Learning Situation Hyper-Graphs for Video Question Answering

Aisha Urooj Khan[1,3], Hilde Kuehne[2,4], Bo Wu[2], Kim Chheu[5],
Walid Bousselham[4], Chuang Gan[2,6], Niels Lobo[1], Mubarak Shah[1]

[1] CRCV, University of Central Florida , [2] MIT-IBM Watson AI Lab, [3] Mayo Clinic, AZ
[4] Goethe University Frankfurt Germany, [5] Western Michigan University, [6] UMass Amherst

Abstract

*Answering questions about complex situations in videos requires not only capturing the presence of actors, objects, and their relations but also the evolution of these relationships over time. A situation hyper-graph is a representation that describes situations as scene sub-graphs for video frames and hyper-edges for connected sub-graphs and has been proposed to capture all such information in a compact structured form. In this work, we propose an architecture for Video Question Answering (VQA) that enables answering questions related to video content by predicting situation hyper-graphs, coined **S**ituation **H**yper-**G**raph based **V**ideo **Q**uestion **A**nswering (SHG-VQA). To this end, we train a situation hyper-graph decoder to implicitly identify graph representations with actions and object/human-object relationships from the input video clip. and to use cross-attention between the predicted situation hyper-graphs and the question embedding to predict the correct answer. The proposed method is trained in an end-to-end manner and optimized by a VQA loss with the cross-entropy function and a Hungarian matching loss for the situation graph prediction. The effectiveness of the proposed architecture is extensively evaluated on two challenging benchmarks: AGQA and STAR. Our results show that learning the underlying situation hyper-graphs helps the system to significantly improve its performance for novel challenges of video question-answering tasks*[1].

1. Introduction

Video question answering in real-world scenarios is a challenging task as it requires focusing on several factors including the perception of the current scene, language understanding, situated reasoning, and future prediction. Visual perception in the reasoning task requires capturing various aspects of visual understanding, e.g., detecting a diverse set of entities, recognizing their interactions, as well as understanding the changing dynamics between these entities over time. Similarly, linguistic understanding has its challenges as some question or answer concepts may not be present in the input text or video.

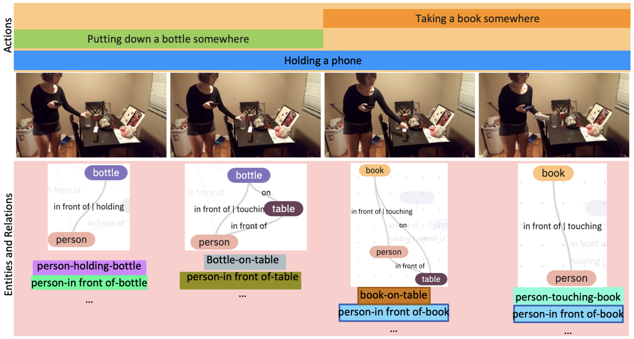

Figure 1. The situation hyper-graph for a video is composed of situations with entities and their relationships (shown as subgraphs in the pink box). These situations may evolve over time. Temporal actions act as hyper-edges connecting these situations into one situation hyper-graph. Learning situation graphs, as well as temporal actions, is vital for reasoning-based video question answering.

Visual question answering, as well as its extension over time, video question answering, have both benefited from representing knowledge in graph structures, e.g., scene graphs [20, 35], spatio-temporal graphs [4, 55], and knowledge graphs [36, 45]. Another approach in this direction is the re-introduction of the concept of "*situation cognition*" embodied in "*situation hyper-graphs*" [47]. This adds the computation of actions to the graphs that capture the interaction between entities. In this case, situations are represented by hyper-graphs that join atomic entities and relations (e.g., agents, objects, and relationships) with their actions (Fig. 1). This is an ambitious task for existing systems as it is impractical to encapsulate all possible interactions in the real-world context.

Recent work [23] shows that transformers are capable

[1] Code will be available at https://github.com/aurooj/SHG-VQA

of learning graphs without adapting graph-specific details in the architectures achieving competitive or even better performance than sophisticated graph-specific models. Our work supports this idea by implicitly learning the underlying hyper-graphs of a video. Thus, it requires no graph computation for inference and uses decoder's output directly for cross attention module. More precisely, we propose to learn situation hyper-graphs, namely framewise actor-object and object-object relations as well as their respective actions, from the input video directly without the need for explicit object detection or other required prior knowledge. While the actions capture events across transitions over multiple frames, such as *Drinking from a bottle*, the relationship encoding actually considers all possible combinations of static, single frame actor-object, and object-object relationships as unique classes, e.g., in the form of *person – hold – bottle* or *bottle – stands on – table*, thus serving as an object and relation classifier. Leveraging this setup allows us to streamline the spatio-temporal graph learning as a set prediction task for predicting relationship predicates and actions in a Situation hyper-graph Decoder block. To train the Situation Graph Decoder, we use a bipartite matching loss between the predicted set and ground truth hyper-graph tokens. The output of the situation graph decoder is a set of action and relationship tokens, which are then combined with the embedding of the associated question to derive the final answer. An overview of the proposed architecture is given in Fig. 2. Note that, compared to other works targeting video scene graph generation, e.g., those listed in [63], we are less focused on learning the best possible scene graph, but rather on learning the representation of the scene which best supports the question answering task. Thus, while capturing the essence of a scene, as well as the transition from one scene to the other, we are not only optimizing the scene graph accuracy but also considering the VQA loss.

We evaluate the proposed method on two challenging video question answering benchmarks: a) STAR [47], featuring four different question types, interaction, sequence, prediction, and feasibility based on a subset of the real-world Charades dataset [44]; and b) Action Genome QA (AGQA) [11] dataset which tests vision focused reasoning skills based on novel compositions, novel reasoning steps, and indirect references. Compared to other VQA datasets, these datasets provide dense ground truth hyper-graph information for each video, which allows us to learn the respective embedding. Our results show that the proposed hyper-graph encoding significantly improves VQA performance as it has the ability to infer correct answers from spatio-temporal graphs from the input video. Our ablations further reveal that achieving high-quality graphs can be critical for VQA performance.

Our contributions to this paper are as follows:

- We introduce a novel architecture that enables the computation of situation hyper-graphs from video data to solve the complex reasoning task of video question-answering;
- We propose a situation hyper-graph decoder module to decode the atomic actions and object/actor-object relationships and model the hyper-graph learning as a transformer-based set prediction task and use a set prediction loss function to predict actions and relationships between entities in the input video;
- We use the resulting high-level embedding information as sole visual information for the reasoning and show that this is sufficient for an effective VQA system.

