Semantic Composition and Alignment with Cross-Modality-Aware Syntactic Hypergraph Convolutional Network for Video Question Answering

Abstract

A key challenge in video question answering is how to realize the cross-modal semantic alignment between textual concepts and corresponding visual objects. Existing methods mostly seek to align the word representations with the video regions. However, word representations are often not able to convey a complete description of textual concepts, which are in general described by the compositions of certain words. To address this issue, we propose to first build a syntactic dependency tree for each question with an off-the-shelf tool and use it to guide the extraction of meaningful word compositions. Based on the extracted compositions, a hypergraph is further built by viewing the words as nodes and the compositions as hyperedges. Hypergraph convolutional networks (HCN) are then employed to learn the initial representations of word compositions. Afterwards, an optimal transport based method is proposed to perform cross-modal semantic alignment for the textual and visual semantic space. To reflect the cross-modal influences, the cross-modal information is incorporated into the initial representations, leading to a model named cross-modality-aware syntactic HCN. Experimental results on three benchmarks show that our method outperforms all strong baselines. Further analyses demonstrate the effectiveness of each component, and show that our model is good at modeling different levels of semantic compositions and filtering out irrelevant information.

1 Introduction

Video question answering (VideoQA) requires systems to understand the visual information and infer an answer for a natural language question from it. It has emerged as an important task with notable development towards bridging the gap between computer vision and natural language. VideoQA is challenging as it needs to understand the complex cross-modal relation between natural language question and video.

To capture the visual-language relation, some works have been proposed to utilize bilinear pooling operation or spatial-temporal attention mechanism to allign the video and textual features [Jang *et al.*, 2019; Seo *et al.*, 2021]. Some methods also proposed to use the co-attention mechanism [Jiang and Han, 2020; Li *et al.*, 2021] to align multi-modal features, or use memory-augmented RNN [Yin *et al.*, 2020] or graph memory mechanism [Liu *et al.*, 2021] to perform relational reasoning in VideoQA. Recently, DualVGR [Wang *et al.*, 2021] devises a graph-based reasoning unit and performs a word-level attention to obtain the question-related video features. In these methods, a cross-modal alignment between textual concepts and visual objects is attempted to be found, which, however, is mostly done at the word level, *i.e.*, aligning the representations of words with visual objects/frames. However, a word representation (even the contextual representation obtained from LSTM) is often not able to convey a complete description of a textual concept, which plays an essential role in video question answering. For instance, to answer the question "*What is the girl in green sitting on?*" as seen in Fig. 1, the model need to understand the textual concepts of "*girl in green*", "*girl sitting on*" etc., then align them with the corresponding video regions. But if the alignment is done at the word level, the word "*green*" will be aligned with all green objects in the video, $e.g.$, the green painting on the wall. Obviously, this is not the real intent of this question. In a question, there are often many textual concepts at different semantic levels, such as "*green*", "*girl in green*" and "*girl in green sitting on*" etc. Therefore, in the task of VideoQA, it is extremely beneficial to identify meaningful textual concepts at different semantic levels. A textual concept is generally described by a compositions of word that are not necessary to be adjacent, thus we cannot simply use the chunks of consecutive words to represent them.

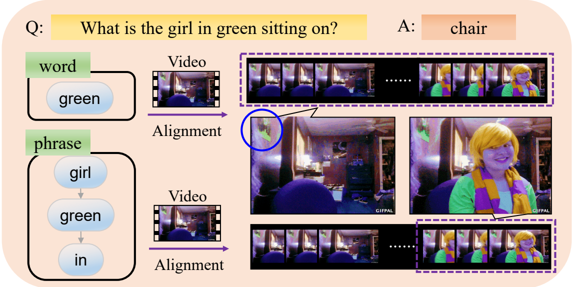


Figure 1: Different levels of semantic composition would be aligned to different frames.

To obtain meaningful compositions of words, we find that this problem is closely related to the syntactic dependency tree [Tai *et al.*, 2015], which describes the dependence structure of words in a sentence. We notice that every subtree can be approximately used to represent a meaningful composition of words that represents a textual concept, as illustrated in Fig. 2. Moreover, different-level textual concepts can be effectively captured by the subtrees of different orders. Thus, by building a syntactic dependency tree for questions with an off-the-shelf tool, we are able to obtain a set of compositions of words that representing different textual concepts. A hypergraph is further built by viewing the words as nodes and the compositions as hyperedges. Hypergraph convolutional networks (HCN) are then employed to learn the initial representations of these compositions (textual concepts). Given the initial representations, an optimal transport (OT) based alignment mechanism is developed to align the textual concepts and visual objects, which has been shown to be better at producing more accurate and sparser alignment than methods of directly using dot-product to compute the similarities ($e.g.$, cosine similarity) [Niculae and Blondel, 2017; Chen *et al.*, 2020]. With the cross-modal alignment, we further propose to update the initial composition and video representations by incorporating the relevant cross-modal information into them. To demonstrate the effectiveness of our approach, we compare our approach with competitive baselines on three benchmark VideoQA datasets, including TGIF-QA [Jang *et al.*, 2017], MSVD-QA [Xu *et al.*, 2017], and MSRVTT-QA [Xu *et al.*, 2016; Xu *et al.*, 2017]. The experimental results show that *SCAN*[1] outperforms all strong baselines and further analyses verify the validity of each component. Qualitative analysis further illustrates that our methods performs better in matching the video information based on semantic composition of text, and also in alleviating noise.

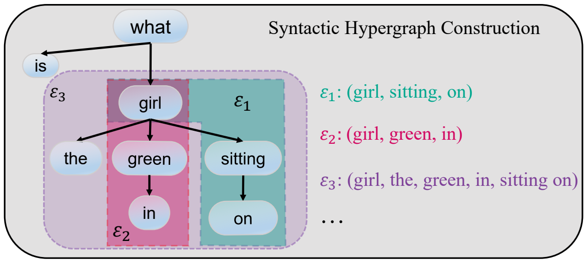
2 Methodology

In this section, we first briefly introduce the VideoQA task with basic notation, and then describe the definition and construction of the syntactic hypergraph . Based on the syntactic hypergraph, we present our cross-modality-aware syntactic hypergraph convolutional network to model the multi-modal interaction between the question and the video, where the optimal transport is employed for the alignment.

