TVQA+: Spatio-Temporal Grounding for Video Question Answering

Jie Lei Licheng Yu Tamara L. Berg Mohit Bansal
Department of Computer Science
University of North Carolina at Chapel Hill
{jielei, licheng, tlberg, mbansal}@cs.unc.edu

Abstract

We present the task of Spatio-Temporal Video Question Answering, which requires intelligent systems to simultaneously retrieve relevant moments and detect referenced visual concepts (people and objects) to answer natural language questions about videos. We first augment the TVQA dataset with 310.8k bounding boxes, linking depicted objects to visual concepts in questions and answers. We name this augmented version as TVQA+. We then propose Spatio-Temporal Answerer with Grounded Evidence (STAGE), a unified framework that grounds evidence in both the spatial and temporal domains to answer questions about videos. Comprehensive experiments and analyses demonstrate the effectiveness of our framework and how the rich annotations in our TVQA+ dataset can contribute to the question answering task. As a side product, by performing this joint task, our model is able to produce more insightful intermediate results. Dataset[1] and code[2] are publicly available.

1. Introduction

We have witnessed great progress in recent years on image-based visual question answering (QA) tasks [2, 43, 48]. One key to this success has been spatial attention [1, 34, 23], where neural models learn to attend to relevant regions for predicting the correct answer. Compared to image-based QA, there has been less progress on the performance of video-based QA tasks. One possible reason is that attention techniques are hard to generalize to the temporal nature of videos. Moreover, due to the high cost of annotation, most existing video QA datasets only contain question-answer pairs, without providing labels for the key moments or regions needed to answer the question. Inspired by previous work on grounded image and video captioning [24, 47, 46], we propose methods that explicitly localize video moments as well as spatial regions for answering video-based questions. Such methods are useful in many scenarios, such as natural language guided spatio-temporal localization, and adding explainability to video question answering, which is potentially useful for decision making and model debugging. To enable this line of research, we collect new annotations for an existing video QA dataset.

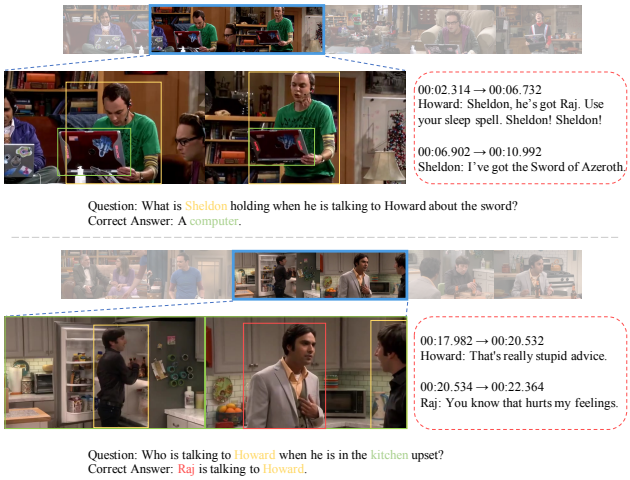


Figure 1. Sample QA pairs from TVQA+ dataset. Questions are both temporally localized to clips, and spatially localized with frame-level bounding box annotations for visual concepts (objects and people) that appear in questions and correct answers. Colors indicate corresponding box-object pairs. Text inside red dashed blocks are subtitles. For brevity, the wrong answers are omitted.

In the past few years, several video QA datasets have been proposed, e.g., MovieFIB [25], MovieQA [35], TGIF-QA [14], PororoQA [17], and TVQA [19]. Among them, TVQA was released most recently, providing a large video QA dataset built on top of 6 famous TV series. Because TVQA was collected on television shows, it is built on natural video content with rich dynamics and realistic social interactions, where question-answer pairs are written by people observing both videos and their accompanying dialogues, encouraging the questions to require both vision and language understanding to answer. One key property of TVQA is it provides temporal annotations denoting which parts of a video clip are necessary for answering a pro-

[1] http://tvqa.cs.unc.edu
[2] https://github.com/jayleicn/TVQA-PLUS

posed question. However, none of the video QA datasets (including TVQA) provide spatial annotation for the answers. Actually, grounding spatial regions correctly could be as important as grounding temporal moments for answering a given question. For example, in Fig. 1, to answer the question of *'What is Sheldon holding when he is talking to Howard about the sword?'*, we need to localize the moment when *'he is talking to Howard about the sword?'*, as well as looking at the specific region of *'What is Sheldon holding'*.

In this paper, we first augment one show, "The Big Bang Theory", from the TVQA dataset with grounded bounding boxes, resulting in a spatio-temporally grounded video QA dataset, TVQA+. TVQA+ consists of 29.4K multiple-choice questions grounded in both the temporal and spatial domains. To collect spatial groundings, we start by identifying a set of visual concept words, *i.e.* objects and people, mentioned in the question or correct answer. Next, we associate the referenced visual concepts with object regions in individual frames, if there are any, by annotating bounding boxes for each referred concept. One example QA pair is shown in Fig. 1. The TVQA+ dataset has a total of 310.8K bounding boxes linked with referred objects and people, spanning across 2.5K categories, more details in Section 3.

With such richly annotated data, we propose the task of spatio-temporal video question answering, which requires intelligent systems to localize relevant moments, detect referred objects and people, and answer questions.

We further design several metrics to evaluate the performance of the proposed task, including QA accuracy, object grounding precision, and a joint temporal localization and answering accuracy. We find that the performance of question answering benefits from both temporal moment and spatial region supervision. Additionally, the visualization of temporal and spatial localization is helpful for understanding what the model has learned.

To address spatio-temporal video question answering, we propose a novel end-to-end trainable model, Spatio-Temporal Answerer with Grounded Evidence (STAGE), which effectively combines moment localization, object grounding, and question answering in a unified framework. Comprehensive ablation studies demonstrate how each of our annotations and model components helps to improve the performance of video question answering.

To summarize, our contributions are:

- We collect TVQA+, a large-scale spatio-temporal video question answering dataset, which augments the original TVQA dataset with frame-level bounding box annotations. To our knowledge, this is the first dataset that combines moment localization, object grounding, and question answering.
- We propose a set of metrics to evaluate the performance of both spatio-temporal localization and question answering.
- We design a novel video question answering framework, Spatio-Temporal Answerer with Grounded Evidence (STAGE), to jointly localize moments, ground objects, and answer questions. By performing all three sub-tasks together, our model achieves significant performance gains over the state-of-the-art, as well as presenting insightful visualized results.

2. Related Work

Question Answering Teaching machines to answer questions is an important problem for AI. In recent years, multiple question answering datasets and tasks have been proposed to facilitate research towards this goal, in both the vision and language communities, in the form of visual question answering [2, 43, 14] and textual question answering [30, 29, 39, 38], respectively. Video question answering [19, 35, 17] with naturally occurring subtitles are particularly interesting, as it combines both visual and textual information for question answering. Different from existing video QA tasks, where a system is only required to predict an answer, we propose a novel task that additionally grounds the answer in both spatial and temporal domains.

Language-Guided Retrieval Grounding language in images/videos is an interesting problem that requires jointly understanding both text and visual data. Earlier works [16, 13, 45, 44, 42, 32] focused on identifying the referred object in an image. Recently, there has been a growing interest in moment retrieval tasks [12, 11, 9], where the goal is to localize a short clip from a long video via a natural language query. Our work integrates the goal of both tasks, requiring a system to ground the referred moments and objects simultaneously.

Temporal and Spatial Attention Attention has shown great success on many vision and language tasks, such as image captioning [1, 40], visual question answering [1, 36], language grounding [42], etc. However, sometimes the attention learned by the model itself may not accord with human expectations [22, 5]. Recent works on grounded image captioning and video captioning [46, 22, 47] show better performance can be achieved by explicitly supervising the attention. In this work, we use annotated frame-wise bounding box annotations to supervise both temporal and spatial attention. Experimental results demonstrate the effectiveness of supervising both domains in video QA.

