question can be answered by purely understanding the semantics of texts in the video (*e.g.* to answer the question of the 1st case in Fig. 1 by outputting the highest price). A "vision" question is answered by jointly considering both the semantics of texts and the visual features of texts (e.g. color and layout) or the video features (e.g. objects and actions) (*e.g.* the 2nd case in Fig. 1). And a "knowledge" question can only be answered by exploiting external knowledge. The workers were encouraged to design "hard" and "vision" questions to increase the difficulty of the benchmark. After this step, we obtained 31,915 QA pairs. Then, we conducted a second step or the verification step, for which 8 additional workers (different from the 11 annotators) were employed to check the previously designed questions. Different from [9], the 8 workers have the right to delete text-unrelated questions to guarantee the quality of the dataset. As for some ambiguous questions that may generate inconsistent or contradictory answers, the authors decided whether or not to correct or delete them. The entire data annotation stage took nearly 50 days. Finally, we obtained 25,123 QA pairs from 7,620 different videos. The statistics of the dataset is given in Tab. 1. Documents on the license, responsibility agreement and accessibility are given in the supplementary materials.

3.2 Statistic and Analysis Results

We first analyze the numbers of questions, answers and OCR tokens in our dataset. Fig. 2(a) shows the distributions of question length and answer length. The lengths of the majority (99%) questions and answers are under 14 and 10, respectively. The average lengths of questions and answers are 6.75 and 1.94, respectively. The distribution of the number of OCR tokens (extracted by methods in [53, 61]) is given in Fig. 2(b). As can be seen, most videos have 1 to 100 OCR tokens. The average number

[3] https://www.youtube.com/

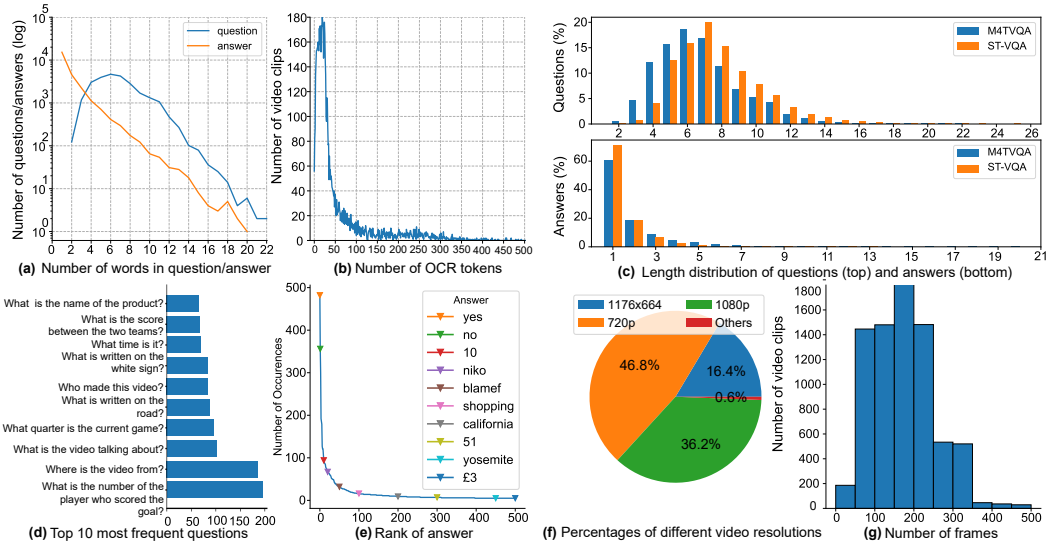


Figure 2: The statistics of questions, answers, OCR tokens and videos of M4-ViteVQA.

of OCR tokens in M4-ViteVQA is 56.92. We also compare the distributions of questions and answers with that of STVQA [9]. As can be seen in Fig. 2(c), the distributions of the two datasets basically follow the same law: the length of the questions first rises and then falls, while the length of the answers shows a downward trend.

Then, we present the statistics of the most frequent questions and the total occurrences of the 500 most common answers in our dataset in Fig. 2(d) and Fig. 2(e), respectively. We can see that there are common questions (*e.g.* "what is the video talking about?"), answers (*e.g.* numbers, shopping) and category-specific QA pairs (*e.g.* names of players and products). The sunburst for the first 4 words in questions is given in Fig. 3. We can see that M4-ViteVQA contains diverse question types for covering various scenes. Besides, the distributions of the two types of labels are: 0.87/0.13 for {"easy", "hard"} and 0.44/0.54/0.02 for {"text", "vision", "knowledge"}. Last, we present two statistics from the perspective of the videos to better demonstrate the diversity of the dataset: 1) The distribution of videos of different resolutions in M4-ViteVQA is shown in Fig. 2(f). There are three resolutions in our dataset: 720p (1280×720), 1080p (1920×1080) and 1176×664. 2) The statistics of the number of frames in the videos is illustrated in Fig. 2(g). The majority of the videos consist of 50 to 250 frames. However, some videos require the models to reason over 300 or even more frames. The different resolutions and lengths of videos not only enrich the diversity of the dataset, but also impose high requirements on models' reading and temporal reasoning abilities.

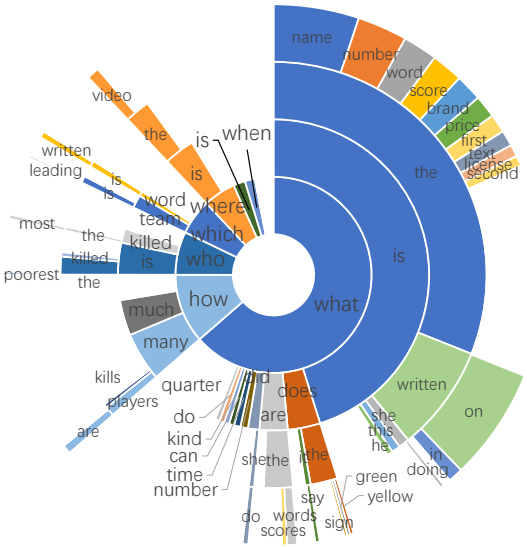


Figure 3: Distribution of the first four words in questions in M4-ViteVQA. Most questions start with "what".

