The Thirty-Seventh AAAI Conference on Artificial Intelligence (AAAI-23)

Efficient End-to-End Video Question Answering with Pyramidal Multimodal Transformer

Min Peng[1,2*], **Chongyang Wang**[3*], **Yu Shi**[1], **Xiang-Dong Zhou**[1]

[1]Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences
[2]University of Chinese Academy of Sciences
[3]Tsinghua University
{pengmin, shiyu, zhouxiangdong}@cigit.ac.cn, mvrjustid@gmail.com

Abstract

This paper presents a new method for end-to-end Video Question Answering (VideoQA), aside from the current popularity of using large-scale pre-training with huge feature extractors. We achieve this with a pyramidal multimodal transformer (PMT) model, which simply incorporates a learnable word embedding layer, a few convolutional and transformer layers. We use the anisotropic pyramid to fulfill video-language interactions across different spatio-temporal scales. In addition to the canonical pyramid, which includes both bottom-up and top-down pathways with lateral connections, novel strategies are proposed to decompose the visual feature stream into spatial and temporal sub-streams at different scales and implement their interactions with the linguistic semantics while preserving the integrity of local and global semantics. We demonstrate better or on-par performances with high computational efficiency against state-of-the-art methods on five VideoQA benchmarks. Our ablation study shows the scalability of our model that achieves competitive results for text-to-video retrieval by leveraging feature extractors with reusable pre-trained weights, and also the effectiveness of the pyramid. Code available at: https://github.com/Trunpm/PMT-AAAI23.

Introduction

Vision-language understanding is basic for a machine to interact with our multimodal reality. The previous success seen in computer vision and natural language processing impacted the ongoing research for a variety of vision-language understanding tasks, e.g., Visual Question Answering (VQA) (Antol et al. 2015; Yu et al. 2019) and text-to-vision retrieval (Yu, Kim, and Kim 2018; Bain et al. 2021). Of our particular interests, Video Question Answering (VideoQA) is quite challenging, which requires accurate semantics reasoning from local-spatio regions to global-temporal dynamics of the video. This point is continuously verified in recent studies, where successes are achieved by methods that are able to capture such a multiscale property considering spatial regions of a specific frame (Xue et al. 2022) or temporal saliency across different frames (Peng et al. 2022; Xiao et al. 2022). It is time to build a VideoQA model that incorporates the learning of multiscale spatial and temporal features, preferably in an end-to-end manner.

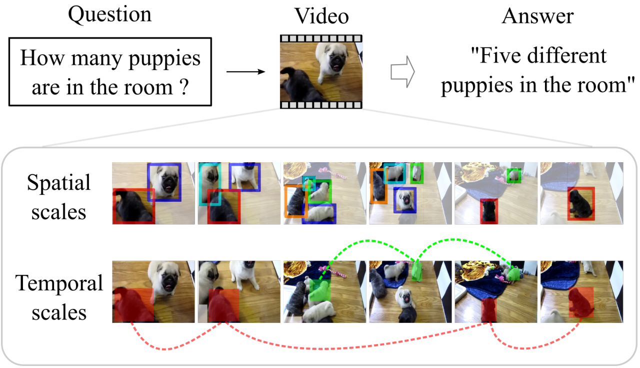
Question
How many puppies are in the room ?
Video
Answer
"Five different puppies in the room"
Spatial scales
Temporal scales

Figure 1: The spatio-temporal multiscale property of VideoQA. While features of local regions (marked by anchors) contribute to the recognition of *objects* per frame, their dynamics described by frames at different temporal locations (marked by anchors connected with dashed lines) help sort out their *relations* or understand the *events*.

Generally, VideoQA includes the processes of feature extraction from each modality and the Video-Language (V-L) interaction for output. Many studies (Xu et al. 2017; Jang et al. 2017; Seo, Nagrani, and Schmid 2021; Yang et al. 2021) isolate these two processes, using fixed Convolutional Neural Network (CNN) (He et al. 2016; Tran et al. 2015; Hara, Kataoka, and Satoh 2018; Xie et al. 2018) for visual feature extraction and Recurrent Neural Network (RNN) (Hochreiter and Schmidhuber 1997) or BERT (Kenton and Toutanova 2019) for language embedding before the V-L interaction. We agree with (Lei et al. 2021) that such suboptimal features acquired in an offline manner may not promise good fitting with the downstream multimodal tasks. For end-to-end VideoQA, more recent efforts (Yu et al. 2021; Li et al. 2022; Xue et al. 2022) are paid to pre-training transformers (Vaswani et al. 2017) on large-scale datasets comprising vision-language pairs, e.g., COCO Captions (Chen et al. 2015), HowTo100M (Miech et al. 2019), and YT-Temporal-180M (Zellers et al. 2021), and transferring knowledge to specific tasks. Despite their promising results, since powerful computational resources are still costly to access, we ask if via model engineering accurate end-to-end VideoQA can be achieved without using large-scale pre-training and huge

[*]These authors contributed equally.