3. Dataset

In this section, we describe the TVQA+ Dataset, the first video question answering dataset with both spatial and temporal annotations. TVQA+ is built on the TVQA dataset introduced in [19]. TVQA is a large-scale video QA dataset based on 6 popular TV shows, containing 152.5K multiple choice questions from 21.8K, 60-90 second long video clips. The questions in the TVQA dataset are composi-

Split	#QAs	#Clips	Avg Span Len (secs)	Avg Video Len (secs)	#Annotated Images	#Boxes	#Categories
Train	23,545	3,364	7.20	61.49	118,930	249,236	2,281
Val	3,017	431	7.26	61.48	15,350	32,682	769
Test	2,821	403	7.18	61.48	14,188	28,908	680
Total	29,383	4,198	7.20	61.49	148,468	310,826	2,527

Table 1. Data Statistics for TVQA+ dataset.

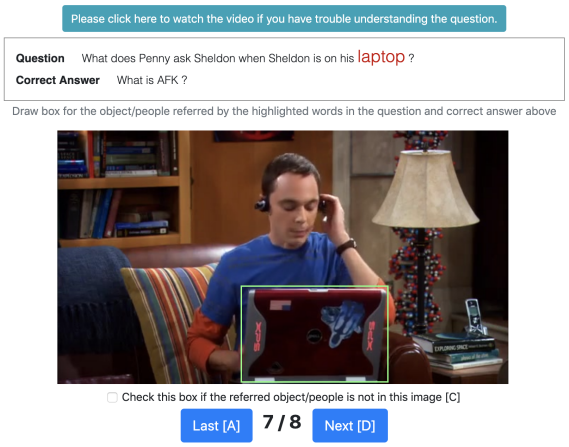


Figure 2. User interface for bounding box annotation. Here, the worker is asked to draw a box around the highlighted "laptop".

Dataset	Origin	Task	#Clips/#QAs (#Querys)	#Boxes	Temporal Annotation
MovieFIB [25]	Movie	QA	118.5K/349K	0	-
MovieQA [35]	Movie	QA	6.8K/6.5K	0	✓
TGIF-QA [14]	Tumblr	QA	71.7K/165.2K	0	-
PororoQA [17]	Cartoon	QA	16.1K/8.9K	0	-
DiDeMo [12]	Flickr	TL	10.5K/40.5K	0	✓
Charades-STA [9]	Home	TL	-/19.5K	0	✓
TVQA [19]	TV Show	QA/TL	21.8K/152.5K	0	✓
TVQA+	TV Show	QA/TL/SL	4.2K/29.4K	310.8K	✓

Table 2. Comparison of TVQA+ dataset with other video-language datasets. QA = Question Answering, TL = Temporal Localization, SL = Spatial Localization.

tional, where each question is comprised of two parts, a question part ("where was Sheldon sitting"), joined via a link word, ("before", "when", "after"), to a localization part that temporally locates when the question occurs ("he spilled the milk"). Models should answer questions using both visual information from the video, as well as language information from the naturally associated dialog (subtitles). Since the video clips on which the questions were collected are usually much longer than the context needed for answering the questions, the TVQA dataset also provides a temporal timestamp annotation indicating the minimum span (context) needed to answer each question.

While the TVQA dataset provides a novel question format and temporal annotations, it lacks spatial grounding information, i.e., bounding boxes of the concepts (objects and people) mentioned in the QA pair. We hypothesize object annotations could provide additional useful training source for models to gain a deeper understanding of visual information in TVQA. Therefore, to complement the original TVQA dataset, we collect frame-wise bounding boxes for visual concepts mentioned in the questions and correct answers. Since the full TVQA dataset is very large, we start by collecting bounding box annotations for QA pairs associated with one of the 6 TV shows - *The Big Bang Theory*, which contains 29,383 QA pairs from 4,198 clips.

3.1. Data Collection

Identify Visual Concepts: To annotate the visual concepts in video frames, the first step is to identify them in the QA pairs. We use the Stanford CoreNLP part-of-speech (POS) tagger [26] to extract all nouns in the questions and correct answers; this gives us a total of 152,722 words from a vocabulary of 9,690 words. We manually label the non-visual nouns (e.g., 'plan', 'time', etc.) in the top 600 nouns, removing 165 frequent non-visual nouns from the vocabulary.
Bounding Box Annotation: For the selected *The Big Bang Theory* videos from TVQA, we first ask Amazon Mechanical Turk (AMT) workers to adjust the start and end timestamps to refine the temporal annotation.[3] We then sample one frame every two seconds from each span for annotation. For each frame, we collect the bounding boxes for the objects/people mentioned in each QA pair. In this step, we show a question, its correct answer, and the sampled video frames to an AMT worker (illustrated in Figure 2). As each QA pair has multiple visual concepts as well as multiple frames, each task shows one pair of a concept word and a sampled frame. For example, in Figure 2, the word "laptop" is highlighted, and workers are instructed to draw a box around it. Note, it is possible that the highlighted word will be a non-visual word or a visual word that is not present in the frame being shown. In that case, the workers are allowed to check the box indicating the object is not present. During annotation, we also provide the original videos (with subtitles) in case they have trouble understanding the given QA pair.

3.2. Dataset Analysis

TVQA+ contains 29,383 QA pairs from 4,198 video clips, with 148,468 images annotated with 310,826 bound-

[3] We provide results of our model trained with original and refined temporal annotation in the supplementary file.

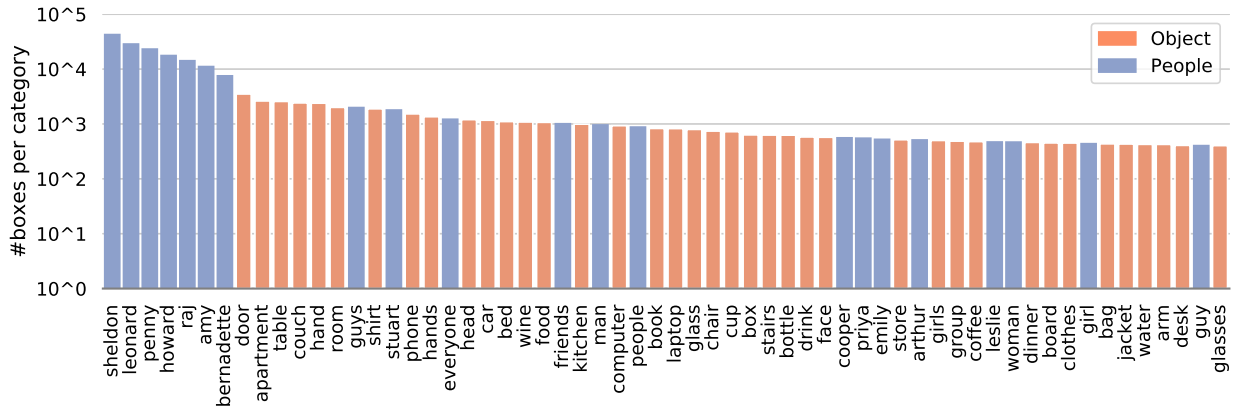


Figure 3. Box distributions for top 60 categories in TVQA+ train set.

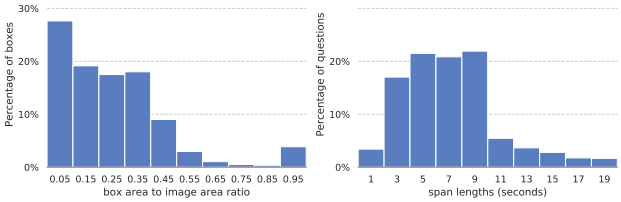


Figure 4. Bounding box/image ratios (left), and span length distributions (right) in TVQA+ The majority of the boxes are very small compared to image size and most spans are less than 10 seconds.

ing boxes. Statistics of the full dataset are shown in Table 1. Note, we follow the same data splits as the original TVQA dataset [19]. Table 2 compares the TVQA+ dataset with other video-language based datasets. The TVQA+ dataset is unique as it contains three different annotations: question answering, temporal localization, and spatial localization.