3.3 Tasks and Evaluation Protocol

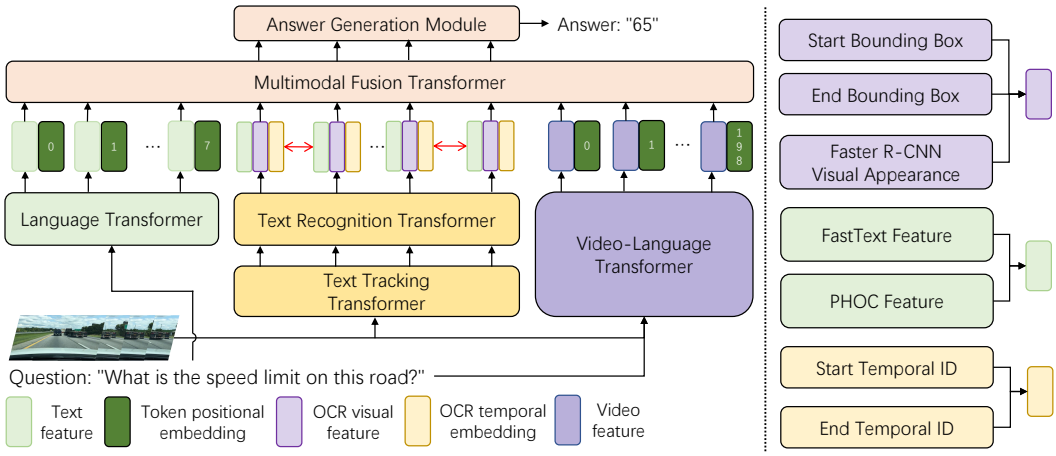


Figure 4: The architecture of T5-ViteVQA.

We define 2 tasks with 3 settings for the ViteVQA problem, namely "regular QA task" (Task1) and "domain adaption task" (Task2). In Task1, the model is trained and tested on all the nine categories of M4-ViteVQA, which is a regular setting. In order to meet the different requirements for the robustness of the model, we consider two data splits for Task1. The first one is called *Task1Split1* that is divided according to the 7,620 cropped videos, the second one is called *Task1Split2* that is divided by the 1,150 raw videos. Task1Split2 is more challenging than Task1Split1 since the content of videos of the same category may be quite different (*e.g.* various shopping venues and sports). In Task2, the model is trained with seven categories while tested on the remaining two categories. Task2 requires the model to deal with unlearned content and completely different category-specific questions, which is very challenging. The statistics of the three settings is given in Tab. 2. It is worth mentioning that for Task2, we provide an extra set, which can be used in different ways (*e.g.* semi-supervised learning, weakly supervised learning etc.) to improve the adaption ability of the model.

Table 2: The numbers of videos (#V) and questions (#Q) in the three settings.

Setting	Task1Split1		Task1Split2		Task2	
	#V	#Q	#V	#Q	#V	#Q
Train	5,373	17,869	5,444	18,220	4,772	14,855
Val	596	1,971	525	1,620	179	762
Test	971	3,183	680	2,103	290	1,221
Extra	-	-	-	-	1,018	4,223

Two metrics are used in this benchmark to evaluate model performance. The first is *accuracy* used in existing TextVQA benchmarks [8, 9]. The second is the *average normalized Levenshtein similarity* (ANLS) [62, 9]. Comparing with the widely used accuracy, ANLS is more tolerant to recognition error, which is more suitable for ViteVQA due to the challenge of reading texts in videos.

4 Baseline

4.1 Architecture

Fig. 4 is the architecture of our baseline T5-ViteVQA, which mainly consists of five transformers and one answer generation module. Given a sample with a video V and a question Q, we first extract the features of the question via a language transformer, the OCR token features with a text tracking transformer and a text recognition transformer, and the video features through a video-language transformer. Then, we adopt a multimodal fusion transformer to fuse these features. Finally, an answer generation module is stacked at the end to generate the answer from the OCR tokens and a given vocabulary. In what follows, we first describe the extraction of multimodal features, then introduce how to fuse them to answer the question, and finally present the training scheme.

4.2 Feature Extraction

Question features. Let $Q = \{Q_i\}_{i=1}^{K}$ be the sequence of the tokenized question tokens, where K is the sequence length of the question, we embed these words into a sequence of d-dimensional feature vectors $X^q = \{x_i^q\}_{i=1}^{K}$ with a pretrained language transformer T_l (*i.e.,* $X^q = T_l(Q)$ where $X^q \in \mathbb{R}^{K\times d}$). In T5-ViteVQA, T_l is implemented by a BERT [63] model.

OCR features. Given the video V with L_v frames, we first employ a text tracking transformer T_t to obtain bounding boxes and tracking ids of the texts in the video frame by frame. After that, we read the texts according to their bounding boxes via a text recognition transformer T_r. Then, for the OCR results that have the same tracking *id*, we merge all the bounding boxes, temporal *ids*, and the recognition results to obtain a temporal OCR token representation. Suppose there are N predicted OCR tokens, the i-th OCR representation O_i can be written as follows: $O_i = \{\{x_{i,j}^{bbox}\}_{j=1}^{L_o^i}, x_i^{s\_tid}, x_i^{e\_tid}, \{x_{i,j}^{ocr\_text}\}_{j=1}^{L_o^i}\}$ where $L_o^i \in [1, L_v]$ is the temporal length of the OCR token, $x_{i,j}^{bbox} \in \mathbb{R}^4$ is the bounding box of the i-th OCR token at time j, $x_i^{s\_tid} \in [1, L_v]$ and $x_i^{e\_tid} \in [1, L_v]$ record when the OCR token appears and disappers (Ergo, we have $L_o^i = x_i^{e\_tid} - x_i^{s\_tid} + 1$), and $x_{i,j}^{ocr\_text} \in \mathbb{R}^{L_{i,j}^{ocr}\times|\mathcal{A}|}$ is the recognition result of the i-th OCR token at time j ($L_{i,j}^{ocr}$ denotes the length of the OCR and $|\mathcal{A}|$ is the size of the alphabet).

After obtaining a set of N OCR tokens in a video through an external OCR system, as shown in Fig. 4, for the i-th token in the N OCR tokens, we further reduce the redundancy in O_i by extracting 1) the start bounding box $x_{i,1}^{bbox}$ and the end bounding box $x_{i,L_o^i}^{bbox}$ as the representative visual geometric features; 2) a 2048-dimensional visual appearance feature x_i^{frcn} from a Faster R-CNN [34] detector via RoI-Pooling the first bounding box of the OCR token whose recognition result occurs the most times (denote the index of this token by k); 3) a 300-dimensional FastText [64] embedding of the recognition result of $x_{i,k}^{ocr\_text}$ to provide essential sub-word information, namely x_i^{ft}; 4) a 604-dimensional Pyramidal Histogram of Characters (PHOC) [65] vector x_i^{phoc} to represent characters presented in the token, which is more robust to OCR error [28]; 5) the start temporal *id* $x_i^{s\_tid}$ and the end temporal *id* $x_i^{e\_tid}$ to provide crucial information for temporal reasoning. It is worth mentioning that for the N OCR tokens, we first sort them according to the reading order and then use a Bi-LSTM (red arrows in Fig. 4) to build their contextual dependence for the enhancement of their semantic features $x_{i,k}^{ocr\_text}$. After rescaling the bounding boxes by dividing the width or height of the video, we project or embed each feature into d-dimensional space, and sum them up [28] (after layer normalization) to get the final representation of the OCR feature:

$$x_i^{ocr} = LN(W_1 x_i^{ft} + W_2 x_i^{frcn} + W_3 x_i^{phoc} + E_1([x_i^{s\_tid}, x_i^{e\_tid}])) + LN(W_4[x_{i,1}^{bbox}, x_{i,L_o^i}^{bbox}]), \quad (1)$$

where W_1, W_2, W_3, and W_4 are learnable projection matrices, E_1 is a embedding layer, $LN(\cdot)$ is layer normalization, and $[\cdot,\cdot]$ denotes the concatenate operation. In T5-ViteVQA, T_t and T_r are implemented by TransVTSpotter [53] and ABINet [61], respectively.

Video features. We apply a video-language transformer T_{vl} to extract question-guided visual information to aid the reasoning. Specifically, we uniformly sample L_{vl} frames from the video V and combine them with the question to extract the video features. Let the sampled L_{vl} frames be V_{vl}. The final video representation is defined as $X^v = T_{vl}(V_{vl}, Q)$ where $X^v \in \mathbb{R}^{M\times d}$ and M is the length of the video features. In our paper, we use All-in-one [40] to implement T_{vl}.

4.3 Feature Fusion and Answer Generation

Given the three multimodal features X^q, $X^{ocr} = \{x_i^{ocr}\}_{i=1}^{N}$ and X^v, we first employ a multimodal fusion transformer T_f that consists of L_f transformer layers [66] to enhance these features in multimodal context via self-attention mechanism. Then, the N enhanced d-dimensional OCR tokens are fed into an answer generation module [28, 44] to infer the answer by selecting words from the OCR tokens in the video and a given vocabulary that is obtained by merging all the tokens in the training set in our experiments (see Sec. 5.1).

Table 3: Performance comparison on M4-ViteVQA dataset.

Method	Task1								Task2			
	Split1				Split2				7 others → *shopping,talking*			
	Val		Test		Val		Test		Val		Test	
	Acc	ANLS	Acc	ANLS	Acc	ANLS	Acc	ANLS	Acc	ANLS	Acc	ANLS
Random	0.56	0.021	0.60	0.025	1.54	0.030	1.38	0.034	1.57	0.030	0.41	0.023
Upper bound	68.44	0.710	68.05	0.714	67.22	0.702	64.76	0.677	66.40	0.698	62.49	0.646
Human	78.08	0.825	85.27	0.893	75.98	0.832	78.41	0.828	83.33	0.859	82.26	0.851
JuskAsk	10.81	0.154	10.05	0.141	7.16	0.100	5.47	0.086	4.86	0.067	3.60	0.067
All-in-one-B	11.47	0.153	10.87	0.148	6.85	0.092	5.66	0.078	4.20	0.050	3.28	0.046
M4C	18.66	0.242	17.91	0.238	13.58	0.172	11.36	0.166	9.16	0.128	7.52	0.125
T5-ViteVQA	**22.07**	**0.282**	**20.23**	**0.266**	**16.42**	**0.226**	**14.31**	**0.207**	**11.81**	**0.152**	**9.17**	**0.136**

4.4 Training Scheme

Following [28, 44], we use teacher-forcing technique [67] and multi-label binary cross-entropy loss $\mathcal{L}_{bce}$ to train the model. Let y_{pred} be the predicted result processed by sigmoid function and y_{gt} be the ground truth, the loss of T5-ViteVQA can be written as follows:

$$\mathcal{L}_{bce} = -y_{gt}\log(y_{pred}) - (1 - y_{gt})\log(1 - y_{pred}). \tag{2}$$

5 Performance Evaluation

5.1 Implementation Details

T5-ViteVQA is implemented in PyTorch1.8. All experiments are conducted on 8 NVIDIA Tesla V100 GPUs with 32GB memory and the same random seed 13. The model is trained using AdamW [68] optimizer with a learning rate of 10^{-4}. The batch size is set to 64. The warm-up learning ratio and warm-up iteration are set as 0.2 and 1,000. K, N, M, L_{vl}, d, and L_f are set to 20, 200, 198, 3, 768 and 4, respectively. The selected frames V_{vl} are resized to 224×224 for saving computational resource. The vocabulary is obtained by merging all the tokens in the training set.

5.2 Experimental Results

We start by giving the performance upper bound and lower bound of ViteVQA as well as human evaluation on the benchmark, then present the performance comparison between our baseline and existing methods of related tasks. Finally, we introduce the results of ablation study.

Performance upper/lower bound and human evaluation. Here, we present the performance upper and lower bounds as well as human evaluation results on the benchmark. All results are given in Tab. 3. The lower bound values (denoted as "Random" in Tab. 3) are obtained by randomly picking texts from the video, and the upper bound values (denoted as "Upper bound" in Tab. 3) are achieved by correctly picking texts, which can be used to evaluate the performance of the OCR system used in the ViteVQA task. We can see that the upper bounds are lower than human evaluation results (indicated by "Human" in Tab. 3), which demonstrates the difficulty of reading texts in videos. Besides, the accuracy of human varies from 75.98% to 85.27%, which is very close to the human evaluation result (84.01%) in TextVQA [8]. This also indicates that human can handle ViteVQA well.

Comparison with existing methods. We reimplement three recent models of related tasks for performance comparison, including M4C [28] for TextVQA, JustAsk [39] for VideoQA and All-in-one [40] for video-language pretraining. The experimental results are given in Tab. 3.

Obviously, JuskAsk and All-in-one perform undesirably in the ViteVQA task since they cannot read texts in videos. M4C performs better than JuskAsk and All-in-one, but is clearly inferior to our method in all the three settings because it cannot do temporal reasoning in videos. Although T5-ViteVQA outperforms the existing techniques in all the three settings, as can be seen in the 3rd row and the last row in Tab. 3, there is still a huge gap between T5-ViteVQA and human evaluation. This indicates that ViteVQA is a difficult task for machine and worthy of further investigation. It is also worth mentioning that the difficulty of the three settings is increasing. All the models perform much worse in Task2 without applying any domain adaption techniques, while human can answer the questions in Task2 well. Therefore, how to use additional data to enhance the generalization power of the model is a significant issue to work on.

Table 4: Detailed performance comparison between M4C and T5-ViteVQA on the validation set of Task1Split1. We do not present the results on the knowledge set because its sample number is too small in the validation set.

