Dynamic Spatio-Temporal Modular Network for Video Question Answering

Zi Qian
qian-z20@mails.tsinghua.edu.cn
Tsinghua University
Beijing, China

Xin Wang*
xin_wang@tsinghua.edu.cn
Tsinghua University
Beijing, China

Xuguang Duan
dxg18@mails.tsinghua.edu.cn
Tsinghua University
Beijing, China

Hong Chen
h-chen20@mails.tsinghua.edu.cn
Tsinghua University
Beijing, China

Wenwu Zhu*
wwzhu@tsinghua.edu.cn
Tsinghua University
Beijing, China

ABSTRACT

Video Question Answering (VideoQA) aims to understand given videos and questions comprehensively by generating correct answers. However, existing methods usually rely on end-to-end black-box deep neural networks to infer the answers, which significantly differs from human logic reasoning, thus lacking the ability to explain. Besides, the performances of existing methods tend to drop when answering compositional questions involving realistic scenarios. To tackle these challenges, we propose a **D**ynamic **S**patio-**T**emporal Modular **N**etwork (DSTN) model, which utilizes a spatio-temporal modular network to simulate the compositional reasoning procedure of human beings. Concretely, we divide the task of answering a given question into a set of sub-tasks focusing on certain key concepts in questions and videos such as objects, actions, temporal orders, etc. Each sub-task can be solved with a separately designed module, *e.g.*, spatial attention module, temporal attention module, logic module, and answer module. Then we dynamically assemble different modules assigned with different sub-tasks to generate a tree-structured spatio-temporal modular neural network for human-like reasoning before producing the final answer for the question. We carry out extensive experiments on the AGQA dataset to demonstrate our proposed DSTN model can significantly outperform several baseline methods in various settings. Moreover, we evaluate intermediate results and visualize each reasoning step to verify the rationality of different modules and the explainability of the proposed DSTN model.

CCS CONCEPTS

• **Computing methodologies → Computer vision**; • **Information systems → Question answering**.

KEYWORDS

video question answering, modular neural network

ACM Reference Format:
Zi Qian, Xin Wang, Xuguang Duan, Hong Chen, and Wenwu Zhu. 2022. Dynamic Spatio-Temporal Modular Network for Video Question Answering. In *Proceedings of the 30th ACM International Conference on Multimedia (MM '22), Oct. 10–14, 2022, Lisboa, Portugal.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3503161.3548061

1 INTRODUCTION

Video Question Answering (VideoQA) aims to correctly answer questions given the related videos. As a challenging extension of the static image question answering (VQA) [4], VideoQA faces the following core challenges: (i) videos usually contain much more information compared with static images, posing more difficulties to feature procession and comprehension, and (ii) VideoQA involves more reasoning operations over both spatial and temporal dimensions. The rich temporal multi-modal information makes VideoQA a challenging research topic in both academia and industry.

Existing VideoQA works can generally be categorized into four categories: (1) Encoder-decoder structure based methods [17, 28, 48, 53, 54], (2) Memory network based methods [10, 24, 33, 42, 46], (3) Spatio-temporal graph neural network (GNN) based methods [36, 37, 39, 43, 45], and (4) Pre-trained Models [1, 30, 51, 52]. However, existing VideoQA models suffer from the following three challenging problems: (i) They are designed with a black-box deep structure without exploring whether each model part is correctly operating based on the designing logic. (ii) They are evaluated on simple questions or scenarios, and their spatio-temporal reasoning performances tend to drop for compositional questions in realistic scenarios. (iii) They are still far away from the process of human logic reasoning [31].

To tackle these challenges and move one step closer to explainable VideoQA, we propose a **D**ynamic **S**patio-**T**emporal Modular **N**etwork (DSTN) model in this paper, which utilizes a hierarchical logic structure with modular design to simulate the compositional procedure of human logic reasoning explicitly and produce more explainable results. Concretely, the proposed DSTN model first decomposes the given question into several sub-tasks with a hierarchical logic structure in a step-by-step manner, where sub-tasks cover a set of key concepts (*e.g.*, object, subject, relation, location, action, temporal order, and duration). In order to handle different sub-tasks, we propose various modules with different functions

[*] Corresponding Authors

MM '22, October 10–14, 2022, Lisboa, Portugal

ACM ISBN 978-1-4503-9203-7/22/10.
https://doi.org/10.1145/3503161.3548061

involving temporal and spatial localization, logic reasoning, relation discovery, etc. Different modules are dynamically assembled into a modular network with rich logical reasoning ability based on the hierarchical logic structure. Then, the assembled modular neural network operates upon textual features and visual features simultaneously in a bottom-up manner to generate the final answer to the given question.

We carry out experiments on the AGQA dataset [11], which is a typical VideoQA dataset for compositional spatio-temporal reasoning under real-world scenarios. Besides the VideoQA accuracy metric widely used in previous works, we also compare the proposed DSTN model with several state-of-the-art approaches in terms of additional settings to demonstrate superior spatio-temporal reasoning ability of DSTN from different aspects, *e.g.*, generalization ability for novel compositional questions. Moreover, we evaluate and visualize intermediate results (results of modules) to verify our modules' reasoning ability and rationality, which further demonstrates the explainability of our model.

To summarize, this paper makes the following contributions:

- We propose the dynamic spatio-temporal modular network (DSTN) model, which is the first modular neural network based approach in VideoQA for explainable video reasoning in real-world scenarios.
- We unify spatial reasoning, temporal reasoning, and logic modules with a dynamically assembled modular framework to simulate the process of human inference.
- We conduct extensive experiments to demonstrate the advantages of DSTN with various settings, and explore the performances of different modules to demonstrate the rationality of modules and the explainability of the overall model.

2 RELATED WORKS

Video Question Answering. Existing methods for video question answering could be generally categorized into four categories: (1) Encoder-decoder structure based methods [17, 28, 48, 53, 54] adopt encoder-decoder frameworks to generate both spatial and temporal contextual features as well as multi-modal representations. (2) Memory network based methods [7, 10, 24, 25, 33, 35, 42, 46, 49] utilize memory network structure to process video and question information, since well-designed write operator and read operator could efficiently generate meaningful multi-modal representations. (3) Spatial-temporal graph neural network (GNN) based methods [12, 15, 19, 20, 36, 37, 39, 43, 45, 50] modify videos into a graph structured format, and then use GNN based network *(e.g.*, Graph Convolutional Network (GCN) and Graph Attention Network (GAT)) to process dynamic information in the video to obtain contextual representations. (4) Pre-trained Models [1, 9, 30, 51, 52] use extensive data (*e.g.*, vision, audio, and text) to generate relatively unbiased multi-modal representations. Besides methods belonging to these categories, there exist some other works using relation networks [8, 26, 47] and neuro-symbolic framework [44, 50].

Although existing approaches have achieved remarkable performance gains, their performances tend to drop when they answer complicated logical questions under realistic scenarios and are still far from real human logic reasoning.

Table 1: Examples of concepts.

Concepts	Examples
Object	bottle, table, dog, milk, window, bed, car
Subject	person, man, woman, kid, children
Relation	put on, pick up, throw, drink
Location	in front of, above, left, right, up, down
Action	put on a coat, drink milk
Temporal Order	first, last, before, after, at the same time
Duration	long, short, continuous, intermittent

Modular Neural Network. Since the procedure of visual (including images and videos) question reasoning is essentially compositional, the modular neural network has been used for several methods in image question answering [2, 3, 3, 6, 13, 14, 22, 31, 34] to increase explainability as well as simulating procedure of human reasoning. These methods explicitly decompose questions into semantic sub-tasks and assemble specialized modules to handle these sub-tasks. However, these methods are limited to image question answering without considering spatio-temporal reasoning.

To conclude, in this work, we address all these difficulties by a dynamic spatio-temporal modular neural network decomposing questions into sub-tasks, where sub-tasks could be solved with a library of general modules designed about key concepts (*e.g.*, object, action, temporal order etc.) in the real-world video scenario.

3 DYNAMIC SPATIO-TEMPORAL MODULAR NETWORK

In this section, we describe the proposed **D**ynamic **S**patio-**T**emporal Modular **N**etwork (DSTN) which targets at explainable spatio-temporal reasoning in real-world videos by decomposing the question into modularized sub-tasks. Figure 1 shows the framework of our proposed DSTN model with an example to illustrate the whole model. In general, DSTN consists of three key components: (1) Sub-tasks modularization, (2) Modular layout policy, and (3) Modular neural network assembly and executions.