On average, we obtain 2.09 boxes per image and 10.58 boxes per question. The annotated boxes cover 2,527 categories. We show the number of boxes (in log scale) for each of the top 60 categories in Figure 3. The distribution has a long tail, e.g., the number of boxes for the most frequent category 'sheldon' is around 2 orders of magnitude larger than the 60th category 'glasses'. We also show the distribution of ratio of bounding box area over image area ratio in Figure 4 (left). The majority of boxes are fairly small compared to the image, which makes object grounding challenging. Figure 4 (right) shows the distribution of localized span length. While most of the spans are less than 10 seconds, the largest spans are up to 20 seconds. The average span length is 7.2 seconds, which is short compared to the average length of the full video clip (61.2 seconds).

4. Methods

Our proposed method, Spatio-Temporal Answerer with Grounded Evidence (STAGE), is a unified framework for moment localization, object grounding and video QA. First, STAGE encodes the video and text (subtitle, QA pairs) via frame-wise regional visual representations and neural language representations, respectively. The encoded video and text representations are then contextualized using a Convolutional Encoder. Second, STAGE computes attention scores from each QA word to the object regions and subtitle words. Leveraging the attention scores, STAGE is able to generate QA-aware representations, as well as automatically detecting the referred objects and people. The attended QA-aware video representation and subtitle representation are then fused together to obtain a frame-wise joint representation. Third, taking the frame-wise representation as input, STAGE learns to predict temporal spans that are relevant to the QA pair, then combines the global and local (span localized) video information to answer the questions. Next, we explain each step in detail.

4.1. Formulation

In our tasks, the inputs are: (1) a question with 5 candidate answers; (2) a 60-second long video; (3) a set of subtitle sentences, and our goal is to predict the correct answer as well as ground the answer both spatially and temporally. Given the question, q, and the answers, $\{a_k\}_{k=1}^{5}$, we first formulate them as 5 hypotheses (QA-pair) $h_k = [q, a_k]$ and predict their correctness scores based on the video and subtitle context, which is similar to [27, 19]. We denote the ground-truth (GT) answer index as y^{ans} and thus the GT hypothesis as $h_{y^{ans}}$. We then extract video frames $\{v_t\}_{t=1}^{T}$ at 0.5 FPS (T is the number of frames for each video), aligning the subtitle sentences temporally with the video frames. Specifically, for each frame v_t, we pair it with two neighboring subtitle sentences based on the subtitle timestamp. We choose two neighbors since this keeps most of the sentences at our current frame rate, and also avoids severe misalignment between the frames and the sentences. The set of aligned subtitle sentences are denoted as $\{s_t\}_{t=1}^{T}$.

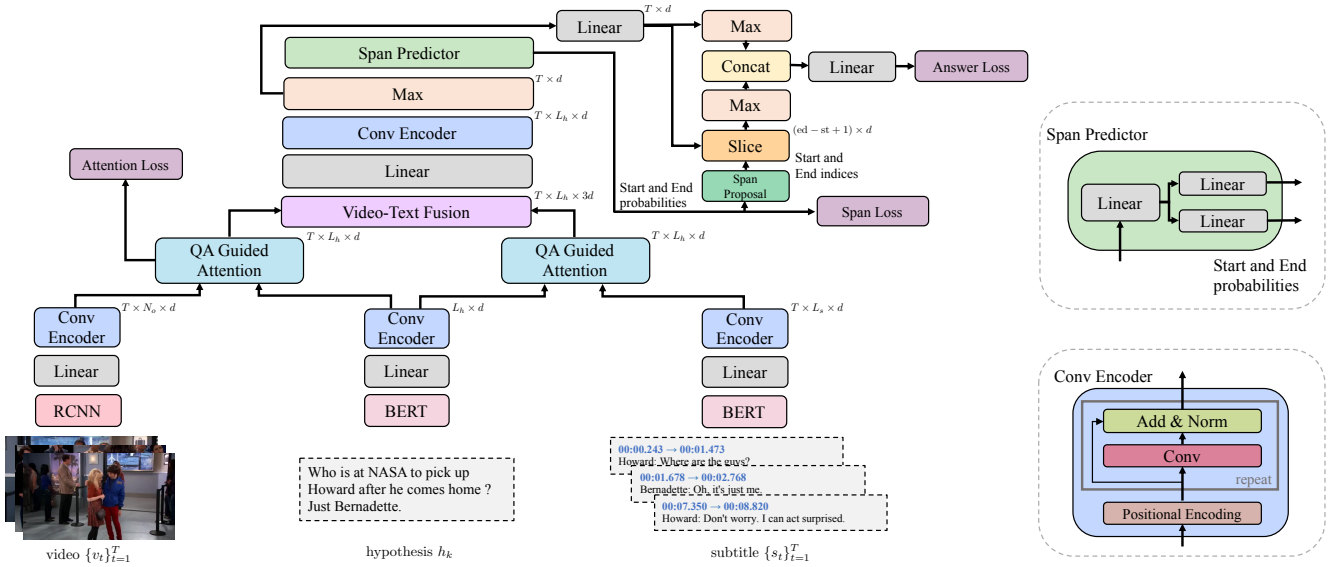


Figure 5. Overview of the proposed framework, Spatio-Temporal Answerer with Grounded Evidence (STAGE) for spatio-temporal video question answering. Given 5 hypotheses (question + answer pairs), a set of video frames and aligned subtitle sentences, STAGE is able to answer the question, as well as providing detections for referred visual concepts (objects and people), and predicting a temporal span for localizing the relevant moment in the video. The full model is trained end-to-end using a combination of attention loss, span loss and answer loss. For brevity, we only show one hypothesis in the figure. We provide output dimensions for some of the modules for clarity.

We denote the number of words in each hypothesis as L_h and the aligned subtitle sentence pair as L_s respectively. We use N_o to denote the number of object regions in a frame, and $d = 128$ as the hidden size.

4.2. STAGE Architecture

Input Embedding Layer: One of our goals is to localize visual concepts. For each frame v_t, we use Faster R-CNN [31] pre-trained on Visual Genome [18] to detect objects and extract their regional embeddings as our visual input feature [1]. We keep top-20 object proposals and use PCA to reduce the feature dimension from 2048 to 300. We denote $o_{t,r} \in \mathbb{R}^{300}$ as the r-th object embedding in the t-th frame. To encode the text input, we use BERT [7], a transformer [37] based language model that achieves state-of-the-art performance on various NLP tasks. Specifically, we first fine-tune the BERT-base model using a masked language model and next sentence prediction on the subtitles and QA pairs from the TVQA+ train set. Then, we fix its parameters and use it to extract 768-dimensional word-level embeddings from the second-to-last layer for the subtitles and each hypothesis. Both the object-level embeddings and the word-level embeddings are then projected into a 128-dimensional space using a linear layer with ReLU activation.

Convolutional Encoder: Inspired by the recent trend of replacing recurrent networks with CNNs [6, 15, 41] and Transformers [37, 7] for sequence modeling, we use positional encoding (PE) [37], CNNs, and layer normalization [3] to build our basic encoding block. As shown in the bottom-right corner of Fig. 5, this is comprised of a positional encoding layer and multiple convolutional layers, each with a residual connection [10] and layer normalization. Specifically, we use $\text{Layernorm}(\text{ReLU}(\text{Conv}(x)) + x)$ as a single Conv unit and stack N_{conv} of such units as the convolutional encoder. x is the input after PE, Conv is a depthwise separable convolution [4]. We use two convolutional encoders at two different levels of STAGE, one with kernel size 7 to encode the raw inputs, and another with kernel size 5 to encode the fused video-text representation. For both encoders, we set $N_{\text{conv}} = 2$.