Set	M4C	T5-ViteVQA
Easy	19.30	**23.23**
Hard	9.02	**14.66**
Text	17.26	**21.63**
Vision	18.36	**22.65**
Total	17.91	**22.07**

Table 5: Ablation study on the features of T5-ViteVQA. The metric is the accuracy on the validation set of Task1Split1.

X^{ocr}			X^v	Val Acc.
Text	Visual	Temporal		
✓	✓	✓	✓	**22.21**
✗	✗	✗	✓	11.51
✓	✗	✓	✓	16.53
✓	✓	✗	✓	19.78
✓	✓	✓	✗	21.15

In addition, we compare the performance of M4C and T5-ViteVQA on the subsets introduced in Sec. 3.1. As can be seen in Tab. 4, our method T5-ViteVQA performs much better than M4C on these different subsets. Among all these subsets, the performance lift on the Hard subset is the largest one (5.64 *v.s.* 3.93, 4.37 and 4.29 on the other three subsets), which demonstrates the advantage of our method in the ViteVQA task because of its temporal reasoning ability.

Ablation study. We also conduct ablation study to check the design of T5-ViteVQA. As mentioned in Sec. 4, three modal features are extracted in our method, and for the OCR features, we extract textual, temporal and visual information to enrich the representation. In order to see how these features affect the performance of ViteVQA, we design some variants that ignore some specific features. The experimental results are given in Tab. 5. As can be seen in Tab. 5, the best performance is achieved when all the features are used (the 1st row). Besides, the variant that ignores all OCR features has the worst performance (the 2nd row), which explains the importance of OCR tokens in ViteVQA. The performance after dropping the video features X^v is also deteriorated (the 5th row), but the degradation is not as much as that of ignoring visual (the 3rd row) and temporal (the 4th row) features of OCR. We notice that the recent TextVQA works [28, 45] also report this observation. This shows that the usage of video features (*i.e.*, the visual object features in TextVQA) should be improved in both TextVQA and ViteVQA tasks.

6 Limitations and Future Work

This paper has two limitations. On the one hand, M4-ViteVQA supports only question answering. On the other hand, T5-ViteVQA does not use pre-training or domain adaptation techniques to further boost the performance. Therefore, it is worthy for future work to extend M4-ViteVQA to text-based video captioning and retrieval tasks to enrich the video text understanding area. For the second limitation, pre-training or domain adaptation solutions are interesting and promising research topics.

In summary, as a new problem, ViteVQA opens a new direction for VQA or TextVQA over videos, and this work may inspire new research momentum to this area.

7 Conclusion

In this paper, we propose and address a novel problem called *video text visual question answering* (ViteVQA), which requires the model to answer a given question by reading texts and visual information from videos and do temporal reasoning over consecutive events or frames in videos. To support ViteVQA research as a novel problem, we curate the first ViteVQA benchmark dataset named M4-ViteVQA that consists of nine categories of videos with three different resolutions, 7,620 video clips and 25,123 question-answer pairs. We also develop a ViteVQA model as the baseline called T5-ViteVQA, which mainly consists of 5 transformers. T5-ViteVQA first extracts question features, OCR features and video features from three different modal inputs, then fuses these features to generate the final answer. Extensive experiments on M4-ViteVQA show the superiority of our method to the existing techniques of TextVQA, VQA and video-language pretraining.

Acknowledgments and Disclosure of Funding

The work was supported in part by a ByteDance Research Collaboration Project. We thank following workers from ByteDance AI Data Service for labeling M4-ViteVQA: Siew Chin Low, Hailey Kwong, Muhammad Nazhan Naqib Bin Mohd Akib, Szvetlana Melnicsenko, Maria Elena Romo Espinoza, Marta Ramos Baonza, Fernando Pereira, Henrique Revez, Maria Pilar Iglesias Pedreira, Za-K, Weng Kin Law, Selvaraj A/L Alurasamy.

References

[1] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh, "Vqa: Visual question answering," in *ICCV*, 2015, pp. 2425–2433.

[2] Q. Wu, D. Teney, P. Wang, C. Shen, A. Dick, and A. van den Hengel, "Visual question answering: A survey of methods and datasets," *Computer Vision and Image Understanding*, vol. 163, pp. 21–40, 2017.

[3] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh, "Making the v in vqa matter: Elevating the role of image understanding in visual question answering," in *CVPR*, 2017, pp. 6904–6913.

[4] D. Gurari, Q. Li, A. J. Stangl, A. Guo, C. Lin, K. Grauman, J. Luo, and J. P. Bigham, "Vizwiz grand challenge: Answering visual questions from blind people," in *CVPR*, 2018, pp. 3608–3617.

[5] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma *et al.*, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," *IJCV*, vol. 123, no. 1, pp. 32–73, 2017.

[6] Y. Zhu, O. Groth, M. Bernstein, and L. Fei-Fei, "Visual7w: Grounded question answering in images," in *CVPR*, 2016, pp. 4995–5004.

[7] K. Marino, M. Rastegari, A. Farhadi, and R. Mottaghi, "Ok-vqa: A visual question answering benchmark requiring external knowledge," in *CVPR*, 2019, pp. 3195–3204.

[8] A. Singh, V. Natarajan, M. Shah, Y. Jiang, X. Chen, D. Batra, D. Parikh, and M. Rohrbach, "Towards vqa models that can read," in *CVPR*, 2019, pp. 8317–8326.

[9] A. F. Biten, R. Tito, A. Mafla, L. Gomez, M. Rusinol, E. Valveny, C. Jawahar, and D. Karatzas, "Scene text visual question answering," in *ICCV*, 2019, pp. 4291–4301.

[10] W. Chen, X. Wang, and W. Y. Wang, "A dataset for answering time-sensitive questions," in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[11] Y. Jiang, V. Natarajan, X. Chen, M. Rohrbach, D. Batra, and D. Parikh, "Pythia v0. 1: the winning entry to the vqa challenge 2018," *arXiv preprint arXiv:1807.09956*, 2018.

[12] A. Mishra, S. Shekhar, A. K. Singh, and A. Chakraborty, "Ocr-vqa: Visual question answering by reading text in images," in *ICDAR*. IEEE, 2019, pp. 947–952.

[13] B. K. Iwana, S. T. R. Rizvi, S. Ahmed, A. Dengel, and S. Uchida, "Judging a book by its cover," *arXiv preprint arXiv:1610.09204*, 2016.

[14] R. Tito, D. Karatzas, and E. Valveny, "Document collection visual question answering," in *ICDAR*. Springer, 2021, pp. 778–792.