2.1 Preliminaries

Task Definition Given a video V and a question Q, the VideoQA task requires the system to find the answer $a \in \mathcal{A}$ with maximum probability $p(a|V, Q, \theta)$, where θ denotes the model parameters. Answers for VideoQA task are usually organized in two forms, $i.e.$, open-ended form and multiple-choice form. The open-ended answer is represented as a free-form text, while the answer of each multiple-choice question comes from a set with fixed number of answer candidates.

Multi-modal Features Following previous works for VideoQA [Park *et al.*, 2021] in which several consecutive frames in a video will be combined into one clip, we will first divide a video into several clips of the same length. Then we separately employ the pre-trained ResNet [He *et al.*, 2016] and ResNeXt-101 [Hara *et al.*, 2018] model on each frame and clip to extract the frame-wise appearance feature matrix $\boldsymbol{F} \in \mathbb{R}^{N_f \times d_v}$ and clip-wise motion feature matrix $\boldsymbol{M} \in \mathbb{R}^{N_c \times d_v}$, where d_v is the dimension of video feature (usually to be 2048), and N_f and N_c denote the number of frames and clips, respectively. For question Q, we follow the previous work [Jiang and Han, 2020] to represent each word with the pre-trained GloVe [Pennington *et al.*, 2014] word embedding and obtain question embedding matrix $\boldsymbol{Q} \in \mathbb{R}^{N_w \times d_w}$, where N_w is the number of word in a question, and d_w denotes the dimension of word embedding (usually to be 300).



what
Syntactic Hypergraph Construction
is
ε_3
girl
ε_1
the
green
sitting
in
on
ε_2
ε_1: (girl, sitting, on)
ε_2: (girl, green, in)
ε_3: (girl, the, green, in, sitting on)
...

Figure 2: Illustration of syntactic hypergraph construction.

2.2 Syntactic Hypergraph

In this part, we first introduce the definition of hypergraph and the advantage of modeling semantic compositons of words with hypergraph. Then, we describe how to construct hypergraph under the guidance of syntactic dependency tree.

Hypergraph Definition Let $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ denote a hypergraph, where $\mathcal{V}$ is a set containing N_v vertices and $\mathcal{E}$ is a set containing N_e hyperedges. Each hyperedge $\varepsilon \in \mathcal{E}$ denotes a set containing any numbers of vertices. The hypergraph can be represented by an incidence matrix $\boldsymbol{H} \in \mathbb{R}^{N_v \times N_e}$ where $\boldsymbol{H}_{i,\varepsilon} = 1$ if the hyperedge $\varepsilon \in \mathcal{E}$ contains the vertex $v_i \in \mathcal{V}$, otherwise 0. For example, if there are vertices $\mathcal{V} = \{``green", ``girl", ``sitting"\}$, one possible hyperedge could be the set $\varepsilon = \{``green", ``girl"\}$. We represent the semantic compositions of words with the hypergraph because the characteristics of hyperedge perfectly fit our assumption that a textual concept is represented by a set-like semantic composition of words. Therefore, it provides us with flexibility and capability to model complex interactions of words in the view of semantic composition.

Syntactic Hypergraph Construction Since the semantic composition phenomenon in nature language is usually related to the syntactic properties [Tai *et al.*, 2015], we take the syntactic information ($i.e.$ dependency syntax tree) as guidance to model the semantic composition of a question. Specifically, we apply off-the-shelf Stanza[2] toolkit on a given question to automatically generate the corresponding syntax tree, which represents the hierarchical syntactic relations of words. As the example shown in Fig.2, each leaf node in the

[1] SCAN stands for **S**emantic **C**omposition and **A**lignment with Cross-Modality-Aware Syntactic Hypergraph Convolution **N**etwork.

[2] https://github.com/stanfordnlp/stanza

syntax tree is a word, then each subtree can be viewed as a case of semantic composition of corresponding words. Therefore, we define each hyperedge as the set of words in each subtree. We travel the syntax tree in a hierarchical bottom-up manner to find all the subtrees of the syntax tree, and finally build all hyperedges with those subtrees.

Specifically, we describe the process of subtree generation method as follows. The algorithm begins by taking each leaf node as a subtree. Then we generate more subtrees by recursively adding higher-order branch nodes to the initial trees in a bottom-up manner. In each step, for each branch node, we add it to all its connected subtrees to generate more trees. Take the branch node "*girl*" in Fig. 2 as an example, we add "*girl*" to two connected subtrees {"*in*", "*green*"} and {"*sitting*", "*on*"} to generate two new subtrees {"*girl*", "*in*", "*green*"} and {"*girl*", "*sitting*", "*on*"}. Due to space limit, the pseudo algorithm of our subtree generation is given in Appendix.

2.3 Cross-Modality-Aware Syntactic Hypergraph Convolutional Network

With the syntactic hypergraph, we now propose a novel cross-modality-aware syntactic hypergraph convolutional network to incorporate cross-modal information into the initial representations, in contrast to the vanilla hypergraph convolutional network that is initially designed to model the information propagation between nodes with hyperedge as the bridge [Feng *et al.*, 2019]. The model mainly includes three steps: 1) **Initial hyperedge representation learning**: learning the initial hyperedge (composition) representations from node (word) representations; 2) **Cross-modal alignment**: finding cross-modal alignment between textual concepts and visual objects; 3) **Cross-modality-aware representations updating**: Updating the representations of hyperedges and videos by incorporating the cross-modal information into them.