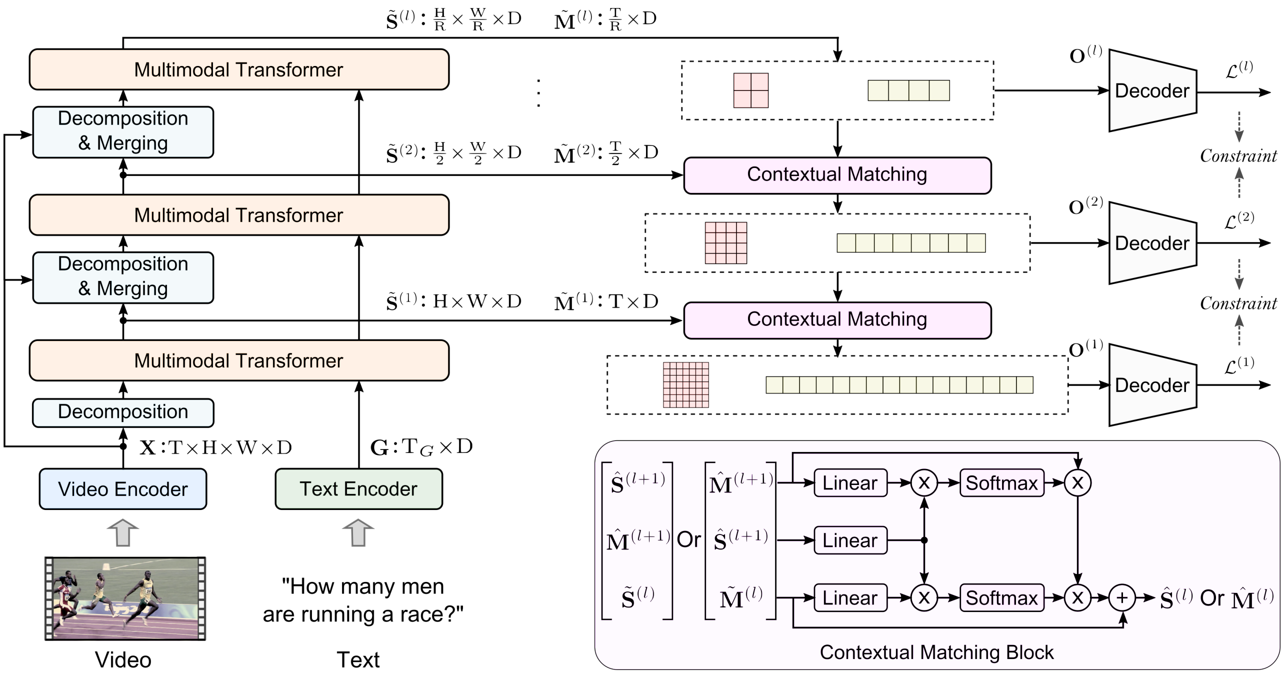
$\tilde{\mathbf{S}}^{(l)}: \frac{\mathrm{H}}{\mathrm{R}} \times \frac{\mathrm{W}}{\mathrm{R}} \times \mathrm{D}$ $\tilde{\mathbf{M}}^{(l)}: \frac{\mathrm{T}}{\mathrm{R}} \times \mathrm{D}$
Multimodal Transformer
Decomposition & Merging
$\tilde{\mathbf{S}}^{(2)}: \frac{\mathrm{H}}{2} \times \frac{\mathrm{W}}{2} \times \mathrm{D}$ $\tilde{\mathbf{M}}^{(2)}: \frac{\mathrm{T}}{2} \times \mathrm{D}$
Multimodal Transformer
Decomposition & Merging
$\tilde{\mathbf{S}}^{(1)}: \mathrm{H} \times \mathrm{W} \times \mathrm{D}$ $\tilde{\mathbf{M}}^{(1)}: \mathrm{T} \times \mathrm{D}$
Multimodal Transformer
Decomposition
$\mathbf{X}: \mathrm{T} \times \mathrm{H} \times \mathrm{W} \times \mathrm{D}$
$\mathbf{G}: \mathrm{T}_G \times \mathrm{D}$
Video Encoder
Text Encoder
"How many men are running a race?"
Video
Text
$\mathbf{O}^{(l)}$
Decoder
$\mathcal{L}^{(l)}$
Constraint
Contextual Matching
$\mathbf{O}^{(2)}$
Decoder
$\mathcal{L}^{(2)}$
Constraint
Contextual Matching
$\mathbf{O}^{(1)}$
Decoder
$\mathcal{L}^{(1)}$
$\hat{\mathbf{S}}^{(l+1)}$ $\hat{\mathbf{M}}^{(l+1)}$ $\tilde{\mathbf{S}}^{(l)}$
Or
$\hat{\mathbf{M}}^{(l+1)}$ $\hat{\mathbf{S}}^{(l+1)}$ $\tilde{\mathbf{M}}^{(l)}$
Linear
Softmax
Linear
Linear
Softmax
$\hat{\mathbf{S}}^{(l)}$ Or $\hat{\mathbf{M}}^{(l)}$
Contextual Matching Block

Figure 2: The PMT model. Bottom-up pathway: the video and text encoders take as input raw video frames and text tokens, respectively. The visual feature $\mathbf{X}$ is decomposed into spatial $\mathbf{S}$ and temporal $\mathbf{M}$ sub-streams, and pooled to reveal their multiscale contents for multimodal learning with linguistic semantics $\mathbf{G}$. Top-down pathway: while separate losses that controlled by a constraint function are used to maintain the integrity of local and global representations, the contextual matching block therein fuses spatio-temporal information at different scales.

feature extractors. We refer to such as ***efficient*** end-to-end VideoQA. We would like to also note that, as we recognize the paradigm of learning from large amount of available data and generalizing to various downstream tasks as the promising route for end-to-end VideoQA, we believe this paper provides new supervised baselines and insights, informative for the future development of VideoQA and possibly V-L understanding.

Targeting the above findings, we achieve efficient end-to-end VideoQA by building a more effective V-L interaction architecture. We deem the anisotropic pyramid structure as a promising candidate. The idea of pyramid is witnessed to help acquire multiscale spatial features for object detection (Lin et al. 2017; Wang et al. 2021b) and multiscale temporal features for dynamic scene classification (Huang et al. 2019), action recognition(Yang et al. 2020), and video grounding (Li, Guo, and Wang 2021) etc. Thereon, for the first time, we aim to incorporate the learning of multiscale spatial as well as temporal features, and leverage such spatio-temporal contexts to build the V-L interaction. As such, we establish multiscale, contextual, and spatio-temporal V-L interactions within a single model, essential for end-to-end VideoQA.

While our evaluations on five VideoQA benchmarks demonstrate better or on-par performances against state-of-the-art methods, the computational complexity and cost of our method remain small and manageable. By simply leveraging feature extractors with reusable pre-trained weights loaded, our model is further improved and achieves competitive results for text-to-video retrieval. Our technical contributions are three-fold: (1) we propose a Pyramidal Multimodal Transformer (PMT) model for efficient end-to-end VideoQA, which works without using large-scale pre-training and huge feature extractors; (2) we propose a decomposition method within the pyramid to enable the learning of spatio-temporal features and their interactions with linguistic semantics at multiple scales; (3) we propose a contextual matching method to fuse spatio-temporal information of different scales, and a constraint function to maintain the integrity of local and global representations.

Method

This section provides details of the video and text encoders, the V-L information passing and interactions within our PMT model, and the loss computations. By default, large-scale pre-training and huge feature extractors are not used. Additionally, the model is designed to be computational-efficient, with a comparably small number of trainable parameters. An overview of our PMT model is shown in Fig.2.

Video Encoder

Earlier studies usually use C3D (Tran et al. 2015), ResNet (He et al. 2016; Hara, Kataoka, and Satoh 2018), and S3D (Xie et al. 2018) for the spatio-temporal feature extraction from densely-sampled (Le et al. 2020; Park, Lee, and Sohn 2021) or multiscale-sampled video frames (Peng et al.

2022). Some also use Faster R-CNN (Ren et al. 2016) to help acquire object-relevant features. While the recent end-to-end methods (Zellers et al. 2021; Li et al. 2022; Xue et al. 2022) use ViT (Dosovitskiy et al. 2021), TimeSformer (Bertasius, Wang, and Torresani 2021), and hybrid ResNet/Vision Transformer as the video encoder, here we simply use X3D (Feichtenhofer 2020) for its better computational efficiency.

In short, we use the first five convolutional blocks of X3D as the video encoder, without using any down-sampling and pooling layers to maintain the rich spatio-temporal information of the grid-based feature map. Given video input $\mathcal{V}$, we acquire feature maps $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_\mathrm{T}] \in \mathbb{R}^{\mathrm{T}\times\mathrm{H}\times\mathrm{W}\times\mathrm{C}}$, where $\mathrm{T}, \mathrm{H}, \mathrm{W}, \mathrm{C}$ denote the number of frames, height, width, and number of channels of the group of feature maps. That is, for each video, we apply uniform sampling to acquire T frames. These feature maps are projected into the D-dimensional feature space using a $1 \times 1 \times 1$ 3D convolutional layer, with learnable positional embedding added to each feature matrix, thus we have $\mathbf{X} \in \mathbb{R}^{\mathrm{T}\times\mathrm{H}\times\mathrm{W}\times\mathrm{D}}$.