[15] X. Wang, Y. Liu, C. Shen, C. C. Ng, C. Luo, L. Jin, C. S. Chan, A. v. d. Hengel, and L. Wang, "On the general value of evidence, and bilingual scene-text visual question answering," in *CVPR*, 2020, pp. 10 126–10 135.

[16] D. Xu, Z. Zhao, J. Xiao, F. Wu, H. Zhang, X. He, and Y. Zhuang, "Video question answering via gradually refined attention over appearance and motion," in *ACM-MM*, 2017, pp. 1645–1653.

[17] J. Xu, T. Mei, T. Yao, and Y. Rui, "Msr-vtt: A large video description dataset for bridging video and language," in *CVPR*, 2016, pp. 5288–5296.

[18] J. Mun, P. Hongsuck Seo, I. Jung, and B. Han, "Marioqa: Answering questions by watching gameplay videos," in *ICCV*, 2017, pp. 2867–2875.

[19] Y. Jang, Y. Song, Y. Yu, Y. Kim, and G. Kim, "Tgif-qa: Toward spatio-temporal reasoning in visual question answering," in *CVPR*, 2017, pp. 2758–2766.

[20] Z. Yu, D. Xu, J. Yu, T. Yu, Z. Zhao, Y. Zhuang, and D. Tao, "Activitynet-qa: A dataset for understanding complex web videos via question answering," in *AAAI*, vol. 33, no. 01, 2019, pp. 9127–9134.

[21] Y. Ye, Z. Zhao, Y. Li, L. Chen, J. Xiao, and Y. Zhuang, "Video question answering via attribute-augmented attention network learning," in *SIGIR*, 2017, pp. 829–832.

[22] K.-H. Zeng, T.-H. Chen, C.-Y. Chuang, Y.-H. Liao, J. C. Niebles, and M. Sun, "Leveraging video descriptions to learn video question answering," in *AAAI*, 2017.

[23] S. Choi, K.-W. On, Y.-J. Heo, A. Seo, Y. Jang, M. Lee, and B.-T. Zhang, "Dramaqa: Character-centered video story understanding with hierarchical qa," *arXiv preprint arXiv:2005.03356*, 2020.

[24] J. Lei, L. Yu, M. Bansal, and T. Berg, "Tvqa: Localized, compositional video question answering," in *EMNLP*, 2018, pp. 1369–1379.

[25] J. Lei, L. Yu, T. L. Berg, and M. Bansal, "Tvqa+: Spatio-temporal grounding for video question answering," *arXiv preprint arXiv:1904.11574*, 2019.

[26] L. Li, Y.-C. Chen, Y. Cheng, Z. Gan, L. Yu, and J. Liu, "Hero: Hierarchical encoder for video+ language omni-representation pre-training," *arXiv preprint arXiv:2005.00200*, 2020.

[27] M. Tapaswi, Y. Zhu, R. Stiefelhagen, A. Torralba, R. Urtasun, and S. Fidler, "Movieqa: Understanding stories in movies through question-answering," in *CVPR*, 2016, pp. 4631–4640.

[28] R. Hu, A. Singh, T. Darrell, and M. Rohrbach, "Iterative answer prediction with pointer-augmented multimodal transformers for textvqa," in *CVPR*, 2020, pp. 9992–10 002.

[29] C. Gao, Q. Zhu, P. Wang, H. Li, Y. Liu, A. Van den Hengel, and Q. Wu, "Structured multimodal attentions for textvqa," *TPAMI*, 2021.

[30] F. Liu, G. Xu, Q. Wu, Q. Du, W. Jia, and M. Tan, "Cascade reasoning network for text-based visual question answering," in *ACM-MM*, 2020, pp. 4060–4069.

[31] Q. Zhu, C. Gao, P. Wang, and Q. Wu, "Simple is not easy: A simple strong baseline for textvqa and textcaps," *arXiv preprint arXiv:2012.05153*, vol. 2, 2020.

[32] Y. Kant, D. Batra, P. Anderson, A. Schwing, D. Parikh, J. Lu, and H. Agrawal, "Spatially aware multimodal transformers for textvqa," in *ECCV*. Springer, 2020, pp. 715–732.

[33] F. Borisyuk, A. Gordo, and V. Sivakumar, "Rosetta: Large scale system for text detection and recognition in images," in *KDD*, 2018, pp. 71–79.

[34] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *NeurIPS*, vol. 28, 2015.

[35] T. M. Le, V. Le, S. Venkatesh, and T. Tran, "Hierarchical conditional relation networks for video question answering," in *CVPR*, 2020, pp. 9972–9981.

[36] J. Park, J. Lee, and K. Sohn, "Bridge to answer: Structure-aware graph interaction network for video question answering," in *CVPR*, 2021, pp. 15 526–15 535.

[37] M. Bain, A. Nagrani, G. Varol, and A. Zisserman, "Frozen in time: A joint video and image encoder for end-to-end retrieval," in *CVPR*, 2021, pp. 1728–1738.

[38] J. Lei, L. Li, L. Zhou, Z. Gan, T. L. Berg, M. Bansal, and J. Liu, "Less is more: Clipbert for video-and-language learning via sparse sampling," in *CVPR*, 2021, pp. 7331–7341.

[39] A. Yang, A. Miech, J. Sivic, I. Laptev, and C. Schmid, "Just ask: Learning to answer questions from millions of narrated videos," in *ICCV*, 2021, pp. 1686–1697.

[40] A. J. Wang, Y. Ge, R. Yan, G. Yuying, X. Lin, G. Cai, J. Wu, Y. Shan, X. Qie, and M. Z. Shou, "All in one: Exploring unified video-language pre-training," *arXiv preprint arXiv:2203.07303*, 2022.

[41] K. Hara, H. Kataoka, and Y. Satoh, "Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet?" in *CVPR*, 2018, pp. 6546–6555.

[42] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy, "Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 305–321.

[43] Y. Xu, M. Li, L. Cui, S. Huang, F. Wei, and M. Zhou, "Layoutlm: Pre-training of text and layout for document image understanding," in *KDD*, 2020, pp. 1192–1200.

[44] Z. Yang, Y. Lu, J. Wang, X. Yin, D. Florencio, L. Wang, C. Zhang, L. Zhang, and J. Luo, "Tap: Text-aware pre-training for text-vqa and text-caption," in *CVPR*, 2021, pp. 8751–8761.

[45] A. F. Biten, R. Litman, Y. Xie, S. Appalaraju, and R. Manmatha, "Latr: Layout-aware transformer for scene-text vqa," *arXiv preprint arXiv:2112.12494*, 2021.