2. Related Work

Video Question Answering: Video question answering is an active area of research with efforts in insightful directions [37,62], such as attention [16,18,29], cross-modal interactions [25,41,52,56,59,60], hierarchical learning [5,24,38], and so on. A few other algorithms classes include modular networks [5, 24, 49], symbolic reasoning [47, 53, 54], and memory networks [8, 10]. Several video QA benchmarks are introduced to evaluate this task from varying perspectives entailing description [34, 51, 57, 61], temporal reasoning [17, 53, 64], causal structures [48, 53], visual-language comprehension [26, 46, 52], relational reasoning [46], and measuring social intelligence [58]. STAR [47] benchmark goes one step further providing a benchmark to perform diagnostic study at additional fronts such as predicting future interactions and feasibility of next possible actions in the unseen future. Existing approaches on the benchmark either rely on object features [40] as nodes explicitly modeling their interactions through a message passing mechanism [15, 47], or benefit from efficient input sampling strategies during training to learn robust visual representations [24, 25]. We, however, take a different approach and focus on inferring the underlying semantic graph structure in the video and use it for QA reasoning. Our method is independent of using pretrained object detectors and uses a simple approach to learn to infer the sets of relationship predicates as well as actions for each frame. The model is trained end-to-end with the frozen backbone.

Graph-based VQA: Another related line of work is graph-based VQA methods [22, 27]. Some of them work on object features extracted from a pretrained detector (e.g., Faster-RCNN [40]) [5, 15, 31, 42, 49], while some operate on frame-level [19, 25, 56] (e.g., ResNet), clip-level features [38, 41, 52](e.g., C3D, S3D) or transformer-based backbones [7, 32, 39]. We focus on predicting atomic actions, and relationship triplets from the frame-level or clip-level features directly instead of an explicit graph.

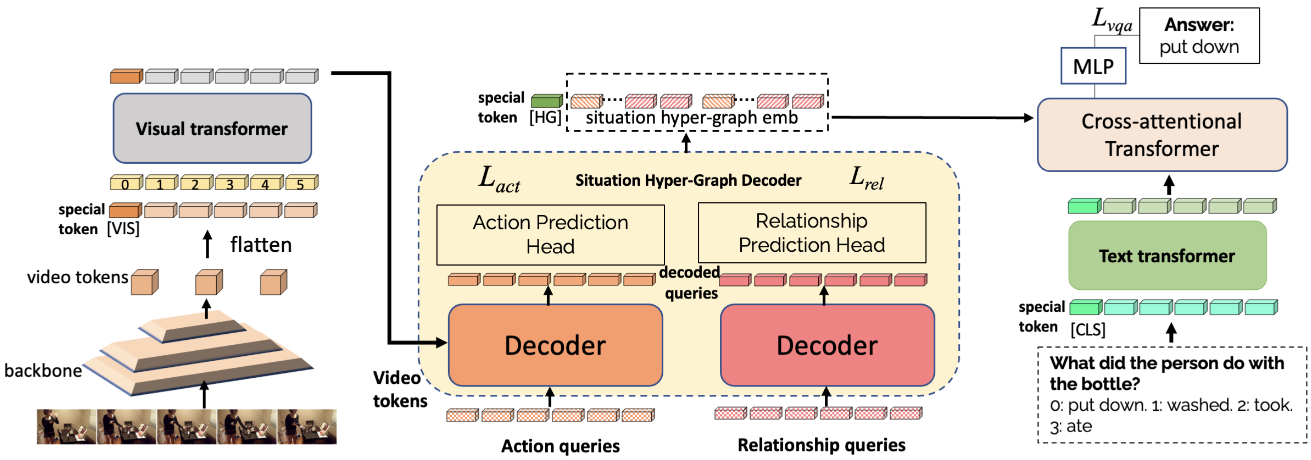

Figure 2. The SHG-VQA architecture: we start with encoding the input video into spatio-temporal features using a pre-trained backbone. These video features are flattened into a sequence of tokens of length $T \times h \times w$ and position encoded to be further processed through a visual encoder. These encoded tokens are input to the action decoder to predict the set of atomic actions from the input action queries as well as to the relationship decoder which takes relationship queries as input along with the video tokens. The action decoder and relationship decoder output the situation graph embeddings. In the text branch, the question and the answer choices are composed into a sequence and passed through a text transformer to obtain encoded word embeddings; for open-ended VQA, only the question is passed to the text transformer. The generated hyper-graph along with the encoded text tokens are then used as input to a cross-attentional transformer and the combined representations are used to predict the correct answer with a classifier. Section 3.4 describes the losses and training objectives.

Scene Graph Generation: The proposed formulation of situation hyper-graph deviates from existing scene-graph generation approaches [2, 47, 63] as we do not require object detections as input or object level supervision, nor do we model it in an explicit graph structure. Our goal is simple: given an input video, predict all object-relation-object and actor-relation-object triplets as well as associated actions for each video frame. Our intuition is that forcing the model to predict these predicates will drive the system to learn a latent graph structure for visual input. The predicted situation hyper-graph is treated as an abstract video representation and used for VQA reasoning in the next step.

3. The SHG-VQA Model

Situation video question answering has three essential steps 1) the visual recognition capacities of visual entities, their relationships, actions, and how these transition over time, 2) the language understanding capacity to the questions, 3) the question-guided reasoning process over the representation learned in the first step. Sub-optimal performance at any of these steps will affect the overall task performance. A key problem here is that capturing the visual structures directly from raw data, e.g., in the form of features often results in a rather noisy signal and does not provide suitable input for high-level language-guided reasoning. To overcome this mismatch, we propose to learn the implicit structure of the visual input as an intermediate step between learning the video representation and question-based reasoning. Forcing the model to learn to predict this implicit structure (actions, relations between entities) not only improves the video representation but also acts as a lightweight, high-level representation of the video content and can be used for the VQA task. We illustrate our architecture in Fig. 2.

3.1. Input Processing

Given the input video and the question, we encode the inputs as described below:

Question Encoder: The question is first tokenized into word tokens using a wordPiece tokenizer. These word tokens along with the special class token $[CLS]$ are the inputs of an embedding layer. The output word embeddings from this layer are input to a transformer encoder that encodes each word using multi-head self-attention between different words at each encoder layer.

