AI Open 3 (2022) 1–13

Contents lists available at ScienceDirect

AI Open

journal homepage: www.sciencedirect.com/journal/ai-open

Survey: Transformer based video-language pre-training

Ludan Ruan, Qin Jin *

School of Information, Renmin University of China, Beijing, China

ARTICLE INFO

Keywords:
Transformer
Muti-modal pre-training
Video-language pre-training

ABSTRACT

Inspired by the success of transformer-based pre-training methods on natural language tasks and further computer vision tasks, researchers have started to apply transformer to video processing. This survey aims to provide a comprehensive overview of transformer-based pre-training methods for Video-Language learning. We first briefly introduce the transformer structure as the background knowledge, including attention mechanism, position encoding etc. We then describe the typical paradigm of pre-training & fine-tuning on Video-Language processing in terms of proxy tasks, downstream tasks and commonly used video datasets. Next, we categorize transformer models into Single-Stream and Multi-Stream structures, highlight their innovations and compare their performances. Finally, we analyze and discuss the current challenges and possible future research directions for Video-Language pre-training.

1. Introduction

Transformer networks (Vaswani et al., 2017) have shown their great advantage on performance and become popular in Deep Learning (DL). Compared to traditional deep learning networks such as Multi-Layer Perceptrons (MLP), Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), transformer is more suitable for pre-training & finetuing, because its network structure is easy to deepen and its smaller model bias. The typical pre-training & fine-tuing paradigm is that the model is first trained on a large amount of (typically self-supervised) training data and then fine-tuned on smaller (typically task specific) datasets for the downstream tasks. The pre-training stage helps the model to learn the universal representation, which benefits downstream tasks.

Transformer based pre-training method was first proposed for Natural Language Processing (NLP) tasks and achieved remarkable performance gains. For example, Vaswani et al., (2017) firstly propose the transformer structure with self-attention mechanism for machine translation and English constituency parsing tasks. BERT - Bidirectional Encoder Representations (Devlin et al., 2018) can be considered as a milestone in NLP, which adopts the transformer network for pre-training on unlabeled text corpus and achieves the state-of-the-art performance on 11 downstream tasks. GPT - Generative Pre-trained Transformer v1-3 (Radford and Narasimhan, 2018; Radford et al., 2019; Brown et al., 2020) are designed as general language models with extended parameters and trained on extended training data, among which GPT-3 is trained on 45 TB of compressed plain text data with 175 billion parameters. Inspired by the breakthrough of transformer based pre-training methods in the NLP field, researchers in computer vision (CV) have also applied transformers in varies tasks in recent years. For example, DETR (Carion et al., 2020) removes the bounding box generation stage for object detection based on transformer networks. Dosovitskiy et al., (2021) apply a pure transformer ViT that directly handles sequences of image patches and proves its effectiveness for image classification based on large training set.

Video analysis and understanding is more challenging, because video naturally carries multi-modal information. For the representative Video-Language tasks such as video captioning (Das et al., 2013) and video retrieval (Xu et al., 2016), existing methods have mainly focused on learning video's semantic representation based on the video frame sequence and corresponding captions. In this paper, we focus on providing a comprehensive overview of the recent advances in transformer based pre-training methods for Video-Language processing, including commonly used metrics of corresponding benchmarks, taxonomy of existing model designs, and some further discussion. We hope to track the progress of this area and provide an introductory summary of related works for peer researchers, especially beginners.

The remainder of this paper is organized as follows: Section 2 introduces the related fundamental concepts, including standard transformer with self-attention mechanism, the paradigm of pre-training & finetuning approach, and commonly used datasets. Section 3 presents the major existing methods according to their model structures and

* Corresponding author.
E-mail addresses: ruanld@ruc.edu.cn (L. Ruan), qjin@ruc.edu.cn (Q. Jin).

https://doi.org/10.1016/j.aiopen.2022.01.001
Received 17 September 2021; Received in revised form 21 December 2021; Accepted 13 January 2022
Available online 28 January 2022

highlights their strength and weakness as well. Section 4 further discusses several research directions and challenges, and Section 5 concludes the survey.

2. Background fundamentals

In this section, we introduce the background fundamentals related to video-language pre-training, including Section 2.1 the key mechanisms and structure of transformer, Section 2.2 the Pre-training & Fine-tuning paradigm and commonly used tasks in video-language pre-training, and Section 2.3 the statistics of related video datasets.

2.1. Transformer

Transformer (Vaswani et al., 2017) was first proposed in the field of Neural Language Processing (NLP) and showed great performance on various tasks (Wang et al., 2018; Rajpurkar et al., 2016; Zellers et al., 2018). It has been successfully applied in other fields ever since, from language (Devlin et al., 2018; W.Rae et al., 2020) to vision (Dosovitskiy et al., 2021).

As illustrated in Fig. 1, the standard transformer consists of several encoder blocks and decoder blocks. Each encoder block contains a self-attention layer and a feed forward layer, while each decoder block contains an encoder-decoder attention layer in addition to the self-attention and feed forward layers.

2.1.1. Self-attention

Self-attention is one of the core mechanisms of transformer, which exists in both encoder and decoder blocks. Taking a sequence of entity tokens $X = \{x_0, x_1, \ldots, x_n\}$ as input (the entity tokens can be word sequence in NLP or video clips in the vision area), self-attention layer first linearly transforms the input tokens into three different vectors: key vector $K \in \mathbb{R}^{n\times d^K}$, query vector $Q \in \mathbb{R}^{n\times d^Q}$ and value vector $V \in \mathbb{R}^{n\times d^V}$ (e.g. $d^K = d^Q = d^V = 512$ in practice). The output is produced via $\mathrm{Att}(X) = \mathrm{softmax}(\frac{Q\cdot K^T}{\sqrt{d^Q}} \times V)$ where $Q \cdot K^T$ is to capture the relevance score between different entities, $\sqrt{d^Q}$ is to reduce the score for gradient stability, softmax operation is to normalize the result for probability distribution and finally, multiplying with V is to obtain the weighted value matrix.

In the decoder block, the encoder-decoder attention is similar to self-attention, with the key vector K and the query vector Q from encoder module and the value vector V from the output of the previous decoder block.

Note that not all self-attention attend to all entities. In the training stage of BERT (Devlin et al., 2018), 15% of the input tokens are randomly masked for prediction and the masked entities should not be attended. When using BERT to output the next word token in the downstream task of sentence generation, the self-attention layer of decoder block only attends to the previous generated entities. Such attention can be realized by a mask $M \in \mathbb{R}^{n\times n}$, where the corresponding masked position of M is set zero. The formula of masked self-attention can be adjusted from the original self-attention to $\mathrm{MaskedAtt}(X) = \mathrm{softmax}(\frac{Q\cdot K^T}{\sqrt{d^Q}} \circ M) \times V$.

2.1.2. Multi-head attention

Multi-head attention mechanism (Vaswani et al., 2017) has been proposed to model the complex relationships of token entities from different aspects. To be specific, the input sequence X is linear transformed into h groups of $\{K_i, Q_i, V_i\}_{i=0}^{h-1}$, each group repeats the self-attention process. The final output is produced by projecting the concatenation of the outputs from the h groups with a weight matrix $W \in \mathbb{R}^{hd^V\times d}$. The overall process can be described as:

$$\mathrm{MultiHeadAtt}(X) = [\mathrm{Att}_0(X), \mathrm{Att}_1(X), \ldots, \mathrm{Att}_{h-1}(X)]W$$

$$\mathrm{Att}_i(X) = \mathrm{softmax}\left(\frac{Q_i\cdot K_i^T}{\sqrt{d_i^Q}}\right) \times V_i$$

2.1.3. Position encoding

Different from CNNs (Lecun et al., 1998) or RNNs (Chung et al., 2014), self-attention lacks the ability to capture the order information of the sequence. To address this problem, position encoding (Vaswani et al., 2017) is added to the input embedding in both the encoder and decoder blocks. The position encoding of tokens are constructed as follows:

$$\mathrm{PE}(pos, 2i) = sin\left(\frac{pos}{10\,000^{2i/d_{model}}}\right)$$

$$\mathrm{PE}(pos, 2i+1) = cos\left(\frac{pos}{10\,000^{2i/d_{model}}}\right)$$

where pos refers to the token's position and i refers to the dimension. Another commonly used way to introduce position information is learned position embedding (Gehring et al., 2017). Experiments in (Vaswani et al., 2017) show that these two position encoding methods achieve similar performance.

