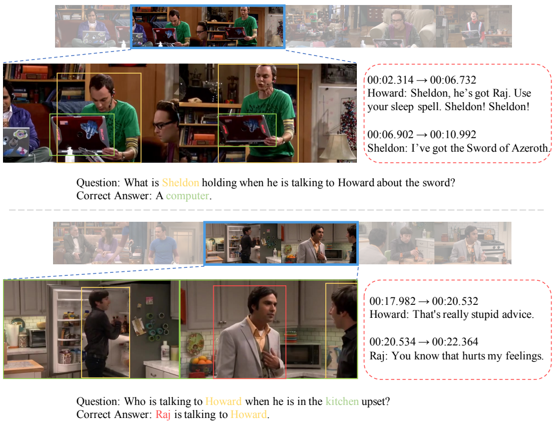
TVQA+: Spatio-Temporal Grounding for Video Question Answering

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 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. Moreover, by performing this joint task, our model is able to produce insightful and interpretable spatio-temporal attention visualizations.[1]



00:02.314 → 00:06.732
Howard: Sheldon, he's got Raj. Use your sleep spell. Sheldon! Sheldon!

00:06.902 → 00:10.992
Sheldon: I've got the Sword of Azeroth.

Question: What is Sheldon holding when he is talking to Howard about the sword?
Correct Answer: A computer.

00:17.982 → 00:20.532
Howard: That's really stupid advice.

00:20.534 → 00:22.364
Raj: You know that hurts my feelings.

Question: Who is talking to Howard when he is in the kitchen upset?
Correct Answer: Raj is talking to Howard.

Figure 1: Samples from TVQA+. Questions and correct answers are temporally localized to clips, and spatially localized to frame-level bounding boxes. Colors indicate corresponding box-object pairs. Subtitles are shown in dashed blocks. Wrong answers are omitted.

1 Introduction

We have witnessed great progress in recent years on image-based visual question answering (QA) tasks (Antol et al., 2015; Yu et al., 2015; Zhu et al., 2016b). One key to this success has been spatial attention (Anderson et al., 2018; Shih et al., 2016; Lu et al., 2016), 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 QA pairs, without providing labels for the key clips or regions needed to answer the question. Inspired by previous work on grounded image and video captioning (Lu et al., 2018; Zhou et al., 2019), we propose methods that explicitly localize video clips 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 also collect new joint spatio-temporal annotations for an existing video QA dataset.

In the past few years, several video QA datasets have been proposed, e.g., MovieFIB (Maharaj et al., 2017), MovieQA (Tapaswi et al., 2016), TGIF-QA (Jang et al., 2017), PororoQA (Kim et al., 2017), MarioQA (Mun et al., 2017), and TVQA (Lei et al., 2018). TVQA is one of the largest video QA datasets, providing a large video QA dataset built on top of 6 famous TV series. Be-

[1] Dataset and code are publicly available: http://tvqa.cs.unc.edu, https://github.com/jayleicn/TVQAplus

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 8211–8225
July 5 - 10, 2020. ©2020 Association for Computational Linguistics

cause TVQA was collected on television shows, it is built on natural video content with rich dynamics and complex 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. Movie (Tapaswi et al., 2016; Maharaj et al., 2017) and television show (Lei et al., 2018) videos come with the limitation of being scripted and edited, but they are still more realistic than cartoon/animation (Kim et al., 2017) and game (Mun et al., 2017) videos, and they also come with richer, real-world-inspired inter-human interactions and span across diverse domains (e.g., medical, crime, sitcom, etc.), making them a useful testbed to study complex video understanding by machine learning models.

One key property of TVQA is that it provides temporal annotations denoting which parts of a video clip are necessary for answering a proposed question. However, none of the existing 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 look at the region of *"What is Sheldon holding"*.

Hence, in this paper, we first augment a subset of the TVQA dataset with grounded bounding boxes, resulting in a spatio-temporally grounded video QA dataset, TVQA+. It consists of 29.4K multiple-choice questions grounded in both the temporal and the 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 concepts with object regions in individual frames, if there are any, by annotating bounding boxes for each referred concept (see examples in Fig. 1). Our TVQA+ dataset has a total of 310.8K bounding boxes linked with referred objects and people, spanning across 2.5K categories (more details in Sec. 3).

With such richly annotated data, we then 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, temporal localization accuracy, and a joint temporal localization and QA accuracy. 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. We find that the QA performance benefits from both temporal moment and spatial region supervision. Additionally, we provide visualization of temporal and spatial localization, which is helpful for understanding what our model has learned. Comprehensive ablation studies demonstrate how each of our annotations and model components helps to improve the performance of the tasks.

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 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 baselines, as well as presents insightful, interpretable visualizations.

2 Related Work

Question Answering In recent years, multiple question answering datasets and tasks have been proposed to facilitate research towards this goal, in both vision and language communities, in the form of visual question answering (Antol et al., 2015; Yu et al., 2015; Jang et al., 2017) and textual question answering (Rajpurkar et al., 2016; Weston et al., 2016), respectively. Video question answering (Lei et al., 2018; Tapaswi et al., 2016; Kim et al., 2017) with naturally occurring subtitles are particularly interesting, as it combines both visual and textual information for question answering. Different from

Dataset	Origin	Task	#Clips/#QAs (#Sentences)	#Boxes	Temporal Annotation
MovieFIB (Maharaj et al., 2017)	Movie	QA	118.5K/349K	-	✗
MovieQA (Tapaswi et al., 2016)	Movie	QA	6.8K/6.5K	-	✓
TGIF-QA (Jang et al., 2017)	Tumblr	QA	71.7K/165.2K	-	✗
PororoQA (Kim et al., 2017)	Cartoon	QA	16.1K/8.9K	-	✗
DiDeMo (Hendricks et al., 2017)	Flickr	TL	10.5K/40.5K	-	✓
Charades-STA (Gao et al., 2017)	Home	TL	-/19.5K	-	✓
TVQA (Lei et al., 2018)	TV Show	QA/TL	21.8K/152.5K	-	✓
ANet-Entities (Zhou et al., 2019)	Youtube	CAP/TL/SL	15K/52K	158K	✓
TVQA+	TV Show	QA/TL/SL	4.2K/29.4K	310.8K	✓

Table 1: Comparison of TVQA+ with other video-language datasets. TL=Temporal Localization, SL=Spatial Localization, CAP=Captioning.