Language Encoder

We use a trainable word embedding layer together with a single LSTM layer as the language encoder for text tokens and semantics extraction. In contrast to BERT-like models (Kenton and Toutanova 2019) used in recent end-to-end V-L methods (Lei et al. 2021; Zellers et al. 2021; Yu et al. 2021; Li et al. 2022; Xue et al. 2022), a single LSTM layer is much efficient and proved to be effective for language encoding.

For each token of the language input, we first acquire the D-dimensional embedding with linear transformation layers. A bidirectional LSTM (Hochreiter and Schmidhuber 1997) layer is further applied to acquire the contextual relations between tokens. The consequential language feature $\mathbf{G} \in \mathbb{R}^{\mathrm{T}_G\times\mathrm{D}}$ is obtained by concatenating the hidden states of LSTM in both directions per timestep, and T_G denotes the number of tokens.

Pyramidal Video-Language Interaction

Apparently, the majority of end-to-end VideoQA research is focused on designing effective self- or semi-supervised tasks using datasets that comprise large amount of vision-language pairs, while improving the generalizability of a model to downstream tasks. Here, we demonstrate how to achieve efficient end-to-end VideoQA solely via a better design of V-L interactions within the proposed PMT model.

The Bottom-up Pathway. While this work is inspired by the success of CNN (He et al. 2016) on acquiring visual features at different levels, for VideoQA, we further consider how to acquire spatial and temporal features at different scales and enable their interaction with linguistic semantics. In the bottom-up pathway of PMT model, L blocks of multimodal transformer layers are stacked to acquire the V-L interaction at different spatio-temporal scales. For $l-$th block ($l \leq \mathrm{L}$), we introduce a ***decomposition*** layer to divide the visual feature map $\mathbf{X}$ into its spatial part $\mathbf{S}^{(l)}$ and temporal part $\mathbf{M}^{(l)}$ as

$$\mathbf{S}^{(l)}_{h',w',d} = \max_{1\leq t\leq\mathrm{T}, h\in\mathrm{R}, w\in\mathrm{R}} \mathbf{X}_{t,h,w,d}, \tag{1}$$

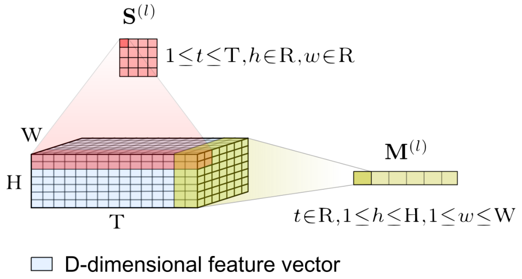
$\mathbf{S}^{(l)}$
$1{\leq}t{\leq}\mathrm{T}, h{\in}\mathrm{R}, w{\in}\mathrm{R}$
W
H
T
$\mathbf{M}^{(l)}$
$t{\in}\mathrm{R}, 1{\leq}h{\leq}\mathrm{H}, 1{\leq}w{\leq}\mathrm{W}$
D-dimensional feature vector

Figure 3: The decomposition layer divides the visual feature into its spatial and temporal sub-features via dimension-wise max-pooling to help acquire multiscale spatio-temporal information within the pyramid.

$$\mathbf{M}^{(l)}_{t',d} = \max_{t\in\mathrm{R}, 1\leq h\leq\mathrm{H}, 1\leq w\leq\mathrm{W}} \mathbf{X}_{t,h,w,d}, \tag{2}$$

where $\mathbf{X}_{t,h,w,d}$ denotes the digit at position (t, h, w, d) of feature map $\mathbf{X}$, and $\mathrm{R} = 2^{(l-1)} \times 2^{(l-1)}$ denotes the size of non-overlapped spatial or temporal pooling region. The consequential $\mathbf{S}^{(l)} \in \mathbb{R}^{\frac{\mathrm{H}}{\mathrm{R}}\times\frac{\mathrm{W}}{\mathrm{R}}\times\mathrm{D}}$ contains complete global-temporal information and local-spatial details, whereas $\mathbf{M}^{(l)} \in \mathbb{R}^{\frac{\mathrm{T}}{\mathrm{R}}\times\mathrm{D}}$ contains complete global-spatial information and local-temporal details. An illustration of the decomposition is shown in Fig.3. While such a decomposition method is based on max-pooling, per se, together with the pyramidal structure it efficiently reveals the multiscale property of the video in either spatial and temporal domains. By doing so, the computational cost is also reduced.

For $l = 1$, $\mathbf{S}^{(l)}$ and $\mathbf{M}^{(l)}$ are input to multimodal transformer block. For $l > 1$, we use a residual merging to make the input for the block as

$$\mathbf{S}^{(l)} = \mathbf{S}^{(l)} + \tilde{\mathbf{S}}^{(l-1)}, \tag{3}$$

$$\mathbf{M}^{(l)} = \mathbf{M}^{(l)} + \tilde{\mathbf{M}}^{(l-1)}, \tag{4}$$

where $\tilde{\mathbf{S}}^{(l-1)}$ and $\tilde{\mathbf{M}}^{(l-1)}$ are the multimodal-interacted output from the previous block. Within the $l-$th block, the V-L interaction of the transformer is implemented as

$$\tilde{\mathbf{S}}^{(l)} = FFN^{(l)}_{\mathrm{S}}(LN(\mathbf{S}^{(l)}_{att})) + \mathbf{S}^{(l)}_{att}, \tag{5}$$

$$\tilde{\mathbf{M}}^{(l)} = FFN^{(l)}_{\mathrm{M}}(LN(\mathbf{M}^{(l)}_{att})) + \mathbf{M}^{(l)}_{att}, \tag{6}$$

with

$$\begin{aligned}\mathbf{S}^{(l)}_{att} = \mathrm{concat}[MCA^{(l)}_h(\mathbf{S}^{(l)}, \mathbf{S}^{(l)}, \mathbf{G})]^{\mathrm{H}}_{h=1}\mathbf{W}^{(l)}_{\mathrm{S}} \\ +\mathbf{S}^{(l)},\end{aligned} \tag{7}$$

$$\begin{aligned}\mathbf{M}^{(l)}_{att} = \mathrm{concat}[MCA^{(l)}_h(\mathbf{M}^{(l)}, \mathbf{M}^{(l)}, \mathbf{G})]^{\mathrm{H}}_{h=1}\mathbf{W}^{(l)}_{\mathrm{M}} \\ +\mathbf{M}^{(l)},\end{aligned} \tag{8}$$