2.1.4. Transformer structure

The original Transformer (Vaswani et al., 2017) follows the encoder-decoder structure with stacks of 6 encoder blocks and 6 decoder

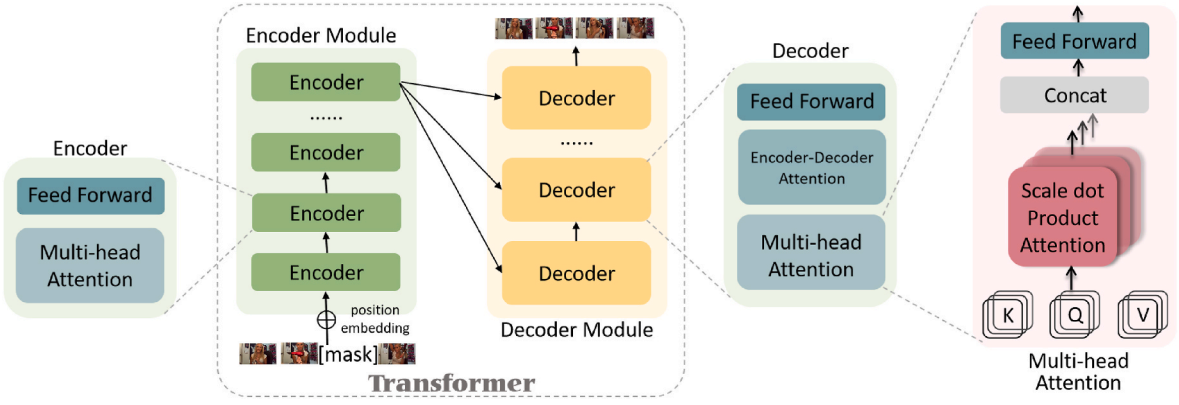

Fig. 1. An overview of the standard transformer architecture. The whole transformer is composed of encoder module and decoder module, with several encoders and decoders stacked in each module respectively. Each encoder consists of a multi-head attention layer and a feed forward layer, while each decoder additionally contains a encoder-decoder attention layer. The multi-head attention mechanism is shown in the right most column, which transfers the input sequence into h groups of {K, Q, V} and concatenates the self-attention outputs of each group as the final output.

blocks respectively. The encoder block consists of a multi-head self-attention sub-layer and a position-wise feed-forward sub-layer, where the position-wise feed-forward sub-layer contains two linear transformations with a ReLU activation. The decoder block additionally inserts a third sub-layer of encoder-decoder attention. What's more, residual connection and layer normalization is added to each single block for further performance promotion. All sub-layers in the model, as well as the embedding layers, produce outputs of dimension $d_{model} = 512$, and the dimension of hidden layer is $d_h = 2048$.

Based on original transformer, multiple variants are proposed to address the problems of cross-modal encoding and spatial-temporal encoding. For the cross-modal encoding, it is common to use different modalities as Query and Key for cross-modal interaction. As shown in Fig. 2, ViLBERT (Lu et al., 2019) proposed a co-attention transformer structure to build cross-modal interaction by exchanging key-value pairs in multi-head attention. Furthermore, ActBERT (Zhu and Yang, 2020) and MulT (Tsai et al., 2019) extend it to three-modal interaction. For the spatial-temporal encoding, as video consists of both spatial and temporal dimensions, while transformer is designed to handle a single sequence information by nature. A few researchers have explored to encode temporal and spatial information synchronously. For example, ViViT (Arnab et al., 2021) introduced four methods to extend ViT (Dosovitskiy et al., 2021) for video processing, which includes 1) extracting non-overlapping, spatial-temporal tokens from the video volume and input into original ViT directly. 2) modeling interactions between image tokens extracted from the same temporal index firstly and then generating a latent representation per time-index. 3) stacking spatial transformer and temporal transformer in a single self-attention block and encoding spatial temporal dimensions alternately. 4) splitting multi-head attention into spatial heads and temporal heads by computing dot-product attention over only the spatial axes or the temporal axis. It is worth noting that due to the computation cost, the works of spatial-temporal encoding on video mainly focus on transferring knowledge from image pre-training to video related tasks.

Compared with CNNs and RNNs, the major advantages of transformer are the ability to simultaneously capture global information and parallel computation. Furthermore, the concise and stackable architecture of transformer enables training on larger datasets, which promotes the development of *pre-training & fine-tuning* self-supervised learning.

2.2. Pre-training & fine-tuning

Pre-training & Fine-tuning has become a typical learning paradigm for transformer based models: first pre-training model on large-scale dataset in supervised or unsupervised way and then adapting the pre-trained model on smaller datasets for specific downstream tasks via fine-tuning. Such paradigm can avoid training new models from scratch for different tasks or datasets. It has been proved that pre-training on larger datasets helps learning universal representations, which improves the performance of downstream tasks. For example, NLP Transformer model GPT (Radford and Narasimhan, 2018) gains average 10% absolute improvement on 9 downstream benchmark datasets (e.g. CoLA (Warstadt et al., 2018), MRPC (Dolan and Brockett, 2005)) after pre-training on BooksCorpus dataset (Zhu et al., 2015) with 7000 unpublished books. Vision Transformer model ViT-L/32 (Dosovitskiy et al., 2021) gains 13% absolute accuracy improvement on the test set of ImageNet (Deng et al., 2009) after pre-training on JFT-300M (Sun et al., 2017) with 300 million images.

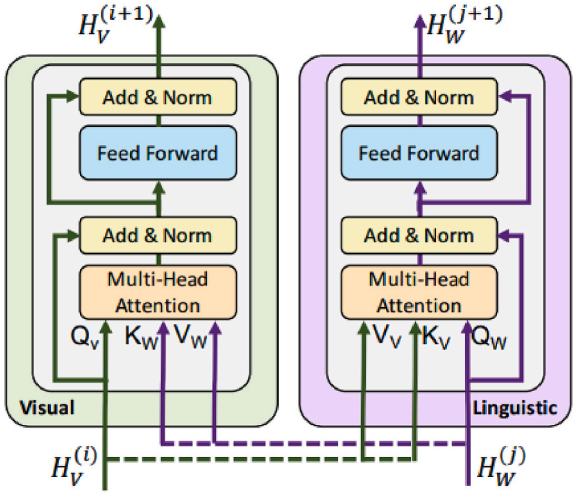

Fig. 2. Illustration of Co-attention Transformer layer. Figure is from Lu et al. (2019).

Owing to the successful application of pre-trained models in NLP and CV tasks, more and more researches explore the cross-modal tasks, including Vision-Language and Video-Language. The main difference between Vision-Language tasks and Video-Language tasks is that the former focus on the image and text modalities such as language based image retrieval (Lee et al., 2018) and image captioning (Vinyals et al., 2015), while the later focuses on the video and text modalities, which adds the temporal dimension over the image modality.

In following subsections, we describe the Pre-training & Fine-tuning methods in Video-Language field, including the commonly used proxy tasks and video-language downstream tasks.

2.2.1. Proxy tasks

Proxy tasks are crucial for the final performance of pre-trained models as they directly determine the models' learning objectives. We classify the proxy tasks into three categories: Completion tasks, Matching tasks and Ordering tasks. 1) *Completion tasks* aim to reconstruct the masked tokens of input, which endow the model with the ability of building intra-modal or inter-modal relationships. Typical tasks include Masked Language Modeling (MLM), Masked Frame Modeling (MFM), Masked Token Modeling (MTM), Masked Modal Modeling (MMM) and Language Reconstruction (LR). We will describe them in details in the following section. 2) *Matching tasks* are designed to learn the alignment between different modalities, originating from Next Sentence Prediction (NSP) of BERT (Devlin et al., 2018). For example, Video Language Matching (VLM) is the classical matching task, which aims at matching video and text modalities. Some researchers also introduce the audio modality for further matching objective (Akbari et al., 2021). 3) *Ordering tasks* are to shuffle the sequence at the input side and force the model to recognize the original sequence order. For example, Frame Ordering Modeling (FOM) is specifically designed to exploit the temporal nature of video sequence and Sentence Ordering Modeling (SOM) is designed for the text modality.