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 modalities. Earlier works (Kazemzadeh et al., 2014; Yu et al., 2017, 2018b; Rohrbach et al., 2016) focused on identifying the referred object in an image. Recently, there has been a growing interest in moment retrieval tasks (Hendricks et al., 2017, 2018; Gao et al., 2017), where the goal is to localize a short clip from a long video via a natural language query. Our work integrates the goals 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 (Anderson et al., 2018; Xu et al., 2015), visual question answering (Anderson et al., 2018; Trott et al., 2018), language grounding (Yu et al., 2018b), etc. However, sometimes the attention learned by the model itself may not agree with human expectations (Liu et al., 2016; Das et al., 2016). Recent works on grounded image captioning and video captioning (Lu et al., 2018; Zhou et al., 2019) 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.

Split	#Clips/#QAs	#Annotated Images	#Boxes	#Categories
Train	3,364/23,545	118,930	249,236	2,281
Val	431/3,017	15,350	32,682	769
Test	403/2,821	14,188	28,908	680
Total	4,198/29,383	148,468	310,826	2,527

Table 2: Data Statistics for TVQA+ dataset.

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 by Lei et al.. 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 compositional, 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 that object annotations could provide an additional useful training signal for models to learn a deeper understanding

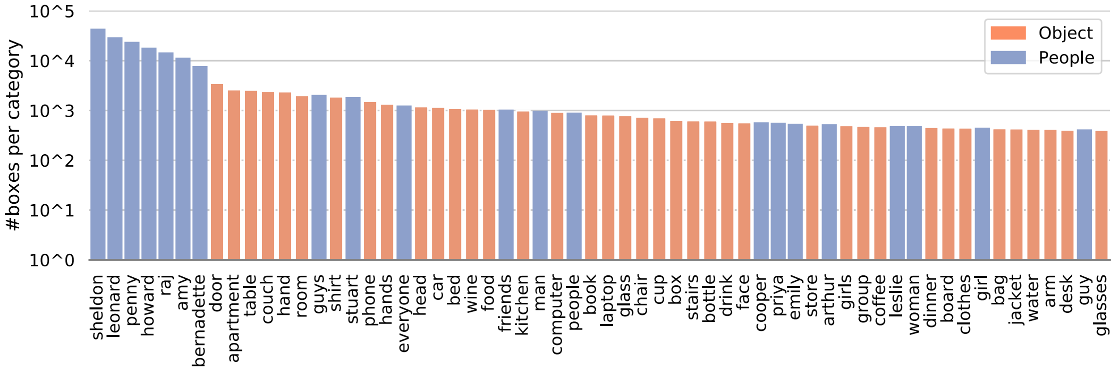


Figure 2: Box distributions for top 60 categories in TVQA+ train set.

of visual information. 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 *The Big Bang Theory*. This subset 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 tagger (Manning et al., 2014) 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 workers to adjust the start and end timestamps to refine the temporal annotation, as we found the original temporal annotation were not ideally tight. We then sample one frame every two seconds from each span for spatial annotation. For each frame, we collect the bounding boxes for the visual concepts in each QA pair. We also experimented with semi-automated annotation for people with face detection (Zhang et al., 2016) and recognition model (Liu et al., 2017), but they do not work well mainly due to many partial occlusion of faces (e.g., side faces) in the frames. During annotation, we provide the original videos (with subtitles) to help the workers understand the context for the given QA pair. More annotation details (including quality check) are presented in the appendix.

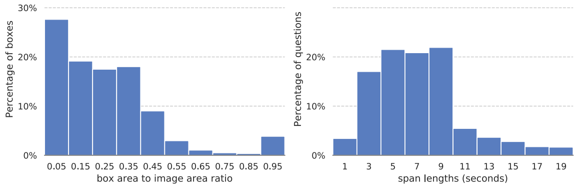


Figure 3: Box/image area ratios (left) and span length distributions (right) in TVQA+.

3.2 Dataset Analysis

TVQA+ contains 29,383 QA pairs from 4,198 videos, with 148,468 images annotated with 310,826 bounding boxes. Statistics of TVQA+ are shown in Table 2. Note that we follow the same data splits as the original TVQA dataset, supporting future research on both TVQA and TVQA+. Table 1 compares TVQA+ dataset with other video-language datasets. TVQA+ is unique as it supports three tasks: question answering, temporal localization, and spatial localization.

It is also of reasonable size compared to the grounded video captioning dataset ANet-Entities (Zhou et al., 2019). 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 Fig. 2. 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 bounding box area over image area ratio in Fig. 3 (left). The majority of boxes are fairly small compared to the image, which makes object grounding challenging. Fig. 3 (right) shows the distribution of localized span length. While most spans are

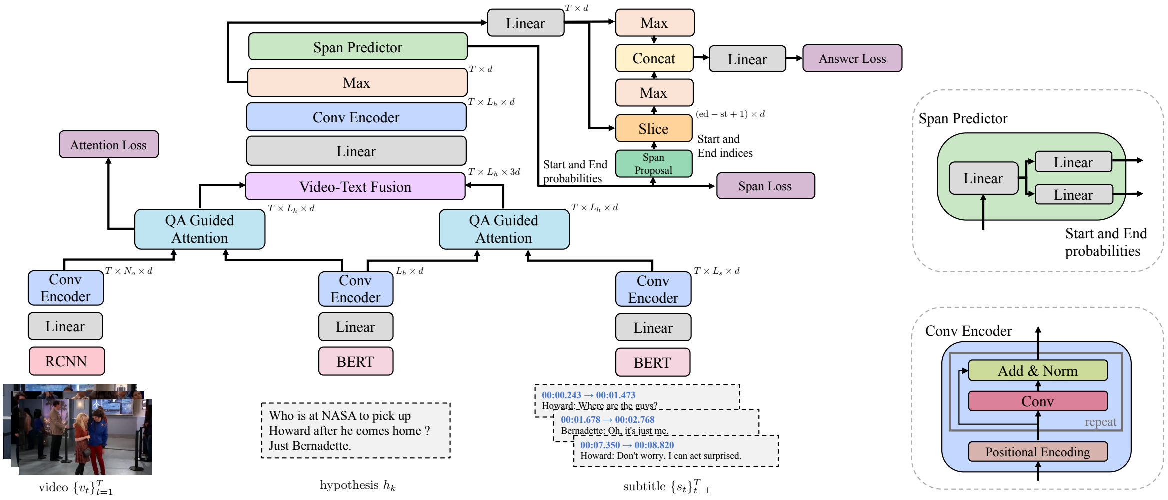


Figure 4: Overview of the proposed STAGE framework.