3.1 Sub-tasks Modularization

Most of our natural language questions could be decomposed into sub-questions. For example, when asked "what did the person do after taking a picture?", we could first find the person that "*has taken pictures*", then we find his current action and answer the question. Similarly, our DSTN contains a series of general modules implementing different sub-tasks. We primitively focus on several key concepts in video question answering as: *object*, *subject*, *relation*, *location*, *action*, *temporal order*, and *duration.* We list several examples for each concept in Table 1. Unlike static images, most of these concepts have a lifecycle. Given a certain video, for example, a person (*subject*) may only take a picture (*action*) in a certain time period. Thus, our modularized sub-tasks are atomic tasks that operate spatial and temporal reasoning for a given concept. Most of our modules take a series of spatio-temporal attention maps and a series of concept embeddings as inputs and output a series of corresponding spatio-temporal attention to represent the lifecycle of a certain concept.

Given video feature $x^v = \mathcal{E}^v(V) \in \mathbb{R}^{T\times dim_v}$, question feature $x^q = \mathcal{E}^q(Q) \in \mathbb{R}^{dim_q}$, and a set of concept parameters $\{C_i\}$ extracted from the question, a module **M** is designed to transform the

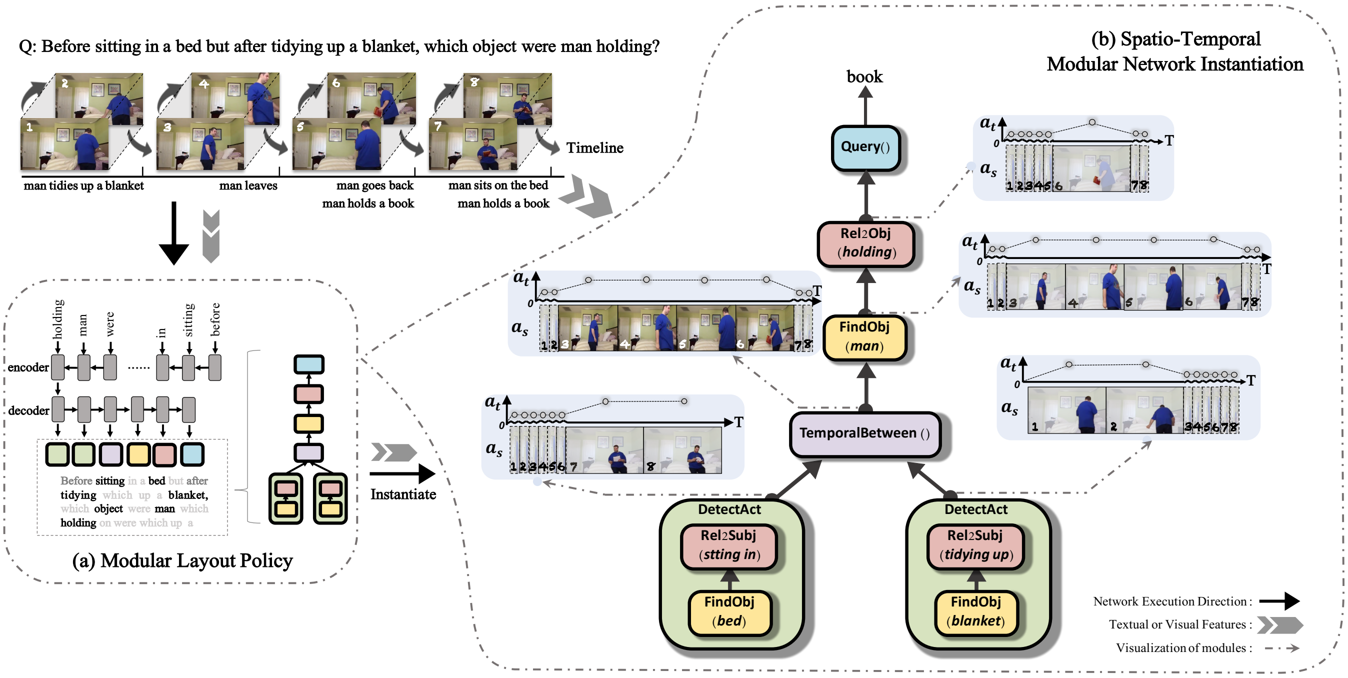
Q: Before sitting in a bed but after tidying up a blanket, which object were man holding?
Timeline
man tidies up a blanket
man leaves
man goes back
man holds a book
man sits on the bed
man holds a book
encoder
decoder
holding
man
were
in
sitting
before
Before sitting in a bed but after tidying which up a blanket, which object were man which holding on were which up a
(a) Modular Layout Policy
Instantiate
(b) Spatio-Temporal Modular Network Instantiation
book
Query()
Rel2Obj (holding)
FindObj (man)
TemporalBetween ()
DetectAct
Rel2Subj (stting in)
FindObj (bed)
DetectAct
Rel2Subj (tidying up)
FindObj (blanket)
a_t
a_s
T
Network Execution Direction :
Textual or Visual Features :
Visualization of modules :

Figure 1: Framework of our proposed Dynamic Spatio-Temporal Modular Network (DSTN) model. Given a pair of video and question, our model constructs a series of perceptual modules with a sequence-to-sequence model (*cf.*, Figure 1a), where each module comes with customized concept parameters and visual features for semantics-to-visual perception. The perceptual modules are assembled into a tree-structured modular layout and instantiated into a spatio-temporal modular neural network (*cf.*, Figure 1b), which executes in a bottom-up manner with visual features and concept parameters to generate the final answer. We visualize the idealized intermediate attention maps for some modules as shown in light blue boxes pointed by the grey dotted arrow. The upside part of light blue boxes is temporal attention map a_t and the bottom part is spatial attention map a_s. For spatial attention map a_s, we highlight the localized concepts and obscure other parts. For temporal attention map a_t, we plot the temporal attention curve to show the importance of each frame.

input attention maps (denoted as attentions for simplicity) $\{a^{(in)}\}$ into another set of attentions $\{a^{(out)}\}$ or the answer *ans*:

$$\{a^{(out)}\} \text{ OR } ans \leftarrow \mathrm{M}\left(\{a^{(in)}\}, \{C_i\}; x^v, x^q\right). \tag{1}$$

M is implemented with a small differentiable neural network with unified interface and specific semantic meaning. All modules are conditioned on the video feature x^v and question feature x^q, and operated on the spatio-temporal attention with respect to a set of given concept parameter $\{C_i\}$.

In this paper, we consider two types of attention $a_t \in \mathbb{R}^T$ representing the temporal attention over T frames of the video, and $a_s \in \mathbb{R}^{T \times dim_v}$ representing the spatial attention over each of the T frames. Based on the functionalities of different modules, they could operate on a_t or a_s, or both. Accordingly, we categorize modules into four categories:

(1) **Spatial Attention Modules**: The modules in this category conduct spatial operations over inputs, which can be regarded as a repetition of operations on every single frame. As shown in Table 2, `FindObj` module outputs a spatial attention map and is used to localize object representing the concept embedding $\mathbf{e}^C$. `Rel2` module and `Loc2` module can be used to localize objects that have a specific relation or on a certain position to a known object. Besides, module `DetectObj`, `DetectRel`, and `DetectAct` are designed to localize all the existing objects, relations, or actions which are given by the concept embedding within specific time period, where the

Table 2: Implementation details of spatial attention modules. Here we denote $a_s^{(in)/(out)}$ and $a_t^{(in)/(out)}$ as input/output spatial and temporal attention respectively, e^C as the embedding of concept C, typically, $e^{(o)/(r)}$ as object/relation embedding. The symbol $\{\}$ represents a list, and symbol $\odot$ represents element wise product. $\tilde{a}_{s/t}$ represents intermediate spatial/temporal attention results. MeanofSameRel **is an average function over the attention of a certain relation.**

Modules & Inputs	Implementation details

`FindObj`
`Rel2[Obj|Subj]`
`Loc2[Obj|Subj]`
$(a_s^{(in)}, a_t^{(in)}, e^C)$
	$a_s^{(out)} = \mathrm{Conv}_2\left(W_2\left(a_s^{(in)} \odot a_t^{(in)} \odot x^v\right) \odot \mathrm{Conv}_1\left(x^v\right) \odot W_1 e^C\right)$