Among all commonly used proxy tasks, *Self-Supervised Learning* (SSL) is the dominant strategy adopted in order to adapt to the situation that pre-training requires massive training data. SSL is one type of *UnSupervised Learning* (USL) that generates labeled data automatically itself, which inspires the model to learn the inherent co-occurrence relationships of data. For example, in the sentence completion task such as "I like __books", a well-trained language model should fill in the blank with the word "reading". In Video-Language pre-training, Masked Language Modeling (MLM) and Mask Frame Modeling (MFM) are two widely used SSL proxy tasks.

Contrastive Learning (CL) (Chen et al., 2020) has recently become an important component in self-supervised learning for Video-Language pre-training. Different from generating masked tokens with measuring L2 distance, it embeds the same samples close to each other while trying to push away the embeddings from different samples. An extensive survey of CL can be found in (Jaiswal et al., 2020).

In the remainder of this section, we introduce some widely used proxy tasks (as summarized in Table 1) during Video-Language pre-training. For the following formulas, we use the general notations of *w*,

Table 1
A summary of proxy tasks that commonly used in Video-Language Pre-training.

Task	Type	Strategy	Sub-task	Description
MLM	Completion	USL		Predicting text tokens that are masked with certain percentage.
MFM	Completion	USL	MFMCE	Predicting masked frame tokens with cross entropy loss.
			MFMR	Reconstructing the masked video tokens with regression loss.
			MFMCL	Identifying the masked video tokens from negative samples constructed by various methods.
MTM	Completion	USL		Identifying the masked tokens (video or text) from negative samples constructed by various methods.
MMM	Completion	USL		Masking either all video tokens or all text tokens and recovering them from the other modality.
LR	Completion	SL		Generate text sequence from left to right according to video modality.
VLM	Matching	USL	GVLM	Globally matching video and text modality.
			LVLM	Matching video and text at the frame level.
SOM	Ordering	USL		Randomly shuffling sentence and reconstruct the sentence order from video modality.
FOM	Ordering	USL		Randomly shuffling video tokens and reconstruct the frame order from text modality.

v, t as word sequence, video sequence and the union tokens of v and w. w_m, v_m, t_m refer to the corresponding masked tokes.

Masked Language Modeling (MLM) was first referred to as a cloze task in (WL, 1953) and then adapted as a proxy task during the pre-training of BERT (Devlin et al., 2018). Original MLM is to randomly mask out a fixed percentage of words from the input sentence, and then predict the masked words based on other word tokens. MLM used in Video-Language pre-training not only learns the inherent co-occurrence relationships of sentence but also combines the visual information with the sentence. For example, as elaborated in ActBERT (Zhu and Yang, 2020), when a verb is masked out, the task forces the model to extract relevant action features for more accurate prediction. When a noun or a description of noun is masked out, visual features of related object can provide contextual information. Empirically, the masking percentage is always set 15%. The loss function of MLM can be defined as:

$$\mathcal{L}_{MLM} = -\mathbb{E}_{w_m \sim w}(logP(w_m|w_{\setminus w_m}, v))$$

Masked Frame Modeling (MFM) is similar to MLM in that it simply changes the sentence to the video sequence. That is, the frame tokens are masked for prediction according to the contextual frames and the input text for semantic constraints.

However, since a video is continuous, with no fixed vocabulary as text, researchers make different adjustments on the input side or loss objective side for the MFM task. We categorize MFM into three sub tasks according to loss functions: 1) MFM with Cross Entropy (MFMCE), 2) MFM with Regression (MFMR), and 3) MFM with Contrastive Learning (MFMCL).

The typical examples of MFMCL can be found in VideoBERT (Sun et al., 2019) and ActBERT (Zhu and Yang, 2020). VideoBERT splits the continuous videos into clip tokens and clusters clip tokens into a fixed size of dictionary by hierarchical k-means. In this way, the masked video feature can be predicted as video word with class likelihood. ActBERT extracts the action concept and local object feature from the video and the model is forced to predict the action category and object category of masked video tokens respectively. The loss function of MFMCL can be defined as:

$$\mathcal{L}_{MFMCL} = -\mathbb{E}_{v_m \sim v}(logP(v_m|v_{\setminus v_m}, w))$$

The typical examples of MFMR can be found in HERO (Li et al., 2020), which learns to regress the output on each masked frame v_m to its visual features. HERO uses L2 regression between the input video feature v_m and the output video feature $h(v_m)$:

$$\mathcal{L}_{MFMR} = \mathbb{E}_{v_m \sim v}(\left\| h(v_m) - v_m \right\|^2)$$

However, it is hard to reconstruct the original video feature with regression as a video contains rich information. MFMCL adapts Contrastive Learning (Chen et al., 2020) to maximize the Mutual Information (MI) between the masked video tokens and the original video tokens:

$$\mathcal{L}_{MFMCO} = -\mathbb{E}_{v_m \sim v}(log\text{NCE}(v_m|v_{\setminus v_m}, w))$$

$$\text{NCE}(v_m|v_{\setminus v_m}, w) = \frac{\exp(h(v_m)v_m^T)}{\mathcal{Z}}$$

$$\mathcal{Z} = \exp(h(v_m)v_m^T) + \sum_{v_j \in v_{\setminus v_m}} \exp(h(v_m)v_j^T)$$

Masked Token Modeling (MTM) unifies MLM and MFM in one loss function. It is proposed by Xu et al., (2021) and the formula is defined as:

$$\mathcal{L}_{MTM} = -\mathbb{E}_{t_m \sim t}(log\text{NCE}(t_m|t_{\setminus t_m}))$$

$$\text{NCE}(t_m|t_{\setminus t_m}) = \frac{\exp(h(t_m)t_m^T)}{\mathcal{Z}}$$

$$\mathcal{Z} = \exp(h(t_m)t_m^T) + \sum_{t_j \in t_{\setminus t_m}} \exp(h(t_m)t_j^T)$$

Compared with MLM and MFM, MTM learns joint token embeddings for both video and text tokens. Furthermore, it also expands the contrasted negative samples in two separate losses for MFM and MLM.

Masked Modal Modeling (MMM) is first used in UniVL (Luo et al. (2020) as part of the pre-training strategy and later is formally proposed by VLM (Xu et al., 2021). It masks either all video tokens or all text tokens, which encourages the model to use tokens from one modality to recover tokens from the other modality. The objective function employs NCE as in MTM, and experiments in VLM (Xu et al., 2021) have proved its effectiveness especially for text-based video retrieval (Xu et al., 2016).

Language Reconstruction (LR) LR is a generative task, which aims to enable the pre-trained model with the ability of video caption generation. The difference between LR and masked language method (MLM and MMM with all text tokens being masked) is that LR generates sentence from left to right, which means the model only attends to the former text tokens and video tokens when predicting the next text token. The loss function is:

$$\mathcal{L}_{LR} = -\mathbb{E}_{w_i \sim w'}(logP(w_i|w_{<i}, w', v))$$

where w' is the groundtruth of word sequence and w is the masked version.