where $FFN(\cdot)$ denotes the feed-forward network implemented with two ELU non-linear activation layers, $LN(\cdot)$ denotes layer normalization, $\mathrm{concat}[\cdot]$ denotes feature concatenation, and $\mathbf{W}^{(l)}_{\mathrm{S}}, \mathbf{W}^{(l)}_{\mathrm{M}} \in \mathbb{R}^{\mathrm{D}\times\mathrm{D}}$ are the trainable parameters. $MCA(\cdot)$ denotes multi-head cross-modal attention layer with H heads totally, which is implemented as

$$MCA_h = \mathrm{softmax}(\frac{\mathbf{F}_q \mathbf{F}_k^\top}{\sqrt{\mathrm{D}}})\mathbf{F}_v, \tag{9}$$

where, for the input $(\mathbf{S}^{(l)}, \mathbf{S}^{(l)}, \mathbf{G})$ as an example, Query $\mathbf{F}_q = LN(\mathbf{S}^{(l)})\mathbf{W}_q^h$, Key $\mathbf{F}_k = LN(\mathbf{S}^{(l)})\mathbf{W}_k^h$, value $\mathbf{F}_v = LN(\mathbf{G})\mathbf{W}_v^h$, and $\mathbf{W}_q^h, \mathbf{W}_k^h, \mathbf{W}_v^h$ are the trainable parameters. Specially, for such an example of spatial-oriented information interaction, $\mathbf{S}^{(l)}$ is first flattened along the spatial dimensions before the computation of Equation 9.

Unlike the isotropic transformer-based architecture, the stack of multimodal transformer blocks together with our decomposition method produces richer semantics information covering V-L interactions along the spatial and temporal dimensions at different scales. Such pyramidal structure also helps reduce computation loads.

The Top-down Pathway. In order to acquire informative answering cues for VideoQA, the global semantics and local spatio-temporal details are both necessary. In our top-down pathway, for such an end, we match the information at different levels coming from both pathways. That is, each time the higher-level semantics with coarser spatio-temporal features are matched with lower-level semantics with higher spatio-temporal resolutions. Thereon, high-resolution as well as strong-semantics information are extracted. Here, we describe how to achieve this via lateral connections between the two pyramidal pathways and the proposed top-down Contextual Matching Block (CMB).

For $l-$th level ($l <$ L) of the top-down pathway, the spatial output $\hat{\mathbf{S}}^{(l)}$ is computed as

$$\hat{\mathbf{S}}^{(l)} = CMB^{(l)}(\tilde{\mathbf{S}}^{(l)}, \hat{\mathbf{S}}^{(l+1)}, \hat{\mathbf{M}}^{(l+1)}), \tag{10}$$

with

$$CMB_{\mathrm{S}}^{(l)}(\cdot) = \tilde{\mathbf{S}}^{(l)} + \mathbf{W}^{(l)}_{\mathrm{M}*\to\mathrm{S}}(\mathbf{W}^{(l)}_{\mathrm{S}\to\mathrm{M}*}\hat{\mathbf{S}}^{(l+1)}), \tag{11}$$

$$\mathbf{W}^{(l)}_{\mathrm{S}\to\mathrm{M}*} = \mathrm{softmax}(f(\hat{\mathbf{M}}^{(l+1)})f(\hat{\mathbf{S}}^{(l+1)})), \tag{12}$$

$$\mathbf{W}^{(l)}_{\mathrm{M}*\to\mathrm{S}} = \mathrm{softmax}(f(\tilde{\mathbf{S}}^{(l)})f(\hat{\mathbf{M}}^{(l+1)})), \tag{13}$$

where $f(\cdot)$ denotes the activated fully-connected layer. Similarly, for the temporal output $\hat{\mathbf{M}}^{(l)}$ we have

$$\hat{\mathbf{M}}^{(l)} = CMB^{(l)}(\tilde{\mathbf{M}}^{(l)}, \hat{\mathbf{M}}^{(l+1)}, \hat{\mathbf{S}}^{(l+1)}). \tag{14}$$

Specially, when $l =$ L, we have $\hat{\mathbf{S}}^{(l)} = \tilde{\mathbf{M}}^{(l)}$ and $\hat{\mathbf{M}}^{(l)} = \tilde{\mathbf{M}}^{(l)}$. In comparison with traditional top-down pathways that use up-sampling or direct attention-based match, the proposed CMB with cross-modal attention uses the spatial or temporal feature as the connection to extract relations between semantic features at different levels, while maintaining the integrity of spatio-temporal representations. Together with global-averaged language semantics $\bar{\mathbf{G}}$ that is acquired by averaging $\mathbf{G}$ across all the tokens, the output of PMT model at level l is

$$\mathbf{O}^{(l)} = \alpha \sum_{t=1}^{\frac{\mathrm{H}}{\mathrm{R}} \times \frac{\mathrm{W}}{\mathrm{R}}} \eta^t \hat{\mathbf{S}}^{(l)} + \beta \sum_{t=1}^{\frac{\mathrm{T}}{\mathrm{R}}} \gamma^t \hat{\mathbf{M}}^{(l)}, \tag{15}$$

with

$$\eta^t = \mathrm{softmax}(\bar{\mathbf{G}} \odot (\hat{\mathbf{S}}^{(l)})), \tag{16}$$

$$\gamma^t = \mathrm{softmax}(\bar{\mathbf{G}} \odot (\hat{\mathbf{M}}^{(l)})), \tag{17}$$

where $\odot$ denotes vector-wise inner product, η^t and γ^t denote the weights between language semantics and spatial or temporal feature, respectively. The learnable coefficients α and β with $\alpha + \beta = 1$ adaptively adjust the importance balance between spatial and temporal features when making the output given different tasks/samples. During inference, only the last output $\mathcal{O}^{(1)}$ is used for the task.

We further introduce a constraint strategy using the multistep loss to help maintain the resolution and semantics integrity of features within this top-down pathway. For $\mathcal{L}^{(l)}$ computed from level l, the multistep loss is computed as

$$\mathcal{L}_{step} = \sum_{l=1}^{\mathrm{L}-1} \max(0, \mathcal{L}^{(l)} - \mathcal{L}^{(l+1)}). \tag{18}$$

This multistep loss tunes the decoders to have smaller loss values along the descent levels. With a penalty factor λ, the total loss is

$$\mathcal{L}_{total} = \mathcal{L}^{(1)} + \lambda \sum_{l=2}^{\mathrm{L}} \mathcal{L}^{(l)} + \mathcal{L}_{step}. \tag{19}$$

Implementation Details

During our experiment, we use the PyTorch deep learning library and merely four NVIDIA GTX 1080 Ti GPUs. The video encoder, X3D-M, is initialized with Kinetics-400 (Kay et al. 2017). The GloVe (Pennington, Socher, and Manning 2014) embedding method is used to initialize the language encoder. In the top-right part of Figure 2, a two-layer fully-connected network with batch normalization is used as the decoder, while different loss functions are used to compute the loss $\mathcal{L}^{(l)}$ per decoder. For open-ended and repetition-count tasks in VideoQA, cross-entropy loss and mean square error are used respectively, and Hinge loss (Gentile and Warmuth 1998) is used for the multi-choice task. For text-to-video retrieval conducted in our ablation study, similar to (Li et al. 2022), we add contrastive loss to the video and language encoders and compute the binary cross-entropy loss given the last output $\mathcal{O}^{(1)}$.