Video Encoder: Let $V \in \mathbb{R}^{T \times H \times W \times 3}$ be the input video clip, where T is the clip length with 3 color channels, height H and width W. First, we extract video features $x_V \in \mathbb{R}^{T \times h \times w \times d_x}$ using a convolutional backbone with d_x being the feature dimension, h and w are the reduced feature's height and width. As transformers process sequential input data, the video features x_V are flattened into a sequence (of size $\mathbb{R}^{Thw \times d_x}$) and reduced to dimension d through a linear layer. Then, we append a trainable vector of dimension d for a special class token $[VIS]$ to this sequence of video features at index 0. These features are combined with position encodings and input to a transformer-based video encoder V_e.

The output features of the video are forwarded to the action decoder and relationship decoder to infer the situation graph capturing the action information, as well as the entities and their relationships. The output of the last layer of both decoders is then combined and augmented with frame positions, forming the final hyper-graph embedding. We further attach a randomly initialized class token $[HG]$. These situation graph embeddings are then input to a multi-layered cross-attentional transformer encoder for more fine-grained interaction between the question words and semantic knowledge extracted from the video in the form of a graph. The output features corresponding to $[CLS]$ and $[HG]$ tokens are input to an answer classifier to produce an answer.

3.2. Situation hyper-graph Generation

Real-world video understanding relies on the scene understanding including changing relationships between the objects over time and the evolving actions. Therefore, we represent the given video as a "situation graph" denoted by G which describes actions and relationships between objects. Let $G = (V, E)$, where V is the set of vertices representing all possible entities in the dataset, and E is the set of edges representing all possible relationships between each pair of entities. Given an input video, we want to learn a situation graph $g_t = (v_t, e_t), v_t \in V, e_t \in E$ for each time step $t \in \{1, 2, ..., T\}$ in the video, that captures the entities (objects, actors), their relationships as well as the associated actions that are present in that frame. The hyper-graph for one video is thus represented by the set of situation graphs $\mathcal{G} = \{g_1, ...g_T\}$. An action may comprise multiple relationships and objects. For each frame, we further have a set of actions $A_t = a_1, a_2, ..., a_N$. The set of actions for the full video is then given as $\mathcal{A} = \{A_1, ...A_T\}$. Note that the actions are predicted in addition to the graph structure and that both will be merged in a *situation hyper-graph* embedding in a separate step after the decoder block. Rather than predicting the set of vertices and edges in the graph g_t, we propose to predict the graph structure by formulating this graph prediction as follows:

Let $x_{V_e} \in \mathbb{R}^{T' \times h \times w \times d}$ be the encoded video features, where T' denotes the temporal length of encoded features, h and w are spatial dimensions, and d is the feature's dimension. A relationship predicate p describes interactions between entities by triplet tokens $object - relation - object$ or $actor - relation - object$. We propose to predict the set of relationship predicates *R* and the set of atomic actions *A* occurring in each video frame. This is intuitive because the ability to predict the atomic actions and relations between entities for each time step benefits the high level reasoning tasks such as video question answering in this work. Therefore, for each time step $t \in \{1, 2, ..., T\}$ in the video, we predict the set of relationship predicates denoted by R_t in each video frame where $R_t = \{p_1, p_2, ..., p_M\}$ and p_i is the i^{th} predicate between two entities, representing the vertices v_x and v_y (object-object or actor-object) and the relation resp. edge e_i represented as $< v_x, e_i, v_y >$, $M = |R_t|$ is the relationship set size. Additionally, we predict the set of N actions A_t for each time step t where $A_t = \{a_1, a_2, ..., a_N\}$ and a_j is the j^{th} action occurring at time step t. $N = |A_t|$ is the actions set size for each step t. M and N are hyperparameters in our system.

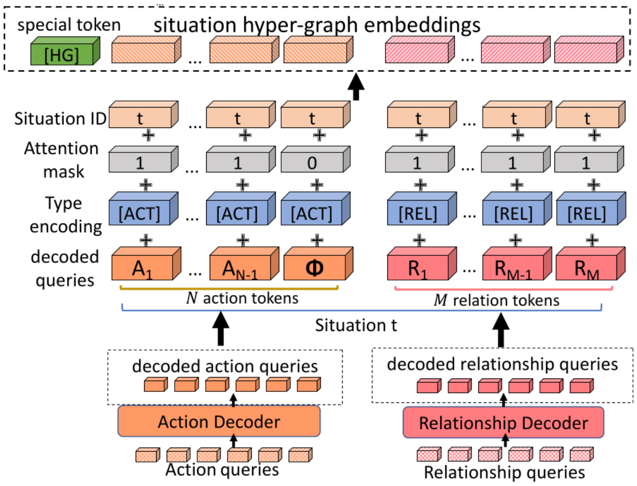

Figure 3. Situation hyper-graph embeddings: We start with decoded queries from action and relationship decoder. Then we add type encoding vectors $[ACT]$ and $[REL]$ for actions and relationships, attention masks, and an embedding vector for the situation ID ($t \in \{1, ..., T\}$). The sums are input to the cross-attentional module. See section 3.2.2 for details).

To obtain actions and relation predicates from the video, we use a transformer decoder that takes video features as memory to learn action queries and relation queries.

3.2.1 Prediction Head

The decoded output embeddings for action and relationship queries are input to respective prediction heads. Prediction heads use a 2-layer feed forward network (FFN) with GELU activation and LayerNorm.

Considering that actions and relationships at a given time step are permutation-invariant, we use optimal bipartite matching between predicted and ground truth actions/relationships. An optimal bipartite matching between the predicted classes and ground-truth labels is the one with the minimum matching cost. Once the optimal matching pairs have been obtained, we use a Hungarian loss function to optimize for ground truth classes [1]. See section 3.4 for details of the loss function.

3.2.2 Situation hyper-graph Embedding

For the decoded queries of actions and relationships (referred as graph token embeddings), *situation hyper-graph*

Table 1. Results on AGQA dataset for different question types. The best results are shown in **bold** font. Numbers are reported in percentages.

	Reasoning								Semantic			Structure					Overall		
Method	obj-rel	rel-action	obj-action	superlative	sequencing	exists	duration	activity	obj	rel	action	query	compare	choose	logic	verify	binary	open	all
PSAC [30]	37.84	49.95	50.00	33.20	49.78	49.94	45.21	4.14	37.97	49.95	46.85	31.63	49.49	46.56	49.96	49.90	48.87	31.63	40.18
HME [9]	37.42	49.90	49.97	33.21	49.77	49.96	47.03	5.43	37.55	49.99	47.58	31.01	49.71	46.42	49.87	49.96	48.91	31.01	39.89
HCRN [24]	40.33	49.86	49.85	33.55	49.70	50.01	43.84	5.52	40.33	49.96	46.41	36.34	49.22	43.42	50.02	50.01	47.97	36.34	42.11
SHG-VQA	**46.42**	**60.67**	**64.63**	**38.83**	**62.17**	**56.06**	**48.15**	**10.12**	**47.61**	**56.19**	**53.83**	**43.42**	**60.68**	**47.76**	**52.86**	**56.63**	**55.04**	**43.42**	**49.20**

embeddings are constructed in order to be used with question features for video question answering. First, these action and relationship graph embeddings are combined for each time step t representing a situation at t. Then, we add token type embedding $[ACT]$ for actions and $[REL]$ for relations to their respective graph embeddings. A situation ID (or frame position t) embedding is also added to these embeddings. An additional attention mask is used to differentiate actual tokens and padded tokens (no-class token ϕ) at training time. At inference time, no attention mask is used as we do not employ any information about the graph at test time. Finally, we add a special class token $[HG]$ to this sequence of features. See Fig. 3 for visualization.