Video Language Matching (VLM) aims to learn the alignment between video and language. There are different task forms of VLM and we classify them into 1) Global Video Language Matching (GVLM) and 2) Local Video Language Matching (LVLM). For the GVLM, one objective function is adapted from the Next Sentence Prediction (NSP) task used by BERT (Devlin et al., 2018), which takes in the hidden state of special token [cls] to a FC layer for binary classification. The objective function is:

$$\mathcal{L}_{GVLM} = -\mathbb{E}_D logP(y|v, w))$$

where y = 1 if v and w are matched. Another VLM is to match the sequence embedding of the two modalities. Specifically, it transfers the 2 embedding sequence of video and language into 2 single feature by mean pooling or linear transfer, then it forces the paired samples closer while pushes away different ones by MIL-NCE (Miech et al., 2020) or other functions. This objective is usually used in pre-training models with multi-stream structure, which does not contain the special token [cls] for direct matching prediction. The example objective function (Luo et al., 2020) is:

$$\mathcal{L}_{GVLM} = -\mathbb{E}_{(v,w)\sim \mathbf{B}} log\text{MIL} - \text{NCE}(v, w)$$

$$\text{MIL} - \text{NCE}(v, w) = \frac{\sum_{(\widehat{v},\widehat{w})\in\mathcal{P}_{v,w}} (\exp(\widehat{v}\widehat{w}^T))}{\mathcal{Z}}$$

$$\mathcal{Z} = \sum_{(\widehat{v},\widehat{w})\in\mathcal{P}_{v,w}} (\exp(\widehat{v}\widehat{w}^T)) + \sum_{(\widetilde{v},\widetilde{w})\in\mathcal{N}_{v,w}} (\exp(\widetilde{v}\widetilde{w}^T))$$

where $\widehat{v}$, $\widetilde{v}$, $\widehat{w}$, $\widetilde{w}$ are mean pooling of video sequence v and text sequence w respectively, the negative pairs $\mathcal{N}_{v,w}$ take negative video clips or captions within the batch **B** after fixing v or w.

Another VLM aims to align video and language locally, thus we abbreviate it as LVLM (Local Video Language Matching). It is first proposed in HERO (Li et al., 2020) that matches video and language at the frame level. That is, computing query-video matching score by dot product: $s = vq \in \mathbb{R}^{N_v}$, where q is the query obtained from language sequence. Two trainable 1D CNNs followed by softmax operation are applied to the matching scores to get two probability vectors p_{st}, p_{ed}, which represent the probability of every position being the start and the end of the ground-truth span. The objective function uses cross-entropy loss and can be summarized as:

$$\mathcal{L}_{LVLM} = -\mathbb{E}_D(log(p_{st}[y_{st}]) + log(p_{ed}[y_{ed}]))$$

Sentence Ordering Modeling (SOM) SOM is first proposed in VICTOR (Lei et al., 2021a), which aims to learn the relationships of text tokens from sequential perspective. Specifically, 15% sentences are selected, randomly split into 3 segments and shuffled by a random permuted order. Therefore, it can be modeled as a 3!-class classification problem. To be specific, after multi-modal fusion, the embedding of special token [cls] is input into the FC layer followed by a softmax operation for classification. The overall objective function is:

$$\mathcal{L}_{SOM} = -\mathbb{E}_D(logP(y|w_s, v))$$

where y is the groundtruth of segment order and w_s is the shuffled word sequence.

Frame Ordering Modeling (FOM) FOM is proposed in VICTOR (Lei et al., 2021a) and HERO (Li et al., 2020). The core idea is to randomly shuffle a fixed percentage of frames and predict their original order. VICTOR (Lei et al., 2021a) randomly selects to shuffle 15% frames. The embedding of each shuffled frame is transformed through a FC layer, followed with softmax operation for N_v-class classification, where N_v is the maximum length of frame sequence. HERO (Li et al., 2020) also randomly selects 15% of frames to be shuffled. The embeddings of all frames are transformed through a FC layer, followed with softmax operation to produce a probability matrix $P \in \mathbb{R}^{N_v \times N_v}$. $P_{i,j}$ represents the scores of the i-th frame that belongs to the j-th time stamp. The two types of FOM can be summarized into one objective function:

$$\mathcal{L}_{FOM} = -\mathbb{E}_D(logP(y|v_s, w))$$

where y is the groundtruth of frame order and v_s is the shuffled frame sequence.

2.2.2. Video-language downstream tasks

The target of pre-training is to better adapt the learned knowledge from a large corpus to downstream tasks via transfer learning (Belinkov et al., 2017). Representative downstream tasks also play the role in evaluating pre-trained models. For better transfer impact, we need to consider the model structure and choose appropriate transferring method for each downstream task. The common downstream tasks that appear in the Video-Language pre-training include generative tasks and classification tasks. We introduce the task requirements and how to transfer the knowledge from pre-training to downstream tasks in the following subsections.

Text-based Video Retrieval (Yu et al., 2018) is defined to retrieve a relevant video/video segment given an input text query. It requires model to map the video and text modality into a common semantic embedding space. Since the proxy task of VLM aims at learning the alignment between video and text, many works (Zhu and Yang, 2020; Li et al., 2020; Luo et al., 2020; Lei et al., 2021b) adapt the proxy task of VLM to calculate the matching score of these two modalities directly.

Action Recognition (Zhu et al., 2020) is defined to classify the action category of the given video/video segment, which is a representative classification task for video understanding. To transfer pre-trained knowledge to action recognition, works in (Sun et al., 2020; Lei et al., 2021a) use the pre-trained models as feature extractors and finetune a linear classifier added on the top of pre-trained model for action recognition.

Action Segmentation (Ding and Xu, 2018) is designed to predict action label of given video/video segment at the frame level. It is also a classification task with video as the only input. To apply pre-trained models to action segmentation, several works (Zhu and Yang, 2020; Xu et al., 2021) use the pre-trained models as feature extractors and add a linear classifier upon the extracted video features.

Action Step Localization is first proposed in Cross Task (Zhukov et al., 2019), which aims to recognize action steps in instructional videos. The difference between action step localization and action recognition is that for the step localization, event is described with manual phrase but not from fixed label dictionary. To apply pre-trained models to action step localization, works in (Zhu and Yang, 2020; Luo et al., 2020; Xu et al., 2021) regard manual phrase as text description and calculate its relevance score with input video/video segment by either dot production or linearly transforming the embedding of [cls].

Video Question Answering (Tapaswi et al., 2016; Lei et al., 2018; Jang et al., 2017) aims to automatically answer natural language questions given a context video. VideoQA applied in Video-Language pre-training can be divided into multiple choices task or fill-in-the-blank task according to the types of the answers, both of which can be handled as classification tasks. For multi-choice VideoQA, works in (Zhu and Yang, 2020; Li et al., 2020) feed candidate answer at the end of query sentence to generate QA-aware global representations, and input the global representations into MLP based classifier to obtain the matching score. The final choice is made by selecting the candidates with the max matching score. For fill-in-the-blank VideoQA, ActBERT (Zhu and Yang, 2020) proposes a similar method, which adds a linear classifier upon the cross-modal feature but without the input of candidate text.

Video Captioning (Chen et al., 2019; Zhou et al., 2018b) is the task of generating a natural-language utterance for the given video, which is the only generative task among the downstream tasks introduced in this paper. It is one of the most typical tasks for multi-modal understanding and nearly all works related to Video-Language pre-training evaluate their pre-trained models on this task. To transfer pre-trained knowledge to video captioning, works in (Sun et al., 2019; Zhu and Yang, 2020; Li et al., 2020) use pre-trained models as video feature extractor or video encoder and add a transformer-based decoder for finetuning. Works in (Xu et al., 2021) transfer a single encoder to generate word sequence by reusing the pre-trained model as prediction heads. Work in (Luo et al., 2020) includes a generative task in the pre-training stage by adding a transformer decoder, which reduces the gap between the proxy task and the video captioning task.

As shown in above introduction, Video-Language pre-training works focus more on classification task. Improving the pre-trained model's ability especially for generation can be further explored. What's more, in addition to the downstream tasks we listed above, other downstream tasks such as multi-modal sentiment analysis (Zadeh et al., 2017), image-based retrieval (Wang et al., 2017) have also been explored recently.

2.3. Video-language datasets

Compared with CNNs, transformer based frameworks rely heavily on massive datasets especially for pre-training. The quality and quantity of video datasets matter a lot to model's performance. In this section, we divide the commonly used video datasets into 3 categories according to the types of their annotations: label-based datasets, caption based datasets and other datasets. Table 2 summarizes all mentioned datasets.

2.3.1. Label Based Datasets

Label Based Datasets are the datasets with labels at the video level. They are widely used for classification tasks such as action recognization. For example, HMDB51 (Kuehne et al., 2011) contains 6,841 videos from 51 action categories in total. UCF101 (Soomro et al., 2012), MPII Cooking (Rohrbach et al., 2012), Kinetics series (Kay et al., 2017) and AVA (Gu et al., 2018) are the other representative datasets.