Initial Hyperedge Representation Learning The first step is to gather node representations to produce the initial hyperedge representation to model the semantic composition process of words. We take contextual features of question words $\boldsymbol{Q}$ (Sec. 2.1) to initialize the node representations. Then, we gather the node representations to produce the corresponding hyperedge representations as:

$$\boldsymbol{X} = \boldsymbol{D}_e^{-1}\boldsymbol{H}^T\boldsymbol{Q}\boldsymbol{W}, \tag{1}$$

where $\boldsymbol{H}$ is the 0-1 incidence matrix representing whether a node is connected by a hyperedge (Sec. 2.1), and $\boldsymbol{X} \in \mathbb{R}^{N_s \times d_w}$ contains the representation of all N_s hyperedges, $\boldsymbol{W} \in \mathbb{R}^{d_w \times d_w}$ is a weight metrix, and $\boldsymbol{D}_e$ is the diagonal matrix denoting the degree of the hyperedge, which is defined as the number of the node connected by a hyperedge. It can be seen that, the multiplication operation $\boldsymbol{H}^T\boldsymbol{Q}$ in (1) performs the information gathering of words in a hyperedge. Since each hyperedge aggregates information from a set of nodes that probably represents a textual concept, the question-video alignment problem is then approximately reduced to the matching problem between hyperedge representations and video features.

Cross-Modal Alignment via Optimal Transport Given the initial hyperedge representations $\boldsymbol{X}$, we now present how to align the textual concepts and video frames. What we want to obtain is an alignment matrix $\boldsymbol{G}_{xf} \in \mathbb{R}^{N_s \times N_f}$, whose (i, j)-th element can reflect the degree of alignment between the i-th textual concept and the j-th frame. In this paper, inspired by recent success of optimal transport (OT) in the alignment of sole textual and visual space [Chen *et al.*, 2020], we propose to apply it to align the cross-modal spaces, *i.e.*, the textual and visual semantic spaces. Specifically, by viewing the hyperedge representations $\{\boldsymbol{x}_i\}_{i=1}^{N_s}$ and video frame representations $\{\boldsymbol{f}_j\}_{j=1}^{N_f}$ as two empirical probability distributions, we can define an optimal transport plan $\boldsymbol{\pi}^* \in \mathbb{R}_+^{N_s \times N_f}$ that will transport the features from textual concept space to video frame space with minimum transportation cost. Mathematically, the optimal transport plan is obtained by solving the following optimization problem

$$\boldsymbol{\pi}^* = \underset{\boldsymbol{\pi} \in \Pi_{\mathcal{I}}}{\arg\min} \left\{ \sum_{ij} \pi_{ij} c(\boldsymbol{x}_i, \boldsymbol{f}_j) \right\}, \tag{2}$$

where $\Pi_{\mathcal{I}}$ denotes the set of all feasible transport plans, which is composed of all $N_s \times N_f$ non-negative matrices whose elements are summed to be 1; and the cost function $c(\cdot, \cdot)$ is defined as

$$c(\boldsymbol{x}_i, \boldsymbol{f}_j) = 1 - \frac{\boldsymbol{x}_i\boldsymbol{T}_x(\boldsymbol{f}_j\boldsymbol{T}_f)^T}{\|\boldsymbol{x}_i\boldsymbol{T}_x\| \cdot \|\boldsymbol{f}_i\boldsymbol{T}_f\|}; \tag{3}$$

and $\boldsymbol{T}_x \in \mathbb{R}^{d_w \times d}$ and $\boldsymbol{T}_f \in \mathbb{R}^{d_v \times d}$ are used to transform the textual and visual features into the same semantic space. It can be seen that if the i-th textual concept and j-th frame are semantically relevant, the cost $c(\boldsymbol{x}_i, \boldsymbol{f}_j)$ will be small, otherwise it will be large. Thus, if the i-th hyperedge is closely relevant to the j-th frame, then π_{ij}^* will be assigned a large value, otherwise a value close to 0 will be assigned. Therefore, in this paper, the optimal transport plan $\boldsymbol{\pi}^*$ is directly used as the alignment weight matrix, that is,

$$\boldsymbol{G}_{xf} = \boldsymbol{\pi}^*. \tag{4}$$

It is shown in our experiments that this method tends to produce a sparser alignment matrix $\boldsymbol{G}_{xf}$ than the method of directly using dot-product to compute the similarity, and also leads to better results. Generally, the optimization problem (2) can not be solved exactly. In this paper, an off-the-shelf differentiable approximate method proposed in [Xie *et al.*, 2020] is borrowed to obtain the optimal matrix $\boldsymbol{\pi}^*$ approximately, where the concrete algorithm is presented in the Appendix. In a similar way, an alignment matrix $\boldsymbol{G}_{xm}$ that reflects the alignment degree between hyperedge representations $\boldsymbol{X}$ and clip features $\boldsymbol{M}$ can also be obtained.

Cross-Modality-Aware Representations Updating Given the alignment matrices $\boldsymbol{G}_{xm}$ and $\boldsymbol{G}_{xf}$, we now leverage them to improve the representations of hyperedges and video by incorporating the relevant cross-modal information into the initial representations. Specifically, we propose to compute the influence from video to hyperedge $\boldsymbol{X}_{v \to x} \in \mathbb{R}^{N_s \times d}$ and the influence from hyperedges to frame $\boldsymbol{F}_{x \to f} \in \mathbb{R}^{N_f \times d}$ as

$$\begin{aligned}\boldsymbol{X}_{v\to x} = &\mathrm{LayerNorm}(\mathrm{softmax}(\boldsymbol{G}_{xm})\boldsymbol{M}\boldsymbol{W}_{xm} \\ &+ \mathrm{softmax}(\boldsymbol{G}_{xf})\boldsymbol{F}\boldsymbol{W}_{xf} + \boldsymbol{X}\boldsymbol{W}_x),\end{aligned} \tag{5}$$

$$\boldsymbol{F}_{x\to f} = \mathrm{LayerNorm}(\mathrm{softmax}(\boldsymbol{G}_{xf}^T)\boldsymbol{X}\boldsymbol{W}_{fx} + \boldsymbol{F}\boldsymbol{W}_f), \quad (6)$$