`DetectObj` $(a_s^{(in)}, a_t^{(in)}, \{e^{(o)}\})$	$\{a_s^{(out)}\} = \texttt{FindObj}\left(e^C = \{e^{(o)}\}, a_s^{(in)} = a_s^{(in)}, a_t^{(in)} = a_t^{(in)}\right)$
`DetectAct` $(a_s^{(in)}, a_t^{(in)}, \{e^{(o)}\}, \{e^{(r)}\})$	$\{\tilde{a}_s\} = \texttt{FindObj}\left(e^C = \{e^{(o)}\}, a_s^{(in)} = a_s^{(in)}, a_t^{(in)} = a_t^{(in)}\right)$ $\{a_s^{(out)}\} = \texttt{Rel2Subj}\left(e^C = \{e^{(r)}\}, a_s^{(in)} = \{\tilde{a}_s\}, a_t^{(in)} = a_t^{(in)}\right)$
`DetectRel` $(a_s^{(in)}, a_t^{(in)}, \{e^{(o)}\}, \{e^{(r)}\})$	$\{\tilde{a}_s\} = \texttt{DetectAct}\Big(e^C = \left(\{e^{(o)}\}, \{e^{(r)}\}\right),$ $a_s^{(in)} = a_s^{(in)}, a_t^{(in)} = a_t^{(in)}\Big)$ $\{a_s^{(out)}\} = \mathrm{MeanofSameRel}\left(\{\tilde{a}_s\}\right)$

Table 3: Implementation details of temporal attention modules. We use the same notations as Table 2. Here *first, last, before, after, shorter* and *longer* are corresponding concept embeddings. $\{a\}_i$ represents i^{th} attention in the list, $i = 1, 2$. $e^{(t)/(d)}$ represents temporal order/duration concept embedding.

Modules & Inputs	Implementation details
`Exist` $(a_s^{(in)})$	$a_t^{(out)} = \text{Sigmoid}\left(W_1 a_s^{(in)}\right)$
`ComputeDuration` $(a_s^{(in)})$	$\tilde{a}_t = \texttt{Exist}\left(a_s^{(in)} = a_s^{(in)}\right)$ $duration = \text{sum}(\tilde{a}_t)$
`TemporalLocalize` $(a_s^{(in)}, e^{(t)})$	$\tilde{a}_t = \text{ReLU}\left(\texttt{Exist}\left(a_s^{(in)} = a_s^{(in)}\right)\right)$ $a_t^{(out)} = \begin{cases} \text{SuffixSum}(\tilde{a}_t), & \text{if } d(e^{(t)}, before) < d(e^{(t)}, after) \\ \text{PrefixSum}(\tilde{a}_t), & \text{otherwise} \end{cases}$
`TemporalFilter` $(a_s^{(in)}, e^{(t)})$	$\tilde{a}_t = \texttt{Exist}\left(a_s^{(in)} = a_s^{(in)}\right)$ $a_s^{(out)} = \begin{cases} a_s^{(in)} \odot [\text{Softmax}(\tilde{a}_t - \beta)], & \text{if } d(e^{(t)}, first) > d(e^{(t)}, last) \\ a_s^{(in)} \odot [\text{Softmax}(\tilde{a}_t + \beta)], & \text{otherwise} \end{cases}$
`TemporalBetween` $(\{a_s^{(in)}\})$	$\{\tilde{a}_t^1\} = \texttt{TemporalLocalize}\left(e^C = \{before, after\}, a_s^{(in)} = \{a_s^{(in)}\}_1\right)$ $\{\tilde{a}_t^2\} = \texttt{TemporalLocalize}\left(e^C = \{before, after\}, a_s^{(in)} = \{a_s^{(in)}\}_2\right)$ $a_t^{(out)} = \max\left(\min\left(\{\tilde{a}_t^1\}_1, \{\tilde{a}_t^2\}_2\right), \min\left(\{\tilde{a}_t^1\}_2, \{\tilde{a}_t^2\}_1\right)\right)$
`CompareDuration` $(\{a_s^{(in)}\}, e^{(d)})$	$\{duration\} = \text{Softmax}\left(\texttt{ComputeDuration}\left(a_s^{(in)} = \{a_s^{(in)}\}\right)\right)$ $a_s^{(out)} = \begin{cases} \text{Sum}\left(\{a_s^{(in)}\} \odot \{duration\}\right), \\ \qquad \text{if } d(e^{(d)}, shorter) > d(e^{(d)}, longer) \\ \text{Sum}\left(\{a_s^{(in)}\} \odot \{1 - duration\}\right), \\ \qquad \text{otherwise.} \end{cases}$

concept embedding is generated from the embedding layer inside the sequence-to-sequence model.

(2) **Temporal Attention Modules**: Rather than operating on a static image, in this paper, our modular network is designed for videos that have temporal dimensions, thus we have questions requiring temporal operations, such as the question "*Who grab a bottle after Leonard talked?*" raised in TVQA+ [29] and the question "*After eating some food, did they touch a table or a chair?*" raised in AGQA [11]. In the above questions, word *after* is a temporal order requiring calculation of weight between frames along the temporal dimension to localize actions and objects more precisely afterward.

Table 3 shows a set of temporal modules, the first two of which are denoted as auxiliary modules, serving as reusable sub-functions for other temporal modules. Module `Exist` uses spatial attention maps to infer the existence of concept C (*e.g.*, object, relation, or action) in each frame, outputting temporal attention serving as a temporal probability mask to mask out the frames that concept C does not occur. Module `ComputeDuration` takes spatial attention of action as input and outputs a score representing the duration of the action.

The bottom four modules in Table 3 are temporal attention modules operating along the temporal dimension based on auxiliary modules. Module `TemporalBetween` is designed to localize the time period between the existing periods of two different concepts. Module `TemporalFilter` is designed to get the first or the last existing frame of a certain concept C in the input spatial attention. Module `CompareDuration` takes a set of spatial attention maps representing a set of different concepts C as input, outputting the longest or shortest concept C representation. Module `TemporalLocalize` highlights temporal weights before or after a certain concept C with *before* or *after* as a temporal order parameter.

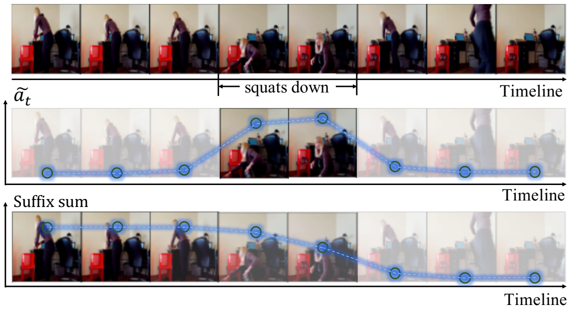

Figure 2: An example for `TemporalLocalize`[*before*], where the module is to localize *"before the woman squats down"*.

Here we take `TemporalLocalize` as an example to illustrate detailed implementation. Given an input spatial attention map $a_s^{(in)}$ representing spatial information of a concept C, and a temporal order concept $C^{(t)}$ (here set as *before*), `TemporalLocalize` [*before*] would first call the auxiliary module `Exist` module to find those frames where concept C exists and output an intermediate temporal attention by $\tilde{a}_t = \texttt{Exist}(a_s^{(in)})$. In order to highlight temporal weights before the concept C, we conduct suffix sum over the temporal weights of the concept C as:

$$a_{t;i}^{(out)} = \text{SuffixSum}(\tilde{a}_t) = \sum_{j=i}^{T} \tilde{a}_{t;j}, \ i = 1, 2, \cdots, T, \tag{2}$$

where i represents the i^{th} frame at temporal attention $a_t^{(out)}$, and j is the j^{th} frame at intermediate temporal attention $\tilde{a}_t$. With this suffix sum operation, $a_{t;i}^{(out)}$ represents the probability that the i^{th} frame is before concept C. Figure 2 shows an example of the `TemporalLocalize`[*before*] computation process.

(3) **Logic Modules**: These modules perform basic logical inference, such as *and* and *or*. They take two one-dimensional vector as inputs and output another one-dimensional vector for possibilities, conducting element-wise logic operations for tensor inputs.

(4) **Answer Modules**: Answer modules serve as top-level modules in the modular neural network and output a one-dimensional score vector for all possible answers. The detailed implementation of all the modules can be found in the appendix.

Most of our designed modules can receive different concepts generated simultaneously with modular layout generation. For example, `LocalizeOrder[before]` and `LocalizeOrder[after]` instantiates the same module `LocalizeOrder` with different concept parameter, where two would localize differently along the temporal dimension. Moreover, some modules may receive more than one concept, *e.g.*, `TemporalBetween` receives two concept information to localize the time period between them.