2.3.2. Caption based datasets

Caption Based Datasets require descriptions for each video or video segment. For example, Activitynet (Krishna et al., 2017a) includes 20k YouTube untrimmed videos with 100k manually caption sentences. Each caption describes the content of the corresponding video segment annotated by start and end time stamps. Caption is the major annotation of video datasets with widely applications. On the one hand, large-scale Caption Based Datasets can be used for Video-Language pre-training. For instance, Howto100M (Miech et al., 2019) is so far the largest English video dataset, which contains 136M video clips paired with captions from YouTube (mostly instructional videos), most works (Sun et al., 2020; Zhu and Yang, 2020; Li et al., 2020) pre-train their models on this dataset. Alivol-10M (Lei et al., 2021a) is a Chinese e-commerce dataset with 10M videos of 98,801 h in total. The descriptions mostly follow the standards of the e-commerce platform to describe the visual content of certain product. Auto-captions on GIF (Pan et al., 2020) is newly designed for generic video understanding based on GIF videos. The paired description is extracted from the Alt-text HTML attribute of each GIF video. On the other hand, datasets with caption annotations are widely used in downstream tasks such as video retrieval/video moment retrieval, video captioning/dense video captioning and text based localization (requires time stamps annotations). As shown in Table 2, ActivityNet (Krishna et al., 2017a), Charades (Sigurdsson et al., 2016), TGIFs (Li et al., 2016), YouCookII (Zhou et al., 2018a), etc. are the representative caption datasets.

2.3.3. Other datasets

In addition to the caption and label annotations, other types of annotations are used for other downstream-tasks. As shown in Table 2, TVQA (Lei et al., 2018) is a videoQA dataset based on 6 popular TV shows, with 460 h of videos and 152.5K human-written QA pairs in total. Each query provides 5 candidates with one correct answer, the correct answer is also marked with start and end time stamps for further inference. COIN (Tang et al., 2019) is designed for **CO**mprehensive **IN**structional video analysis, which is organized with a 3-hierarchical structure, from domain, task, to step. The dataset contains 11,827 instructional videos in total with 12 domains, 180 tasks, and 778 pre-defined steps. As all the videos are annotated with a series of step

Table 2
Commonly used Datasets in Video-Language Pre-training and finetuning. suffix * for Alivol-10 means that the dataset is not released yet. We divide the datasets into 3 groups according to the type of their annotations: Label Based Datasets, Caption Based Datasets and Other Datasets.

Dataset	videos	clips	annotations	duration	source	year
Label Based Datasets						
HMDB51	3.3k	6.8k	labels	24h	Web/Other Datasets	2011
UCF101	2.5k	13.3k	labels	27h	YouTube	2012
MPII Cooking	44	5.6k	labels	8h	Kitchen	2012
Kinetics400	306k	306k	labels	817h	YouTube	2017
AVA	430	230k	labels	717h	YouTube	2018
Caption Based Datasets						
Howto100M	1.22M	136M	136M captions	134,472h	YouTube	2019
Alivol-10M*	10.3M	11M	11M captions	98,801h	e-commerce	2020
Auto-captions on GIF	163k	163k	164k	–	GIF Web	2020
ActivityNet	20k	100k	100k captions	849h	YouTube	2015
Charades	10k	18k	16k captions	82h	Home	2016
TGIF	102k	102k	126k captions	103h	Tumblr GIFs	2016
YouCookII	2k	14k	14k captions	176h	YouTube	2016
MSR-VTT	7.2k	10k	200k captions	40h	YouTube	2016
Didemo	10k	27k	41k captions	87h	Fliker	2017
LSMDC	200	128k	128k captions	150h	Movies	2017
How2	13k	185k	185k captions	298h	YouTube	2018
TVR	21.8k	21.8K	109k queries	460h	TV shows	2020
TVC	21.8k	21.8k	262k captions	460h	TV shows	2020
VIOLIN	6.7k	16k	95k captions	582h	Movie & TV show	2020
Other Datasets						
TVQA	925	21.8k	152.5k QAs	460h	TV shows	2018
COIN	12k	46k	segment labels	476h	YouTube	2019
CrossTask	4.7k	20k	20k steps	376h	YouTube	2019

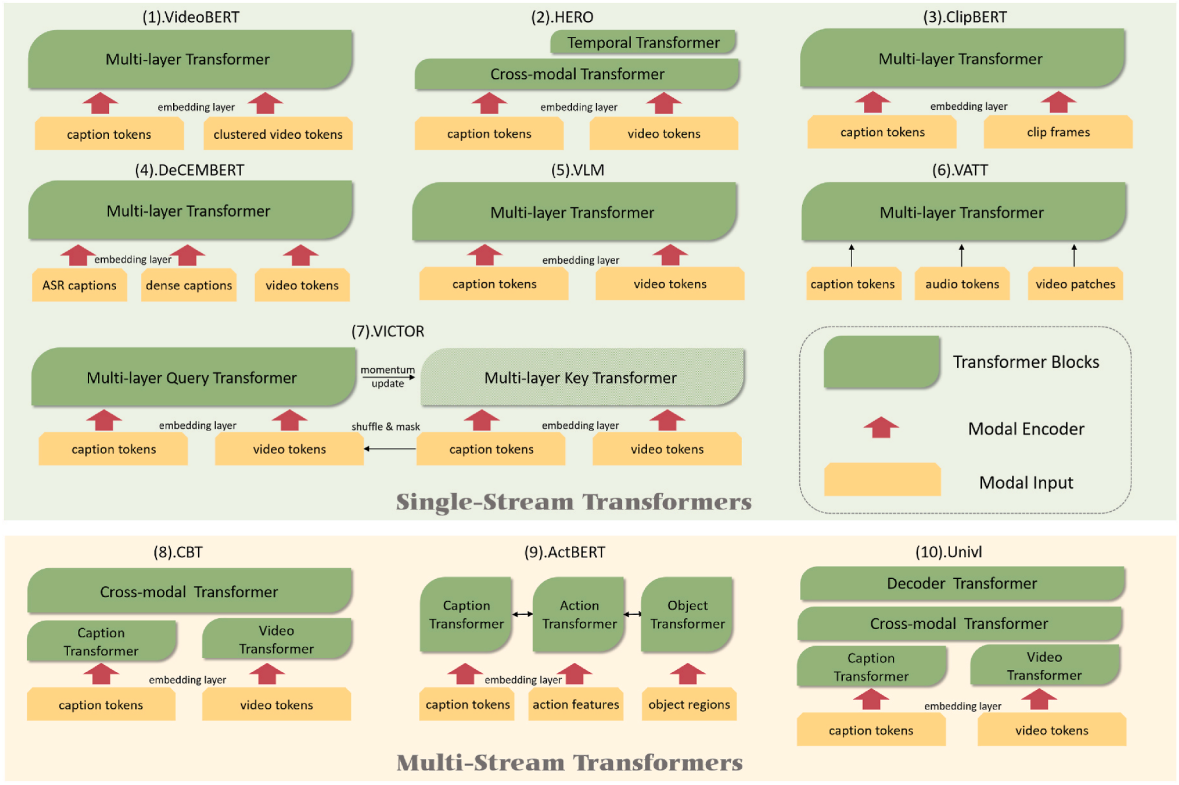

Fig. 3. An overview of Transformer models used for Video-Language representation learning. All models are divided into **Single-Stream Transformers** (VideoBERT (Sun et al., 2019), HERO (Li et al., 2020), ClipBERT (Lei et al., 2021b), DeCEMBERT (Lei et al., 2021b), VLM (Xu et al., 2021), VATT (Akbari et al., 2021), VICTOR (Lei et al., 2021a)) and **Multi-Stream Transformers** (CBT (Sun et al., 2020), ActBERT (Zhu and Yang, 2020), UniVL (Luo et al., 2020)) according to their structure. Despite the differences in model structure, most models take caption tokens and video tokens as inputs, while DeCEMBERT takes ASR captions as additional text information, ActBERT takes object regions as additional visual information and VATT takes audio as additional modality information. As for modal encoders, most models apply modality encoders to extract modality features while VATT abandons them.

descriptions and the corresponding temporal boundaries, COIN is commonly used for action segmentation task. CrossTask (Zhukov et al., 2019) contains 4.7k instructional videos crawled from YouTube, related to 83 tasks. For each task, an ordered list of steps with short descriptions are provided. Works in (Zhu and Yang, 2020; Luo et al., 2020) evaluate their pre-trained models on the task of Action Step Localization (Zhukov et al., 2019) based on this dataset.