3.3. Cross-attentional Transformer Module

The situation hyper-graph embeddings obtained at previous step (section 3.2.2) are input along with the question to a cross-attentional transformer module which allows fine-grained computation between the question features and the graph features. A standard co-attentional transformer module is used for cross-attention between the two sequential feature inputs. The feature outputs corresponding to the $[HG]$ token and $[CLS]$ token from the cross-attentional transformer block are fed to a feed-forward network (FFN) for answer prediction.

3.4. Learning Objective

The SHG-VQA model is trained with the following training objective i.e.,

$$L = L_{act} + L_{rel} + L_{vqa} \tag{1}$$

where L_{act} and L_{rel} are the set prediction loss terms for predicting the action set and relationship set for the video, and L_{vqa} is the cross-entropy loss over the predicted situation graph and question.

Actions and relationships set prediction loss: The situation graph prediction module infers fixed sets of sizes $|N| \times T$ actions and $|M| \times T$ relationships for T-length video clip in a single pass through the action decoder and relationship decoder respectively. We modify the set prediction loss used in [1] as follows. Let A be the set of ground-truth actions and $\hat{A} = \{\hat{a}_i\}_{i=1}^{|N|\times T}$ be the predicted set of actions. In a scenario where $\hat{A}$ is larger than the set of actions present in the video, a special class ϕ (no class) is padded to the ground truth set A. We obtain a bipartite matching between ground-truth and predicted set for each timestep t as follows:

$$\hat{\sigma_a} = \sum_{t}^{T} argmin_{\sigma_t \in \zeta_{|N|}} \sum_{i}^{|N|} \mathcal{L}_{match}(a_{t_i}, \hat{a}_{\sigma_t(i)}) \tag{2}$$

Where, σ_t is a permutation of N elements for frame t, $\mathcal{L}_{match}(a_{ti}, \hat{a}_{\sigma_t(i)})$ is a pair-wise matching cost between i^{th} ground-truth action label in t^{th} frame i.e., a_{t_i} and a predicted action label at index $\sigma_t(i)$. This optimal assignment for each step t is computed using the Hungarian algorithm and the cost is summed over all T steps. The proposed set prediction loss takes into account only the class predictions for all video frames with no bounding box ground truths being used, different from the original set prediction loss used in [1] for object detection in images. Let $\hat{p}(c_{t(i)})$ be the class probability for the action prediction at $\sigma_t(i)$, $\mathcal{L}_{match}(a_{t_i}, \hat{a}_{\sigma_t(i)})$ would be $-\mathbb{1}_{\{c_{t(i)} \neq \phi\}} \hat{p}_{\sigma_t(i)}(c_{t(i)})$. After we obtain a one-to-one optimal matching between ground-truth and predicted set items without duplicates at each time step, we can compute the loss between the matched pairs using a Hungarian loss as follows:

$$\mathcal{L}_{act}(a, \hat{a}) = \sum_{i=1}^{|N|\times T} -\log \hat{p}_{\hat{\sigma}(i)}(c_i) \tag{3}$$

Likewise, R is the set of ground-truth relations and $\hat{R} = \{\hat{p_i}\}_{i=1}^{|M|\times T}$ denotes the predicted relationships. L_{rel} is formulated as follows:

$$\hat{\sigma_p} = \sum_{t}^{T} argmin_{\sigma_t \in \zeta_{|M|}} \sum_{i}^{|M|} \mathcal{L}_{match}(p_{t_i}, \hat{p}_{\sigma_t(i)}) \tag{4}$$

$$\mathcal{L}_{rel}(p, \hat{p}) = \sum_{i=1}^{|M|\times T} -\log \hat{p}_{\hat{\sigma}(i)}(c_i) \tag{5}$$

where $\hat{\sigma}(i)$ is the optimal matching obtained in the previous step for each frame. The inferred graph is input to the cross-attentional transformer module along with the question and the answer choices as explained in section 3.2.2. The proposed network is trained in an end-to-end manner.

Table 2. Results on AGQA's novel compositions test metric.

	Sequencing			Superlative			Duration			Obj-relation			Overall		
Method	B	O	All	B	O	All	B	O	All	B	O	All	B	O	All
PSAC [30]	49.19	29.33	40.96	45.23	17.76	33.32	47.89	34.84	42.06	**43.76**	0.01	24.28	46.49	19.34	34.71
HME [9]	49.33	28.06	40.53	44.06	13.8	30.95	48.45	34.72	42.31	39.58	0.00	21.96	45.42	17.17	33.15
HCRN [24]	48.31	30.00	40.73	45.12	17.30	33.06	46.15	39.11	43.01	37.15	2.86	21.88	44.88	20.12	34.13
SHG-VQA	**50.88**	**38.59**	**45.79**	**51.14**	**23.64**	**39.25**	**51.84**	**49.21**	**50.66**	39.73	**6.23**	**24.82**	**49.07**	**26.68**	**39.37**

Table 3. Evaluation on AGQA's more compositional steps.

More Compositional Steps	Binary	Open	All
PSAC [30]	47.65	14.81	47.19
HME [9]	**48.09**	20.98	**47.72**
HCRN [24]	46.96	**23.70**	46.63
SHG-VQA	47.13	22.66	46.97

Table 4. Evaluation on AGQA's indirect references test metric.