3.2 Modular Layout Policy

Given a question and a set of predefined modules, we need a modular layout policy to translate the question into a modular layout such that we can organize these modules into a modular neural network. In this section, we describe the policy we adopt to generate the modular layout. For better understanding, we use an example to illustrate the function of our Modular Layout Policy as shown in Figure 1. Given an input question, such as:

"Before sitting in a bed but after tidying up a blanket, which object were man holding?"

We expected the Modular Layout Policy part would generate the parameterized modular layout sequence as:

`Query(Rel2Obj(FindObj(LocalizeBetween(DetectAct(`*sitting in*`, `*bed*`), DetectAct(`*tidying up*`, `*blanket*`)), `*man*`), `*holding*`))`

where `Name(`*param1*`, `*param2*`, ⋯)` is the module `Name` with parameters *param1, param2,⋯*. For example, `FindObj(`*person*`)` is the module `FindObj` and concept parameter *person* for the sub-task of "find the person".

Based on the semantic meaning and logic inside the given question, we construct a tree-structured modular layout, where the leaf modules are usually used for low-level visual perceptions, such as spatial attention modules, while the root modules are usually used to generate answers such as logic modules and answer modules.

We transform the tree-structure layout into a layout sequence using Reverse Polish Notation [5], and learn the layout policy using an attentive sequence-to-sequence Recurrent Neural Network (RNN). Formally, given an input question $\{w_i\}_{i=1}^{T_q}$, where w_i is the i^{th} word in the question and T_q is the question length, our goal is to transform $\{w_i\}_{i=1}^{T_q}$ into the module layout sequence $\{m_i\}_{i=1}^{T_l}$ and its corresponding concept parameter as $\{C_i\}_{i=1}^{T_l}$, where T_l is the layout length. To clarify, **M** in Equation (1) is module m with specific concept parameters.

Firstly, the question is encoded with a RNN encoder as:

$$\mathrm{h}_i^{enc} = \mathrm{RNN}^{enc}\left(e(w_i), \mathrm{h}_{i-1}\right), \quad i = 1, 2, \cdots T_q. \quad \mathrm{h}_0 = \mathbf{0}. \tag{3}$$

Here $e(w_i)$ is the word embedding for w_i. Then we first compute the dynamic attention weight as:

$$u_{ti} = v^T \tanh\left(W_1 h_i^{enc} + W_2 h_{t-1}^{dec}\right), \quad \alpha_{ti} = \frac{\exp(u_{ti})}{\sum_{j=1}^{T}\exp(u_{tj})}, \tag{4}$$

where W_1 and W_2 are learnable parameters. Based on the attention weight, the dynamic context, and hidden state are:

$$c_t = \sum_{i=1}^{T_q} \alpha_t h_i^{enc}, \quad \mathrm{h}_t^{dec} = \mathrm{RNN}^{dec}(\mathrm{h}_{t-1}^{dec}, c_t), \tag{5}$$

where $\mathrm{h}_0^{dec} = 0$ is the initial decoding state, RNN^{dec} is the Recurrent Decoder.

At each decoding time step t, the dynamic context c_t is also used to decode the current module $\hat{m}_t$ and construct current concept parameter C_t as:

$$p(\hat{m}_t|\hat{m}_{<t}; q) = \mathrm{softmax}\left(W_3 c_t + W_4 h_t^{dec}\right), \quad C_t = \sum_{i=1}^{T_q} \alpha_{ti} e(w_i), \tag{6}$$

where W_3 and W_4 are learnable parameters and $\hat{m}_{<t}$ are predicted tokens before timestamp t. In order to ensure the validity of the generated token, where the validity means that the amount of module inputs is no more than the total amount of its children modules outputs, we add masks to the invalid tokens in the probability p.

Finally, let θ be all the parameters in our model, given question q, the probability of a chosen layout $\hat{l} = \{\hat{m}_1, \hat{m}_2, \dots\}$ is given by

$$p_{\mathrm{seq}}(\hat{l}|q;\theta) = \prod_{t=1}^{T_{\hat{l}}} p\left(\hat{m}_t|\hat{m}_{<t}, q\right), \tag{7}$$

where $T_{\hat{l}}$ is length of layout $\hat{l}$. Given $l = \{m_i\}_{i=1}^{T_l}$ as the ground truth layout, the loss L^{seq} on our sequence-to-sequence model can be written as the negative log likelihood of the ground truth l, *i.e.*,

$$L^{\mathrm{seq}}(q, l; \theta) = -\log p_{\mathrm{seq}}(l|q;\theta) = -\sum_{t=1}^{T_l} \log p(m_t|m_{<t}, q). \tag{8}$$

3.3 Modular Neural Network Instantiation and Execution

We have so far designed a set of modules and elaborated on how to generate a tree-structured modular layout for reasoning. In this section, we turn to modular neural network instantiation and execution, where modules in the layout will be assembled and executed in a bottom-up manner.

As we have mentioned in Section 3.2, the output layout from the sequence-to-sequence model is in the Reverse Polish Notation format, which will be used to generate the tree-structured modular layout. Modules are instantiated with concept parameters during the modular neural network instantiation and a tree-structured modular neural network is dynamically assembled based on the tree-structured modular layout. We apply random sampling over answer set for the rare cases when invalid modular neural network are generated. Each module assembled in the valid modular neural network takes the outputs from its children modules as inputs and sends outputs to its parent module as inputs until obtaining a final answer from the top-most module (root module).

Given question q, video v, and learned layout $\hat{l}$ from the sequence-to-sequence part of our model, with model parameters θ, the final answer $\hat{y}$ is generated by:

$$\hat{y} = \arg\max_{y} \mathcal{M}_{\hat{l}}(y|v, q; \theta), \tag{9}$$

where $\mathcal{M}_{\hat{l}} = \mathcal{I}\left(\{\mathbf{m_i}\}, \{C_i\}, \hat{l}\right)$ is the instanced modular neural network with layout l and a set of predefined modules $\{\mathbf{M_i}\}$. Similar to the existing method [13, 21], we draw the loss L^{ans} between the ground truth answer and the predicted answer as:

$$L^{\mathrm{ans}}(q, v, \hat{l}; \theta) = \ell_{\mathrm{ce}}\left(\mathcal{M}_{\hat{l}}(\cdot|v, q; \theta), y^*\right), \tag{10}$$

where ℓ_{ce} is the cross entropy loss and y^* is the ground truth answer for question q on video v.

3.4 Training Strategies

As we have described all the components of our model in detail, we will introduce our training strategies for the entire model in this section. Since our DSTN is an independent structure free from training strategies, we adopt two strategies [13, 31] by varying the way of choosing the layout $\hat{l}$ in Equation (10).

Strategy 1: DSTN-E2E. Following Li et al. [31], strategy 1 chooses the best layout generated from the sequence-to-sequence model. To be specific, given the question q, video v, and ground truth layout l, we first generate the layout $\hat{l}^*$ with the highest probability, *i.e.*,

$$\hat{l}^* = \arg\max_{\hat{l}} p_{\text{seq}}(\hat{l}|q;\theta). \tag{11}$$

Afterward, the total loss could be written as:

$$L^{(1)}(q,v,l;\theta) = L^{\text{seq}}(q,l;\theta) + L^{\text{ans}}(q,v,\hat{l}^*;\theta). \tag{12}$$

In practice, we first use loss L^{seq} in Equation (8) to pre-train our sequence-to-sequence model for modular layout policy merely from questions. While we do not want to fix the parameters to the ground truth layout, we then train the entire model in an end-to-end manner with total loss obtained in Equation (12), from which we also fine-tune the parameters in the policy searching space at the same time when learning the parameters in neural modules.

Strategy 2: DSTN-RL. Following Hu et al. [13], given the question q, video v, the total loss of strategy 2 is the expected loss of L^{ans} in Equation (10) by sampling $\hat{l}$ from $p_{\text{seq}}(\hat{l}|q;\theta)$ in Equation (7), *i.e.*,

$$L^{(2)}(q,v;\theta) = \mathbb{E}_{\hat{l}\sim p_{\text{seq}}(\hat{l}|q;\theta)}\left[L^{\text{ans}}(q,v,\hat{l};\theta)\right]. \tag{13}$$

Due to the non-differentiability of Equation (13), we follow [13] to conduct a Monte-Carlo sampling and Reinforcement-Learning strategy to optimize (13). In addition, optimizing Equation (13) from scratch is challenging and we use the ground truth layout l as an expert policy for the first several epochs, and then switch to the policy gradient learning [41].