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 clips (61.49 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) 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 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/people. The attended QA-aware video and subtitle representation are then fused together to obtain a joint frame-wise representation. Third, taking the frame-wise representation as input, STAGE learns to predict QA relevant temporal spans, then combines the global and local (span localized) video information to answer the questions. In the following, we describe STAGE 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. Our goal is to predict the answer and ground it 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 (Onishi et al., 2016). 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). Subtitle sentences are then temporally aligned with the video frames. Specifically, for each frame v_t, we pair it with two neighboring sentences based on the subtitle timestamps. 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 $\{s_t\}_{t=1}^{T}$. We denote the number of words in each hypothesis and subtitle as L_h, 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 For each frame v_t, we use Faster R-CNN (Ren et al., 2015) pre-trained on Visual Genome (Krishna et al., 2017) to detect objects and extract their regional representation as our visual features (Anderson et al., 2018). We keep the top-20 object proposals and use PCA to reduce the feature dimension from 2048 to 300, to save GPU memory and computation. 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 (Devlin et al., 2019), a transformer-based language model (Vaswani et al., 2017) that achieves state-of-the-art performance on various NLP tasks. Specifically, we first fine-tune the BERT-base model using the masked language model and next sentence prediction objectives on the subtitles and QA pairs from TVQA+ train set. Then, we fix its parameters and use it to extract 768D word-level embeddings from the second-to-last layer for the subtitles and each hypothesis. Both embeddings are projected into a 128D space using a linear layer with ReLU.

Convolutional Encoder Inspired by the recent trend of replacing recurrent networks with CNNs (Dauphin et al., 2016; Yu et al., 2018a) and Transformers (Vaswani et al., 2017; Devlin et al., 2019) for sequence modeling, we use positional encoding (PE), CNNs, and layer normalization (Ba et al., 2016) to build our basic encoding block. As shown in the bottom-right corner of Fig. 4, it is comprised of a PE layer and multiple convolutional layers, each with a residual connection (He et al., 2016) and layer normalization. We use $\mathrm{Layernorm}(\mathrm{ReLU}(\mathrm{Conv}(x)) + x)$ to denote 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 (Chollet, 2017). 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 $N_{\mathrm{conv}} = 2$.

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:

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

where $\odot$ denotes hadamard product, $W_F \in \mathbb{R}^{3d \times d}$ 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. After collecting $F_{k,t}^{att}$ from all time steps, we get $F_k^{att} \in \mathbb{R}^{T \times L_h \times d}$. 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 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 $\mathbf{p_k^1} \in \mathbb{R}^T$ and end probabilities $\mathbf{p_k^2} \in \mathbb{R}^T$ using two linear layers with softmax, as shown in the top-right corner of Fig. 4. Different from existing works (Seo et al., 2017; Yu et al., 2018a) that used the span predictor for text only, we use it for a joint localization of both video and text, which requires properly-aligned joint embeddings.

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 $G_k^g \in \mathbb{R}^d$. With the start and end probabilities from the span predictor, we generate span proposals using dynamic programming (Seo et al., 2017). 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 $\{st_p, ed_p\}$ (Ren et al., 2015). At inference time, we take the proposals with the highest confidence scores for each hypothesis. For each proposal, we generate a local representation $G_k^l \in \mathbb{R}^d$ by max-pooling $A_{k,st_p:ed_p}$. The local and global representations are concatenated to obtain $G_k \in \mathbb{R}^{2d}$. We then forward $\{G_k\}_{k=1}^5$ through softmax to get the answer scores $\mathbf{p}^{ans} \in \mathbb{R}^5$. Compared with existing works (Jang et al., 2017; Zhao et al., 2017) that use soft temporal attention, we use more interpretable hard attention, extracting local features (together with global features) for question answering.

4.3 Training and Inference

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.

Spatial Supervision While the attention described in Sec. 4.2 can be learned in a weakly supervised end-to-end manner, we can also train it with supervision from 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 $\{o_{t,r}\}_{r=1}^{N_o}$ at frame v_t. We expect the attention on positive boxes to be higher than the negative ones, and therefore use LSE (Li et al., 2017) loss for the supervision:

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

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. LSE loss is a smoothed alternative to the widely used hinge loss, it is easier to optimize than the original hinge loss (Li et al., 2017). 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 $\mathcal{L}^{att}_{t,j}$. We define the overall attention loss $\mathcal{L}^{att} = \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}^{att}_i$. At inference time, we choose the boxes with scores higher than 0.2 as the predictions.

Temporal Supervision Given softmax normalized start and end probabilities $\mathbf{p}^1$ and $\mathbf{p}^2$, we apply cross-entropy loss:

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

where y^1_i and y^2_i are the GT start and end indices.

Answer Prediction Similarly, given answer probabilities $\mathbf{p}^{ans}$, our answer prediction loss is:

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

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

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

5 Experiments

As introduced, 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 define the evaluation metrics, then compare STAGE against several baselines, and finally provide a comprehensive analysis of our model. Additionally, we also evaluate STAGE on the full TVQA dataset.

Model	QA Acc.	Grd. mAP	Temp. mIoU	ASA
ST-VQA (Jang et al., 2017)	48.28	-	-	-
two-stream (Lei et al., 2018)	68.13	-	-	-
STAGE (video)	52.75	26.28	10.90	2.76
STAGE (sub)	67.99	-	30.16	20.13
STAGE	**74.83**	**27.34**	**32.49**	**22.23**
Human (Lei et al., 2018)	90.46	-	-	-

Table 3: TVQA+ test set results.

5.1 Metrics

To measure QA performance, we use classification accuracy (QA Acc.). We evaluate *span prediction* using temporal mean Intersection-over-Union (Temp. mIoU) following previous work (Hendricks et al., 2017) 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 Answer-Span joint Accuracy (ASA), that jointly evaluates both answer prediction and span prediction. For this metric, 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 (Everingham et al., 2015) 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 two-stream model (Lei et al., 2018) 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 both streams. We retrain the model using the official code[2] on TVQA+ data, with the same feature as STAGE. We also consider ST-VQA (Jang et al., 2017) model, which is primarily designed for question answering on short videos (GIFs). We also provide STAGE variants that use only video or subtitle to study the effect of using only one of the modalities. Table 3 shows the test results of STAGE and the baselines. STAGE outperforms the baseline model (two-stream) by a large margin in QA Acc.,[3] with 9.83% relative gains. Additionally, STAGE also lo-

[2] `https://github.com/jayleicn/TVQA`

[3] This also holds true when considering mean (standard-deviation) of 5 runs: 74.20 (0.42).