QA-Guided Attention: For each hypothesis $h_k = [q, a_k]$, we compute its attention scores w.r.t. the object embeddings in each frame and the words in each subtitle sentence, respectively. Given the encoded hypothesis $H_k \in \mathbb{R}^{L_h \times d}$ for the hypothesis h_k with L_h words, and encoded visual feature $V_t \in \mathbb{R}^{N_o \times d}$ for the frame v_t with N_o objects, we compute their matching scores $M_{k,t} \in \mathbb{R}^{L_h \times N_o} = H_k V_t^T$. We then apply softmax at the second dimension of $M_{k,t}$ to get the normalized scores $\bar{M}_{k,t}$. Finally, we compute the QA-aware visual representation $V_{k,t}^{att} \in \mathbb{R}^{L_h \times d} = \bar{M}_{k,t} V_t$. Similarly, we compute QA-aware subtitle representation $S_{k,t}^{att}$.

Video-Text Fusion: The above two QA-aware representations are then fused together as:

$$F_{k,t} = [S_{k,t}^{att}; V_{k,t}^{att}; S_{k,t}^{att} \odot V_{k,t}^{att}] W_F + b_F, \qquad (1)$$

where $\odot$ denotes element-wise multiplication, $W_F \in \mathbb{R}^{3d \times d}$ and $b_F \in \mathbb{R}^d$ are trainable weights and bias, $F_{k,t} \in \mathbb{R}^{L_h \times d}$ is the fused video-text representation. Note that the

frame and subtitle representations are temporally aligned, which is essential for the downstream span prediction task. Collecting $F^{att}_{k,t}$ at all time steps, we have $F^{att}_{k} \in \mathbb{R}^{T \times L_h \times d}$. We then apply another convolutional encoder with a max-pooling layer to obtain the output $A_k \in \mathbb{R}^{T \times d}$.

Span Predictor: To predict temporal spans, we follow existing works [21, 33, 41] to predict the probability of each position being the start or end of the span. Given the fused input $A_k \in \mathbb{R}^{T \times d}$, we produce start probabilities $\mathbf{p^1_k} \in \mathbb{R}^T$ and end probabilities $\mathbf{p^2_k} \in \mathbb{R}^T$ using two linear layers with softmax, as shown in the top-right corner of Fig. 5.

Span Proposal and Answer Prediction: Given the max-pooled video-text representation A_k, we use a linear layer to further encode it. We run max-pool across all the time steps to get a global hypothesis representation $G^g_k \in \mathbb{R}^d$. With the start and end probabilities from the span predictor, we generate span proposals using dynamic programming as [41, 33]. At training time, we combine the set of proposals with $IoU \geq 0.5$ with the GT spans, as well as the GT spans to form the final proposals $\{st_p, ed_p\}$ [31]. At inference time, we take the proposals with the highest confidence scores for each hypothesis. For each proposal, we generate a local representation $G^l_k \in \mathbb{R}^d$ by max-pooling $A_{k,st_p:ed_p}$. The local and the global representations are concatenated to obtain $G_k \in \mathbb{R}^{2d}$. We then forward $\{G_k\}^5_{k=1}$ through softmax to get the answer scores $\mathbf{p}^{ans} \in \mathbb{R}^5$.

4.3. Training and Inference Objective Functions

In this section, we describe the objective functions used in the STAGE framework. Since our spatial and temporal annotations are collected based on the question and GT answer, we only apply the attention loss and span loss on the targets associated with the GT hypothesis (question + GT answer), i.e., $M_{k=y^{ans},t}$, $\mathbf{p}^1_{k=y^{ans}}$ and $\mathbf{p}^2_{k=y^{ans}}$. For brevity, we omit the subscript k=y^{ans} in the following.

Explicit Attention Supervision: While the attention described in Section 4.2 can be learned in a weakly supervised end-to-end manner, we can also train it with the supervision of available GT boxes. We define a box as positive if it has an $IoU \geq 0.5$ with the GT box. Consider the attention scores $M_{t,j} \in \mathbb{R}^{N_o}$ from a concept word w_j in GT hypothesis $h_{y^{ans}}$ to the set of proposal boxes' representations $\{o_{t,r}\}^{N_o}_{r=1}$ at frame v_t. We expect the attention on positive boxes to be higher than the negative ones, thus we use a ranking loss for the supervision. Recent work [20] suggests using log-sum-exp (LSE) as a smooth approximation of the non-smooth hinge loss, as it is easier to optimize. The LSE formulation of ranking loss is:

$$\mathcal{L}^{lse}_{t,j} = \sum_{r_p \in \Omega_p, r_n \in \Omega_n} \log\Big(1+\exp(M_{t,j,r_n} - M_{t,j,r_p})\Big), \quad (2)$$

where M_{t,j,r_p} is the r_p-th element of the vector $M_{t,j}$. Ω_p and Ω_n denote the set of positive and negative box indices, respectively. During training, we randomly sample two negatives for each positive box. We use $\mathcal{L}^{att}_i$ to denote the attention loss for the i-th example, which is obtained by summing over all the annotated frames $\{v_t\}$ and concepts $\{w_j\}$ for $\mathcal{L}^{att}_{t,j}$ in the example. We define the overall attention loss $\mathcal{L}^{att} = \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}^{lse}_i$. At inference time, we choose the box with the highest score as the prediction.

Span Prediction: Given the softmax normalized start and end probabilities $\mathbf{p}^1$ and $\mathbf{p}^2$, we use cross-entropy loss:

$$\mathcal{L}^{span} = -\frac{1}{2N}\sum_{i=1}^{N}\big(\log \mathbf{p}^1_{y^1_i} + \log \mathbf{p}^2_{y^2_i}\big), \quad (3)$$

where y^1_i and y^2_i are the indices of the GT start and end positions, respectively. To predict the span $(st, ed), st \leq ed$ for each QA pair, we follow previous works [33, 41] to find the one with maximum $\mathbf{p}^1_{st}\mathbf{p}^2_{ed}$.

Answer Prediction: Similar to span prediction loss, given answer probabilities $\mathbf{p}^{ans}$, answer prediction loss is:

$$\mathcal{L}^{ans} = -\frac{1}{N}\sum_{i=1}^{N}\log \mathbf{p}^{ans}_{y^{ans}_i}, \quad (4)$$

where y^{ans}_i is the index of the GT answer.

Finally, the overall loss is a weighted combination of the above three objectives: $\mathcal{L} = \mathcal{L}^{ans} + w_{att}\mathcal{L}^{att} + w_{span}\mathcal{L}^{span}$, where w_{att} and w_{span} are set as 0.1 and 0.5 based on validation set tuning.

5. Experiments

Our task is spatio-temporal video question answering, requiring systems to temporally localize relevant moments, spatially detect referred objects and people, and answer questions. In this section, we first introduce our metrics, then compare STAGE against several baselines, and finally provide a comprehensive analysis of our model. Additionally, we evaluate our STAGE on the full original TVQA dataset and achieve rank-1 in the TVQA Codalab leaderboard[4] at the time of submission, outperforming the second best method by 1.5%.

5.1. Metrics

To measure *question answering* performance, we use classification accuracy (QA Acc.). We evaluate *span prediction* using temporal mean Intersection-over-Union (Temp. mIoU) following previous works [12, 11] on language-guided video moment retrieval. Since the span depends on the hypothesis (QA pair), each QA pair provides a predicted span, but we only evaluate the span of the predicted answer. Additionally, we propose a new metric, Answer-Span joint Accuracy (ASA), that jointly evaluates both answer prediction and span prediction. For this metric,

[4] https://competitions.codalab.org/competitions/20687#results

	Model	vfeat	tfeat	QA Acc.	Grd. mAP	Temp. mIoU	ASA
1	Longest Answer [19]	-	-	33.32	-	-	-
2	TFIDF Answer-Subtitle [19]	-	-	50.97	-	-	-
3	two-stream [19]	reg	GloVe	64.73	-	-	-
4	two-stream [19]	cpt	GloVe	66.47	-	-	-
5	backbone + Attn. Sup. + Temp. Sup. + local (STAGE)	reg	BERT	**74.83**	**27.34**	**32.49**	**22.23**
6	Human Performance [19]	-	-	90.46	-	-	-

Table 3. Comparison with existing methods on TVQA+ test set. vfeat = video feature, tfeat = text feature. We follow the convention in [19] to use reg to denote detected object embeddings, cpt to denote detected object labels and attributes. Grd. mAP = grounding mAP, Temp. mIoU = temporal mIoU, ASA = Answer-Span joint Accuracy.