where $\boldsymbol{W}_x$ and $\boldsymbol{W}_{fx}$, $\boldsymbol{W}_f$, $\boldsymbol{W}_{xm}$, and $\boldsymbol{W}_{xf}$ are trainable model parameters; the $\mathrm{softmax}(\cdot)$ is applied to the matrix in row-wise; and $\mathrm{LayerNorm}(\cdot)$ [Ba *et al.*, 2016] denotes layer normalization, which is used for more stable training. Similar to $\boldsymbol{F}_{x\to f}$, we can also compute the influence from hyperedges to clip features $\boldsymbol{M}_{x\to m}$. With the cross-modal influences $\boldsymbol{X}_{v\to x}$, $\boldsymbol{F}_{x\to f}$ and $\boldsymbol{M}_{x\to m}$, we can now incorporate them into the original representations and obtain cross-modality-aware representations, that is,

$$\tilde{\boldsymbol{X}} = \mathrm{LayerNorm}(\boldsymbol{X}_{v\to x}\boldsymbol{W}_{v\to x} + \boldsymbol{X}), \quad (7)$$

$$\tilde{\boldsymbol{F}} = \mathrm{LayerNorm}(\boldsymbol{F}_{x\to f}\boldsymbol{W}_{x\to f} + \boldsymbol{F}), \quad (8)$$

$$\tilde{\boldsymbol{M}} = \mathrm{LayerNorm}(\boldsymbol{M}_{x\to m}\boldsymbol{W}_{x\to m} + \boldsymbol{M}), \quad (9)$$

where $\boldsymbol{W}_{v\to x}$, $\boldsymbol{W}_{x\to m}$ and $\boldsymbol{W}_{x\to f}$ are trainable model parameters. Actually, $\tilde{\boldsymbol{X}}$ are the video-aware hyperedge representations, while $\tilde{\boldsymbol{F}}$ and $\tilde{\boldsymbol{M}}$ are the question-aware frame and clip representations. Finally, given the video-aware hyperedge representations $\tilde{\boldsymbol{X}}$, we can use it to get the video-aware node (word) representations

$$\tilde{\boldsymbol{Q}} = \boldsymbol{D}_v^{-1}\boldsymbol{H}\tilde{\boldsymbol{X}}\tilde{\boldsymbol{W}}, \quad (10)$$

where $\tilde{\boldsymbol{W}}$ is a trainable model weights, and $\boldsymbol{D}_v$ is the diagonal matrix with diagonal element being the node degree, which is defined as the number of hyperedges connecting to each node. It can be seen that multiplying the incidence matrix $\boldsymbol{H}$ in (10) can be viewed as updating the node representation via aggregating information from all the connected hyperedges. The computation above can be seen as a transformation block from $\{\boldsymbol{Q}, \boldsymbol{F}, \boldsymbol{M}\}$ to $\{\tilde{\boldsymbol{Q}}, \tilde{\boldsymbol{F}}, \tilde{\boldsymbol{M}}\}$. Obviously, we can stack more such transformation blocks and constitute a deeper model. The influence of depth will be discussed in the experiments.

2.4 Prediction and Training

Given the cross-modality-aware representations $\tilde{\boldsymbol{Q}}$, $\tilde{\boldsymbol{F}}$ and $\tilde{\boldsymbol{M}}$, we first project them into a common output space of dimension d_o, and then concatenate the projected representations into one matrix $\boldsymbol{Y} \in \mathbb{R}^{(N_w+N_f+N_c)\times d_o}$. Afterwards, a self-attention pooling function is applied on $\boldsymbol{Y}$ to obtain the final representation of the entire task $\boldsymbol{y} \in \mathbb{R}^{d_o}$ as

$$\boldsymbol{y} = [\mathrm{softmax}(\mathrm{LeakyReLU}(\boldsymbol{Y}\boldsymbol{W}_1^o)\boldsymbol{W}_2^o)]^T\boldsymbol{Y}, \quad (11)$$

where $\boldsymbol{W}_1^o \in \mathbb{R}^{d_o\times d_o}$ and $\boldsymbol{W}_2^o \in \mathbb{R}^{d_o\times 1}$ are trainable model parameters. We design different classifiers for different types of VideoQA tasks, into which the output vector $\boldsymbol{y}$ will be fed to predict the answer. For the case of the open-ended format, the output $\boldsymbol{y}$ is fed into a linear classifier that outputs the probabilities over the candidates in an answer set $\mathcal{A}$. The classifier is trained by minimizing the cross-entropy loss. It is worth noting that we treat the number counting problem (ranging from 0 to 10) as a regression problem and a L_2 regularization is added into the training loss. On the other hand, for the multiple-choice format, we follow previous work HGA [Jiang and Han, 2020] to concatenate the question with every answer candidate from $\mathcal{A}_q$ and generate N_a candidate question-answer sequences for each question. Then, these sequences are fed into our model to produce the output vector $\{\boldsymbol{y}_i\}_1^{N_a}$, which are then fed into a linear regression function to output N_a scores for every candidate answer. We train our model by minimizing the hinge loss as

$$\mathcal{L} = \frac{1}{|\mathcal{Q}|}\sum_{j=1}^{|\mathcal{Q}|}\sum_{i=1}^{N_a}\max(0, 1 + s_i^j - s_t^j), \quad (12)$$

where $\mathcal{Q}$ is the set of questions; s_i^j denotes the output score for the i-th answer of the j-th question; and t represent the number of ground-truth answer of the j-th question. In practice, we only need to use a mini-batch from the question set $\mathcal{Q}$ for every iteration.