Method		Object			Action			Temporal		
		B	O	All	B	O	All	B	O	All
Precision	PSAC	63.69	53.77	56.64	61.01	52.46	53.24	57.52	53.74	54.39
	HME	62.95	52.31	55.39	58.21	48.12	49.04	55.99	52.42	53.04
	HCRN	54.06	67.24	63.43	53.87	64.43	63.47	52.35	66.84	64.34
	SHG-VQA	**76.93**	**86.27**	**81.59**	**79.55**	**87.40**	**84.90**	**69.08**	**87.35**	**82.87**
Recall	PSAC	45.06	27.36	38.80	40.91	22.18	25.96	35.13	26.84	30.64
	HME	46.03	26.80	39.23	41.32	21.71	25.67	37.96	26.59	31.80
	HCRN	44.84	35.46	41.52	44.01	30.43	33.17	35.11	34.38	34.71
	SHG-VQA	**53.86**	**43.98**	**49.85**	**54.16**	**37.83**	**43.10**	**52.21**	**43.57**	**46.78**

4. Experiments

4.1. Datasets

AGQA Benchmark [11]: The Action Genome Question Answering benchmark is a visual dataset comprising 192M hand-crafted questions about 9.6K videos from the Charades dataset [43]. In addition to VQA accuracy, AGQA presents three testing metrics for testing the VQA methods: indirect references, novel compositions, and more compositional steps. We use the AGQA 2.0 Balanced dataset, which consists of 2.27M question-answer pairs as a result of balancing the original dataset using stricter procedures to reduce as much language bias as possible. Of the 2.27M questions, there are approximately 1.6M training questions and 669K test questions [12]. To have a standard train-val-test setup for our experiments, we randomly sampled 10% QA pairs from training data for validation of hyperparameters.

STAR Benchmark [47]: The STAR dataset provides 60K situated reasoning questions based on 22K trimmed situation video clips, also based on the Charades dataset [43]. They further provide ∼144K ground-truth situation graphs including 111 actions, 37 unique objects, and 24 relationships. The dataset is split into training, validation, and test sets where test evaluation can be done on the evaluation server a limited number of times. We perform ablations and analysis on the validation set and report test set results to compare with the baselines in Table 5.

4.2. Implementations

On the VQA task, we report accuracy; we also report mAP for situation hyper-graph predictions. For AGQA's *indirect references* testing metric, precision and recall are reported. For STAR benchmark, we follow the same training protocol as [47] and train SHG-VQA from scratch on each question type separately unless specified otherwise. Training details are shared in the supplementary document.

Visual Embeddings: The video frames are resized to size 224×224 with a clip length of 16 frames. We use RandAugment [3] for data augmentation during training. The video clip is input to a pretrained convolutional network with freeze weights to obtain video features of size $16 \times 7 \times 7 \times 2048$. A 2-layer 3D convolutional block with the kernel of size $5\times$ 3×3 further processes the zero-padded extracted features yielding features of size $8 \times 7 \times 7 \times d$. The spatio-temporal dimensions are then flattened to obtain a sequence of length $Thw = 392$ $d-$dimensional tokens where $d = 768$. The input encoders as well as situation hyper-graph decoders use $L = 5$ transformer encoder layers with non-shared weights.

Query embeddings for action and relationship predicates: The features output from the video encoder is then input to a situation hyper-graph decoder comprising transformer-based action and relationship decoders. Our best model uses $M = 8$ relation queries and $N = 3$ action queries for each situation; T is set to 16. **AGQA** has 157 total raw actions, 36 unique objects, and 44 unique relationships which obtain 456 relationship triplets $< v_x, e_i, v_y >$ in the training and validation set. For **STAR** dataset, there are 37 objects, and 24 relationships yielding 563 unique relation predicates; it also has 111 action classes.

Situation graph embeddings: The decoded situation graph queries are input to a cross-attentional transformer. The input situation graph embedding is a sum of 4 different encoding types: 1) decoded query embeddings for predicted actions and relationship predicates, 2) situation IDs denoting the situation (or frame) number for each query, 3) attention mask set to 1 for actual tokens and 0 for padded tokens, and 4) token type embeddings to distinguish between action tokens and relationship tokens as shown in Fig. 3. Our network uses a 2-layer co-attentional module [33] for cross-attention.

Table 5. Results on STAR dataset. **Best results** are shown in bold font and second best results are underlined. Numbers are reported for VQA accuracy in percentages.

Method(test)	Backbone	Obj.	Hyper.	Question Type				
				Inter.	Seq.	Pred.	Feas.	Overall
Q-type (Random) [21]	-	✗	✗	25.06	24.93	24.79	24.81	24.89
Q-type (Frequent) [21]	-	✗	✗	19.09	19.45	12.90	18.31	17.44
Blind Model (LSTM) [14]	GloVe	✗	✗	32.24	32.17	28.56	28.41	30.34
Blind Model (BERT) [6]	BERT	✗	✗	32.68	34.21	29.98	29.26	31.53
CNN-LSTM [53]	ResNext101-K400	✗	✗	33.25	32.67	30.69	30.43	31.76
CNN-BERT [28]	ResNext101-K400	✗	✗	33.59	37.16	30.95	30.84	33.14
LCGN [15]	ResNext101-K400	✓	✗	39.01	37.97	28.81	26.98	33.19
HRCN [24]	ResNext101-K400	✓	✗	39.10	38.17	28.75	27.27	33.32
ClipBERT [25]	ResNext101-K400	✗	✗	39.81	**43.59**	32.24	31.42	36.70
NS-SR [47]	ResNext101-K400	✓	✓	30.88	31.76	30.23	29.73	30.65
SHG-VQA (Ours)	SlowR50-K400	✗	✓	**47.98**	42.03	**35.34**	**32.52**	**39.47**
SHG-VQA (Ours)	ResNext101-ImageNet1K	✗	✓	45.8	42.77	34.64	29.91	38.28

AGQA: For AGQA, we train our model with SlowR50 backbone and report results for VQA accuracy on the test set. We also report our model's generalization capability to indirect references. Furthermore, we train our network to report its generalization to novel compositions and to more compositional steps. For more compositional steps, we train our network with randomly sampled 100K QA pairs.

STAR: Following [47], we train the model for each question type separately to compare with the baselines. However, this protocol is expensive in terms of time and resources. To address the matter of limited resources, we also tried combining all questions together after the questions filtering from interaction and sequence types and train a single model instead of multiple trainings. We use this training regime to study models ablations.