Model	QA Acc.	Grd. mAP	Temp. mIoU	ASA
baseline	65.79	2.74	-	-
+ CNN	67.25	3.16	-	-
+ Aligned Fusion (backbone)	68.31	7.31	-	-
+ Temp. Sup.	71.40	10.86	30.77	20.09
+ Spat. Sup.	71.99	24.10	31.16	20.42
+ Local Feature (STAGE)	**72.56**	**25.22**	**31.67**	**20.78**
STAGE with GT Span	73.28	-	-	-

Table 4: Ablation study of STAGE on TVQA+ val set. *Each row adds an extra component to the row above it.*

Model	baseline	+CNN	+AF	+TS	+SS	+LF
what (60.52%)	65.66	66.43	67.58	70.76	71.25	**72.34**
who (10.24%)	65.37	64.08	64.72	72.17	73.14	**74.11**
where (9.68%)	65.41	64.38	68.49	71.58	71.58	**74.32**
why (9.55%)	74.31	78.82	77.43	**79.86**	78.12	76.39
how (9.05%)	60.81	67.03	69.23	66.30	**69.96**	67.03
total (100%)	65.79	67.25	68.31	71.40	71.99	**72.56**

Table 5: QA Acc. by question type on TVQA+ val set. For brevity, we only show top-5 question types (percentage in brackets). AF=Aligned Fusion, TS=Temp. Sup., SS=Spat. Sup., LF=Local Feature. *Each column adds an extra component to the column before it.*

calizes the relevant moments with temporal mIoU of 32.49% and detects referred objects and people with mAP of 27.34%. However, a large gap is still observed between STAGE and human, showing space for further improvement.

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 remove the span predictor along with the span proposal module, as well as the explicit attention supervision. We further replace the CNN encoders with RNN encoders, and remove the aligned fusion from the backbone model. This baseline model uses RNN to encode input sequences and interacts QA pairs with subtitles and videos separately. The final confidence score is the sum of the confidence scores from the two modalities. In the backbone model, we align subtitles with video frames from the start, fusing their representation conditioned on the input QA pair, as in Fig. 4. 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. With both changes, our backbone model obtains 68.31% on QA Acc., significantly higher than the baseline's 65.79%. The results are shown in Table 4.

Model	Temp. Sup.	val	test-public
two-stream (Lei et al., 2018)	✗	65.85	66.46
PAMN (Kim et al., 2019b)	✗	66.38	66.77
multi-task (Kim et al., 2019a)	✓	66.22	67.05
STAGE backbone (GloVe)	✗	66.46	-
STAGE backbone + Temp. Sup. (GloVe)	✓	66.92	-
STAGE backbone	✗	68.56	69.67
STAGE backbone + Temp. Sup.	✓	**70.50**	**70.23**

Table 6: QA Acc. on the full TVQA dataset.

Temporal and Spatial Supervision In Table 4, we also show the results when using temporal and spatial supervision. After adding temporal supervision, the model is be able to ground on the temporal axis, which also improves the model's performance on other tasks. Adding spatial supervision gives additional improvements, particularly for Grd. mAP, with 121.92% relative gain.

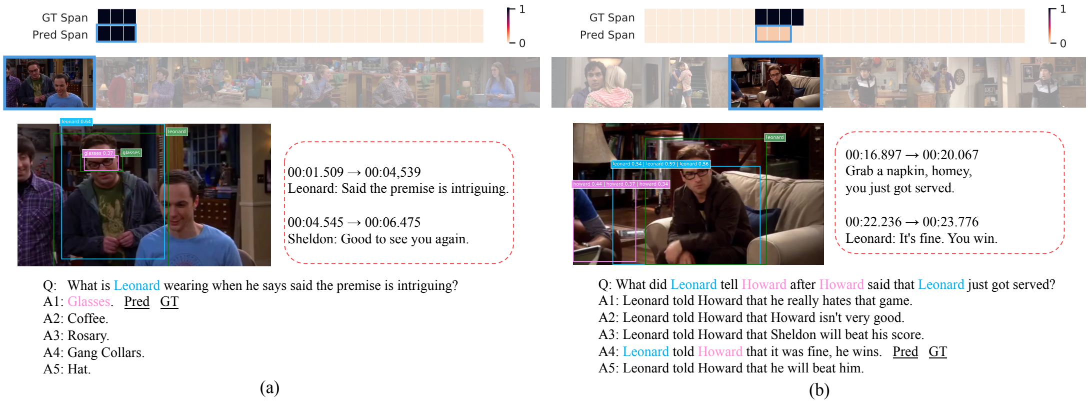
Span Proposal and Local Feature In the second-to-last row of Table 4, we show our full STAGE model, which is augmented with local features G^l for question answering. Local features are obtained by max-pooling the span proposal regions, which contain more relevant cues for answering the questions. With G^l, we achieve the best performance across all metrics, indicating the benefit of using local features.

Inference with GT Span The last row of Table 4 shows our model uses 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 question type. We observe a clear increasing trend on "what", "who", and "where" questions after using the backbone net and adding attention/span modules in each column. Interestingly, for "why" and "how" questions, 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. 5, where Fig. 5(a) uses grounded objects to answer the question, and Fig. 5(b) uses text. More examples (including failure cases) are provided in the appendix.

TVQA Results We also conduct experiments on the full TVQA dataset (Table 6), without relying on the bounding boxes and refined timestamps in TVQA+. Without temporal supervision, STAGE backbone is able to achieve 3.91% relative gain from the best published result (multi-task) on


GT Span
Pred Span
1
0
leonard 0.64
leonard
glasses 0,37
glasses
GT Span
Pred Span
1
0
leonard
leonard 0.54 | leonard 0.59 | leonard 0.56
howard 0.44 | howard 0.37 | howard 0.34

00:01.509 → 00:04,539
Leonard: Said the premise is intriguing.

00:04.545 → 00:06.475
Sheldon: Good to see you again.

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)

00:16.897 → 00:20.067
Grab a napkin, homey,
you just got served.

00:22.236 → 00:23.776
Leonard: It's fine. You win.