4 EXPERIMENTS

We evaluate our model on the recent-proposed AGQA [11] dataset for its real-world dynamic visual reasoning and rich semantic information. AGQA is a large-scale compositional video question answering dataset containing $3.9M$ balanced and $192M$ unbalanced question pairs associated with $9.6K$ videos. The videos in the AGQA involve everyday human activities, while questions in the dataset require a comprehensive understanding of the objects, relations, actions, temporal order, etc., much more challenging than previous datasets [17, 28, 29, 47, 53] and closer to the way human beings think and reason. In this dataset, each video is annotated with a spatio-temporal scene-graph generated from Action Genome [18], providing a structural-semantic representation. Each question is associated with a functional program that points out the necessary reasoning steps to answer the question. Besides, the dataset contains indirect references and novel composition questions, making this dataset more challenging.

The experiments demonstrate the following advantages of our proposed Dynamic Spatio-Temporal Neural Network (DSTN) model. Firstly, our model outperforms the state-of-the-art methods on different metrics and splitting, reflecting the excellent reasoning ability on complex questions and scenarios (Section 4.2). Secondly, our model could provide clear reasoning evidence with semantic meaning at each reasoning step (Section 4.3).

4.1 Implementation and Baselines

Implementation. We use standard video features supplied by AGQA dataset [11], including appearance features $x_a^v \in \mathbb{R}^{8\times 2048}$ extracted from ResNet pool5 layer and motion features $x_m^v \in \mathbb{R}^{8\times 2048}$ extracted from ResNeXt-101. We concatenate appearance features x_a^v with motion features x_m^v to obtain our video feature $x^v \in \mathbb{R}^{8\times 4096}$. Our question features $x^q \in \mathbb{R}^{1000}$ are extracted from the last hidden state of a Bi-LSTM. Moreover, we use the functional programs provided to train initial parameters for sequence-to-sequence model.

Table 4: Overall results on the test dataset. DSTN-E2E and DSTN-RL are the results of two training strategies introduced in Section 3.4. The best results of all methods are highlighted with the bold font and the second with underscore.

Methods	Binary	Open-ended	Overall
PSAC [32]	53.56	32.19	42.44
HME [8]	57.21	36.57	46.47
HCRN [27]	56.01	40.27	47.82
DualVGR [43]	55.48	40.75	47.80
HQGA [45]	56.15	39.49	47.48
DSTN-E2E	**57.38**	42.43	**49.60**
DSTN-RL	56.75	**42.52**	49.34

Our model is trained with the learning rate 2e-5, batch size 32, and the optimizer is Adam optimizer with a weight decay of 1e-5. The experiments are run on the GPU and the total running time is about 120 GPU hours. More details about the space and time complexity of our model and baselines can be found in the appendix.

Baselines. In total, we compare our methods with five state-of-the-art baselines. Besides the baseline methods used in AGQA, we also select two state-of-the-art methods. PSAC [32] and HME [8] propose memory network based methods for important visual and textual features. DualVGR [43] and HQGA[1] [45] adopt graph attention mechanism and encoder-decoder structure for contextual representation. HCRN [27] builds conditional relation network as reusable blocks to construct a hierarchy network to generate contextual multi-modal representation. In order to maintain consistency across different methods, we use the same input visual features for all baseline methods, and rerun the baselines selected by the dataset (HCRN, HME, PSAC) on the latest released version (claimed having the same distribution with the previous version), achieving similar results with those shown in [11].

4.2 Model Performance

We use the official metrics from [11] to evaluate the model performance: (1) **Accuracy Metric**: measures the general model performance for different types of questions. (2) **Indirect Reference Metric**: measures whether the model could figure out the indirect reference of temporal order, actions and objects through precision and recall. (3) **Novel Composition Metric** for Out-of-distribution (OOD) Setting: measures the model's ability to solve dataset shift for novel compositions of word groups. Later in this section, we describe these metrics in detail and analysis quantitative model performance under these metrics.

Accuracy Metric. The quantitative model performances of baselines methods and ours (DSTN-E2E and DSTN-RL) are listed in Table 4. We use standard accuracy metric to evaluate the general performance. Compared with state-of-the-art baselines, DSTN-E2E

[1] We use the HQGA w/o G_O version since object-level features are not provided.

Table 5: Results on indirect reference metrics on the test set, where the metrics are following the definitions in AGQA dataset. Precision values are the accuracy on these indirect questions when the corresponding direct questions were answered correctly, while recall values are the accuracy on all questions with that kind of indirect reference.

Methods	Object		Action		Temporal	
	Precision	Recall	Precision	Recall	Precision	Recall
PSAC [32]	66.67	40.53	66.59	34.50	68.07	36.78
HME [8]	73.86	45.59	78.15	39.51	74.33	41.35
HCRN [27]	81.55	46.39	**86.45**	41.09	84.78	43.25
DualVGR [43]	82.13	46.49	85.81	40.44	85.21	43.56
HQGA [45]	79.51	45.69	81.45	41.38	82.49	42.19
DSTN-E2E	**82.26**	**48.50**	86.34	41.57	**87.47**	**45.76**
DSTN-RL	82.17	48.24	85.12	**42.08**	87.39	45.65

Table 6: Results on novel composition metrics on the novel composition test split. "Seq.", "Sup.", "Dur.", "Obj.", and "Open." denote Sequencing, Superlative, Duration, Obj-rel, and Open-ended separately.

Methods	Seq.	Sup.	Dur.	Obj.	Binary	Open.	Overall
PSAC [32]	36.18	30.51	36.83	20.21	40.63	15.66	31.30
HME [8]	43.22	39.22	45.82	24.17	49.34	20.91	38.72
HCRN [27]	43.94	37.61	50.27	24.87	46.29	25.56	38.55
DualVGR [43]	44.71	36.92	51.47	**28.67**	46.09	**27.17**	39.02
HQGA [45]	43.35	35.93	51.32	24.14	44.90	25.27	37.57
DSTN-E2E	**45.63**	39.62	52.50	25.04	49.42	24.77	40.21
DSTN-RL	45.36	**40.62**	**53.94**	26.14	**50.40**	24.44	**40.70**

and DSTN-RL gain 1.78% and 1.52% absolute improvement over the best baseline model overall, respectively.

Baseline models could perform relatively well in certain question types, while our model outperforms all the baselines in both question types (*i.e.*, binary and open-ended questions), showing the reasoning ability on complex questions and scenarios of our designed model.

Indirect Reference Metric. Indirect reference metric is aimed at measuring the model’s ability to understand objects, actions, and temporal semantic meanings in indirect reference formats. In short, the model has to distinguish the detailed concepts before it can reason on it. We denote questions containing indirect references as indirect questions (e.g., *Did they contact the object they were watching?*) and their corresponding direct questions replace the indirect references with certain entities (e.g., *Did they contact a television?*). In Table 5, we report the precision and recall for each concept type as a measurement of indirect reference similar to [11]. Precision values are the accuracy on these indirect questions when the corresponding direct questions were answered correctly, while recall values are the accuracy on all questions with that kind of indirect reference.

In Table 5, DSTN-E2E and DSTN-RL significantly outperform baselines in object indirect reference questions and temporal indirect reference questions. Although HCRN achieves the highest precision score in action indirect reference, it does not achieve the corresponding highest recall value, meanwhile, DSTN achieves a competitive precision score in action indirect reference, which means DSTN has a good performance in general action indirect reference questions.

Table 7: Temporal module performances.

Module	Metrics	Baseline	DSTN-E2E	DSTN-RL
`TemporalFilter`	Kendall’s τ	0.001	0.221	0.330
`TemporalLocalize`	IOU	0.363	0.578	0.597
`TemporalBetween`	IOU	0.237	0.460	0.458
`CompareDuration`	IOU	0.296	0.447	0.440

Novel Composition Metric for OOD setting. Novel composition metric for OOD setting measures the model’s ability to answer an Out-Of-Distribution (OOD) question: the model is trained on the questions with certain compositions but tested on the questions with unseen novel compositions [40]. For example, the word “first” and “behind” do not co-occur in any question during the training process, while they do co-occur in a question during testing. As shown in Table 6, DSTN-E2E and DSTN-RL outperform all the baselines in the overall accuracy and achieve the highest two accuracies in most cases, demonstrating DSTN has a strong generalization ability.

Discussions about DSTN-RL and DSTN-E2E. We could find that: (1) The E2E strategy performs better in the i.i.d. setting (independent and identically distributed) where the training and testing data follow the same distribution (Tables 4, 5, and 7) because the E2E strategy tends to imitate the training data. (2) The RL strategy works better in the o.o.d. setting (out-of-distribution) where the training and testing data may follow different distributions (Table 6) because the RL strategy would freely explore these unseen layouts and thus works better.