5. Results and Analysis

5.1. Comparison to State-of-the-Art

AGQA: We compare our method with the existing state-of-the-art methods on AGQA benchmark. The best baseline on AGQA is HCRN [24] for overall accuracy. HCRN uses appearance features from ResNet101 as well as motion features from ResNext101-Kinetics400 backbones. Our model outperforms HCRN by a significant margin of 7.09% (HCRN: 42.11% vs. SHG-VQA: 49.20%) in terms of overall accuracy (see Table 1). We observe the biggest improvement of 14.63% absolute points on object-action reasoning questions compared to the best model in that category i.e., PSAC [30]: 50.00% vs. SHG-VQA: 64.63%. We further report results on the three novel testing metrics as follows:

a) Novel Compositions: For novel composition at test time, we observe an overall gain of 4.70% when compared to the best contender, i.e., PSAC [30]. For open-ended questions, we outperform HCRN by 6.56% (see Table 2);

b) Indirect References: When tested for indirect references questions in Table 4, we outperform all baselines by an absolute 7.83%-16.62% in terms of recall; For precision, we also gain similar improvements;

c) Compositional Steps: Our model is trained on only 15% (100K QA pairs) of the training data and is still able to perform on par with the baselines (trained on 1.6M QA pairs) achieving second best results for each category of more compositional steps (table 3).

STAR: For STAR dataset, we first compare the proposed architecture to other state-of-the-art works in the field using SlowR50 [13] video backbone and a ResNext101 [50] as a frame-level backbone to evaluate accuracy based on 3D video and 2D image architectures. It shows that for both settings, SHG-VQA significantly outperforms other baseline methods even with weaker backbones (see Table 5). Concretely, we obtain an absolute gain of 8.53% over NS-SR [47], 5.86% compared to HCRN [24], 5.99% improvement over LCGN [15], and $\sim$ 2.5% over ClipBERT [25] which is a SOTA model in terms of overall VQA accuracy. We notice the substantial gain of 7.86% for interaction questions which test the understanding of interactions between entities in a situation. Prediction is the next category of questions that benefits the most with 2.48% improvement.

5.2. Ablation and Hyperparameter Analysis

We perform our ablation studies on the STAR benchmark as discussed below:

Impact of situation graphs quality: To assess the effect of situation graphs' quality on VQA accuracy, we train a baseline version of our system, where the model is trained on ground truth situation graphs for VQA tasks only (Table 6). Since the ground truth situation graphs are not available for the test set, we can only compare SHG-VQA and this baseline on the validation set. As expected, when taking ground truth situation graphs as input, the performance is significantly improved for interaction (GT=91.9% vs. predicted=46.78%) and sequence (GT=80.5% vs. predicted=42.52%) questions. However, for the questions about the unseen part of the video, the model with ground truth graph tokens still struggles despite better performance compared to SHG-VQA: predic-

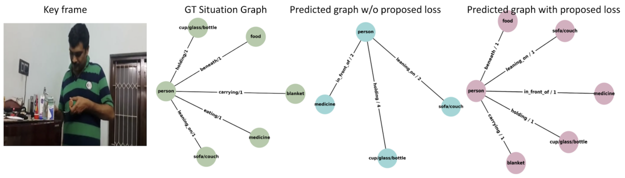

Figure 4. Qualitative example for using frame-wise set prediction loss. Col. 1 shows the frame, col. 2 shows the ground-truth situation graph, col. 3 shows predicted graphs when trained with set prediction loss for the full video, and col. 4 shows the predicted graph when the model is trained by matching each timestep t. The edges show the person-object relationship labels along with the number of times it was predicted.

Table 6. **Impact of hyper-graphs (HG) quality.** Results shown for STAR val set with SlowR50 backbone.

Method	Interaction	Sequence	Prediction	Feasibility	Overall
Predicted hyper-graphs	47.08	42.52	37.82	33.61	40.26
GT hyper-graphs	**91.9**	**80.5**	**41.22**	**35.42**	**62.46**

tion (GT=41.22% vs. predicted=37.82%) and feasibility (GT=35.42% vs. predicted=33.61%). We also report the mAP scores for the prediction of action and relationship predicates using our best model in Table 7. We obtain an overall mAP of 87.63 for actions and 72.9 for relationships respectively.

Input to cross-attentional transformer: To evaluate the choice of input to the cross-attentional transformer, we experiment with three settings: a) question and video embeddings, b) question and situation graphs embeddings, c) question, situation graphs, and video embeddings. We observe no gain in the overall VQA accuracy when adding video embeddings to the cross-attentional transformer and get our best results with (b) (table 8).

Situation hyper-graph components: To study the impact of different components of situation graphs, we train our system with only action predicate tokens with objective $L_{act} + L_{vqa}$, only relationship predicates (with $L_{rel} + L_{vqa}$), and the full model (eq. 1). The action predicates are more effective compared to the relationship predicates. When compared in terms of predicate classification, we observe high accuracy for action predicates. Nonetheless, using the full situation graphs perform better than omitting actions or relationship prediction task (table 9).

Number of queries: Number of action queries M and relationship queries N is a hyperparameter for SHG-VQA. We report the performance of the SHG-VQA with a varying number of queries for actions and relationships at each timestep in Table 9. We report our best results with $M = 3$ and $N = 8$. Additional results are reported in the supplementary document.

Frame-wise set prediction loss: Using the notion of time t in the set prediction loss alleviates the problem of duplicate predictions within each situation. See Fig. 4 for qualitative examples of produced hyper-graphs with and without this loss. Extra examples are in the supplementary document.

Table 7. **Predicate classification results** for situation hyper-graphs in terms of mAP for STAR validation set from SHG-VQA with SlowR50 backbone. Numbers are reported in percentages.

	Interact	Sequence	Prediction	Feasibility	Overall
Actions	84.77	89.43	85.30	91.46	87.63
Relationships	72.83	73.5	70.05	72.82	72.9

Table 8. **Results for cross-attention input.** Results shown for STAR test set with SlowR50 backbone.

Method	Interaction	Sequence	Prediction	Feasibility	Overall
Q + V	33.28	35.60	27.93	26.43	30.81
Q + HG	**47.98**	42.03	**35.34**	32.52	**39.47**
Q + V + HG	45.45	**44.19**	34.22	**32.87**	39.18

Table 9. Model variations on STAR validation set with a single model using SlowR50 backbone for all question types. **Best results** are shown in bold font and second best results are underlined. Numbers are reported for VQA accuracy in percentages.

Method (val)	Interaction	Sequence	Prediction	Feasibility	Overall
hyper-graph components					
Action only – Act=3	40.94	38.08	35.58	30.56	38.68
Relation. only – Rel=8	35.94	35.79	34.78	27.86	35.16
Both – Act=3, Rel=8	**42.93**	**38.20**	**36.06**	**30.56**	**39.20**
Number of queries					
Act=2, Rel=8	40.32	40.13	38.78	29.73	38.34
Act=3, Rel=8	**42.93**	38.20	36.06	30.56	39.20
Act=4, Rel=8	38.40	35.79	35.58	30.35	37.06
Act=3, Rel=12	41.32	**40.80**	**39.42**	**31.39**	**39.90**
Act=4, Rel=12	40.40	39.05	36.86	29.11	38.39

6. Conclusion

We presented a novel approach to model situation graph prediction as an underlying sub-task for video question answering. The proposed method predicts a situation hyper-graph structure composed of existing actions and relationships in the input video. The input question can then reason over the predicted graph to solve VQA. We show the impact of the proposed approach by evaluating on two video question-answering benchmarks and achieving significant performance gains overall baseline methods. Our method demonstrates promise for further research in this direction to improve VQA systems even further.