For video processing, the number of frames is T $= 16$, the size of each frame is H $=$ W $= 224$. Simple yet efficient data augmentation (Hara, Kataoka, and Satoh 2018) is used to acquire more frames for training. That is, each input video is uniformly divided into 16 segments and each segment contributes one frame, which is randomly cropped with width-height ratios of $0.8 - 1.2$, rotated, masked, and blurred. In testing, we directly use the frame in the middle of each segment and resize them to be 224×224. The feature dimension is D $= 512$. The number of heads is set to be H $= 8$ in the multimodal transformer block. The penalty factor λ is set to 0.1. Adam optimizer is used with initial learning rate of 10^{-4}, which is cut by half when the loss is not decreased for 10 epochs. The maximum number of epochs is 50, and the batch size is 32 for VideoQA, and 8 for text-to-video retrieval.

Method	Video Rep.	Text Rep.	PT	Action↑	Trans.↑	FrameQA↑	Count↓
ST-TP (Jang et al. 2017)	ResNet, C3D	Glove	-	62.9	69.4	49.5	4.32
Co-Mem (Gao et al. 2018)	ResNet-152, Flow CNN	Glove	-	68.2	74.3	51.5	4.10
PSAC (Li et al. 2019)	ResNet-152	Glove	-	70.4	76.9	55.7	4.27
STA (Gao et al. 2019)	ResNet-152	Glove	-	72.3	79.0	56.6	4.25
L-GCN (Huang et al. 2020)	ResNet-152, Mask R-CNN	Glove	-	74.3	81.1	56.3	3.95
QueST (Jiang et al. 2020)	ResNet-152	Glove	-	75.9	81.0	59.7	4.19
HCRN (Le et al. 2020)	ResNet, ResNeXt-101	Glove	-	75.0	81.4	55.9	3.82
B2A (Park, Lee, and Sohn 2021)	ResNet, ResNeXt-101	Glove	-	75.9	82.6	57.5	3.71
HAIR (Liu et al. 2021)	ResNet, Faster R-CNN	Glove	-	77.8	82.3	60.2	3.88
PVI-Net (Wang et al. 2021a)	ResNet, C3D	BERT	-	79.2	84.7	60.3	3.79
MASN (Seo et al. 2021)	I3D, Faster R-CNN	Glove	-	84.4	87.4	59.5	3.75
HQGA (Xiao et al. 2022)	ResNeXt-101, Faster R-CNN	BERT	-	76.9	85.6	61.3	-
MHN (Peng et al. 2022)	ResNet, 3D ResNet152	Glove	-	83.5	90.8	58.1	3.58
ClipBERT (Lei et al. 2021)	ResNet-50	BERT	COCO,VG	(82.8)	(87.8)	(60.3)	-
SiaSamRea (Yu et al. 2021)	ResNet-50	BERT	COCO,VG	(79.7)	(85.3)	(60.2)	(3.61)
HD-VILA (Xue et al. 2022)	ResNet, TimeSformer	BERT	HD-VILA-100M	(84.3)	(90.0)	(60.5)	-
PMT (ours)	X3D-M	Glove	-	**87.6**	**92.9**	60.6	**3.41**

Table 1: Comparison with state-of-the-art methods on TGIF-QA datasets. Video Rep. and Text Rep. denote the video and text encoders, respectively. PT denotes the datasets used for pre-training. Results in brackets denote what acquired with large-scale pre-training. Only our PMT model and methods placed in the second section are end-to-end.

Experiments

We first report the comparison results of our method against a series of state-of-the-art methods on five VideoQA benchmarks, with more insights provided in the ablation study.

Datasets

We use the state-of-the-art benchmarks for VideoQA in our experiment: 1) TGIF-QA (Jang et al. 2017), a large-scale VideoQA dataset comprising 72K animated GIFs and 165K question-answer pairs, which are divided into four task types, namely multi-choice tasks of Action and Transition (Trans.), open-ended task of FrameQA, and Count that requires answering the exact integer number; 2) MSVD-QA (Xu et al. 2017), with 1,970 short clips and 50,505 open-ended question-answer pairs; 3) MSRVTT-QA (Xu et al. 2017, 2016), with 10K videos and 243K question-answer that have the same setting of MSVD-QA; 4) ActivityNet-QA (Yu et al. 2019), with 5.8K complex videos (average duration of 3 mins) downloaded from the internet and 58K question-answer pairs; and 5) Youtube2Text-QA (Ye et al. 2017), where the video data come from MSVD-QA with 9.9K question-answer pairs, which are divided into three task types, namely *what, who, other*, we experiment with the multi-choice task. We use the official split of training, validation, and testing sets provided by the datasets, and report results acquired on the testing set.

For each videoQA dataset, we pre-define a vocabulary comprising the top K frequent words appeared in the training set, and K is set to 4000 for MSVD-QA and 8000 for the rest. We report accuracy for open-ended and multi-choice tasks. To compute the mean squared error for Count task in TGIF-QA dataset, we apply a rounding-function to the model output to acquire the predicted integer result. For the text-to-video retrieval experiment, following (Yu, Kim, and Kim 2018; Lei et al. 2021; Miech et al. 2019; Li et al. 2020), we use the 7K training set and report results on the 1K testing set. Therein, the first k (R@k) and median rank (MdR) accuracies are reported.

Method	MSVD-QA↑	MSRVTT-QA↑	ActivityNet-QA↑
AMU (2017)	32.0	32.5	-
E-SA (2019)	27.6	29.3	31.8
HME (2019)	33.7	33.0	-
HGA (2020)	34.7	35.5	-
HCRN (2020)	36.1	35.6	-
B2A (2021)	37.2	36.9	-
HAIR (2021)	37.5	36.9	-
SSML (2021)	(35.1)	(35.1)	-
CoMVT (2021)	35.7 (42.6)	37.3 (39.5)	36.6 (38.8)
VQA-T (2021)	41.2 (46.3)	39.6 (41.5)	36.8 (38.9)
MHN (2022)	40.4	38.6	-
HQGA (2022)	41.2	38.6	-
ClipBERT (2021)	-	(37.4)	-
SiaSamRea (2021)	(45.5)	(41.6)	(39.8)
ALPRO (2022)	41.5 (45.9)	39.6 (42.1)	-
HD-VILA (2022)	-	(40.0)	-
PMT (ours)	41.8	40.3	**42.6**

Table 2: Comparison with state-of-the-art methods on MSVD-QA, MSRVTT-QA, and ActivityNet-QA datasets. Results in brackets denote what acquired with large-scale pre-training. Only our PMT model and methods placed in the second section are end-to-end.