3. Video-language transformer models

In this section, we provide an overview of Transformer based models for Video-Language pre-training in Fig. 3. We roughly divide different models into two categories based on their model structure: Single-Stream Transformers and Multi-Stream Transformers.

For the Single-Stream Transformers, features/embeddings of different modalities are input into a single transformer to capture their intra and inter modality information. Multi-Stream Transformers input each modality into independent transformers to capture information within modalities and then build cross-modal relationship via for example another transformer. Single-Stream and Multi-Stream Transformers are not new to video-language pre-training as it is commonly used in all cross-modal pre-training. B2T2 (Alberti et al., 2019) and VideoBERT (Sun et al., 2019) are first to use the Single-Stream architecture for cross-modal learning. ViLBERT (Lu et al., 2019) and LXMERT (Tan and Bansal, 2019) followed with Multi-Stream architecture.

In addition to the model structure, the distinctions across different methods relate to their inputs, proxy tasks and downstream tasks and benchmarks, which we summarize in Table 3 and describe in details below.

3.1. Single-stream structure

3.1.1. VideoBERT

VideoBERT (Sun et al., 2019) is the first to explore Video-Language representation with transformer based pre-training method. It follows the single-stream structure, porting the original BERT structure to the multi-modal domain as illustrated in Fig. 3-(1). Specifically, it inputs the combination of video tokens and linguistic sentence into multi-layer transformers, training the model to learn the correlation between video and text by predicting masked tokens. VideoBERT shows the ability of simple transformer structure to learn high level video representations that capture semantic meaning and long-range temporal dependencies.

To discretize continuous videos as discrete word tokens, they cut the video into small clips of fixed length and cluster the tokens to build a video dictionary. In pre-training stage, the model is trained with proxy tasks of MLM, MFM and VLM, corresponding to the feature learning in text-only domain, video-only domain, and video-text domain. Although with the simple proxy tasks and plain model structure, VideoBERT shows great performance on the downstream tasks of zero-shot action classification and video captioning. The model is initialized with the pre-trained BERT weights, the video token is generated based on the S3D (Xie et al., 2018) backbone. All experiments are applied on the cooking domain, with pre-training on the large scale of cooking videos crawled from YouTube by authors themselves and evaluating on the YouCookII benchmark dataset (Zhou et al., 2018a).

3.1.2. HERO

As illustrated in Fig. 3-(2), Li et al., (2020) propose HERO, a **H**ierarchical **E**ncode**R** for **O**mni representation learning, which contains a

Table 3
A summary of Video-Language Pre-training methods.

Method	Inputs	Proxy Tasks	Pre-train Dataset	Downstream Task(dataset)	Source
Single-Stream					
VideoBERT	video + text	MLM, MFM, VLM	Cooking	Action Classification(YoucookII), Video Caption(YoucookII)	ICCV
HERO	video + text	MLM, MFM,	Howto100M, TV	Video retrieval(MSR-VTT, TVR,DiDeMo,How2R), VideoQA(TVQA,How2QA), Video-and-Language Inference (VIOLIN), Video Caption(TVC)	EMNLP
ClipBERT	clip + text	MLM, VLM	COCO Captions, VG Captions	Video Retrieval(MSR-VTT, DiDeMo, ActivityNet), VideoQA(TGIF-QA,MSRVTT)	CVPR
DeCEMBERT	video + text	MLM, VLM, Constrained	Howto100M	Video Retrieval(MSR-VTT, YouCookII), Video Caption(MSR-VTT, YouCookII),	NAACL
VLM	video + text	MTM, MMM	Howto100M	Video retrieval(MSR-VTT, YouCookII), Action Segmentation(COIN), Action Step Localization(CrossTask), VideoQA(MSR-VTT), Video Caption(YouCookII)	ACL
VATT	video + text +audio	Multi-modal Contrastive Learning	Howto100M, AudioSet	Action Recognition(UCF101, HMDB51,kenitics-400,600), Audio Event Classification (ESC50,AudioSet),	NeurIPS
VICTOR	video + text	MLM, MFOM, MSOM, dual-VSA,	Alivol-10M	Video Retrieval(Alivol-10M), Video Classification(Alivol-10M), Video Recommendation (users' video viewing logs), Video Caption(Alivol-10M)	arxiv
Multi-Stream					
CBT	video + text	MLM, MFM,	Howto100M	Action Recognition(UCF101, HMDB51), Action Anticipation(Breakfast, 50Salads, ActivityNet), Video Caption(YouCookII), Action Segmentation(COIN)	ECCV
ActBERT	action + region +text	MLM, MAM,	Howto100M	Video Retrieval(YouCookII, MSR-VTT), Video Caption(YouCookII), Action Segmentation(COIN), Action Step Localization(CrossTask), VideoQA(LSMDC, MSR-VTT)	CVPR
UniVL	video + text	MFM, MLM,	Howto100M	Video Retrieval(YouCookII, MSR-VTT), Video Caption(YoucookII), Action Segmentation(COIN), Action Step Localization(CrossTask), Multi-modal Sentiment Analysis (CMU-MOSI)	arxiv

cross-modal transformer to fuse video frame sequence and corresponding sentence, and a temporal transformer to learn contextualized video embeddings from the global context. Previous works simply adapt proxy tasks of masking (MLM) and matching (VLM) that originated from NLP domain. HERO firstly designs the proxy tasks of LVLM (Local Video Language Matching) and FOM (Frame Order Modeling), which consider the sequential nature of videos. These two proxy tasks have been described in Section 2.2.1. The experiments of HERO prove that hierarchical transformer structure and new proxy tasks are both beneficial to downstream tasks. Li et al., (2020) also expand the pre-training datasets from instructional video domain to TV or movie domain. They find that text-based video-moment retrieval is more sensitive to domain gaps. In other words, keeping dataset domain consistent, text-based video retrieval could achieve the same or better performance with less pre-training data.

To be more specific, HERO extracts both 2D and 3D video features with ResNet (He et al., 2016) and Slow-Fast (Feichtenhofer et al., 2019) respectively. The cross-modal transformer takes the combination of video sequence and text sequence as input to learn contextualized embeddings through cross-modal attention. The output of visual embeddings are further input into temporal transformer to learn contextualized embeddings from the global video context. HERO applies the proxy tasks of MLM, MFM, VLM and FOM in pre-training stage and transfers to downstream tasks of video retrieval, videoQA, video-and-language inference and video captioning. The ablation study shows that FOM can effectively benefit downstream tasks that rely on temporal reasoning (such as QA tasks), VLM for both global and local alignment can benefit the retrieval tasks.

3.1.3. *ClipBERT*

Lei et al., (2021b) propose a generic framework ClipBERT for video-text representation learning that could be trained in end-to-end manner. Different from previous works that extract video features from pre-trained backbone such as S3D (Xie et al., 2018), ClipBERT directly samples a few frames from each video clip, using 2D CNN as backbone instead of 3D CNN for lower memory cost and better computation efficiency. Based on 2D visual backbone, they also demonstrate that image-text pre-training on COCO (Chen et al., 2015) and Visual Genome Captions (Krishna et al., 2017b) benefits video-text tasks. ClipBERT adopts a sparse sampling strategy, including sampling a few frames from each clip and using only a single or a few sampled clips instead of full-length videos. The experiments show 1 or 2 frames per clip and 2 clips per video is sufficient for effective Video-Language pre-training.

The concrete structure of ClipBERT is single-stream (Fig. 3-(3)), the video input is patch sequence of a single clip. After 2D backbone

generates T visual feature map for T frames of each single clip, a temporal fusion layer is applied to aggregate the frame-level feature maps into a single clip-level feature map. A cross transformer is then applied to combine the clip feature map and text sequence to capture the cross-modal relationship. During the inference, when multiple clips are used, the predictions are fused together as the final output. ClipBERT uses MLM and VLM objectives to optimize the model, the pre-trained weights are further finetuned to text-based video retrieval and videoQA on 6 benchmarks.