[46] A. J. Wang, Y. Ge, G. Cai, R. Yan, X. Lin, Y. Shan, X. Qie, and M. Z. Shou, "Object-aware video-language pre-training for retrieval," *arXiv preprint arXiv:2112.00656*, 2021.

[47] J. Wang, Z. Yang, X. Hu, L. Li, K. Lin, Z. Gan, Z. Liu, C. Liu, and L. Wang, "Git: A generative image-to-text transformer for vision and language," *ArXiv*, vol. abs/2205.14100, 2022.

[48] X.-C. Yin, Z.-Y. Zuo, S. Tian, and C.-L. Liu, "Text detection, tracking and recognition in video: a comprehensive survey," *TIP*, vol. 25, no. 6, pp. 2752–2773, 2016.

[49] P. X. Nguyen, K. Wang, and S. Belongie, "Video text detection and recognition: Dataset and benchmark," in *WACV*. IEEE, 2014, pp. 776–783.

[50] X. Zhou, S. Zhou, C. Yao, Z. Cao, and Q. Yin, "Icdar 2015 text reading in the wild competition," *arXiv preprint arXiv:1506.03184*, 2015.

[51] S. Reddy, M. Mathew, L. Gomez, M. Rusinol, D. Karatzas, and C. Jawahar, "Roadtext-1k: Text detection & recognition dataset for driving videos," in *ICRA*. IEEE, 2020, pp. 11 074–11 080.

[52] Z. Cheng, J. Lu, B. Zou, S. Zhou, and F. Wu, "Icdar 2021 competition on scene video text spotting," in *ICDAR*. Springer, 2021, pp. 650–662.

[53] W. Wu, Y. Cai, D. Zhang, S. Wang, Z. Li, J. Li, Y. Tang, and H. Zhou, "A bilingual, open-world video text dataset and end-to-end video text spotter with transformer," *arXiv preprint arXiv:2112.04888*, 2021.

[54] X. Wang, Y. Jiang, S. Yang, X. Zhu, W. Li, P. Fu, H. Wang, and Z. Luo, "End-to-end scene text recognition in videos based on multi frame tracking," in *ICDAR*, vol. 1. IEEE, 2017, pp. 1255–1260.

[55] Z. Cheng, J. Lu, Y. Niu, S. Pu, F. Wu, and S. Zhou, "You only recognize once: Towards fast video text spotting," in *ACM-MM*, 2019, pp. 855–863.

[56] Z. Cheng, J. Lu, B. Zou, L. Qiao, Y. Xu, S. Pu, Y. Niu, F. Wu, and S. Zhou, "Free: A fast and robust end-to-end video text spotter," *TIP*, vol. 30, pp. 822–837, 2020.

[57] H. Yu, Y. Huang, L. Pi, C. Zhang, X. Li, and L. Wang, "End-to-end video text detection with online tracking," *PR*, vol. 113, p. 107791, 2021.

[58] Z. Li, W. Wu, M. Z. Shou, J. Li, S. Li, Z. Wang, and H. Zhou, "Contrastive learning of semantic and visual representations for text tracking," *arXiv preprint arXiv:2112.14976*, 2021.

[59] W. Feng, F. Yin, X.-Y. Zhang, and C.-L. Liu, "Semantic-aware video text detection," in *CVPR*, 2021, pp. 1695–1705.

[60] W. Wu, D. Zhang, Y. Fu, C. Shen, H. Zhou, Y. Cai, and P. Luo, "End-to-end video text spotting with transformer," *arXiv preprint arXiv:2203.10539*, 2022.

[61] S. Fang, H. Xie, Y. Wang, Z. Mao, and Y. Zhang, "Read like humans: autonomous, bidirectional and iterative language modeling for scene text recognition," in *CVPR*, 2021, pp. 7098–7107.

[62] V. I. Levenshtein *et al.*, "Binary codes capable of correcting deletions, insertions, and reversals," in *Soviet physics doklady*, vol. 10, no. 8. Soviet Union, 1966, pp. 707–710.

[63] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *NAACL*, 2019, pp. 4171–4186.

[64] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *TACL*, vol. 5, pp. 135–146, 2017.

[65] J. Almazán, A. Gordo, A. Fornés, and E. Valveny, "Word spotting and recognition with embedded attributes," *TPAMI*, vol. 36, no. 12, pp. 2552–2566, 2014.

[66] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *NeurIPS*, vol. 30, 2017.

[67] A. M. Lamb, A. G. ALIAS PARTH GOYAL, Y. Zhang, S. Zhang, A. C. Courville, and Y. Bengio, "Professor forcing: A new algorithm for training recurrent networks," *NeurIPS*, vol. 29, 2016.

[68] I. Loshchilov and F. Hutter, "Fixing weight decay regularization in adam," 2018.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See Sec. 1.
   (b) Did you describe the limitations of your work? [Yes] See Sec. 6.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Supplementary Material.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments (e.g. for benchmarks)...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See Supplementary Material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Sec. 5.1.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] We pick same random seed.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Sec. 5.1.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See Sec. 5.2.
   (b) Did you mention the license of the assets? [Yes] See Supplementary Material.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] See Sec. 3.1.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] See Sec. 3.1.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [Yes] See Supplementary Material.
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [Yes] See Supplementary Material.

A License and Copyright

As mentioned in our paper, the videos are collected from YouTube. During collection, workers were told to try their best to download only those videos that were available with a Creative Commmons CC-BY (v3.0) License. Since we do not own the copyright for these videos, we provide M4-ViteVQA for non-commercial research purposes only. The detailed license and responsibility agreement are given in the next section.

B Responsibility Agreement

In order to ensure that researchers can reasonably use the data for research purposes only, all researchers must sign the following agreement when using M4-ViteVQA.

- The researcher shall use the M4-ViteVQA dataset only for non-commercial algorithm research and educational purposes. The researcher can not use the M4-ViteVQA dataset for any other purposes, including but not limited to distribution, commercial usage, etc...
- The researcher takes full responsibility for his or her use of the M4-ViteVQA dataset and shall defend and indemnify the dataset, including their affiliates, employees, trustees, officers and agents, against any and all claims arising from the researcher's use of the M4-ViteVQA dataset.
- The researcher agrees and confirms that authors reserve the right to terminate the researcher's access to the M4-ViteVQA dataset at any time.
- If the researcher is employed by a for-profit business entity, the researcher's employer shall also be bound by these terms and conditions, and the researcher hereby shall represent that he or she is fully authorized to enter into this agreement on behalf of such employer.

C Accessibility

To access M4-ViteVQA, researchers must sign Sec. B to get the download links of M4-ViteVQA.