	Model	vfeat	tfeat	QA Acc.	Grd. mAP	Temp. mIoU	ASA
1	two-stream [19]	reg	GloVe	62.28	-	-	-
2	two-stream [19]	cpt	GloVe	62.25	-	-	-
3	backbone	reg	GloVe	67.29	4.46	-	-
4	backbone	reg	BERT	68.31	7.31	-	-
5	backbone + Attn. Sup.	reg	BERT	71.03	24.8	-	-
6	backbone + Temp. Sup.	reg	BERT	71.4	10.86	30.77	20.09
7	backbone + Attn. Sup. + Temp. Sup.	reg	BERT	71.99	24.1	31.16	20.42
8	backbone + Attn. Sup. + Temp. Sup. + local (STAGE)	reg	BERT	**72.56**	**25.22**	**31.67**	**20.78**
9	STAGE with GT Span	reg	BERT	73.28	-	-	-

Table 4. Ablation study of our proposed STAGE framework on TVQA+ val set. In row 9, we show a model with GT spans at inference. Attn. Sup. = spatial attention supervision, Temp. Sup. = span predictor with temporal supervision. Models in row 3-7 use only global feature G^g for question answering, while the one in row 8 additionally use local feature G^l.

we define a prediction to be correct if the predicted span has an $IoU \geq 0.5$ with the GT span, provided that the answer prediction is correct. Finally, to evaluate *object grounding* performance, we follow the standard metric from the PASCAL VOC challenge [8] and report the mean Average Precision (Grd. mAP) at IoU threshold 0.5. We only consider the annotated words and frames when calculating the mAP.

5.2. Comparison with Baseline Methods

We consider the previous two-stream model [19] as our main baseline. In this model, two streams are used to predict answer scores from subtitles and videos respectively and final answer scores are produced by summing scores from the two streams. We retrain the model using the official code[5] on TVQA+ data. We also evaluate the two most representative non-neural baselines from [19], i.e., Longest Answer and TFIDF Answer-Subtitle matching.

Table 3 shows the test results of STAGE and the baseline methods. Our best QA model (row 5) outperforms previous state-of-the-art (row 4) by a large margin in QA accuracy, with 12.58% relative gains. Additionally, our model also localizes the relevant moments and detect referred objects and people. Table 3 shows our model achieves the best mAP of 27.34% on object grounding, and the best temporal mIoU of 32.49% on temporal localization. However, a large gap is still observed between our best model and humans (row 6), showing there is space for further improvement.

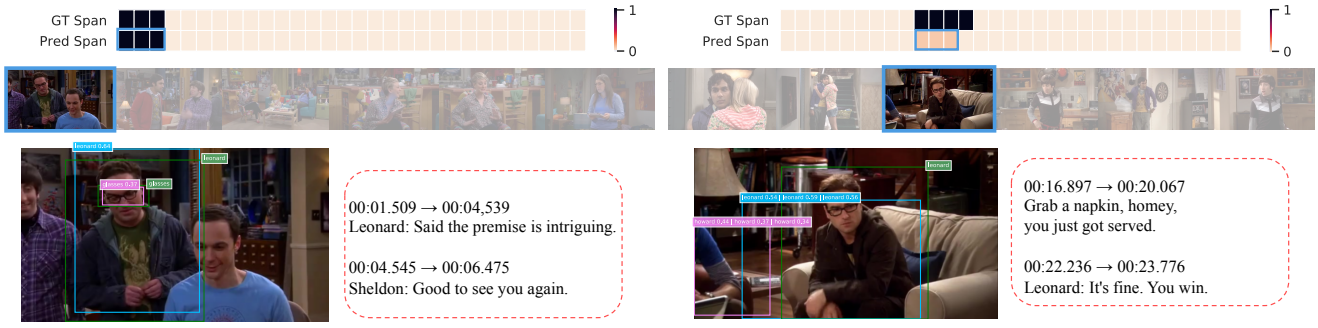
[5] https://github.com/jayleicn/TVQA

5.3. Model Analysis

Backbone Model: Given the full STAGE model defined in Sec. 4, we define the backbone model as the ablated version of it, where we removed span predictor along with the span proposal module, as well as the explicit attention supervision. Different from the baseline two-stream model [19] which uses RNNs to model text and video sequences, in our backbone model, we use CNN to encode both modalities. The two-stream [19] model interacts QA pairs with subtitles and videos separately, then sums the confidence score from each modality, while we align subtitles with video frames from the start, fusing their representation conditioned on the input QA pair, as in Fig. 5. We believe this aligned fusion is essential for improving QA performance, as the latter part of STAGE has a joint understanding of both video and subtitles. Using the same visual and text features, we observe our backbone model (row 3) far outperforms two-stream (row 1) in Table 4.

BERT as Feature: BERT [7] has primarily been used in NLP tasks. In Table 4, we show it is also useful for video QA task. Compared to the model with GloVe [28] as a text feature, BERT improves the backbone model by 1.52% in QA Acc. (row 4 *vs* row 3). We also find it improves the grounding performance of the model by 63.9%, relatively.

Spatial Attention Supervision: On top of the backbone model, we use annotated bounding boxes to provide attention supervision. We compare the model with attention supervision (row 5) with the backbone model (row 4) in Ta-



Q: What is Leonard wearing when he says said the premise is intriguing?
A1: Glasses. Pred GT
A2: Coffee.
A3: Rosary.
A4: Gang Collars.
A5: Hat.

(a)

Q: What did Leonard tell Howard after Howard said that Leonard just got served?
A1: Leonard told Howard that he really hates that game.
A2: Leonard told Howard that Howard isn't very good.
A3: Leonard told Howard that Sheldon will beat his score.
A4: Leonard told Howard that it was fine, he wins. Pred GT
A5: Leonard told Howard that he will beat him.

(b)

Figure 6. Example predictions from STAGE. The span predictions are shown on the top of each example, each block represents a frame, the color indicates the model's confidence for the predicted spans. For each QA, we show grounding examples and scores for one frame in GT span, GT boxes are shown in green. Model predicted answers are labeled by Pred, GT answers are labeled by GT.

ble 4. After adding such supervision, we observe a relative gain of 3.98% in QA Acc. and 239.26% in Grd. mAP.

Temporal Supervision: In Table 4, we also show the results of our model with span prediction under temporal supervision. For the backbone model with span prediction (with global feature G^g for question answering), we have a relative gain of 4.52% in QA Acc. and 48.56% in Grd. mAP (row 6 *vs* row 4). For our backbone model with both attention supervision and span predictor, we have a relative gain of 1.35% in QA Acc. (row 7 *vs* row 5).

Span Proposal and Local Feature: In row 8 and row 7 of Table 4, we compare the models with and without local features G^l for answer classification. Local features are obtained by max-pooling the span proposal regions, which should contain more relevant cues for answering the questions. With additional local features, we achieve the best performance across all metrics, indicating the benefit of using a span proposal module, as well as its provided local features.

Inference with GT Span: The last row of Table 4 shows our model with GT spans instead of predicted spans at inference time. We observe better QA Acc. with GT spans.

Accuracy by Question Type: In Table 5 we show a breakdown of QA Acc. by different question types. We observe a clear increasing trend on "what", "who", and "where" questions after replacing the backbone net and adding attention/span modules in each column. Interestingly, for "why" and "how" question types, our full model fails to present overwhelming performance, indicating some reasoning (textual) module to be incorporated as future work.

Qualitative Examples: We show two correct predictions in Fig. 6, where Fig. 6(a) uses text to answer the question, and Fig. 6(b) uses grounded objects to answer. More examples (including failure cases) are provided in the supplementary.