3.1.4. DeCEMBERT

Tang et al., (2021) propose the approach of **De**nse **C**aptions and **E**ntropy **M**inimization (DeCEMBERT) to alleviate the problem that the automatically generated captions in pre-training dataset like Howto100M (Miech et al., 2019) are noisy and sometimes unaligned with video content. To be specific, the original caption may not describe the rich content of the corresponding video or contains only irrelevant words due to recognition error of ASR. Therefore, DeCEMBERT uses dense captions (Johnson et al., 2016) generated from (Yang et al., 2017) as additional language input for the model learning. To better align video with ASR captions, DeCEMBERT propose a constrained attention loss that encourages the model to select the best matched ASR caption from a pool of continuous caption candidates.

As illustrated in Fig. 3-(4), DeCEMBERT applies the single-stream structure, using a BERT like transformer to encode the relationship of video features, dense captions and a set of continuous ASR captions. The whole model is pre-trained with MLM, VLM tasks and finetuned on video captioning, text-based video retrieval and videoQA. Comprehensive experiments demonstrate that DECEMBERT is an improved pre-training method for learning from noisy, unlabeled dataset.

3.1.5. VLM

Previous methods (Luo et al., 2020; Li et al., 2020) propose either multiple transformer encoders or a single cross-modal encoder but requires both modalities as inputs, What's more, existing pre-training tasks tend to be more and more task-specific, limiting the extensibility and generalization ability of pre-trained models. In contrast, VLM (**V**ideo-**L**anguage **M**odel) is a task-agnostic model with BERT like cross-model Transformer that can accept text, video, or both as input.

VLM introduces two new schemes of masked tasks: As illustrated in Fig. 4 Masked Modality Modeling (MMM) and Masked Token Modeling (MTM). MMM is to randomly mask a whole modality for a portion of training examples, which forces the encoder to reconstruct the masked modality based on the tokens from the other modality. MTM is to randomly mask a fixed portion of tokens (both video or language tokens) and predict them from negative candidates, which unifies the losses on MLM and MVM. MMM has been validated to be effective especially for text-based video retrieval and MTM performs better than MLM + MVM. VLM is evaluated on the downstream tasks of text-based video retrieval, action segmentation, action step localization and videoQA. To apply the BERT like model with single encoder to generative tasks such as video captioning, VLM uses a masked attention map to make the future text tokens unavailable. Based on that, VLM re-use the language model heads as prediction heads for generation with no extra decoder architecture. Experimental results show that VLM can maintain competitive performance while requiring less parameters.

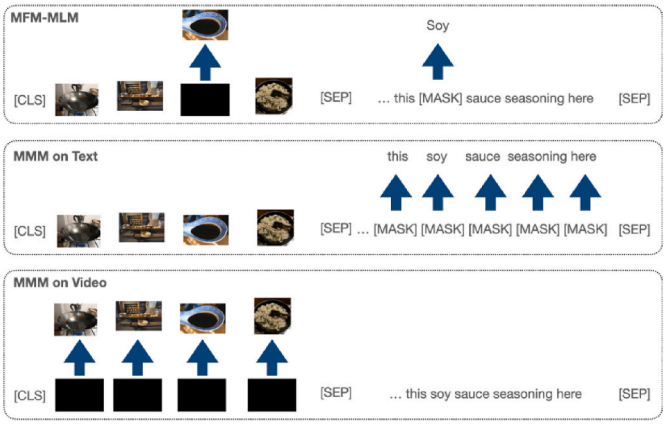

Fig. 4. Illustration of MMM and MTM (on text or video) of VLM, Figure is from (Xu et al., 2021).

3.1.6. VATT

Akbari et al., (2021) present an end-to-end framework VATT (**V**ideo-**A**udio-**T**ext **T**ransformer) for leaning multi-modal representations from raw video, audio and text. To be specific, they partition the raw video frames into a sequence of $[T/t] \times [H/h] \times [W/w]$ patches, where T, H, W correspond to video's temporal, height, width dimension respectively. The raw audio waveform is segmented on its temporal dimension. The word token is represented by one-hot vector. These three modality sequences are transformed by linear projection but not pre-trained backbones as previous works do. To obtain inherent co-occurrence relationships of three modalities, Akbari et al., (2021) adopt the most widely used transformer architecture (ViT) except keeping the layer of tokenization and linear projection reserved for each modality separately. VATT is optimized by matching video-audio pairs and video-text pairs with common space projection and contrastive learning. The whole model is pre-trained on Howto100M (Miech et al., 2019) providing video-audio-text triplets and AudioSet (Gemmeke et al., 2017) providing audio-text pairs. After pre-training, VATT is finetuned on the downstream tasks of action recognition, audio event classification (Dai et al., 2017), text-based video retrieval and image classification. The experiment results of image classification on ImageNet (Deng et al., 2009) demonstrate that VATT can be adapted from video domain to image domain.

In conclusion, VATT validates that large-scale self supervised pre-training is a promising direction to learn multi-modal representation (video, text, audio) with pure attention-based model and end-to-end training.

3.1.7. VICTOR

VICTOR (Lei et al., 2021a) stands for **VI**deo-language understanding via **C**ontrastive mul**T** im**O**dal p**R**e-training, which is trained on Chinese Video-Language dataset. VICTOR follows the single-stream model structure, with an encoder transformer to obtain the cross-modal relationship, a decoder transformer for generative tasks. What's more, inspired by MoCo (He et al., 2020) that expands negative samples with memory bank and momentum updating for better contrastive learning, VICTOR involves memory queues that save the negative samples for calculating contrastive losses. Synchronously, another network symmetric to the main Query network named Key network is applied to embed negative samples.

Due to the absence of Chinese pre-training dataset, Lei et al., (2021a) collect Alivol-10M from e-commerce platform with standard descriptions and corresponding product videos. The details have been described in Section 2.3. Lei et al., (2021a) design new proxy tasks of Masked Frame Order Modeling (MFOM), Masked Sentence Order Modeling (MSOM) and Dual Video and Sentence Alignment (dual-VSA) for pre-training, where MFOM is to explore the sequential structure of videos by reordering the shuffled video sequence, MSOM is similar to MFOM but from the text perspective. For the dual-VSA (similar with VLM), they only take matched video-text pairs as inputs, utilizing the representation of frames/text to retrieve the representation of corresponding text/frames. In other words, the negative samples come from memory bank would only go through Key transformer network as the authors point out that inputting in the mismatched video and text would hamper the pre-training process of multi-modal encoder. The pre-trained weights of VICTOR are further transferred to downstream

tasks of multi-level video classification, content-based video recommendation, multi-modal video captioning, and cross-modal retrieval that with both text and image as input query.

3.2. Multi-stream structure

3.2.1. CBT

CBT (Sun et al., 2020) propose noise contrastive estimation (NCE) (Józefowicz et al., 2016) as the loss objective for Video-Language learning, which preserves the fine-grained information of video compared to vector quantization (VQ) and softmax loss in VideoBERT. The model contains 3 components as shown in Fig. 3-(8): one text transformer (BERT) to embed discrete text features, one visual transformer that takes in the continuous video features and a third cross-modal transformer to embed mutual information between two modalities. CBT extends the BERT structure to multi-stream structure and verifies the effectiveness of NCE loss for learning cross-modal features.

In pre-training stage, two single modal transformers learn video and text representations respectively via contrastive learning. The third cross-modal transformer combines the two modal sequences, computes their similarity score and learns the relationship of paired video and sentence by NCE loss. Sun et al., (2020) propose curriculum learning strategy by first pre-training the S3D (Xie et al., 2018) backbone and then finetuning the last block of S3D with visual transformer using visual loss. Both pre-trained visual features and cross-modal features are evaluated on downstream tasks of action recognition, action anticipation, video captioning and video segmentation.