D Ethical Issues and Potential Negative Societal Impacts

All the workers in this paper are employees of ByteDance and are paid according to local standards. Therefore, there are no ethical issues for M4-ViteVQA.

The possible negative societal impact is the personal information in M4-ViteVQA. We have utilized the internal algorithm from ByteDance to preprocess all the videos to mask this information and pass the check.

E Maintenance Plan

Zhao, as the first author, is a Ph.D. student focusing on video and OCR research topics at Fudan University since 2020 and will graduate at least after 2025. Zhao will maintain the benchmark [4] at least until 2025.

F Reproducibility

The code is given in the supplementary material.

G Annotation Instruction

In this section, we give the detailed annotation instructions of M4-ViteVQA. We use ByteDance's internal annotation platform to complete the labeling. Therefore, we cannot provide the annotation interface.

[4] https://github.com/bytedance/VTVQA

G.1 Annotation Step

In brief, given a video clip, we should write 3 to 7 question-answer (QA) pairs based on the texts and visual information from the video. And the answer must come from the video or 'Yes' or 'No'. If the answer is 'Yes' or 'No', it must be answered via reasoning texts in the video.

There are four items you should label for each QA pair.

- Question: An interrogative sentence. That is, a question should start with various types of words including What, How, Where, Is, Does, Do, etc... And ends with '?'. The type of question should be diverse. Nevertheless, it is hard to raise some types of questions for some video clips. Therefore, there should be at least two different question types (Where, Is, Does, etc...) in one video clip. We recommend raising some meaningful and practical questions, which can benefit its downstream applications. We require at least three QA pairs for each video clip. For some text-rich and meaningful video clips, we recommend you write down up to seven questions. We hope that on average there are five questions for each clip. The question should be a text-related question. That is, the question should be answered by reasoning texts in the video.
- A selection of '{Easy,Hard}: Easy: This question can be answered via one static frame. Hard: This question can only be answered via jointly understanding multiple frames.
- A selection of '{Text, Vision, Knowledge}'. Vision: This question should be answered via the texts and the visual information from the video. Text: This question can be answered by purely understanding the semantics of texts in the video. Knowledge: This question should be answered via some external knowledge from life.
- Answer: Standard answers to questions. There are only two sources of answers, and other sources are not allowed: Texts from video; 'Yes' or 'No'. If the answer is from the video, it may be diverse. In this case, please write down the most detailed and original form (Case sensitive). You can write down up to 2 answers split by ';' (semicolon). If the answer is 'Yes' or 'No', we hope the proportion is half-half. That is, the number of 'Yes' and 'No' should be similar.

G.2 Verification Step

In this phase, we should check the quality of the labeling. The key point of this phase is to check whether the question can be answered and whether the provided answer is correct. That is, for each labeled question, we should watch the video, write down your own answer and check whether it is the same as the original one. Besides, please also check whether this answer comes from the video or is 'Yes' or 'No'. If the QA pair is unqualified (the answer does not come from the video), please delete it.

H Introduction of nine categories in M4-ViteVQA

As mentioned in our paper, M4-ViteVQA has 9 categories. These 9 categories cover scene texts recorded from daily life (*i.e.,* shopping and driving) and embedded texts in online media (*i.e.,* game and movie). It also contains various video themes and scenes. Here, we introduce each category to highlight the diversity of our dataset.

H.1 Shopping

This category focuses on understanding the events of shopping (both offline and online). The main questions include asking about prices, products, product features, etc... Both online shopping and offline shopping are included in this category. Besides, wide shopping venues are selected in M4-ViteVQA, including supermarket, shopping mall and etc... Two examples are given in Fig. 5.

H.2 Traveling

The traveling category records some street view videos and descriptions of natural scenes. Two representative cases are given in Fig. 6.

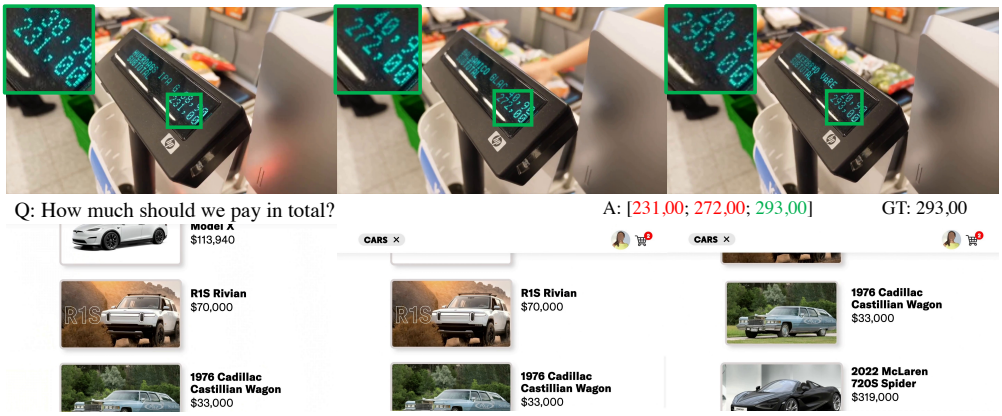


Figure 5: Two examples from shopping category. 'A' indicates the answers returned by TextVQA models and wrong answers are colored in red.

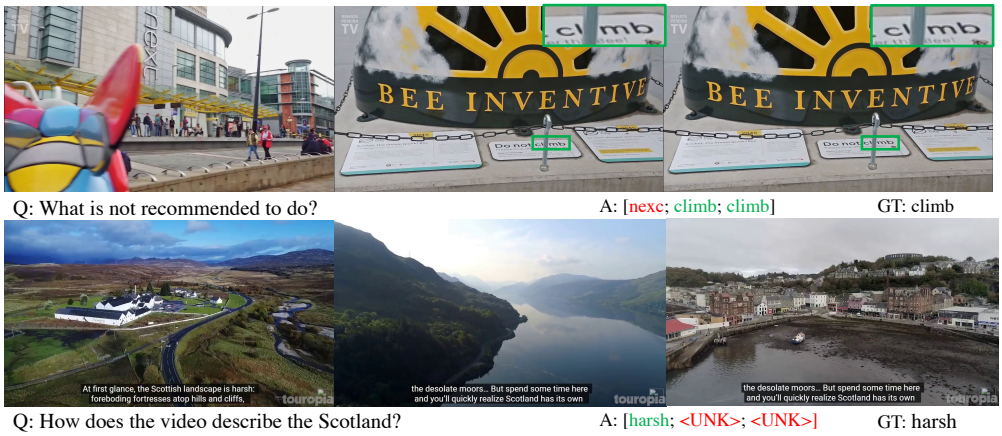


Figure 6: Two examples from traveling category. 'A' indicates the answers returned by TextVQA models and wrong answers are colored in red.