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 5: Example predictions from STAGE. Span predictions are shown on the top, each block represents a frame, the color indicates the model's confidence for the spans. For each QA, we show grounding examples and scores for one frame in GT span. GT boxes are in green. Predicted and GT answers are labeled by Pred and GT, respectively.

TVQA test-public set. Adding temporal supervision, performance is improved to 70.23%. For a fair comparison, we also provided STAGE variants using GloVe (Pennington et al., 2014) instead of BERT (Devlin et al., 2019) as text feature. Using GloVe, STAGE models still achieve better results.

6 Conclusion

We collected the TVQA+ dataset and proposed the spatio-temporal video QA task. This task requires systems to jointly localize relevant moments, detect referred objects/people, and answer questions. We further introduced STAGE, an end-to-end trainable framework to jointly perform all three tasks. Comprehensive experiments show that temporal and spatial predictions help improve QA performance, as well as providing explainable results. Though our STAGE achieves state-of-the-art performance, there is still a large gap compared with human performance, leaving space for further improvement.

Acknowledgement

We thank the reviewers for their helpful feedback. This research is supported by NSF Awards #1633295, 1562098, 1405822, DARPA MCS Grant #N66001-19-2-4031, DARPA KAIROS Grant #FA8750-19-2-1004, Google Focused Research Award, and ARO-YIP Award #W911NF-18-1-0336.

References

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, and Stephen Gould. 2018. Bottom-up and top-down attention for image captioning and vqa. In *CVPR*.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *ICCV*.

Jimmy Ba, Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. *CoRR*, abs/1607.06450.

François Chollet. 2017. Xception: Deep learning with depthwise separable convolutions. In *CVPR*.

Abhishek Das, Harsh Agrawal, C. Lawrence Zitnick, Devi Parikh, and Dhruv Batra. 2016. Human attention in visual question answering: Do humans and deep networks look at the same regions? In *EMNLP*.

Yann Dauphin, Angela Fan, Michael Auli, and David Grangier. 2016. Language modeling with gated convolutional networks. In *ICML*.

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

Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. 2015. The pascal visual object classes challenge: A retrospective. *IJCV*.

Jiyang Gao, Chen Sun, Zhenheng Yang, and Ramakant Nevatia. 2017. Tall: Temporal activity localization via language query. In *ICCV*.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *CVPR*.

Chunhui Gu, Chen Sun, David A. Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, Cordelia Schmid, and Jitendra Malik. 2018. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *CVPR*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*.

Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. 2018. Localizing moments in video with temporal language. In *EMNLP*.

Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan C. Russell. 2017. Localizing moments in video with natural language. In *ICCV*.

Drew A Hudson and Christopher D Manning. 2019. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *CVPR*.

Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. 2017. Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In *CVPR*.

Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara L. Berg. 2014. Referitgame: Referring to objects in photographs of natural scenes. In *EMNLP*.

Junyeong Kim, Minuk Ma, Kyungsu Kim, Sungjin Kim, and Chang Dong Yoo. 2019a. Gaining extra supervision via multi-task learning for multi-modal video question answering. *IJCNN*.

Junyeong Kim, Minuk Ma, Kyungsu Kim, Sungjin Kim, and Chang Dong Yoo. 2019b. Progressive attention memory network for movie story question answering. In *CVPR*.

Kyung-Min Kim, Min-Oh Heo, Seong-Ho Choi, and Byoung-Tak Zhang. 2017. Deepstory: Video story qa by deep embedded memory networks. In *IJCAI*.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*.

Jie Lei, Licheng Yu, Mohit Bansal, and Tamara L Berg. 2018. Tvqa: Localized, compositional video question answering. In *EMNLP*.

Yuncheng Li, Yale Song, and Jiebo Luo. 2017. Improving pairwise ranking for multi-label image classification. In *CVPR*.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *ECCV*.

Chenxi Liu, Junhua Mao, Fei Sha, and Alan Loddon Yuille. 2016. Attention correctness in neural image captioning. In *AAAI*.

Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. 2017. Sphereface: Deep hypersphere embedding for face recognition. In *CVPR*.

Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. 2016. Hierarchical question-image co-attention for visual question answering. In *NeurIPS*.

Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. 2018. Neural baby talk. In *CVPR*.

Tegan Maharaj, Nicolas Ballas, Aaron C. Courville, and Christopher Joseph Pal. 2017. A dataset and exploration of models for understanding video data through fill-in-the-blank question-answering. In *CVPR*.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *ACL*.

Jonghwan Mun, Paul Hongsuck Seo, Ilchae Jung, and Bohyung Han. 2017. Marioqa: Answering questions by watching gameplay videos. In *ICCV*.

Takeshi Onishi, Hai Wang, Mohit Bansal, Kevin Gimpel, and David A. McAllester. 2016. Who did what: A large-scale person-centered cloze dataset. In *EMNLP*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy S. Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. In *EMNLP*.

Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *TPAMI*.

Anna Rohrbach, Marcus Rohrbach, Ronghang Hu, Trevor Darrell, and Bernt Schiele. 2016. Grounding of textual phrases in images by reconstruction. In *ECCV*.

Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. In *ICLR*.

Kevin J. Shih, Saurabh Singh, and Derek Hoiem. 2016. Where to look: Focus regions for visual question answering. In *CVPR*.

Hao Tan and Mohit Bansal. 2019. Lxmert: Learning cross-modality encoder representations from transformers. In *EMNLP*.

Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2016. Movieqa: Understanding stories in movies through question-answering. In *CVPR*.

Alexander Trott, Caiming Xiong, and Richard Socher. 2018. Interpretable counting for visual question answering. In *ICLR*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*.

Jason Weston, Antoine Bordes, Sumit Chopra, and Tomas Mikolov. 2016. Towards ai-complete question answering: A set of prerequisite toy tasks. *CoRR*, abs/1502.05698.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan R. Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*.

Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V. Le. 2018a. Qanet: Combining local convolution with global self-attention for reading comprehension. In *ICLR*.

Licheng Yu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, Mohit Bansal, and Tamara L. Berg. 2018b. Mattnet: Modular attention network for referring expression comprehension. In *CVPR*.

Licheng Yu, Eunbyung Park, Alexander C. Berg, and Tamara L. Berg. 2015. Visual madlibs: Fill in the blank description generation and question answering. In *ICCV*.

Licheng Yu, Hao Tan, Mohit Bansal, and Tamara L Berg. 2017. A joint speaker-listener-reinforcer model for referring expressions. In *CVPR*.

Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. From recognition to cognition: Visual commonsense reasoning. In *CVPR*.

Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. 2016. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*.

Zhou Zhao, Qifan Yang, Deng Cai, Xiaofei He, and Yueting Zhuang. 2017. Video question answering via hierarchical spatio-temporal attention networks. In *IJCAI*.

Luowei Zhou, Yannis Kalantidis, Xinlei Chen, Jason J. Corso, and Marcus Rohrbach. 2019. Grounded video description. In *CVPR*.

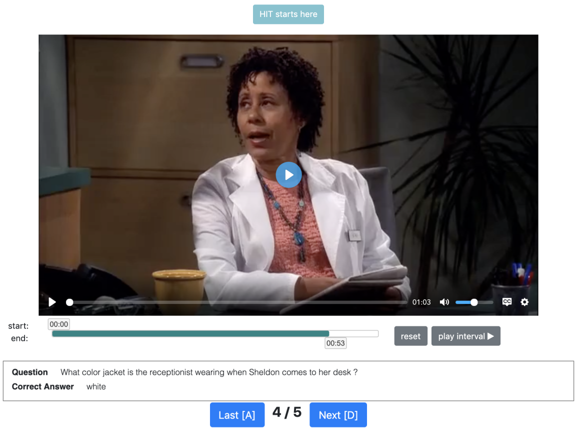
Yuke Zhu, Oliver Groth, Michael Bernstein, and Li Fei-Fei. 2016a. Visual7W: Grounded Question Answering in Images. In *CVPR*.

Yuke Zhu, Oliver Groth, Michael S. Bernstein, and Li Fei-Fei. 2016b. Visual7w: Grounded question answering in images. In *CVPR*.

A Appendices

A.1 Timestamp Annotation

During our initial analysis, we find the original timestamp annotations from the TVQA (Lei et al., 2018) 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. 6, 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.



start: 00:00
end: 00:53

Question What color jacket is the receptionist wearing when Sheldon comes to her desk ?
Correct Answer white

Figure 6: Timestamp refinement interface.

We show span length distributions of the original and the refined timestamps from TVQA+ train set in Fig. 7. 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 STAGE performance on TVQA+ val set using the original timestamps and the refined timestamps. Models with the refined timestamps performs consistently better than the ones with the original timestamps.

A.2 Bounding Box Annotation

At each step, we show a question, its correct answer, and the sampled video frames to an AMT

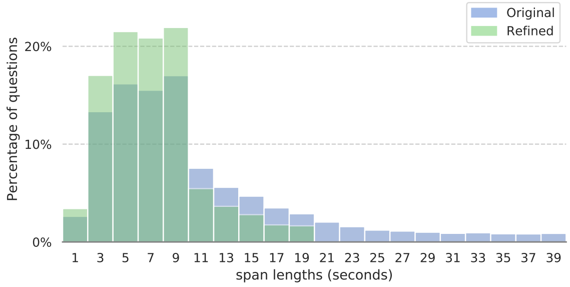
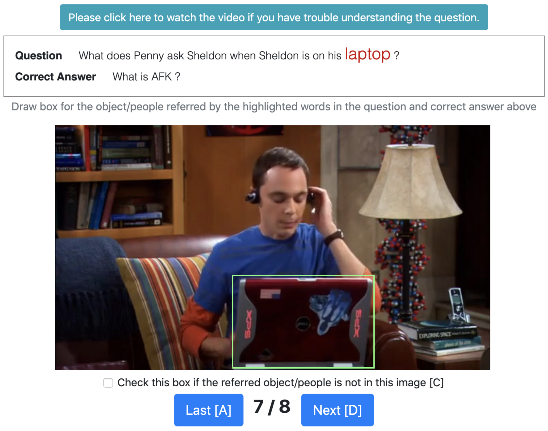


Figure 7: Comparison between the original and the refined timestamps in the TVQA+ train set. The refined timestamps are generally tighter than the original ones.

Model	QA Acc.	
	Original	Refined
STAGE backbone	68.31	68.31
+ Temp. Sup.	70.87	71.40
+ Spat. Sup.	71.23	71.99
+ Local Feature (STAGE)	70.63	**72.56**

Table 7: STAGE performance comparison between the original timestamps and the refined timestamps on TVQA+ val set. *Each row adds an extra component to the row above it.*


Please click here to watch the video if you have trouble understanding the question.
Last [A]
7 / 8
Next [D]

Question What does Penny ask Sheldon when Sheldon is on his laptop ?

Correct Answer What is AFK ?

Draw box for the object/people referred by the highlighted words in the question and correct answer above

□ Check this box if the referred object/people is not in this image [C]

Figure 8: Bounding box annotation interface. Here, the worker is asked to draw a box around the highlighted word "laptop".

worker. (illustrated in Fig. 8). We do not annotate the wrong answers as most of them cannot be grounded in the video. We checked 200 sampled QAs - only 3.13% of the wrong answers could be grounded, while 46% of the correct answers could be grounded. 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 Fig. 8, the word "laptop" is highlighted, and workers are instructed to draw a box around it. In our MTurk instructions, we required workers to draw boxes for each instance of a plural word. E.g., for the word "everyone", the worker need to draw a box for each person in the frame. 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. Recent works (Zellers et al., 2019; Gu et al., 2018) suggest the use of pre-trained detectors for semi-automated annotation. However, since TVQA+ has a wide range of categories (see Fig. 2 and Table 1), it is challenging to use off-the-shelf detectors in the annotation process. As face detection and recognition might be easier than recognizing open set objects, we initially also tried using strong face detection (Zhang et al., 2016) and recognition (Liu et al., 2017) model for character face annotation, but the quality was much poorer than expected. Thus, we decided to invest the required funds to collect boxes manually and ensure their accuracy. After the collection, with the GT labels, we again used the above models to test face retrieval performance for 12 most frequently appeared characters in TVQA+. To allow (Liu et al., 2017) to work, we manually collected 5 GT faces for each character as our gallery set. At test time, we assign each test face the label of its closest neighbor from the gallery set in the learned embedding space. This method achieves 55.6 F1/74.4 Precision/44.4 Recall. Such performance is not strong enough to support further research. We found the main reason is due to many partial occlusion of faces (e.g., side faces) in TV shows.