3 Related Work

The VideoQA task requires machine to understand the visual-language correlation [Jang *et al.*, 2019; Le *et al.*, 2020; Zhang *et al.*, 2021; Wang *et al.*, 2021; Guo *et al.*, 2021]. To do this, MDAM [min Kim *et al.*, 2018] and PSAC [Li *et al.*, 2019] proposes to adopt the self-attention based approaches to learn the correlation between each frame and question. To enhance the frame-question correlation, L-GCN [Huang *et al.*, 2020] and HAIR [Liu *et al.*, 2021] first extracts the object information from each frame and integrate both object-level and frame-level information to enhance the frame-question correlation. Some researches have attempted to capture more fine-grained visual-language correlation. MASN [Seo *et al.*, 2021] introduce frame-level and clip-level modules to simultaneously model different-level correlation between visual information and question. RHA [Li *et al.*, 2021] proposed to use hierarchical attention network to further model the video subtitle-question correlation. There are also researches that adopt the memory-augmented approaches to capture the correlation [Fan *et al.*, 2019; Yin *et al.*, 2020]. Although these works have effectively capture the visual-language correlation, they ignore the syntactic compositional semantics of fine-grained concepts in the question, our *SCAN* bridges this gap by introducing a syntactic hypergraph and performing visual-aware hypergraph convolution.

4 Experiments

4.1 Datasets and Baselines

Datasets: Experiments are conducted on three benchmarks, including TGIF-QA [Jang *et al.*, 2017], MSVD-QA [Xu *et al.*, 2017], and MSRVTT-QA [Xu *et al.*, 2017] datasets, where the TGIF-QA dataset involves four sub-tasks (*i.e.*, *Action*, *Transition*, *FrameQA* and *Count*). Details of the three datasets can be found in the Appendix. Please note that we use accuracy (Acc.) as the evaluation metric for all the experiments, except repetition count task on TGIF-QA dataset, which uses the Mean Squared Error (MSE).

Baselines: We compare our model with the following strong baselines: AMU [Xu *et al.*, 2017], ST-VQA [Jang *et al.*, 2017], Co-Mem [Gao *et al.*, 2018], HME [Fan *et al.*, 2019],

Method	Action	Transition	FrameQA	Count$^{\downarrow}$
ST-VQA	60.8	67.1	49.3	4.40
Co-Mem	68.2	74.3	51.5	4.10
HME	73.9	77.8	53.8	4.02
HGA	75.4	81.0	55.1	4.09
HCRN	75.0	81.4	55.9	3.82
L-GCN	74.3	81.1	56.3	3.95
QueST	75.9	81.0	59.7	4.19
B2A	75.9	82.6	57.5	**3.71**
HAIR	77.8	82.3	60.2	3.88
SCAN	**79.8**	**84.3**	**61.0**	3.89

Table 1: Performances on TGIF-QA dataset.

Method	What	Who	How	When	Where	All
AMU	20.6	47.5	83.5	72.4	53.6	32.0
ST-VQA	18.1	50.0	**83.8**	72.4	28.6	31.3
Co-Mem	19.6	48.7	81.6	74.1	31.7	31.6
HME	22.4	50.1	73.0	70.7	42.9	33.7
TSN	25.0	51.3	**83.8**	**78.4**	**59.1**	36.7
HGA	23.5	50.4	83.0	72.4	46.4	34.7
HCRN	—	—	—	—	—	36.1
QueST	24.5	52.9	79.1	72.4	50.0	36.1
B2A	—	—	—	—	—	37.2
HAIR	—	—	—	—	—	37.5
DualVGR	28.7	53.8	80.0	70.7	46.4	39.0
SCAN	**29.5**	**55.7**	82.4	72.4	42.9	**40.3**

Table 2: Performances on MSVD-QA dataset.

Method	What	Who	How	When	Where	All
AMU	26.2	43.0	82.4	72.5	30.0	32.5
ST-VQA	24.5	41.2	78.0	76.5	34.9	30.9
Co-Mem	23.9	42.5	74.1	69.0	42.9	32.0
HME	26.5	43.6	82.4	76.0	28.6	33.0
TSN	27.9	46.1	**84.1**	77.8	**37.6**	35.4
HGA	29.2	45.7	83.5	75.2	34.0	35.5
HCRN	—	—	—	—	—	35.6
QueST	27.9	45.6	83.0	75.7	31.6	34.6
B2A	—	—	—	—	—	36.9
HAIR	—	—	—	—	—	36.9
DualVGR	29.4	45.6	79.8	76.7	36.4	35.5
SCAN	**30.3**	**48.8**	81.5	**78.0**	37.2	**37.1**

Table 3: Performances on MSRVTT-QA dataset.

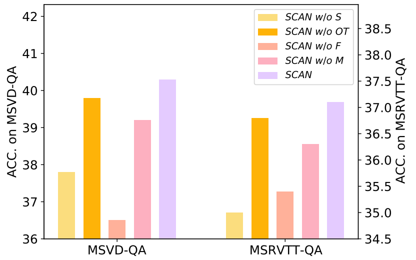
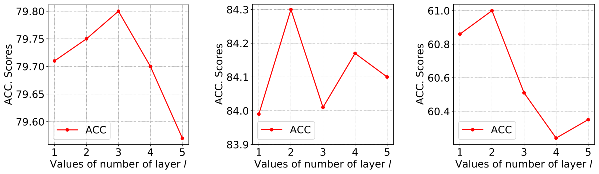


Figure 3: Performances of *SCAN* variants that exclude or replace certain components on MSVD-QA and MSRVTT-QA datasets.



(a) Action (b) Transition (c) FrameQA

Figure 4: The performance of *SCAN* under different number l of the video-aware hypergraph convolutional network layers on $Action$, $Transition$, and $FrameQA$ tasks.

TSN [Yang *et al.*, 2019], HGA [Jiang and Han, 2020], HCRN [Le *et al.*, 2020], L-GCN [Huang *et al.*, 2020], QueST [wen Jiang *et al.*, 2020], *Bridge to Answer* (shorted as B2A) [Park *et al.*, 2021], HAIR [Liu *et al.*, 2021], DualVGR [Wang *et al.*, 2021]. It is worth mentioning that the performance of baselines on certain datasets are taken from the corresponding papers.