Model	two-stream [19]		backbone					
					+C1	+C2	+C3	+C4
vfeat	reg	cpt	reg	reg	reg	reg	reg	reg
tfeat	GloVe	GloVe	GloVe	BERT	BERT	BERT	BERT	BERT
what (60.52%)	62.71	62.60	67.63	67.58	69.99	70.76	71.25	**72.34**
who (10.24%)	53.07	55.66	61.49	64.72	72.60	72.17	73.14	**74.11**
where (9.68%)	51.37	55.82	62.33	68.49	71.52	71.58	71.58	**74.32**
why (9.55%)	78.46	75.35	76.74	77.43	**79.86**	**79.86**	78.12	76.39
how (9.05%)	65.20	61.90	67.40	69.23	68.50	66.30	**69.96**	67.03
total (100%)	62.28	62.25	67.29	68.31	71.03	71.40	71.99	**72.56**

Table 5. QA accuracy breakdown for different approaches across each question type on TVQA+ val set. For brevity, we only show top-5 question types (with the percentage of each type). C1 = Attn. Sup., C2 = Temp. Sup., C3 = Attn. Sup. + Temp. Sup., C4 = Attn. Sup. + Temp. Sup. + local (STAGE).

	Model	QA Acc.	
		val	test-public
1	two-stream [19]	65.85	66.46
2	anonymous 1 (JunyeongKim)	66.22	67.05
3	anonymous 2 (jeyki)	68.90	68.77
4	backbone	68.56	69.67
5	backbone + Temp. Sup. + local	70.50	70.23

Table 6. Model performance on full TVQA dataset. The results are from TVQA Codalab leaderboard.

TVQA Leaderboard Results: We also conduct experiments on the Leaderboard's full TVQA dataset (Table 6), without relying on the bounding box annotations and refined timestamps in TVQA+. Without span predictor (row 4), STAGE backbone is able to achieve 4.83% relative gain from the best published result (row 1) on TVQA test-public set. Adding span predictor (row 5), performance is improved to 70.23%, a new state-of-the-art for the task.

6. Conclusion

We presented the TVQA+ dataset and corresponding spatio-temporal video question answering task. The proposed task requires intelligent systems to localize relevant moments, detect referred objects and people, and answer questions. We further introduced STAGE, a novel, end-to-end trainable framework to jointly perform all three tasks. Comprehensive experiments show that temporal and spatial predictions help improve question answering performance as well as producing more explainable results. Though STAGE performs well, there is still a large gap to human performance that we hope will inspire future research.

Acknowledgments

This research is supported by NSF Awards #1633295, 1562098, 1405822, Google Faculty Research Award, Salesforce Research Deep Learning Grant, Facebook Faculty Research Award, and ARO-YIP Award #W911NF-18-1-0336.

A. Appendix

A.1. Timestamp Refinement.

During our initial analysis, we find the original timestamp annotations from the TVQA [19] dataset to be somewhat loose, i.e., around 8.7% of 150 randomly sampled training questions had a span that was at least 5 seconds longer than what is needed. To have better timestamps, we asked a set of Amazon Mechanical Turk (AMT) workers to refine the original timestamps. Specifically, we take the questions that have a localized span length of more than 10 seconds (41.33% of the questions) for refinement while leaving the rest unchanged. During annotation, we show a question, its correct answer, its associated video (with subtitle), as well as the original timestamp to the AMT workers (illustrated in Fig. 7, with instructions omitted). The workers are asked to adjust the start and end timestamps to make the span as small as possible, but need to contain all the information mentioned in the QA pair.

We show span length distributions of the original and the refined timestamps from TVQA+ train set in Fig. 8. The average span length of the original timestamps is 14.41 secs, while the average for the refined timestamps is 7.2 secs.

In Table 7 we show model performance on TVQA+ val set using the original timestamps and the refined timestamps. Models with the refined timestamps perform consistently better than the ones with the original timestamps.

A.2. More Examples

We show 6 correct prediction examples from STAGE in Fig. 9. As can be seen from the figure, correct examples usually have correct temporal and spatial localization. In Fig. 10 we show 6 incorrect examples. Incorrect object localization is one of the most frequent failure reason, while

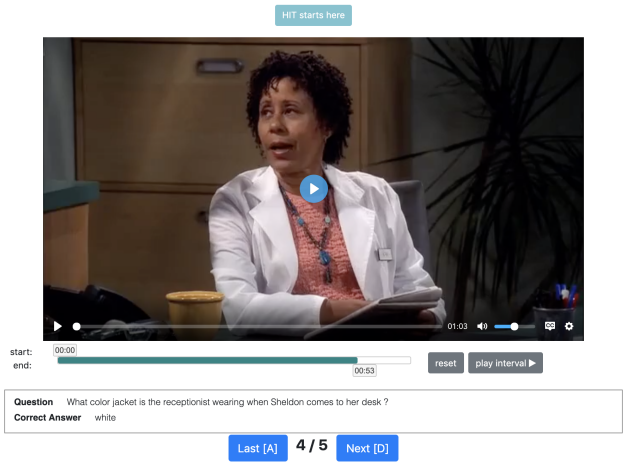


Figure 7. Timestamp refinement interface.

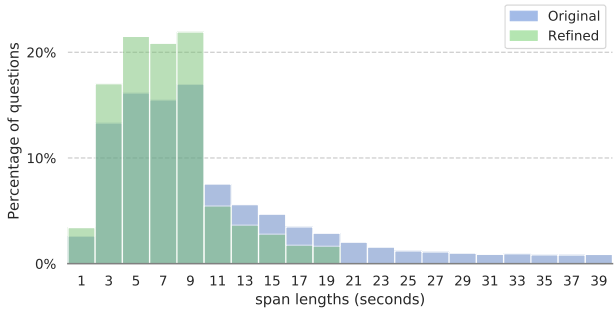


Figure 8. Comparison between the original and the refined timestamps in the TVQA+ train set. The refined timestamps are generally tighter than the original timestamps

Model	QA Acc.	
	Original	Refined
backbone	68.56	68.56
backbone + Attn. Sup.	71.03	71.03
backbone + Temp. Sup.	70.87	71.40
backbone + Attn. Sup. + Temp. Sup.	71.23	71.99
backbone + Attn. Sup. + Temp. Sup. + local (STAGE)	70.63	**72.56**

Table 7. Model performance comparison between the original timestamps and the refined timestamps on TVQA+ val set. We use reg as video feature, BERT as text feature for all the experiments in this table.

the model is able to localize common objects, it is difficult for it to localize unusual objects (Fig. 10(a, d)), small objects (Fig. 10(b)). Incorrect temporal localization is another most frequent failure reason, e.g., Fig. 10(c, f). There are also cases where the objects being referred are not present in the sampled frame, as in Fig. 10(e). Such failures indicate that using more densely sampled frames for question answering would be advantageous.