Comparison with State-of-the-arts

Table 1 reports the results on TGIF-QA dataset. Without using large-scale pre-training (e.g., using COCO Captions (Chen et al. 2015), Visual Genome Captions (Krishna et al. 2017), HD-VILA-100M video-text pairs (Xue et al. 2022)) and huge or complex feature extractors (e.g., C3D (Tran et al. 2015), I3D (Carreira and Zisserman 2017), ResNeXt (Hara, Kataoka, and Satoh 2018), Faster R-CNN (Ren et al. 2016), and even a tool for text analysis (Manning et al. 2014) seen in B2A (Park, Lee, and Sohn 2021)), our efficient end-to-end PMT model achieves the best performances in Action (+3.2), Trans. (+2.1), and Count (−0.17) tasks of TGIF-QA dataset. We also find that HQGA, ClipBERT, and HD-VILA ignored the Count task of this dataset. It is possible that their methods do not work well on searching global and local semantics in data of long temporal durations.

Method	What↑	Who↑	Other↑	All↑
r-ANL (2017)	63.3	36.4	84.5	52.0
HME (2019)	83.1	77.8	86.6	80.8
L-GCN (2020)	86.0	81.5	80.6	83.9
HAIR (2021)	87.8	82.4	81.4	85.3
PMT (ours)	**90.8**	80.4	**99.0**	**86.4**

Table 3: Comparison with state-of-the-art methods on Youtube2Text-QA dataset. Only PMT model is end-to-end.

Table 2 reports the results on MSVD-QA, MSRVTT-QA, and ActivityNet-QA datasets. For comparison, extra methods are added, namely AMU (Xu et al. 2017), E-SA (Yu et al. 2019; Xu et al. 2017), HME (Fan et al. 2019), HGA (Jiang and Han 2020) SSML (Amrani et al. 2021), CoMVT (Seo, Nagrani, and Schmid 2021), VQA-T (Yang et al. 2021), and ALPRO (Li et al. 2022). It should be mentioned that, the offline methods CoMVT and VQA-T are pre-trained on HowTo100M (Miech et al. 2019) and HowToVQA69M (Yang et al. 2021), respectively. For ALPRO, WebVid2M (Bain et al. 2021) and CC3M (Sharma et al. 2018) are used in pre-training. When pre-training is not used in these methods, our method achieves the best performances across the three datasets. We also recognize that pre-training indeed largely improves the performances of several methods on these three datasets. However, for the ActivityNet-QA dataset, our method outperforms the methods that are pre-trained on large-scale datasets.

In table 3, we draw another comparison on Youtube2Text-QA dataset with another method r-ANL (Ye et al. 2017) added, where we also achieve improved performances when the three attributes (*what, who,* and *other*) are pooled together on multi-choice (+1.1) tasks. Here, L-GCN and HAIR all require the use of extra object detectors.

In general, while large-scale pre-training contribute to VideoQA particularly in MSVD-QA and MSRVTT-QA datasets, the better or on-par performances achieved by our method across the five VideoQA benchmarks so far shall demonstrate the equal importance of proposing effective V-L interactions in a model for VideoQA. This is important at this moment when expensive computational resources are not easily accessible.

Comparison on Computational Efficiency

We propose efficient end-to-end VideoQA in this work with our PMT model, in comparison with methods that normally adopt large-scale pre-training and huge feature extractors. We compute the number of trainable parameters (nParams) and GFLOPs (He et al. 2016; Feichtenhofer 2020; Xie et al. 2018) of our model and several recent methods, which are shown in Figure 4. We set the number of input frames as 16 for computing GFLOPs. It should be noted that parameters and computational loads created by object detectors used in some of these methods are left out. Our PMT model (nParams=18.1M, GFLOPs=5.01B) is more efficient than other end-to-end methods, e.g., ClipBERT (nParams=110.7M, GFLOPs=72.2B), ALPRO (nParams=148.5M, GFLOPs=201.1B) and HD-VILA (nParams=233.6M, GFLOPs=203.6B). Additionally, except for being suboptimal, methods that adopt pretrained and *frozen* encoders may have smaller compute need in training.

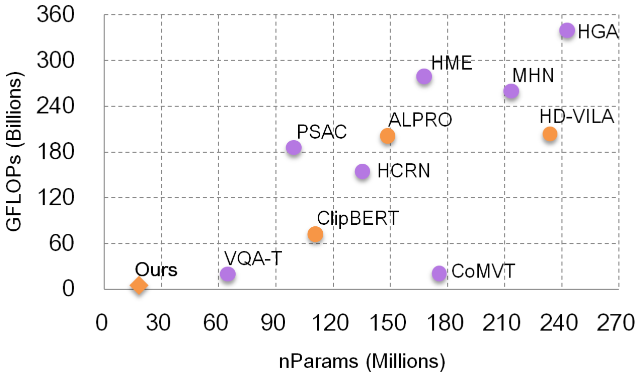

Figure 4: Comparison on computational efficiency. The end-to-end methods are marked in yellow, and those using offline feature extractors are marked in purple.

Ablation Study

The Scalability of our PMT Model. One way to alleviate the dependency on expensive computational resources and take the advantage of large-scale pre-training is to import the feature extractors that have pre-trained weights loaded. Here, we conduct an experiment with our PMT model, where we use TimeSformer and BERT that used in (Li et al. 2022) to replace the vanilla feature extractors used in our experiments above. These two feature extractors were pre-trained on WebVid2M and CC3M video-text pairs by the authors of (Li et al. 2022). We experiment with both the VideoQA and text-to-video retrieval tasks of MSRVTT dataset (Xu et al. 2017, 2016).

As reported in Table 4, by taking the advantage of feature extractors that have more powerful representation learning capacities, our method outperforms other methods on VideoQA, and achieves competitive if not better performances on text-to-video retrieval. Specifically, we acquire an +0.3 improvement on VideoQA, and outperforms SSML, ATP (Buch et al. 2022), HERO (Li et al. 2020), ClipBERT, and VideoCLIP (Xu et al. 2021). For FiT (Bain et al. 2021), * denotes the results acquired by training on the 9K training set. It should be noted that, despite the feature extractors, our PMT model is not pre-trained on large-scale datasets.

Aside from the promising scalability of our method suggested in this experiment, the generalizability of our method for different V-L tasks is demonstrated. This experiment also implies that model engineering (as conducted in this work) together with the reusable prior knowledge (weights from a one-shot pre-training) may enable the end-to-end V-L understanding in an efficient and environmental-friendly manner.

The Impact of the Proposed Pyramidal Structure. This paper proposes to enable multiscale, contextual, and spatio-temporal V-L interactions within a single pyramidal network. Here, we use the default PMT model adopted in the experiments above as the baseline method, and examine the performance of its several candidate variants on TGIF-QA

Method	PT	VideoQA	Text-to-video retrieval			
		acc ↑	R1↑	R5↑	R10↑	MdR↓
SSML (2021)	HowTo100M	35.1	17.4	41.6	53.6	8
VQA-T (2021)	HowToVQA69M	41.5	-	-	-	-
ATP (2022)	WebImageText	-	27.8	49.8	-	-
HERO (2020)	HowTo100M	-	16.8	43.4	57.7	-
ClipBERT (2021)	COCO,VG	37.4	22.0	46.8	59.9	6
SiaSamRea (2021)	COCO,VG	41.6	-	-	-	-
VideoCLIP (2021)	HowTo100M	-	30.9	55.4	66.8	-
FiT* (2021)	WebVid2M, CC3M	-	31.0	59.5	70.5	3
ALPRO (2022)	WebVid2M, CC3M	42.1	33.9	60.7	73.2	3
PMT (ours)	Weights from WebVid2M, CC3M	**42.4**	31.0	55.5	66.2	4

Table 4: The experiment on scalability of our PMT model using MSRVTT dataset. PT denotes the datasets used for pre-training. Only our PMT model and methods placed in the second section are end-to-end.