Acknowledgements. Aisha Urooj is supported by the ARO grant W911NF-19-1-0356. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. This work was partially supported under NSF grant CNS-2050731, Research Experience for Undergraduates in Computer Vision. Dr. Gan is supported by the MIT-IBM Watson AI Lab, DSO grant DSOCO21072, and gift funding from MERL, Cisco, Sony, and Amazon.

Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of ARO, IARPA, DOI/IBC, or the U.S. Government.

References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In European conference on computer vision, pages 213–229. Springer, 2020. 4, 5

[2] Xiaojun Chang, Pengzhen Ren, Pengfei Xu, Zhihui Li, Xiaojiang Chen, and Alexander G Hauptmann. A comprehensive survey of scene graphs: Generation and application. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021. 3

[3] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pages 702–703, 2020. 6

[4] Vinay Damodaran, Sharanya Chakravarthy, Akshay Kumar, Anjana Umapathy, Teruko Mitamura, Yuta Nakashima, Noa Garcia, and Chenhui Chu. Understanding the role of scene graphs in visual question answering. arXiv preprint arXiv:2101.05479, 2021. 1

[5] Long Hoang Dang, Thao Minh Le, Vuong Le, and Truyen Tran. Hierarchical object-oriented spatio-temporal reasoning for video question answering. arXiv preprint arXiv:2106.13432, 2021. 2

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018. 7

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929, 2020. 2

[8] Chenyou Fan, Xiaofan Zhang, Shu Zhang, Wensheng Wang, Chi Zhang, and Heng Huang. Heterogeneous memory enhanced multimodal attention model for video question answering. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 1999–2007, 2019. 2

[9] Chenyou Fan, Xiaofan Zhang, Shu Zhang, Wensheng Wang, Chi Zhang, and Heng Huang. Heterogeneous memory enhanced multimodal attention model for video question answering. CoRR, abs/1904.04357, 2019. 5, 6

[10] Jiyang Gao, Runzhou Ge, Kan Chen, and Ram Nevatia. Motion-appearance co-memory networks for video question answering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 6576–6585, 2018. 2

[11] Madeleine Grunde-McLaughlin, Ranjay Krishna, and Maneesh Agrawala. Agqa: A benchmark for compositional spatio-temporal reasoning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 11287–11297, 2021. 2, 6

[12] Madeleine Grunde-McLaughlin, Ranjay Krishna, and Maneesh Agrawala. Agqa 2.0: An updated benchmark for compositional spatio-temporal reasoning, 2022. 6

[13] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Learning spatio-temporal features with 3d residual networks for action recognition. In Proceedings of the IEEE International Conference on Computer Vision Workshops, pages 3154–3160, 2017. 7

[14] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. Neural computation, 9(8):1735–1780, 1997. 7

[15] Ronghang Hu, Anna Rohrbach, Trevor Darrell, and Kate Saenko. Language-conditioned graph networks for relational reasoning. In Proceedings of the IEEE International Conference on Computer Vision, pages 10294–10303, 2019. 2, 7

[16] Yunseok Jang, Yale Song, Chris Dongjoo Kim, Youngjae Yu, Youngjin Kim, and Gunhee Kim. Video question answering with spatio-temporal reasoning. International Journal of Computer Vision, 127(10):1385–1412, 2019. 2

[17] Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2758–2766, 2017. 2

[18] Jianwen Jiang, Ziqiang Chen, Haojie Lin, Xibin Zhao, and Yue Gao. Divide and conquer: Question-guided spatio-temporal contextual attention for video question answering. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, pages 11101–11108, 2020. 2

[19] Pin Jiang and Yahong Han. Reasoning with heterogeneous graph alignment for video question answering. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, pages 11109–11116, 2020. 2

[20] Juanzi Li Jiaxin Shi, Hanwang Zhang. Explainable and explicit visual reasoning over scene graphs. In CVPR, 2019. 1

[21] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In CVPR, 2017. 7

[22] Yash Kant, Dhruv Batra, Peter Anderson, Alexander Schwing, Devi Parikh, Jiasen Lu, and Harsh Agrawal. Spatially aware multimodal transformers for textvqa. In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX 16, pages 715–732. Springer, 2020. 2

[23] Jinwoo Kim, Tien Dat Nguyen, Seonwoo Min, Sungjun Cho, Moontae Lee, Honglak Lee, and Seunghoon Hong. Pure transformers are powerful graph learners. arXiv preprint arXiv:2207.02505, 2022. 1

[24] Thao Minh Le, Vuong Le, Svetha Venkatesh, and Truyen Tran. Hierarchical conditional relation networks for video question answering. CoRR, abs/2002.10698, 2020. 2, 5, 6, 7

[25] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L Berg, Mohit Bansal, and Jingjing Liu. Less is more: Clipbert for video-and-language learning via sparse sampling. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 7331–7341, 2021. 2, 7

[26] Jie Lei, Licheng Yu, Mohit Bansal, and Tamara L Berg. Tvqa: Localized, compositional video question answering. In EMNLP, 2018. 2

[27] Linjie Li, Zhe Gan, Yu Cheng, and Jingjing Liu. Relation-aware graph attention network for visual question answering. In Proceedings of the IEEE/CVF international conference on computer vision, pages 10313–10322, 2019. 2

[28] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language. arXiv preprint arXiv:1908.03557, 2019. 7

[29] Xiangpeng Li, Jingkuan Song, Lianli Gao, Xianglong Liu, Wenbing Huang, Xiangnan He, and Chuang Gan. Beyond rnns: Positional self-attention with co-attention for video question answering. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, pages 8658–8665, 2019. 2

[30] Xiangpeng Li, Jingkuan Song, Lianli Gao, Xianglong Liu, Wenbing Huang, Xiangnan He, and Chuang Gan. Beyond rnns: Positional self-attention with co-attention for video question answering. In AAAI, 2019. 5, 6, 7

[31] Fei Liu, Jing Liu, Weining Wang, and Hanqing Lu. Hair: Hierarchical visual-semantic relational reasoning for video question answering. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 1698–1707, 2021. 2