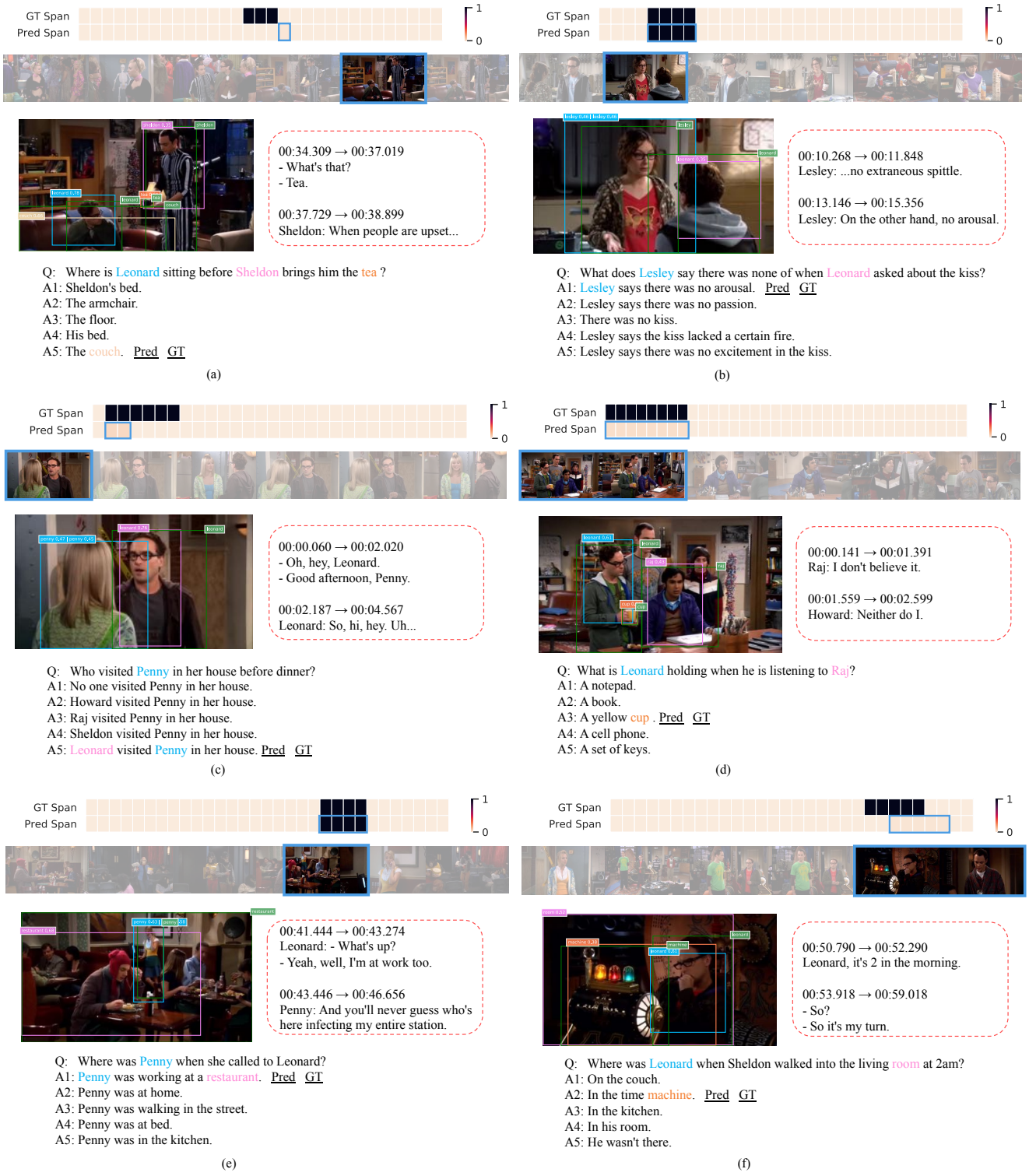
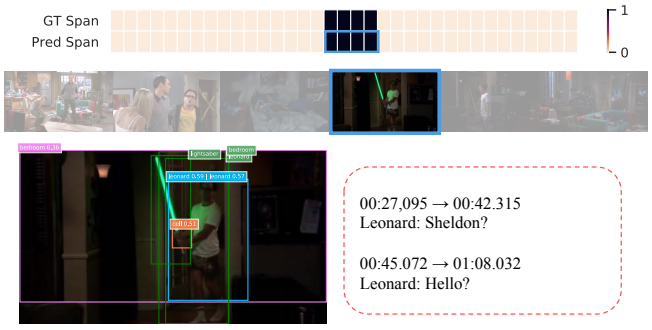
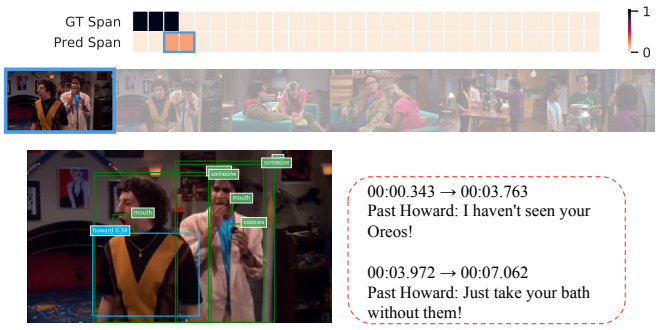


Figure 9. Correct prediction examples from STAGE. The span predictions are shown on the top of each example, each block represents a frame, the color indicates the model's confidence for the predicted spans. For each QA, we show grounding examples and scores for one frame in GT span, GT boxes are shown in green. Model predicted answers are labeled by Pred, GT answers are labeled by GT.



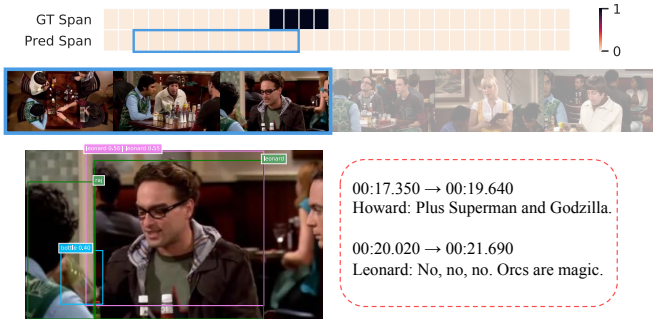
Q: What is Leonard holding when he comes out of the bedroom?
A1: Leonard is holding his cell phone. Pred
A2: Leonard is holding a baseball bat.
A3: Leonard is holding a shovel.
A4: Leonard is holding a coat hanger.
A5: Leonard is holding a mock lightsaber. GT

(a)



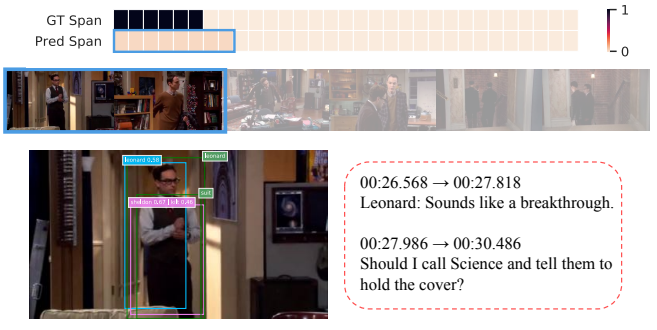
Q: What was Raj doing when Howard was shouting at someone?
A1: Raj was playing some music.
A2: Raj was seated in the couch.
A3: Raj was taking a shower. Pred
A4: Raj was not in the room.
A5: Raj was eating lots of cookies in his mouth as he watched Howard. GT

(b)



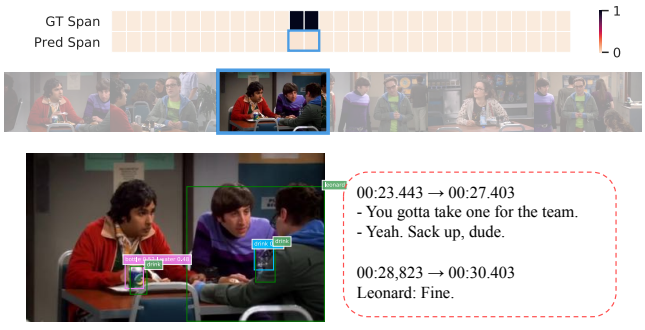
Q: Who grab a bottle after Leonard talked?
A1: Sheldon.
A2: Howard.
A3: Penny.
A4: Raj. GT
A5: Leonard. Pred

(c)



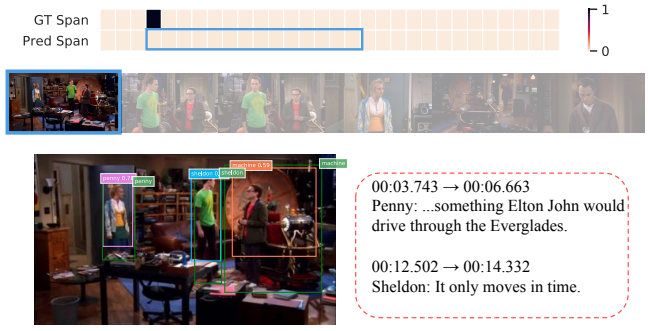
Q: What is Leonard wearing when he is talking to Sheldon?
A1: A scarf.
A2: A hat.
A3: A suit. GT
A4: A kilt. Pred
A5: Jogging pants.