Method	Action↑	Trans.↑	FrameQA↑	Count↓
PMT w/o decomp.	85.4	90.1	58.6	3.50
PMT w/ up-sample	86.7	92.3	59.5	3.46
PMT w/ attention	86.5	92.4	60.1	3.46
PMT w/o constraint	87.5	92.6	60.4	3.45
PMT (default)	**87.6**	**92.9**	**60.6**	**3.41**

Table 5: Ablation study on our pyramidal

dataset. We remove the decomposition method used in the bottom-up pathway (w/o decomp.), thus features become isotropic. For others, we replace the proposed top-down contextual matching block with up-sampling (w/ up-sample), or attention-based fusion (w/ attention), which are used in previous pyramidal networks, and remove the multistep loss (w/o constraint) to show its importance on maintaining the feature integrity. As shown in Table 5, performances drop for all the tasks when features become isotropic in the network, since the learning of multiscale information becomes difficult. The use of up-sampling and attention-based fusion in the top-down pathway also leads to worse results on all the tasks, showing the importance of our proposed contextual matching block. Without the constraint created by the multistep loss on feature integrity across local and global semantics, performances decrease for all the tasks.

The Impact of Tunable Hyperparameters. One of the noticeable hyperparameter is the number of input frames that balances the amount of information provided to the model and the computational load or even the introduction of irrelevant noise. Here, we experiment with $\mathrm{T} = \{8, 16, 32, 64\}$, using MSVD-QA and MSRVTT-QA datasets, while default values of other hyperparameters are still used. As shown in Figure 5, our PMT model reaches the best performance at $\mathrm{T} = 16$, and the change of such a hyperparameter leads to obvious performance fluctuations. Another hyperparameter that could be of interest is the number of levels L of our pyramidal architecture that sets another balance between informative representation learning and computational loads or ever the risk of overfitting. Particularly, the maximum value of L is set by $\log_2(\mathrm{T})$, given the decomposition method used in our model. We conduct another experiment on MSVD-QA dataset, and do not find further improvements on performance for different values

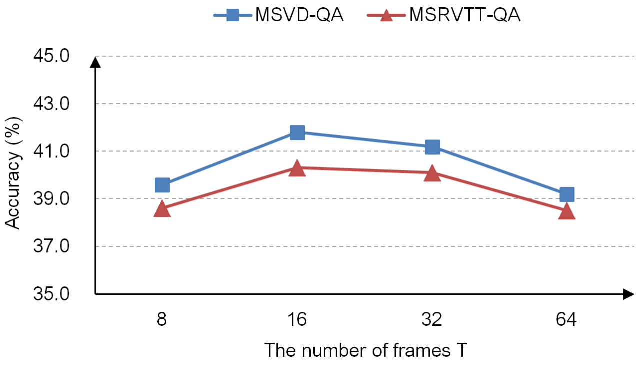

Figure 5: The impact of number of input frames.

other than $\mathrm{L} = 3$ when $\mathrm{T} = 16$.

Conclusion

This paper proposes a new Pyramidal Multimodal Transformer (PMT) model for efficient end-to-end Video Question Answering (VideoQA). Particularly, we enable multiscale, contextual, and spatio-temporal V-L interactions within this single model. Without using large-scale pre-training and huge or complex feature extractors, our PMT model achieves better or on-par performances against state-of-the-art approaches across five VideoQA benchmarks. We demonstrate the scalability and generalizability of our method on text-to-video retrieval, by leveraging feature extractors that have reusable pre-trained weights loaded. Our work suggests that, aside from the booming of large-scale pre-training, model engineering is equally important to V-L understanding, especially when powerful computational resources are still expensive to access and environmental-friendly research is encouraged. Our future work will consider the evaluation with datasets like Next-QA (Xiao et al. 2021) and AGQA (Grunde-McLaughlin, Krishna, and Agrawala 2021), which are certainly more demanding on spatio-temporal and commonsense reasoning.

Acknowledgments

This work is funded by the National Natural Science Foundation of China (62106247).

References

Amrani, E.; Ben-Ari, R.; Rotman, D.; and Bronstein, A. 2021. Noise estimation using density estimation for self-supervised multimodal learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 6644–6652.

Antol, S.; Agrawal, A.; Lu, J.; Mitchell, M.; Batra, D.; Zitnick, C. L.; and Parikh, D. 2015. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, 2425–2433.

Bain, M.; Nagrani, A.; Varol, G.; and Zisserman, A. 2021. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 1728–1738.

Bertasius, G.; Wang, H.; and Torresani, L. 2021. Is space-time attention all you need for video understanding? In *ICML*, volume 2, 4.

Buch, S.; Eyzaguirre, C.; Gaidon, A.; Wu, J.; Fei-Fei, L.; and Niebles, J. C. 2022. Revisiting the" Video" in Video-Language Understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2917–2927.

Carreira, J.; and Zisserman, A. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6299–6308.

Chen, X.; Fang, H.; Lin, T.-Y.; Vedantam, R.; Gupta, S.; Dollár, P.; and Zitnick, C. L. 2015. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*.

Fan, C.; Zhang, X.; Zhang, S.; Wang, W.; Zhang, C.; and Huang, H. 2019. Heterogeneous memory enhanced multimodal attention model for video question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 1999–2007.

Feichtenhofer, C. 2020. X3d: Expanding architectures for efficient video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 203–213.

Gao, J.; Ge, R.; Chen, K.; and Nevatia, R. 2018. Motion-appearance co-memory networks for video question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6576–6585.

Gao, L.; Zeng, P.; Song, J.; Li, Y.-F.; Liu, W.; Mei, T.; and Shen, H. T. 2019. Structured two-stream attention network for video question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 6391–6398.

Gentile, C.; and Warmuth, M. K. 1998. Linear Hinge Loss and Average Margin. In *Proceedings of the 11th International Conference on Neural Information Processing Systems*, 225–231.

Grunde-McLaughlin, M.; Krishna, R.; and Agrawala, M. 2021. Agqa: A benchmark for compositional spatio-temporal reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11287–11297.

Hara, K.; Kataoka, H.; and Satoh, Y. 2018. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 6546–6555.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural computation*, 9(8): 1735–1780.

Huang, D.; Chen, P.; Zeng, R.; Du, Q.; Tan, M.; and Gan, C. 2020. Location-aware graph convolutional networks for video question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 11021–11028.