4.3 Module Evaluation

As a modular neural network based model, our DSTN model has good explainability aside from good performance. In this section, we evaluate intermediate results of temporal modules with the ground truth to verify the rationality of our designed modules. Moreover, we visualize the spatial attention map and temporal attention map of several showcases to further demonstrate the explainability of our DSTN model.

Temporal Module Evaluation. We firstly evaluate the consistency between our temporal modules and the ground truth. For each temporal module, we first extract the ground truth temporal region from the AGQA annotation, and then we compute the consistency between our module prediction and the ground truth. We use Kendall’s τ coefficient metric [23] to measure the consistency between the output of the `TemporalFilter` module and the ground truth. Besides, we use Intersection-over-union (IOU) metric [16] to measure the consistency between predicted region and the ground truth region for `TemporalLocalize`, `TemporalBetween`, and `CompareDuration` modules. As there are not other available baselines, we implement a baseline that randomly generates predictions to simulate the function of the module. Take the baseline for `TemporalBetween` as an example. We randomly select two frames as the start and end frames of the localized time period. More details about the implementation of the baselines can be found in the appendix. The results are shown in Table 7.

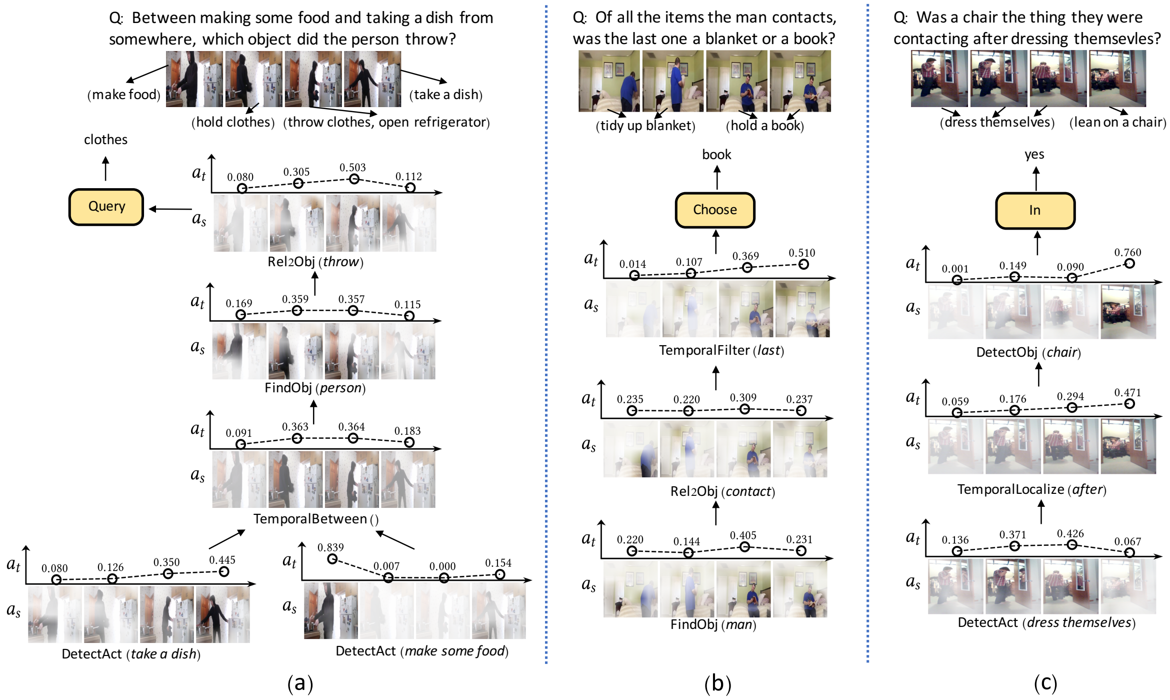

Figure 3: Real examples visualizing the reasoning process of our DSTN model. The temporal attention map a_t and spatial attention map a_s of intermediate modules are depicted. The upside part of the module is the temporal attention map a_t plotted as a curve line with weights, and the bottom part is the spatial attention map with highlighted localized concepts. The top modules (depicted as yellow rectangles) are answer modules. For clarification, we annotate actions contained in each frame.

(1) `TemporalFilter` aims to localize the first or last concepts (actions, objects etc.). Given ground truth temporal attention a_t^* and output temporal a_t, we compute the Kendall's τ coefficient [23] between our localized results and ground truth. Compared with the baseline, our `TemporalFilter` module has a remarkably high Kendall's τ coefficient, which shows the rationality of this model.
(2) `TemporalLocalize`, `TemporalBetween`, and `CompareDuration` aim to find a corresponding temporal region for the given concept. We measure the performance of these modules with IOU between the predicted temporal region and ground truth region. Compared with the baseline, our method has a much higher IOU with ground truth, meaning that these modules indeed learn the desired mapping during the implicit training process, demonstrating the reliability and explainability of our method.

Visualization We visualize the intermediate results of our designed modules as shown in Figure 3. For temporal modules that output temporal attention maps, we use the calculated temporal attention maps to determine the importance of each frame. For other modules that output spatial attention maps, we adopt the Grad-Cam method [38] to produce a coarse localization map highlighting the important regions. Specifically, we first calculate the inner product between the visual features and the spatial attention maps outputted by each module. Then we evaluate its gradient on the final convolutional layer of the visual feature extraction CNN. Finally, a bi-linear interpolation step is adopted to obtain a rough pixel-level saliency map.

From Figure 3, both our temporal modules and spatial modules can localize given concepts precisely, *e.g.*, `DetectAct`[*take a dish*] and `DetectAct`[*make some food*] (*cf.*, Figure 3a) highlight the related actions in the related frame, `TemporalFilter`[*last*] module (*cf.*, Figure 3b) picks last object out from several objects, `TemporalLocalize`[*before*] module (*cf.*, Figure 3c) highlights the frames after the action. As a result, we verify that each part (module) of our model is correctly operating based on the human logic rather than a black-box deep structure.

5 CONCLUSION

In summary, we propose the dynamic spatio-temporal modular network (DSTN) model, which is the first modular neural network based approach in VideoQA for explainable video reasoning in real-world scenarios. Moreover, we conduct extensive experiments to demonstrate the advantages of DSTN with various metrics and settings, and explore the performances of different modules to demonstrate the rationality of modules and the explainability of the overall model.

ACKNOWLEDGMENT

This work is supported by the National Key Research and Development Program of China No. 2020AAA0106300 and National Natural Science Foundation of China No. 62250008, No. 62102222.

REFERENCES

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *arXiv preprint arXiv:2204.14198*, 2022.

[2] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Learning to compose neural networks for question answering. *arXiv preprint arXiv:1601.01705*, 2016.

[3] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Neural module networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 39–48, 2016.

[4] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015.

[5] Arthur W Burks, Don W Warren, and Jesse B Wright. An analysis of a logical machine using parenthesis-free notation. *Mathematical tables and other aids to computation*, 8(46):53–57, 1954.

[6] Wenhu Chen, Zhe Gan, Linjie Li, Yu Cheng, William Wang, and Jingjing Liu. Meta module network for compositional visual reasoning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 655–664, 2021.

[7] Xuguang Duan, Wenbing Huang, Chuang Gan, Jingdong Wang, Wenwu Zhu, and Junzhou Huang. Weakly supervised dense event captioning in videos. *Advances in Neural Information Processing Systems*, 31, 2018.

[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.

[9] Tsu-Jui Fu, Linjie Li, Zhe Gan, Kevin Lin, William Yang Wang, Lijuan Wang, and Zicheng Liu. Violet: End-to-end video-language transformers with masked visual-token modeling. *arXiv preprint arXiv:2111.12681*, 2021.

[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.

[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.

[12] Mao Gu, Zhou Zhao, Weike Jin, Richang Hong, and Fei Wu. Graph-based multi-interaction network for video question answering. *IEEE Transactions on Image Processing*, 30:2758–2770, 2021.

[13] Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko. Learning to reason: End-to-end module networks for visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 804–813, 2017.

[14] Ronghang Hu, Jacob Andreas, Trevor Darrell, and Kate Saenko. Explainable neural computation via stack neural module networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 53–69, 2018.

[15] Deng Huang, Peihao Chen, Runhao Zeng, Qing Du, Mingkui Tan, and Chuang Gan. Location-aware graph convolutional networks for video question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11021–11028, 2020.