(d)



Q: What was Leonard 's drink when they are talking about taking one for the team?
A1: Fanta.
A2: bottle of water. Pred
A3: Sprite.
A4: Gatorade.
A5: Coke Cola. GT

(e)



Q: What direction did Sheldon turn to when Penny insulted their time machine ?
A1: He looked at his hands.
A2: To the left.
A3: Up towards the ceiling. Pred
A4: He turned to Penny.
A5: To the right. GT

(f)

Figure 10. Wrong prediction examples from STAGE. The span predictions are shown on the top of each example, each block represents a frame, the color indicates the model's confidence for the predicted spans. For each QA, we show grounding examples and scores for one frame in GT span, GT boxes are shown in green. Model predicted answers are labeled by Pred, GT answers are labeled by GT.

References

[1] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, and S. Gould. Bottom-up and top-down attention for image captioning and vqa. *CoRR*, abs/1707.07998, 2017. 1, 2, 5

[2] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. Vqa: Visual question answering. In *ICCV 2015*, 2015. 1, 2

[3] J. Ba, R. Kiros, and G. E. Hinton. Layer normalization. *CoRR*, abs/1607.06450, 2016. 5

[4] F. Chollet. Xception: Deep learning with depthwise separable convolutions. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1800–1807, 2017. 5

[5] A. Das, H. Agrawal, C. L. Zitnick, D. Parikh, and D. Batra. Human attention in visual question answering: Do humans and deep networks look at the same regions? In *EMNLP*, 2016. 2

[6] Y. Dauphin, A. Fan, M. Auli, and D. Grangier. Language modeling with gated convolutional networks. In *ICML*, 2016. 5

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. 5, 7

[8] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111(1):98–136, 2015. 7

[9] J. Gao, C. Sun, Z. Yang, and R. Nevatia. Tall: Temporal activity localization via language query. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5277–5285, 2017. 2, 3

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 5

[11] L. A. Hendricks, O. Wang, E. Shechtman, J. Sivic, T. Darrell, and B. Russell. Localizing moments in video with temporal language. In *EMNLP*, 2018. 2, 6

[12] L. A. Hendricks, O. Wang, E. Shechtman, J. Sivic, T. Darrell, and B. C. Russell. Localizing moments in video with natural language. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5804–5813, 2017. 2, 3, 6

[13] R. Hu, H. Xu, M. Rohrbach, J. Feng, K. Saenko, and T. Darrell. Natural language object retrieval. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4555–4564, 2016. 2

[14] Y. Jang, Y. Song, Y. Yu, Y. Kim, and G. Kim. Tgif-qa: Toward spatio-temporal reasoning in visual question answering. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1359–1367, 2017. 1, 2, 3

[15] L. Kaiser, A. N. Gomez, and F. Chollet. Depthwise separable convolutions for neural machine translation. *CoRR*, abs/1706.03059, 2018. 5

[16] S. Kazemzadeh, V. Ordonez, M. Matten, and T. L. Berg. Referitgame: Referring to objects in photographs of natural scenes. In *EMNLP*, 2014. 2

[17] K.-M. Kim, M.-O. Heo, S.-H. Choi, and B.-T. Zhang. Deepstory: Video story qa by deep embedded memory networks. In *IJCAI*, 2017. 1, 2, 3

[18] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. S. Bernstein, and L. Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123:32–73, 2016. 5

[19] J. Lei, L. Yu, M. Bansal, and T. L. Berg. Tvqa: Localized, compositional video question answering. In *EMNLP*, 2018. 1, 2, 3, 4, 7, 8, 9

[20] Y. Li, Y. Song, and J. Luo. Improving pairwise ranking for multi-label image classification. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1837–1845, 2017. 6

[21] T. Lin, X. Zhao, H. Su, C. Wang, and M. Yang. Bsn: Boundary sensitive network for temporal action proposal generation. In *ECCV*, 2018. 6

[22] C. Liu, J. Mao, F. Sha, and A. L. Yuille. Attention correctness in neural image captioning. In *AAAI*, 2016. 2

[23] J. Lu, J. Yang, D. Batra, and D. Parikh. Hierarchical question-image co-attention for visual question answering. In *NIPS*, 2016. 1

[24] J. Lu, J. Yang, D. Batra, and D. Parikh. Neural baby talk. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7219–7228, 2018. 1

[25] T. Maharaj, N. Ballas, A. C. Courville, and C. J. Pal. A dataset and exploration of models for understanding video data through fill-in-the-blank question-answering. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7359–7368, 2017. 1, 3

[26] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60, 2014. 3

[27] T. Onishi, H. Wang, M. Bansal, K. Gimpel, and D. McAllester. Who did what: A large-scale person-centered cloze dataset. *EMNLP*, 2016. 4

[28] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014. 7

[29] P. Rajpurkar, R. Jia, and P. S. Liang. Know what you don't know: Unanswerable questions for squad. In *ACL*, 2018. 2

[30] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. S. Liang. Squad: 100, 000+ questions for machine comprehension of text. In *EMNLP*, 2016. 2

[31] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:1137–1149, 2015. 5, 6

[32] A. Rohrbach, M. Rohrbach, R. Hu, T. Darrell, and B. Schiele. Grounding of textual phrases in images by reconstruction. In *ECCV*, 2016. 2

[33] M. J. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi. Bidirectional attention flow for machine comprehension. *CoRR*, abs/1611.01603, 2017. 6

[34] K. J. Shih, S. Singh, and D. Hoiem. Where to look: Focus regions for visual question answering. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4613–4621, 2016. 1

[35] M. Tapaswi, Y. Zhu, R. Stiefelhagen, A. Torralba, R. Urtasun, and S. Fidler. Movieqa: Understanding stories in movies through question-answering. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4631–4640, 2016. 1, 2, 3

[36] A. Trott, C. Xiong, and R. Socher. Interpretable counting for visual question answering. *CoRR*, abs/1712.08697, 2018. 2

[37] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *NIPS*, 2017. 5

[38] J. Welbl, P. Stenetorp, and S. Riedel. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association of Computational Linguistics*, 06:287–302, 2018. 2

[39] J. Weston, A. Bordes, S. Chopra, and T. Mikolov. Towards ai-complete question answering: A set of prerequisite toy tasks. *CoRR*, abs/1502.05698, 2016. 2

[40] K. Xu, J. Ba, R. Kiros, K. Cho, A. C. Courville, R. R. Salakhutdinov, R. S. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015. 2

[41] A. W. Yu, D. Dohan, M.-T. Luong, R. Zhao, K. Chen, M. Norouzi, and Q. V. Le. Qanet: Combining local convolution with global self-attention for reading comprehension. *CoRR*, abs/1804.09541, 2018. 5, 6

[42] L. Yu, Z. Lin, X. Shen, J. Yang, X. Lu, M. Bansal, and T. L. Berg. Mattnet: Modular attention network for referring expression comprehension. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[43] L. Yu, E. Park, A. C. Berg, and T. L. Berg. Visual madlibs: Fill in the blank description generation and question answering. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2461–2469, 2015. 1, 2

[44] L. Yu, P. Poirson, S. Yang, A. C. Berg, and T. L. Berg. Modeling context in referring expressions. In *ECCV*, 2016. 2

[45] L. Yu, H. Tan, M. Bansal, and T. L. Berg. A joint speaker-listener-reinforcer model for referring expressions. In *CVPR*, 2017. 2

[46] Y. Yu, J. Choi, Y. Kim, K. Yoo, S.-H. Lee, and G. Kim. Supervising neural attention models for video captioning by human gaze data. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6119–6127, 2017. 1, 2

[47] L. Zhou, Y. Kalantidis, X. Chen, J. J. Corso, and M. Rohrbach. Grounded video description. *CoRR*, abs/1812.06587, 2018. 1, 2

[48] Y. Zhu, O. Groth, M. S. Bernstein, and L. Fei-Fei. Visual7w: Grounded question answering in images. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4995–5004, 2016. 1