A.3 Quality

To ensure the quality of the collected bounding boxes, we only allow workers from English-speaking countries to participate the task. Besides, we set high requirements for workers – they needed to have at least 3000 accepted HITs and 95% accept rate. Qualified workers were well paid. We also kept track of the quality of the data during collection - workers with poor annotations were disqualified to work on our task. After collection, we further conducted an in-house check, 95.5% of 200 sampled QAs are correctly labeled, indicating the high quality of our data.

A.4 Training Details

We optimize our model using Adam with an initial learning rate of 1e-3, weight decay 3e-7. A mini-

Model	QA Acc.	Grd. mAP	Temp. mIoU	ASA
STAGE-LXMERT	71.46	21.01	26.31	18.04
STAGE	**74.83**	**27.34**	**32.49**	**22.23**

Table 8: TVQA+ test set results with LXMERT.

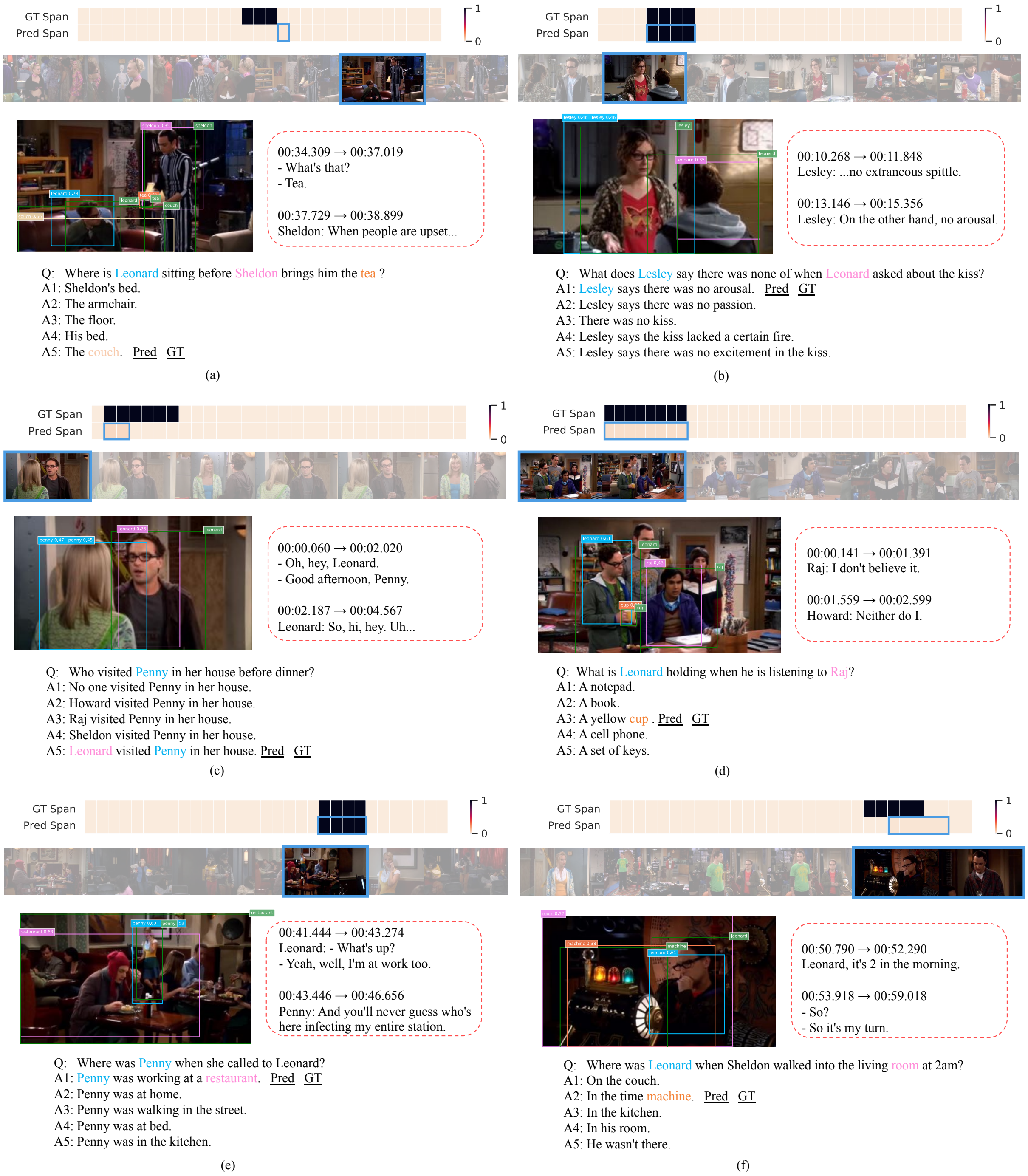
batch contains 16 questions. We train the model for maximum 100 epochs with early stop – if QA Acc. is not improving for consecutive 5 epochs, the training is stopped. CNN hidden size is set to 128.

A.5 Vision-Language Pretrained Features

In addition, we also consider features from LXMERT (Tan and Bansal, 2019). This model is pretrained on a large amount of image-text pairs from multiple image captioning (Lin et al., 2014; Krishna et al., 2017) and image question answering (Goyal et al., 2017; Hudson and Manning, 2019; Zhu et al., 2016a) datasets. Specifically, we use video frame-question pairs as input to LXMERT, and use the extracted features to replace Faster R-CNN object features and BERT question features. For answers and subtitles, we still use the original BERT features. The results are shown in Table 8. We notice that using LXMERT feature lowers STAGE's performance. This is not surprising, as the domains in which the LXMERT model are pretrained on are very different from TVQA+: (captions/questions+image) vs (subtitles+QAs+videos). Future work includes more investigation into adapting these pre-trained vision-language models for more challenging video+dialogue domains.

A.6 More Prediction 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 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).



00:34.309 → 00:37.019
- What's that?
- Tea.

00:37.729 → 00:38.899
Sheldon: When people are upset...

Q: Where is Leonard sitting before Sheldon brings him the tea ?
A1: Sheldon's bed.
A2: The armchair.
A3: The floor.
A4: His bed.
A5: The couch. Pred GT

(a)

00:10.268 → 00:11.848
Lesley: ...no extraneous spittle.

00:13.146 → 00:15.356
Lesley: On the other hand, no arousal.

Q: What does Lesley say there was none of when Leonard asked about the kiss?
A1: Lesley says there was no arousal. Pred GT
A2: Lesley says there was no passion.
A3: There was no kiss.
A4: Lesley says the kiss lacked a certain fire.
A5: Lesley says there was no excitement in the kiss.