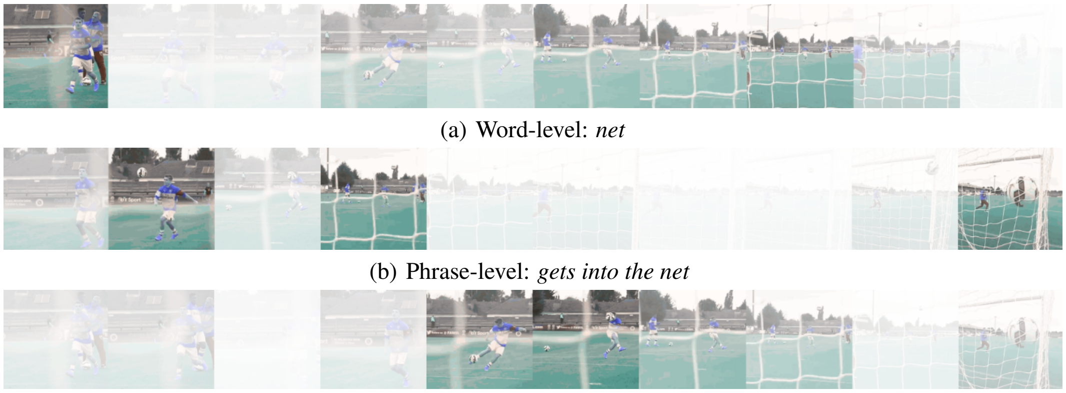
4.2 Main Results

The experimental results of our model and the strong baselines on TGIF-QA, MSVD-QA and MSRVTT-QA datasets are shown in Table 1, Table 2 and Table 3, respectively, with the best performance highlighted in bold. For TGIF-QA dataset, our model outperforms the recent baseline HAIR by 2.6% on *Action*, 2.4% on *Transition*, and 1.3% on *FrameQA*. It is worth noting that the B2A baseline also involves syntactic information for VideoQA, but it only takes the whole syntactic graph as a tree, without considering the multi-level compositional semantics of the syntactic information. It can be seen that our model outperforms B2A model by 5.1%, 2.1%, and 6.1% on *Action*, *Transition*, and *FrameQA* sub-tasks, respectively, showing the effectiveness of modeling multi-level compositional semantics of questions. The MSVD-QA and MSRVTT-QA benchmarks are more challenging as they only provide open-ended questions. It can be observed that our model also outperforms the most recent DualVGR model targeting on computing question-attended video representation by a large margin (3.3% and 4.5% acc. on MSVD-QA and MSRVTT-QA), which shows that semantic composition is essential for representing the semantic meaning of the question.

4.3 Ablation Studies

We conduct in-depth analyses on how different components and parameters contribute to the model performance. To this end, we evaluate the performance with different variants of our model from two aspects: (1) excluding or replacing certain components, (2) changing the values of hyperparameters.

Impacts of Different Components We evaluate the effectiveness of different components by eliminating four modules. *1) Syntactic tree (SCAN w/o S)*: we remove the syntactic hypergraph and only consider the word-level cross-modality alignment. *2) Optimal transport alignment (SCAN w/o OT)*: we remove the optimal transport alignment module and replace it with a simple dot-product similarity module. *3) Frame-level feature (SCAN w/o F)*: we remove the frame-level video feature and only keep the motion-level feature. *4) Clip-level feature (SCAN w/o M)*: we remove the clip-level video feature. We compare these four tailored models with

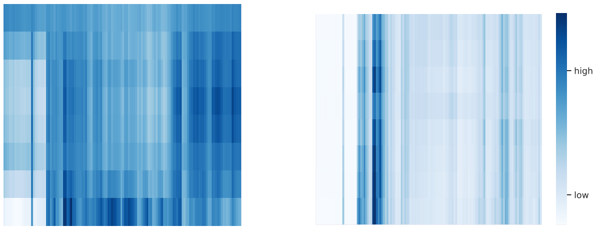


(a) Word-level: *net*

(b) Phrase-level: *gets into the net*

(c) Sentence-level: *what is the soccer player wearing a uniform gets into the net*

Figure 5: Visualization of visual-language alignments between semantic composition and video frames learned by our model.



(a) Simple Dot-production

(b) Optimal Transport

Figure 6: Visualization of different attention matrices.

the complete *SCAN* model on the MSVD-QA and MSRVTT-QA datasets. The results are showed in Fig. 3. It can be seen that eliminating syntactic hypergraph *(SCAN w/o S)* leads to significant performance drop, suggesting the importance of modeling compositional semantics of the question. It can be further observed that OT alignment *(SCAN w/o OT)* also largely contributes to the model performance, demonstrating the necessity of filtering out irrelevant video information by using a more sparser alignment matrix. Also, eliminating frame-level and clip-level features, $i.e.$, *SCAN (w/o F)* and *SCAN (w/o M)*, also harms the final performance, which indicates the importance of simultaneously modeling visual features at different levels.

Impacts of the Number of Computational Blocks We analyze the impact of using different number of computation blocks in the proposed HCN. Fig. 4 shows the performance on *Action*, *Transition* and *FrameQA* sub-tasks of TGIF-QA dataset with the number changing from 1 to 5. It can be observed that the optimal number varies from task to task, *e.g.*, $l = 2$ for *Transition* and *Frame QA*, and $l = 3$ for the *Action* subtask. This phenomenon indicates that different question types emphasize different levels of compositional semantics of questions, and we can increase the depth of the interaction layers to support more complex task, or find a tradeoff between performance and efficiency.