Huang, Y.; Cao, X.; Zhen, X.; and Han, J. 2019. Attentive temporal pyramid network for dynamic scene classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 8497–8504.

Jang, Y.; Song, Y.; Yu, Y.; Kim, Y.; and Kim, G. 2017. Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2758–2766.

Jiang, J.; Chen, Z.; Lin, H.; Zhao, X.; and Gao, Y. 2020. Divide and conquer: Question-guided spatio-temporal contextual attention for video question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 11101–11108.

Jiang, P.; and Han, Y. 2020. Reasoning with heterogeneous graph alignment for video question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 11109–11116.

Kay, W.; Carreira, J.; Simonyan, K.; Zhang, B.; Hillier, C.; Vijayanarasimhan, S.; Viola, F.; Green, T.; Back, T.; Natsev, P.; et al. 2017. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*.

Kenton, J. D. M.-W. C.; and Toutanova, L. K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL-HLT*, 4171–4186.

Krishna, R.; Zhu, Y.; Groth, O.; Johnson, J.; Hata, K.; Kravitz, J.; Chen, S.; Kalantidis, Y.; Li, L.-J.; Shamma, D. A.; et al. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1): 32–73.

Le, T. M.; Le, V.; Venkatesh, S.; and Tran, T. 2020. Hierarchical conditional relation networks for video question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9972–9981.

Lei, J.; Li, L.; Zhou, L.; Gan, Z.; Berg, T. L.; Bansal, M.; and Liu, J. 2021. 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*, 7331–7341.

Li, D.; Li, J.; Li, H.; Niebles, J. C.; and Hoi, S. C. 2022. Align and Prompt: Video-and-Language Pre-training with Entity Prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4953–4963.

Li, K.; Guo, D.; and Wang, M. 2021. Proposal-free video grounding with contextual pyramid network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 1902–1910.

Li, L.; Chen, Y.-C.; Cheng, Y.; Gan, Z.; Yu, L.; and Liu, J. 2020. HERO: Hierarchical Encoder for Video+ Language Omni-representation Pre-training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2046–2065.

Li, X.; Song, J.; Gao, L.; Liu, X.; Huang, W.; He, X.; and Gan, C. 2019. Beyond rnns: Positional self-attention with co-attention for video question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 8658–8665.

Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; and Belongie, S. 2017. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2117–2125.

Liu, F.; Liu, J.; Wang, W.; and Lu, H. 2021. Hair: Hierarchical visual-semantic relational reasoning for video question answering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 1698–1707.

Manning, C. D.; Surdeanu, M.; Bauer, J.; Finkel, J. R.; Bethard, S.; and McClosky, D. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, 55–60.

Miech, A.; Zhukov, D.; Alayrac, J.-B.; Tapaswi, M.; Laptev, I.; and Sivic, J. 2019. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2630–2640.

Park, J.; Lee, J.; and Sohn, K. 2021. 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*, 15526–15535.

Peng, M.; Wang, C.; Gao, Y.; Shi, Y.; and Zhou, X.-D. 2022. Multilevel Hierarchical Network with Multiscale Sampling for Video Question Answering. In *Proceedings of the International Joint Conference on Artificial Intelligence*.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2016. Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6): 1137–1149.

Seo, A.; Kang, G.-C.; Park, J.; and Zhang, B.-T. 2021. Attend What You Need: Motion-Appearance Synergistic Networks for Video Question Answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 6167–6177.

Seo, P. H.; Nagrani, A.; and Schmid, C. 2021. Look before you speak: Visually contextualized utterances. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16877–16887.

Sharma, P.; Ding, N.; Goodman, S.; and Soricut, R. 2018. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2556–2565.

Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; and Paluri, M. 2015. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, 4489–4497.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Wang, H.; Guo, D.; Hua, X.-S.; and Wang, M. 2021a. Pairwise VLAD Interaction Network for Video Question Answering. In *Proceedings of the 29th ACM International Conference on Multimedia*, 5119–5127.

Wang, W.; Xie, E.; Li, X.; Fan, D.-P.; Song, K.; Liang, D.; Lu, T.; Luo, P.; and Shao, L. 2021b. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 568–578.

Xiao, J.; Shang, X.; Yao, A.; and Chua, T.-S. 2021. Next-qa: Next phase of question-answering to explaining temporal actions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9777–9786.

Xiao, J.; Yao, A.; Liu, Z.; Li, Y.; Ji, W.; and Chua, T.-S. 2022. Video as Conditional Graph Hierarchy for Multi-Granular Question Answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Xie, S.; Sun, C.; Huang, J.; Tu, Z.; and Murphy, K. 2018. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of the European conference on computer vision (ECCV)*, 305–321.

Xu, D.; Zhao, Z.; Xiao, J.; Wu, F.; Zhang, H.; He, X.; and Zhuang, Y. 2017. Video question answering via gradually refined attention over appearance and motion. In *Proceedings of the 25th ACM international conference on Multimedia*, 1645–1653.

Xu, H.; Ghosh, G.; Huang, P.-Y.; Okhonko, D.; Aghajanyan, A.; Metze, F.; Zettlemoyer, L.; and Feichtenhofer, C. 2021. VideoCLIP: Contrastive Pre-training for Zero-shot Video-Text Understanding. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 6787–6800.

Xu, J.; Mei, T.; Yao, T.; and Rui, Y. 2016. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5288–5296.

Xue, H.; Hang, T.; Zeng, Y.; Sun, Y.; Liu, B.; Yang, H.; Fu, J.; and Guo, B. 2022. Advancing High-Resolution Video-Language Representation with Large-Scale Video Transcriptions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5036–5045.

Yang, A.; Miech, A.; Sivic, J.; Laptev, I.; and Schmid, C. 2021. Just ask: Learning to answer questions from millions of narrated videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 1686–1697.

Yang, C.; Xu, Y.; Shi, J.; Dai, B.; and Zhou, B. 2020. Temporal pyramid network for action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 591–600.

Ye, Y.; Zhao, Z.; Li, Y.; Chen, L.; Xiao, J.; and Zhuang, Y. 2017. Video question answering via attribute-augmented attention network learning. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, 829–832.

Yu, W.; Zheng, H.; Li, M.; Ji, L.; Wu, L.; Xiao, N.; and Duan, N. 2021. Learning from inside: Self-driven siamese sampling and reasoning for video question answering. *Advances in Neural Information Processing Systems*, 34: 26462–26474.

Yu, Y.; Kim, J.; and Kim, G. 2018. A joint sequence fusion model for video question answering and retrieval. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 471–487.

Yu, Z.; Xu, D.; Yu, J.; Yu, T.; Zhao, Z.; Zhuang, Y.; and Tao, D. 2019. Activitynet-qa: A dataset for understanding complex web videos via question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 9127–9134.

Zellers, R.; Lu, X.; Hessel, J.; Yu, Y.; Park, J. S.; Cao, J.; Farhadi, A.; and Choi, Y. 2021. Merlot: Multimodal neural script knowledge models. *Advances in Neural Information Processing Systems*, 34: 23634–23651.