[16] Paul Jaccard. The distribution of the flora in the alpine zone. 1. *New phytologist*, 11(2):37–50, 1912.

[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*, 2017.

[18] Jingwei Ji, Ranjay Krishna, Li Fei-Fei, and Juan Carlos Niebles. Action genome: Actions as compositions of spatio-temporal scene graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10236–10247, 2020.

[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.

[20] Weike Jin, Zhou Zhao, Xiaochun Cao, Jieming Zhu, Xiuqiang He, and Yueting Zhuang. Adaptive spatio-temporal graph enhanced vision-language representation for video qa. *IEEE Transactions on Image Processing*, 30:5477–5489, 2021.

[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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2901–2910, 2017.

[22] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Judy Hoffman, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Inferring and executing programs for visual reasoning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2989–2998, 2017.

[23] Maurice G Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2): 81–93, 1938.

[24] Junyeong Kim, Minuk Ma, Kyungsu Kim, Sungjin Kim, and Chang D Yoo. Progressive attention memory network for movie story question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8337–8346, 2019.

[25] Kyung-Min Kim, Min-Oh Heo, Seong-Ho Choi, and Byoung-Tak Zhang. Deepstory: Video story qa by deep embedded memory networks. *arXiv preprint arXiv:1707.00836*, 2017.

[26] Thao Minh Le, Vuong Le, Svetha Venkatesh, and Truyen Tran. Learning to reason with relational video representation for question answering. *arXiv preprint arXiv:1907.04553*, 2, 2019.

[27] Thao Minh Le, Vuong Le, Svetha Venkatesh, and Truyen Tran. Hierarchical conditional relation networks for video question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9972–9981, 2020.

[28] Jie Lei, Licheng Yu, Mohit Bansal, and Tamara L Berg. Tvqa: Localized, compositional video question answering. *arXiv preprint arXiv:1809.01696*, 2018.

[29] Jie Lei, Licheng Yu, Tamara L Berg, and Mohit Bansal. Tvqa+: Spatio-temporal grounding for video question answering. *arXiv preprint arXiv:1904.11574*, 2019.

[30] 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.

[31] Guohao Li, Xin Wang, and Wenwu Zhu. Perceptual visual reasoning with knowledge propagation. In *Proceedings of the 27th acm international conference on multimedia*, pages 530–538, 2019.

[32] 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.

[33] 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.

[34] David Mascharka, Philip Tran, Ryan Soklaski, and Arjun Majumdar. Transparency by design: Closing the gap between performance and interpretability in visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4942–4950, 2018.

[35] Seil Na, Sangho Lee, Jisung Kim, and Gunhee Kim. A read-write memory network for movie story understanding. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 677–685, 2017.

[36] Jungin Park, Jiyoung Lee, and Kwanghoon Sohn. Bridge to answer: Structure-aware graph interaction network for video question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15526–15535, 2021.

[37] 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.

[38] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[39] Ahjeong Seo, Gi-Cheon Kang, Joonhan Park, and Byoung-Tak Zhang. Attend what you need: Motion-appearance synergistic networks for video question answering. *arXiv preprint arXiv:2106.10446*, 2021.

[40] Zheyan Shen, Jiashuo Liu, Yue He, Xingxuan Zhang, Renzhe Xu, Han Yu, and Peng Cui. Towards out-of-distribution generalization: A survey. *arXiv preprint arXiv:2108.13624*, 2021.

[41] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[42] Hui Wang, Dan Guo, Xian-Sheng Hua, and Meng Wang. Pairwise vlad interaction network for video question answering. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 5119–5127, 2021.

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

[44] 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 Datasets and Benchmarks Track (Round 2)*, 2021.

[45] 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.

[46] 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.

[47] Li Xu, He Huang, and Jun Liu. Sutd-trafficqa: A question answering benchmark and an efficient network for video reasoning over traffic events. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages

9878–9888, 2021.

[48] Hongyang Xue, Zhou Zhao, and Deng Cai. Unifying the video and question attentions for open-ended video question answering. *IEEE Transactions on Image Processing*, 26(12):5656–5666, 2017.

[49] Hongyang Xue, Wenqing Chu, Zhou Zhao, and Deng Cai. A better way to attend: Attention with trees for video question answering. *IEEE Transactions on Image Processing*, 27(11):5563–5574, 2018.

[50] Kexin Yi, Chuang Gan, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B Tenenbaum. Clevrer: Collision events for video representation and reasoning. *arXiv preprint arXiv:1910.01442*, 2019.

[51] 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*, 34:23634–23651, 2021.

[52] 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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16375–16387, 2022.

[53] Zhou Zhao, Qifan Yang, Deng Cai, Xiaofei He, Yueting Zhuang, Zhou Zhao, Qifan Yang, Deng Cai, Xiaofei He, and Yueting Zhuang. Video question answering via hierarchical spatio-temporal attention networks. In *IJCAI*, volume 2, 2017.

[54] Zhou Zhao, Zhu Zhang, Shuwen Xiao, Zhou Yu, Jun Yu, Deng Cai, Fei Wu, and Yueting Zhuang. Open-ended long-form video question answering via adaptive hierarchical reinforced networks. In *IJCAI*, volume 2, page 8, 2018.

A MORE EXPERIMENTAL RESULTS

A.1 Baselines in Temporal Module Evaluation

We implement baselines that randomly generate predictions to simulate the function of the module for temporal module evaluation:
(1) **`TemporalFilter` Module** uses Kendall’s τ coefficient as the metric. Therefore we randomly generate an index set as our baseline.
(2) **`TemporalLocalize` Module** uses Intersection-over-union (IOU) as the metric. As shown in Figure 4(a), we randomly select an index a as the boundary and a direction d, then we highlight the frames guided by the boundary and the direction.
(3) **`Temporalbetween` Module** uses Intersection-over-union (IOU) as the metric. As shown in Figure 4(b), we randomly select two indexes a and b as the boundary, then we highlight the frames between the two indexes.
(4) **`CompareDuration` Module** uses Intersection-over-union (IOU) as the metric. We randomly highlight or mask each frame for our baseline.

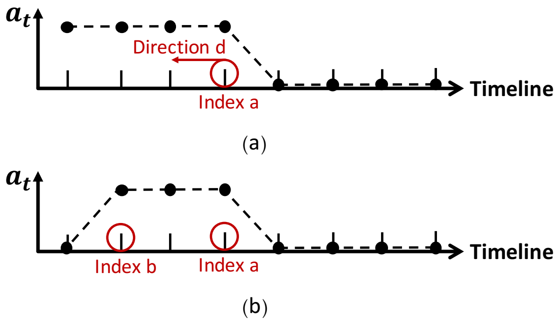

Figure 4: The generation of baselines. a_t represents the temporal attention map. Figure 1a is the baseline of `TemporalLocalize` module, and Figure 1b is the baseline of `TemporalBetween` module.

A.2 Space and Time Complexity

We have listed our model’s training cost and other training details in Section 4.1 of the main paper. To further demonstrate our model’s time and space complexity, we calculate the number of parameters and inference time of our model and other baselines. In order to maintain consistency across different methods, we run the experiments on the same machine (Intel(R) Xeon(R) Gold 6240 CPU, Nvidia GTX 3090 GPU) with batch size 32. We use the balanced test split in AGQA for inference, and the number of QA pairs is 1,041,600. As listed in Table 8, we can find as follows:

(i) The numbers of parameters of all the models are of the same order of magnitude.

(ii) The inference time results can be categorized into three classes: (1) HQGA, which needs less than 1 hour. (2) DSTN (ours), DualVGR, HCRN, and PSAC all need a couple of hours. (3) HME, which needs significantly more time than other baselines.

To conclude, splitting the reasoning process of a problem into multiple sub-tasks does not lead to a significant increase in time or space complexity compared with other state-of-the-art baselines.

B IMPLEMENTATION DETAILS OF MODULES

We design four types of modules: spatial attention modules, temporal attention modules, logic modules, and answer models. Table 9 lists all the notations we use, and then Table 10 lists implementation details of all modules.

Table 8: Space and time complexity of models.

Method	Number of parameters (M)	Inference time (hours)	Accuracy
PSAC	39.34	1.3	42.44
HME	42.89	32.5	46.47
HCRN	41.46	1.7	47.82
DualVGR	14.24	1.2	47.80
HQGA	11.79	0.7	47.48
DSTN (ours)	36.87	4.0	49.60

Table 9: Notations and corresponding meanings.