4.4 Qualitative Analysis

Visualization of language-vision Alignments In order to visualize the alignments between different semantic composition and the visual feature, we take the question "*what is the soccer player wearing a uniform gets into the net*" extracted from $frameQA$ as an example. We visualize the aligned videos based on different levels of semantic composition (i.e., word-level, phrase-level and sentence-level). Then we extract 10 frames, which in temporally ranked, with the highest alignment weights for each semantic level. The visualization is shown in Fig. 5, where frames with higher attention weights are clearer and vice verse. It can be seen that, the word net is aligned with most frames containing nets. When it is composed to phrase-level semantics ("*gets into the net*"), the matched frames intend to focus on the scene that the soccer goes into the net. Finally, for sentence-level, we find that key frames focus on the entire process of a football player kicking the ball into the net. The results show that *SCAN* can better model the semantic composition phenomenon and its multi-modal alignment with the visual information.

Visualization of Different Alignment Matrices To analyze whether our OT Alignment mechanism indeed generate sparse alignment scores, we visualize the OT alignment matrix and compare it with the matrix generated by simply dot-product approach, and plots one typical example in Fig. 6. It can be seen that the OT method returns a sparser alignment matrix, while dot-product-based attention is effectively dense. This emphasizes the effectiveness of OT-based soft alignment in concentrating more relevant cross-modal information.

5 Conclusion

In this paper, we propose to model the semantic composition phenomenon of question with a syntactic hypergraph for VideoQA. We first build the syntactic hypergraph based on the syntactic dependency tree in a hierarchical bottom-up manner. Then we propose a cross-modality-aware syntactic hypergraph convolution network to align the cross-modal semantic information. To enhance the cross-modal alignment, we adopt the optimal transport attention mechanism to obtain a sparse matching. Experiments show that our method outperforms strong baselines on three benchmark datasets and verify the effectiveness of each component.

References

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

[Chen *et al.*, 2020] Liqun Chen, Guoyin Wang, Chenyang Tao, Dinghan Shen, Pengyu Cheng, Xinyuan Zhang, Wenlin Wang, Yizhe Zhang, and Lawrence Carin. Improving textual network embedding with global attention via optimal transport. In *ACL*, pages 5193–5202, 2020.

[Fan *et al.*, 2019] Chenyou Fan, Xiaofan Zhang, Shu Zhang, Wensheng Wang, Chi Zhang, and Heng Huang. Heterogeneous memory enhanced multimodal attention model for video question answering. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1999–2007, 2019.

[Feng *et al.*, 2019] Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. Hypergraph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3558–3565, 2019.

[Gao *et al.*, 2018] J. Gao, Runzhou Ge, Kan Chen, and Ramakant Nevatia. Motion-appearance co-memory networks for video question answering. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6576–6585, 2018.

[Guo *et al.*, 2021] Zhicheng Guo, Jiaxuan Zhao, Licheng Jiao, Xu Liu, and Fang Liu. A universal quaternion hypergraph network for multimodal video question answering. *IEEE Transactions on Multimedia*, 2021.

[Hara *et al.*, 2018] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? *CVPR*, pages 6546–6555, 2018.

[He *et al.*, 2016] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[Huang *et al.*, 2020] Deng Huang, Peihao Chen, Runhao Zeng, Qing Du, Mingkui Tan, and Chuang Gan. Location-aware graph convolutional networks for video question answering. In *AAAI*, volume 34, pages 11021–11028, 2020.

[Jang *et al.*, 2017] Y. Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. Tgif-qa: Toward spatio-temporal reasoning in visual question answering. *CVPR*, pages 1359–1367, 2017.

[Jang *et al.*, 2019] Y. 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*, 127:1385 – 1412, 2019.

[Jiang and Han, 2020] Pin Jiang and Yahong Han. Reasoning with heterogeneous graph alignment for video question answering. In *AAAI*, 2020.

[Le *et al.*, 2020] Thao Minh Le, Vuong Le, Svetha Venkatesh, and T. Tran. Hierarchical conditional relation networks for video question answering. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9969–9978, 2020.

[Li *et al.*, 2019] 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.

[Li *et al.*, 2021] Fangtao Li, Ting Bai, Chenyu Cao, Zihe Liu, Chenghao Yan, and Bin Wu. Relation-aware hierarchical attention framework for video question answering. *Proceedings of the 2021 International Conference on Multimedia Retrieval*, 2021.

[Liu *et al.*, 2021] 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.

[min Kim *et al.*, 2018] Kyung min Kim, Seongho Choi, Jin-Hwa Kim, and Byoung-Tak Zhang. Multimodal dual attention memory for video story question answering. In *ECCV*, 2018.

[Niculae and Blondel, 2017] Vlad Niculae and Mathieu Blondel. A regularized framework for sparse and structured neural attention. In *NIPS*, 2017.

[Park *et al.*, 2021] Jungin Park, Jiyoung Lee, and Kwanghoon Sohn. Bridge to answer: Structure-aware graph interaction network for video question answering. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15521–15530, 2021.

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

[Seo *et al.*, 2021] Ahjeong Seo, Gi-Cheon Kang, Joon Ki Park, and Byoung-Tak Zhang. Attend what you need: Motion-appearance synergistic networks for video question answering. In *ACL/IJCNLP*, 2021.

[Tai *et al.*, 2015] Kai Sheng Tai, Richard Socher, and Christopher D Manning. Improved semantic representations from tree-structured long short-term memory networks. In *ACL-IJCNLP*, pages 1556–1566, 2015.

[Wang *et al.*, 2021] Jianyu Wang, Bingkun Bao, and Changsheng Xu. Dualvgr: A dual-visual graph reasoning unit for video question answering. *IEEE Transactions on Multimedia*, 2021.

[wen Jiang *et al.*, 2020] Jian wen Jiang, Ziqiang Chen, Haojie Lin, Xibin Zhao, and Yue Gao. Divide and conquer: Question-guided spatio-temporal contextual attention for video question answering. In *AAAI*, 2020.

[Xie *et al.*, 2020] Yujia Xie, Xiangfeng Wang, Ruijia Wang, and Hongyuan Zha. A fast proximal point method for computing exact wasserstein distance. In *Uncertainty in Artificial Intelligence*, pages 433–453. PMLR, 2020.

[Xu *et al.*, 2016] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. *2016 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5288–5296, 2016.