(b)

00:00.060 → 00:02.020
- Oh, hey, Leonard.
- Good afternoon, Penny.

00:02.187 → 00:04.567
Leonard: So, hi, hey. Uh...

Q: Who visited Penny in her house before dinner?
A1: No one visited Penny in her house.
A2: Howard visited Penny in her house.
A3: Raj visited Penny in her house.
A4: Sheldon visited Penny in her house.
A5: Leonard visited Penny in her house. Pred GT

(c)

00:00.141 → 00:01.391
Raj: I don't believe it.

00:01.559 → 00:02.599
Howard: Neither do I.

Q: What is Leonard holding when he is listening to Raj?
A1: A notepad.
A2: A book.
A3: A yellow cup . Pred GT
A4: A cell phone.
A5: A set of keys.

(d)

00:41.444 → 00:43.274
Leonard: - What's up?
- Yeah, well, I'm at work too.

00:43.446 → 00:46.656
Penny: And you'll never guess who's here infecting my entire station.

Q: Where was Penny when she called to Leonard?
A1: Penny was working at a restaurant. Pred GT
A2: Penny was at home.
A3: Penny was walking in the street.
A4: Penny was at bed.
A5: Penny was in the kitchen.

(e)

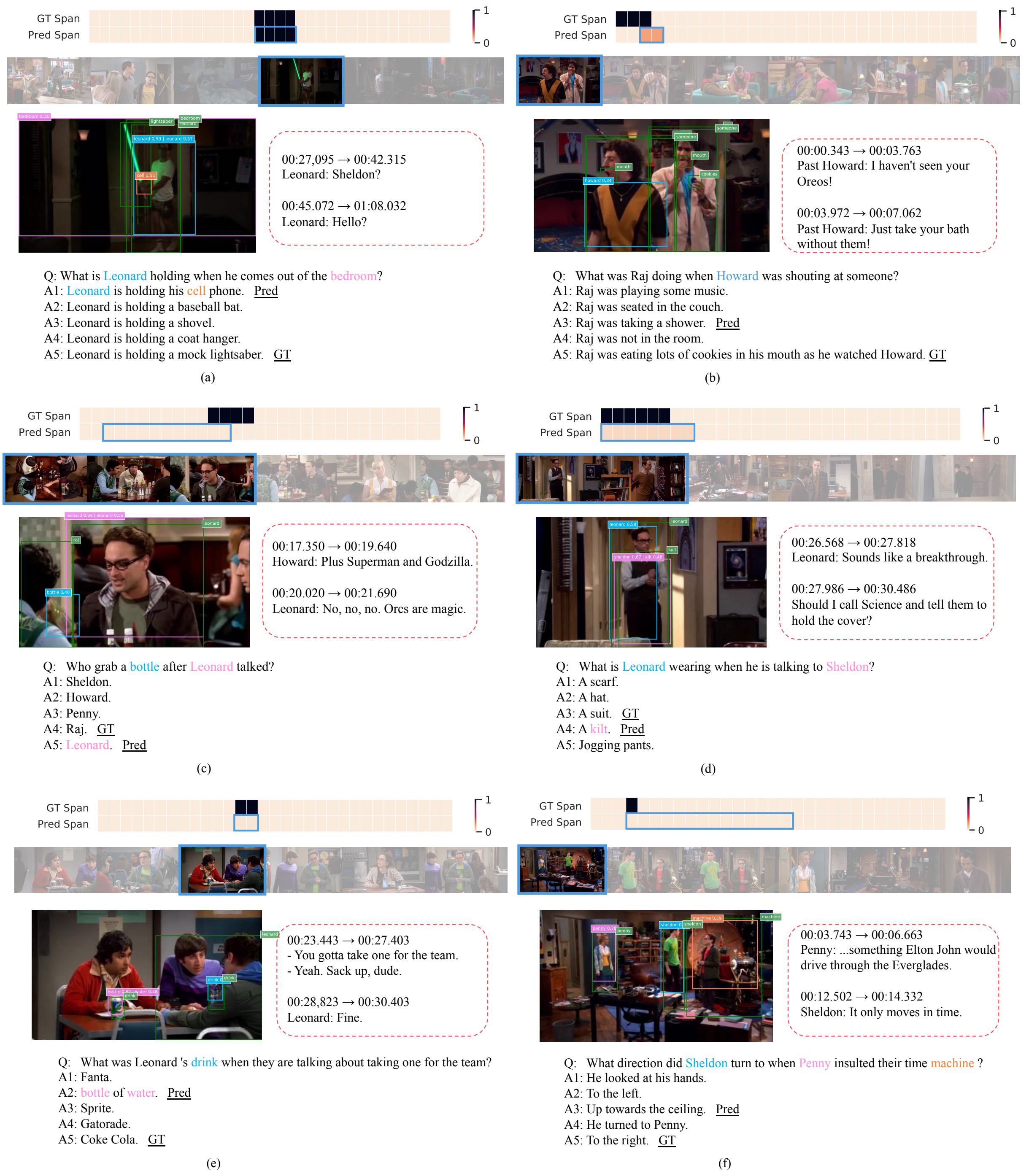
00:50.790 → 00:52.290
Leonard, it's 2 in the morning.

00:53.918 → 00:59.018
- So?
- So it's my turn.

Q: Where was Leonard when Sheldon walked into the living room at 2am?
A1: On the couch.
A2: In the time machine. Pred GT
A3: In the kitchen.
A4: In his room.
A5: He wasn't there.

(f)

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.



00:27,095 → 00:42.315
Leonard: Sheldon?

00:45.072 → 01:08.032
Leonard: Hello?

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)

00:00.343 → 00:03.763
Past Howard: I haven't seen your Oreos!

00:03.972 → 00:07.062
Past Howard: Just take your bath without them!

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)

00:17.350 → 00:19.640
Howard: Plus Superman and Godzilla.

00:20.020 → 00:21.690
Leonard: No, no, no. Orcs are magic.

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

(c)

00:26.568 → 00:27.818
Leonard: Sounds like a breakthrough.

00:27.986 → 00:30.486
Should I call Science and tell them to hold the cover?

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)

00:23.443 → 00:27.403
- You gotta take one for the team.
- Yeah. Sack up, dude.

00:28,823 → 00:30.403
Leonard: Fine.

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)

00:03.743 → 00:06.663
Penny: ...something Elton John would drive through the Everglades.

00:12.502 → 00:14.332
Sheldon: It only moves in time.

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.