H.3 Driving

Driving category videos are mainly captured by vehicles. As shown in Fig. 7, this category requires the model to answer information about landmarks, road signs, etc... This category is suffering from motion blur and low resolution issues, which bring huge challenges to ViteVQA models. We hope this category will spur research in related fields such as auto-driving.

H.4 Vlog

This category includes filmed videos which recorded people's daily lives and street scenes. The photographer's personal private information has been processed. One example is given in Fig. 8.

H.5 Sport

This category includes sports videos including football, basketball, rugby and many other sports. Models need to understand temporal events to answer questions about scores, goals, etc... An example is given in Fig. 9.

Q: What's the license plate number of the car on the left? A: [<UNK>; T654186C; <UNK>] GT: T654186C

Figure 7: An example from driving category. 'A' indicates the answers returned by TextVQA models and wrong answers are colored in red.

Q: What does the button he press at the end say? A: [<UNK>; <UNK>; PUSH TO OPEN] GT: PUSH TO OPEN

Figure 8: An example from vlog category. 'A' indicates the answers returned by TextVQA models and wrong answers are colored in red.

H.6 Advertisement

As shown in Fig. 10, this category consists of advertisements collected from the network.

H.7 Movie

This category mainly consists of dialogues in movies and TV series. Unlike natural texts, the movie class requires the model to have the ability to read and understand contextual dialogue from the subtitle. One example is given in Fig. 11.

H.8 Game

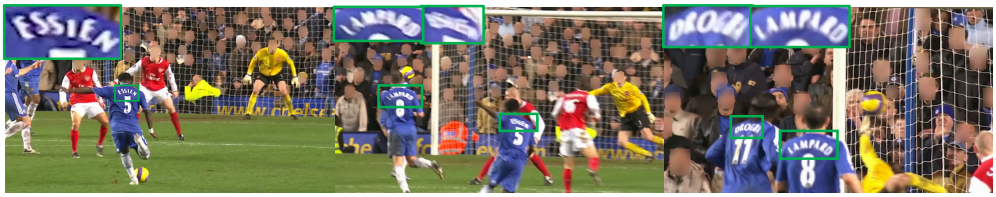
This category consists of videos collected from several popular video games. This category requires the model to have an understanding of the events in the game. The layout of the texts in the game also poses a huge challenge for ViteVQA models. One example is given in Fig. 12.

H.9 Talking

This category requires the model to understand the information in news and speeches. An example is shown in Fig. 13.

I Visualization

In this section, we visualize some cases in M4C [28] and T5-ViteVQA. The results are given in Fig. 14. As can be checked in Fig. 14, M4C can not solve some temporal action-related (1st case) and temporal layout-related (2nd case) questions. Although T5-ViteVQA has a better temporal reasoning ability, it still fails in some cases where texts change rapidly. For example, in the 3rd case of Fig. 14 the rapid change of the number of the items in the cart introduces huge challenges in reading and tracking these OCR tokens as well as understanding them. This indicates that our proposed T5-ViteVQA still has great space for improvement.



Q: Who is shooting the goal? A: [ESSIEN; LAMPARD; LAMPARD] GT: ESSIEN

Figure 9: An example from sport category. 'A' indicates the answers returned by TextVQA models and wrong answers are colored in red.

Q: What brand is the beer? A: [<UNK>; Heineken; <UNK>] GT: Heineken

Figure 10: An example from advertisement category. 'A' indicates the answers returned by TextVQA models and wrong answers are colored in red.

Q: What does he want for drink? A: [<UNK>; <UNK>; Cognac] GT: Cognac

Figure 11: An example from movie category. 'A' indicates the answers returned by TextVQA models and wrong answers are colored in red.

Q: What was the second thing he picked up? A: [PainKiller; Adrenaline Syringe; Cognac] GT: Adrenaline Syringe

Figure 12: An example from game category. 'A' indicates the answers returned by TextVQA models and wrong answers are colored in red.

Q: What is the S&P 500 index shown at last? A: [3,364.30; 3,364.23; 3,364.21] GT: 3,364.21

Figure 13: An example from talking category. 'A' indicates the answers returned by TextVQA models and wrong answers are colored in red.

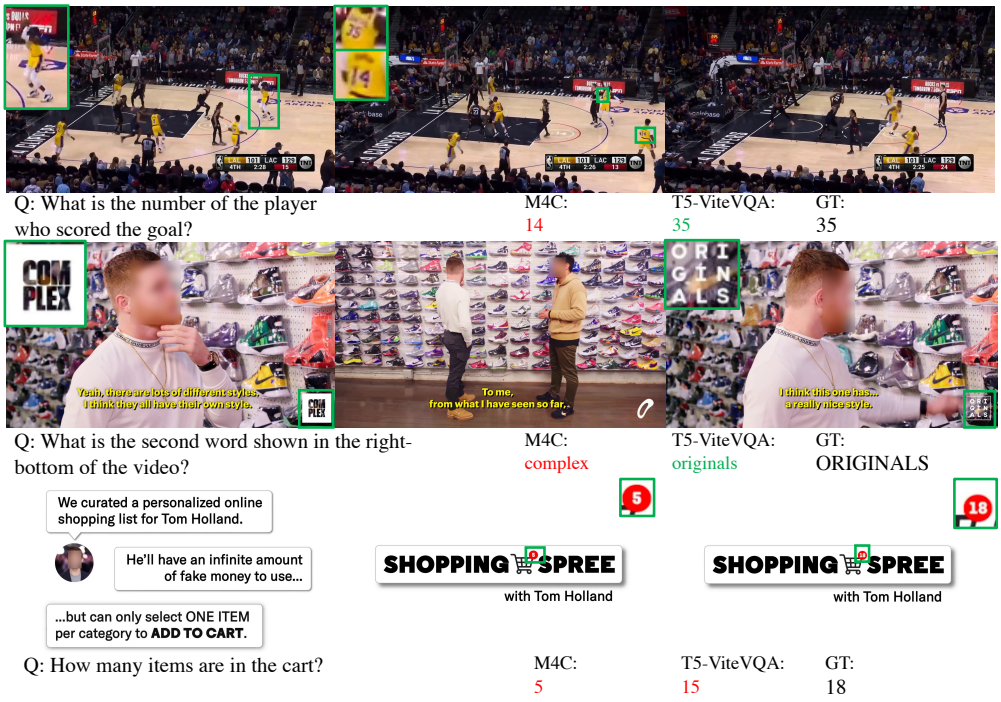


Figure 14: Some qualitative examples on the M4-ViteVQA dataset. We compare our method with M4C [28] and find that our method performs better than M4C.