[Xu *et al.*, 2017] D. 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. *Proceedings of the 25th ACM international conference on Multimedia*, 2017.

[Yang *et al.*, 2019] Tianhao Yang, Zhengjun Zha, Hongtao Xie, Meng Wang, and Hanwang Zhang. Question-aware tube-switch network for video question answering. *Proceedings of the 27th ACM International Conference on Multimedia*, 2019.

[Yin *et al.*, 2020] Chengxiang Yin, Jian Tang, Zhiyuan Xu, and Yanzhi Wang. Memory augmented deep recurrent neural network for video question answering. *IEEE Transactions on Neural Networks and Learning Systems*, 31:3159–3167, 2020.

[Zhang *et al.*, 2021] Fuwei Zhang, Ruomei Wang, Songhua Xu, and Fan Zhou. Fusing temporally distributed multi-modal semantic clues for video question answering. *ICME*, 2021.

Algorithm 1 OT-based Attention Mechanism $\mathcal{OT}(\boldsymbol{X}, \boldsymbol{F})$

```
Input: Hyperedges matrices $\boldsymbol{X} = \{\boldsymbol{x}_i\}_1^{N_s}$ frame-level matrices $\boldsymbol{F} = \{\boldsymbol{f}_j\}_1^{N_f}$.
$\boldsymbol{b} = \frac{1}{N_s}\mathbf{1}_{N_s}$, $\boldsymbol{\pi}^{(1)} = \mathbf{1}_{N_s}\mathbf{1}_{N_f}^T$, $\boldsymbol{C}_{ij} = e^{-c(\boldsymbol{x}_i, \boldsymbol{f}_j)}$.
for $t = 1, 2, \cdots, 10$ do
    $\boldsymbol{\Gamma} = \boldsymbol{C} \odot \boldsymbol{\pi}^{(t)}$.
    $\boldsymbol{a} = \frac{1}{N_s \boldsymbol{\Gamma} \boldsymbol{b}}$, $\boldsymbol{b} = \frac{1}{N_f \boldsymbol{\Gamma}^T \boldsymbol{a}}$.
    $\boldsymbol{\pi}^{(t+1)} = \mathrm{diag}(\boldsymbol{a})\boldsymbol{\Gamma}\mathrm{diag}(\boldsymbol{b})$.
end for
Return $\boldsymbol{\pi}$
```

A More Details for the Datasets

Experiments are conducted on three benchmark datasets, including TGIF-QA [Jang *et al.*, 2017], MSVD-QA [Xu *et al.*, 2017], and MSRVTT-QA [Xu *et al.*, 2017] datasets.

1) TGIF-QA [Jang *et al.*, 2017] is a prominent large-scale benchmark dataset for VideoQA task that consists of $165K$ Q&A pairs based on $72K$ animated GIFs. The dataset defines four tasks: (1) Repeating action ($Action$) requires to identify the action repeated for a given number of times from 5 candidate answers; (2) State transition ($Transition$) also deals with 5 candidate answers aiming to identify the transition of two states; (3) Frame QA ($FrameQA$) is an open-ended task that needs to find a key frame in the video to indicate the correct answer from a pre-defined dictionary; (4) Repetition count ($Count$) also contains an open-ended numbers of task to count the number of occurrences of an action.

2) MSVD-QA [Xu *et al.*, 2017] is an open-ended VideoQA dataset, which is divided into 5 different types, including $what$, who, how, $when$, and $where$. The dataset contains 1970 short video clip, 50505 Q&A pairs and 1000 pre-defined answers.

3) MSRVTT-QA [Xu *et al.*, 2017] is similar to MSVD-QA. It is also divided into the same five types with 1000 pre-defined answers. The MSRVTT-QA is generated from the MSRVTT [Xu *et al.*, 2016] dataset, containing 10K videos and $243K$ Q&A.

B Algorithm for OT

We adopt an off-the-shelf differentiatable approximate method [Xie *et al.*, 2020] to obtain the OT matrix $\boldsymbol{\pi}^*$, which is summarized in Algorithm 1.

C Subtree Generation Algorithm

Algorithm 2 shows our subtree generation algorithm SubTreeGen$(\cdot, \cdot)$ used in our syntactic hypergraph construction. The algorithm begins by taking each leaf node as a subtree. Then, for the branch node v_i, our algorithm first adopts the recursive GetSubRree$(\cdot, \cdot)$ on each of the child node of node v_i to obtain the connected subtrees. Secondly, we add node v_i to all connected subtrees to generate more trees. Finally, we directly apply recursive GetSubRree$(\cdot, \cdot)$ on node v_i to obtain higher level semantic composition.

Algorithm 2 Subtree Generation Algorithm SubTreeGen$(\cdot, \cdot)$

```
Input: Syntactic dependency Tree $\mathcal{T}$, set of vertices $\mathcal{V}$ containing all vertices on the tree.
$\mathcal{T}_s = \{\}$ // The set containing all found subtrees.
for $v_i$ in $\mathcal{V}$ do
    if $v_i$ is leaf node then
        APPEND($\mathcal{T}_s$, $\{v_i\}$)
    end if
    if $v_i$ is branch node then
        for $c$ in Child($\mathcal{T}, v_i$) do // The child of $v_i$
            APPEND($\mathcal{T}_s$, $\{v_i$, GetSubTree$(\mathcal{T}, c)\}$)
        end for
        APPEND($\mathcal{T}_s$, $\{$GetSubTree$(\mathcal{T}, v_i)\}$)
    end if
end for
Return $\mathcal{T}_s$
```

Algorithm 3 GetSubTree($\mathcal{T}$,c)

```
Input: Syntactic dependency Tree $\mathcal{T}$, node $c$.
if $c$ is leaf node then
    Return $\{c\}$
end if
$\mathcal{T}_s' = \{c\}$
for $c'$ in Child($\mathcal{T}, c$) do
    APPEND($\mathcal{T}_s'$, GetSubTree($\mathcal{T}$,$c'$))
end for
Return $\mathcal{T}_s'$
```