Notation	Meaning
$a_t^{(in)}$	input temporal attention map
$a_s^{(in)}$	input spatial attention map
$a_t^{(out)}$	output temporal attention map
$a_s^{(out)}$	output spatial attention map
e^C	embedding of concept C
$e^{(o)}$	embedding of object concept
$e^{(r)}$	embedding of relation concept
$e^{(t)}$	embedding of temporal order concept
$e^{(d)}$	embedding of duration concept
ans	an one-dimensional score vector for all possible answers
$ans^{(in)}$	input ans
$ans^{(out)}$	output ans
$\{\}$	a list, $\{a\}$ represents a list of a
$\tilde{a}$	intermediate attention map results
x^v	visual features
Conv	convolution operator
W	weight matrix
$\odot$	element-wise product operator
MeanofSameRel	for each relation, calculate the average of related attention maps
Sigmoid	Sigmoid activation function
ReLU	ReLU activation function
PrefixSum	$\text{PrefixSum}(x) = y = \{y_i = \sum_{j=1}^{i} x_j\}$, where y_i and x_j are the i^{th} and j^{th} number in y and x, respectively
SuffixSum	$\text{SuffixSum}(x) = y = \{y_i = \sum_{j=i}^{T} x_j\}$, where y_i and x_j are the i^{th} and j^{th} number in y and x respectively. T is the length of x
Softmax	Softmax function
$d(x, y)$	distance between x and y using inner product

Table 10: The list of modules in our model. The modules are categorized into five categories (including Auxiliary modules) and implemented by different functions. Meanings of each notation are listed in Table 9.

Module Type	Module Name	Inputs	Outputs	Implementation Details

Spatial Attention	`FindObj`
	`Rel2[Obj|Subj]`
	`Loc2[Obj|Subj]`
		$e^C, a_s^{(in)}, a_t^{(in)}$	$a_s^{(out)}$	$a_s^{(out)} = \text{Conv}_2\left(W_2(a_s^{(in)} \odot a_t^{(in)} \odot x^v) \odot \text{Conv}_1(x^v) \odot W_1 e^C\right)$

	`DetectObj`	$\{e^{(o)}\}, a_s^{(in)}, a_t^{(in)}$	$\{a_s^{(out)}\}$	$\{a_s^{(out)}\} = \texttt{FindObj}\left(e^C = \{e^{(o)}\}, a_s^{(in)} = a_s^{(in)}, a_t^{(in)} = a_t^{(in)}\right)$
	`DetectAct`	$\{e^{(o)}\}, \{e^{(r)}\}, a_s^{(in)}, a_t^{(in)}$	$\{a_s^{(out)}\}$	$\{\tilde{a}_s\} = \texttt{FindObj}\left(e^C = \{e^{(o)}\}, a_s^{(in)} = a_s^{(in)}, a_t^{(in)} = a_t^{(in)}\right)$ $\{a_s^{(out)}\} = \texttt{Rel2Subj}\left(e^C = \{e^{(r)}\}, a_s^{(in)} = \{\tilde{a}_s\}, a_t^{(in)} = a_t^{(in)}\right)$
	`DetectRel`	$\{e^{(o)}\}, \{e^{(r)}\}, a_s^{(in)}, a_t^{(in)}$	$\{a_s^{(out)}\}$	$\{\tilde{a}_s\} = \texttt{DetectAct}\left(e^C = \left(\{e^{(o)}\}, \{e^{(r)}\}\right), a_s^{(in)} = a_s^{(in)}, a_t^{(in)} = a_t^{(in)}\right)$ $\{a_s^{(out)}\} = \text{MeanofSameRel}\left(\{\tilde{a}_s\}\right)$
Auxiliary	`Exist`	$a_s^{(in)}$	$a_t^{(out)}$	$a_t^{(out)} = \text{Sigmoid}\left(W_1 a_s^{(in)}\right)$
	`ComputeDuration`	$a_s^{(in)}$	duration	$\tilde{a}_t = \texttt{Exist}\left(a_s^{(in)} = a_s^{(in)}\right)$ duration = sum($\tilde{a}_t$)
Temporal Attention	`TemporalLocalize`	$e^{(t)}, a_s^{(in)}$	$a_t^{(out)}$	$\tilde{a}_t = \text{ReLU}\left(\texttt{Exist}\left(a_s^{(in)} = a_s^{(in)}\right)\right)$ $a_t^{(out)} = \begin{cases} \text{SuffixSum}(\tilde{a}_t), & \text{if } d(e^{(t)}, before) < d(e^{(t)}, after) \\ \text{PrefixSum}(\tilde{a}_t), & \text{otherwise} \end{cases}$
	`TemporalFilter`	$e^{(t)}, a_s^{(in)}$	$a_s^{(out)}$	$\tilde{a}_t = \texttt{Exist}\left(a_s^{(in)} = a_s^{(in)}\right)$ $a_s^{(out)} = \begin{cases} a_s^{(in)} \odot [\text{Softmax}(\tilde{a}_t - \beta)], & \text{if } d(e^{(t)}, first) > d(e^{(t)}, last) \\ a_s^{(in)} \odot [\text{Softmax}(\tilde{a}_t + \beta)], & \text{otherwise} \end{cases}$
	`TemporalBetween`	$\{a_s^{(in)}\}$	$a_t^{(out)}$	$\{\tilde{a}_t^1\} = \texttt{TemporalLocalize}\left(e^{(t)} = \{before, after\}, a_s^{(in)} = \{a_s^{(in)}\}_1\right)$ $\{\tilde{a}_t^2\} = \texttt{TemporalLocalize}\left(e^{(t)} = \{before, after\}, a_s^{(in)} = \{a_s^{(in)}\}_2\right)$ $a_t^{(out)} = \max\left(\min\left(\{\tilde{a}_t^1\}_1, \{\tilde{a}_t^2\}_2\right), \min\left(\{\tilde{a}_t^1\}_2, \{\tilde{a}_t^2\}_1\right)\right)$
	`CompareDuration`	$e^{(d)}, \{a_s^{(in)}\}$	$a_s^{(out)}$	$\{duration\} = \text{Softmax}\left(\texttt{ComputeDuration}\left(a_s^{(in)} = \{a_s^{(in)}\}\right)\right)$ $a_s^{(out)} = \begin{cases} \text{Sum}\left(\{a_s^{(in)}\} \odot \{duration\}\right), \text{if } d(e^{(d)}, shorter) > d(e^{(d)}, longer) \\ \text{Sum}\left(\{a_s^{(in)}\} \odot \{1 - duration\}\right), \text{otherwise.} \end{cases}$
Logic	`And`	$ans_1^{(in)}, ans_2^{(in)}$	$ans^{(out)}$	$sim = \text{Sigmoid}(ans_1^{(in)}) \odot \text{Sigmoid}(ans_2^{(in)})$ $ans^{(out)} = \text{padding}(sim, 1 - sim)$
	`Xor`	$ans_1^{(in)}, ans_2^{(in)}$	$ans^{(out)}$	$sim = \text{Sigmoid}(ans_1^{(in)}) \odot (1 - \text{Sigmoid}(ans_2^{(in)})) + (1 - \text{Sigmoid}(ans_1^{(in)})) \odot \text{Sigmoid}(ans_2^{(in)})$ $ans^{(out)} = \text{padding}(sim, 1 - sim)$
Answer	`Query`	$a_s^{(in)}, a_t^{(in)}$	$ans^{(out)}$	$ans^{(out)} = \max(W_y^T(W(a_s^{(in)} \odot a_t^{(in)} \odot x^v)))$
	`QueryCompare`	$ans_1^{(in)}, ans_2^{(in)}$	$ans^{(out)}$	$ans^{(out)} = \text{padding}(ans_1^{(in)}, ans_2^{(in)})$
	`Choose`	$e_1^C, e_2^C, a_s^{(in)}, a_t^{(in)}$	$ans^{(out)}$	$ans^{(out)} = W_y^T\left(d\left(W_1 e_1^C, W_3(a_s^{(in)} \odot a_t^{(in)} \odot x^v)\right) \odot W_1 e_1^C + d\left(W_2 e_2^C, W_3(a_s^{(in)} \odot a_t^{(in)} \odot x^v)\right) \odot W_2 e_2^C\right)$
	`In`	$e^C, a_s^{(in)}$	$ans^{(out)}$	$ans^{(out)} = \text{padding}\left(d(W_1 e^C, W_2(a_s^{(in)} \odot x^v)), -d(W_1 e^C, W_2(a_s^{(in)} \odot x^v))\right)$