[32] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. arXiv preprint arXiv:2106.13230, 2021. 2

[33] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. arXiv preprint arXiv:1908.02265, 2019. 6

[34] Tegan Maharaj, Nicolas Ballas, Anna Rohrbach, Aaron Courville, and Christopher Pal. A dataset and exploration of models for understanding video data through fill-in-the-blank question-answering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 6884–6893, 2017. 2

[35] Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B Tenenbaum, and Jiajun Wu. The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. In International Conference on Learning Representations, 2018. 1

[36] Kenneth Marino, Xinlei Chen, Devi Parikh, Abhinav Gupta, and Marcus Rohrbach. Krisp: Integrating implicit and symbolic knowledge for open-domain knowledge-based vqa. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 14111–14121, 2021. 1

[37] Devshree Patel, Ratnam Parikh, and Yesha Shastri. Recent advances in video question answering: A review of datasets and methods. In International Conference on Pattern Recognition, pages 339–356. Springer, 2021. 2

[38] Liang Peng, Shuangji Yang, Yi Bin, and Guoqing Wang. Progressive graph attention network for video question answering. In Proceedings of the 29th ACM International Conference on Multimedia, pages 2871–2879, 2021. 2

[39] Tianwen Qian, Jingjing Chen, Shaoxiang Chen, Bo Wu, and Yu-Gang Jiang. Scene graph refinement network for visual question answering. IEEE Transactions on Multimedia, 2022. 2

[40] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, 28, 2015. 2

[41] Ahjeong Seo, Gi-Cheon Kang, Joonhan Park, and Byoung-Tak Zhang. Attend what you need: Motion-appearance synergistic networks for video question answering. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 6167–6177, Online, Aug. 2021. Association for Computational Linguistics. 2

[42] Paul Hongsuck Seo, Arsha Nagrani, and Cordelia Schmid. Look before you speak: Visually contextualized utterances. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 16877–16887, 2021. 2

[43] Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In European Conference on Computer Vision, pages 510–526. Springer, 2016. 6

[44] Gunnar A. Sigurdsson, Gül Varol, X. Wang, Ali Farhadi, Ivan Laptev, and Abhinav Kumar Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. ECCV, 2016. 2

[45] Ajeet Kumar Singh, Anand Mishra, Shashank Shekhar, and Anirban Chakraborty. From strings to things: Knowledge-enabled vqa model that can read and reason. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 4602–4612, 2019. 1

[46] Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. Movieqa: Understanding stories in movies through question-answering. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 4631–4640, 2016. 2

[47] Bo Wu, Shoubin Yu, Zhenfang Chen, Joshua B Tenenbaum, and Chuang Gan. Star: A benchmark for situated reasoning in real-world videos. In Thirty-fifth Conference on Neural Information Processing Systems, 2021. 1, 2, 3, 6, 7

[48] Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. Next-qa: Next phase of question-answering to explaining temporal actions. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 9777–9786, 2021. 2

[49] Junbin Xiao, Angela Yao, Zhiyuan Liu, Yicong Li, Wei Ji, and Tat-Seng Chua. Video as conditional graph hierarchy for multi-granular question answering. arXiv preprint arXiv:2112.06197, 2021. 2

[50] Saining Xie, Ross Girshick, Piotr Dollar, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017. 7

[51] Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueting Zhuang. Video question answering via gradually refined attention over appearance and motion. In Proceedings of the 25th ACM international conference on Multimedia, pages 1645–1653, 2017. 2

[52] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Just ask: Learning to answer questions from millions of narrated videos. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 1686–1697, 2021. 2

[53] Kexin Yi, Chuang Gan, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B Tenenbaum. Clevrer: Collision events for video representation and reasoning. ICLR, 2020. 2, 7

[54] Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Joshua B Tenenbaum. Neural-Symbolic VQA: Disentangling Reasoning from Vision and Language Understanding. In Advances in Neural Information Processing Systems (NIPS), 2018. 2

[55] Fei Yu, Jiji Tang, Weichong Yin, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. Ernie-vil: Knowledge enhanced vision-language representations through scene graph. arXiv preprint arXiv:2006.16934, 2020. 1

[56] Weijiang Yu, Haoteng Zheng, Mengfei Li, Lei Ji, Lijun Wu, Nong Xiao, and Nan Duan. Learning from inside: Self-driven siamese sampling and reasoning for video question answering. Advances in Neural Information Processing Systems, 34, 2021. 2

[57] Zhou Yu, Dejing Xu, Jun Yu, Ting Yu, Zhou Zhao, Yueting Zhuang, and Dacheng Tao. Activitynet-qa: A dataset for understanding complex web videos via question answering. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, pages 9127–9134, 2019. 2

[58] Amir Zadeh, Michael Chan, Paul Pu Liang, Edmund Tong, and Louis-Philippe Morency. Social-iq: A question answering benchmark for artificial social intelligence. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 8807–8817, 2019. 2

[59] Rowan Zellers, Jiasen Lu, Ximing Lu, Youngjae Yu, Yanpeng Zhao, Mohammadreza Salehi, Aditya Kusupati, Jack Hessel, Ali Farhadi, and Yejin Choi. Merlot reserve: Neural script knowledge through vision and language and sound. arXiv preprint arXiv:2201.02639, 2022. 2

[60] Rowan Zellers, Ximing Lu, Jack Hessel, Youngjae Yu, Jae Sung Park, Jize Cao, Ali Farhadi, and Yejin Choi. Merlot: Multimodal neural script knowledge models. Advances in Neural Information Processing Systems, 34, 2021. 2

[61] Kuo-Hao Zeng, Tseng-Hung Chen, Ching-Yao Chuang, Yuan-Hong Liao, Juan Carlos Niebles, and Min Sun. Leveraging video descriptions to learn video question answering. CoRR, abs/1611.04021, 2016. 2

[62] Yaoyao Zhong, Wei Ji, Junbin Xiao, Yicong Li, Weihong Deng, and Tat-Seng Chua. Video question answering: Datasets, algorithms and challenges, 2022. 2

[63] Guangming Zhu, Liang Zhang, Youliang Jiang, Yixuan Dang, Haoran Hou, Peiyi Shen, Mingtao Feng, Xia Zhao, Qiguang Miao, Syed Afaq Ali Shah, and Mohammed Bennamoun. Scene graph generation: A comprehensive survey. CoRR, abs/2201.00443, 2022. 2, 3

[64] Linchao Zhu, Zhongwen Xu, Yi Yang, and Alexander G Hauptmann. Uncovering the temporal context for video question answering. International Journal of Computer Vision, 124(3):409–421, 2017. 2