3.2.2. ActBERT

Zhu et al. (Zhu and Yang, 2020) introduce global action and local regional objects as visual inputs to learn joint video-text representations. ActBERT is a multi-stream model (Fig. 3-(9)) with Tangled Transformer block to enhance the communications between different sources, which is illustrated in Fig. 5. Previous multi-stream structure always use an extra transformer layer to encode inter relationship of multi-modal information, while the Tangled Transformer block uses co-attentional transformer layer (Lu et al., 2019) that the key-value pairs from one modality could pass through the other modality. Experiments on various Video-Language related downstream tasks verify that the global action information and local object clues are complementary.

For the global action input, they extract verbs from the corresponding descriptions of each video clip and build a verb dictionary.

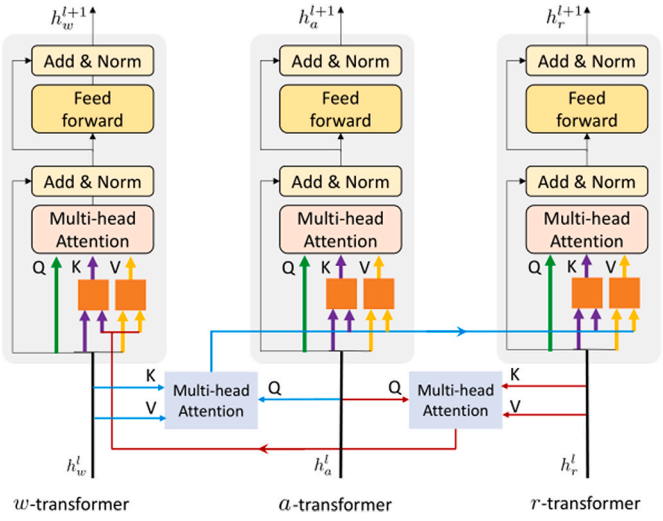

Fig. 5. Illustration of Tangled Transformer block. Figure is from Zhu and Yang (2020).

Then a 3D network classifier is trained to predict the each video clip's verb labels. The action feature of each clip is extracted from the 3D network classifier after global averaging layer. For the input of local object regions, authors use pre-trained Faster-RCNN (Ren et al., 2015) to extract the bounding boxes and the corresponding visual features. ActBERT is pre-trained on the proxy tasks of MLM, MAM (Masked Action Modeling), MOM (Masked Object Modeling) and VLM. The pre-trained weights are further transferred to 5 downstream tasks of video captioning, action segmentation, action step localization, video retrieval and videoQA.

3.2.3. UniVL

Previous multi-modal models are pre-trained on understanding tasks, which leads to discrepancy for generative downstream tasks such as video captioning. UniVL (Luo et al., 2020) is the first one to pre-train model on both understanding and generative proxy tasks. UniVL follows multi-stream structure as illustrated in Fig. 3–10, which contains two single transformer encoders to embed video and text respectively, a cross-modal transformer to fully interact the text and video embeddings, a decoder for generation tasks.

UniVL uses VLM, MFM, MLM and LR (Language Reconstruction) as proxy tasks for pre-training, transfers to the downstream tasks of text-based video retrieval, multi-modal video captioning, action segmentation, action step localization and multi-modal sentiment analysis. There are two types of VLM in UniVL. The first one is to train two single modal encoders by matching their video and text sequence with NCE loss. The other is to train the cross modal transformer by inputting the special token [cls] to predict the alignment score of given video and sentence. The experiments show that the later type of VLM applied on the cross-modal transformer benefits more on retrieval tasks. UniVL develops a three stage training strategy for pre-training. Firstly, UniVL trains the weights of text BERT and video transformer by matching their output sequences with NCE objective. Next, the whole model is trained by all objectives with smaller learning rate. Furthermore, UniVL enhances its video representations by masking the whole text tokens with a 15% possibility. The step-by-step training strategy improves the pre-training process consistently.

3.3. Summary & comparison

In this part, we provide a summary and comparison of the above mentioned methods from the perspectives of model structure, proxy tasks, training strategy, and performance on widely used benchmarks: MSR-VTT for text-based video retrieval and YouCookII for video captioning. The paradigm of the above methods can be summarized as building models containing transformer encoders to learn intra and inter modality representations, pre-training models on pre-designed proxy tasks and finetuning/evaluating on varies downstream tasks.

Model Structure For the transformer blocks, most works apply the original transformer structure directly, while some works make adjustments to adapt to multi-modal processing and temporal modeling. For example, VATT shares the weight of self attention layer across different modalities, but keeps the layer of tokenization and linear projection independent for each modality. VLM uses different attention masks to accommodate downstream tasks that require different modalities. ActBERT use Tangled Transformer blocks to build relationship of different modalities across independent transformer blocks. HERO use an extra temporal transformer for the temporal encoding of videos.

For the word embedding, most works apply WordPiece embeddings (Wu et al., 2016) with a 30,000 token vocabulary provided by BERT (Devlin et al., 2018). For the video embedding, most works extract video features with fixed visual backbone S3D (Xie et al., 2018), which is pre-trained by Miech (Miech et al., 2020). There exists exceptions, for example, VICTOR (Lei et al., 2021a) utilizes 2D backbone of Inception-V4 (Szegedy et al., 2017) pre-trained on ImageNet (Deng et al., 2009) to extract visual features for each frame. HERO (Li et al.,

2020) combines 2D features from Resnet (He et al., 2016) and 3D features from SlowFast (Feichtenhofer et al., 2019). ClipBERT (Lei et al., 2021b) and VATT (Akbari et al., 2021) design end-to-end frameworks without fixed visual backbone.

Proxy Tasks The selection or designing of the proxy tasks directly determines the model training objectives and further affects the performance on downstream tasks. Most pre-training works inherit the masking based tasks and matching based tasks from BERT, learning the correlation within the same modality and across different modalities. HERO (Li et al., 2020) and VICTOR (Lei et al., 2021a) design ordering tasks to explore the sequential structure of videos, which have been demonstrated beneficial to downstream tasks that rely on temporal reasoning such as videoQA. UniVL (Luo et al., 2020) and VLM (Xu et al., 2021) both demonstrate that masking out the whole modalities and reconstruct it based on other modalities benefits the retrieval task.

Training Strategy A few works develop the stage-by-stage pre-training methods instead of training the whole model in one step. For example, UniVL (Luo et al., 2020), the representative of Multi-Stream transformer, trains the transformer encoder for each modality first and then the whole model with decreasing learning rate. CBT (Sun et al., 2020) uses a curriculum learning strategy by first pre-training the visual feature extractor S3D and then jointly fine-tuning the last blocks of S3D with the visual transformer using the CBT visual loss. Compared to training in one step, training stage-by-stage makes the pre-training progress more smoothing.

Downstream Tasks To evaluate the pre-trained models, the standard approach is to transfer the pre-trained weight to other down-stream tasks. We compare the above methods on matching task of text-based video retrieval and generative task of video captioning. The results are shown in Tables 4 and 5 respectively. We divide models according to their structure. VLM (Xu et al., 2021) generally performs the best across Single-Stream models for both retrieval and captioning tasks. Among Multi-Stream models, UniVL (Luo et al., 2020) outperforms other models generally. VICTOR (Lei et al., 2021a) is not included since it is pre-trained and evaluated only on Chinese dataset.

Result Analysis Due to the different settings and implementation details, it is hard to draw a unified conclusion from above methods. For example, it is still an open question whether it is better to add a cross-encoder as late-fusion or to directly fuse multi-modal information with single-stream model structure. Although from Table 4 and Table 5, VLM (Xu et al., 2021) achieves better performance on both text-video retrieval and video caption, the choices of proxy tasks, settings of hyper-parameters also introduce in uncontrollable variables. However, we can observe some common trends from various methods. For example, UniVL (Luo et al., 2020) and HERO (Li et al., 2020) both prove that multi-task learning brings benefits. ClipBERT (Lei et al., 2021b) and VLM (Xu et al., 2021) both indicate that video has a lot of redundancy in the temporal dimension. ClipBERT reduces redundancy by reducing sampling frames, while VLM by increasing the proportion of MASK.

4. Discussion

Pre-training has shown obvious improvements on various Video-Language tasks compared to traditional methods. Nevertheless, the potential of transformer structure on Video-Language has not been fully explored. There still exists several challenges to be tackled. In this section, we discuss these challenges and possible future directions.

Table 4
Performance of text-based video retrieval on MSR-VTT.

Methods	R@1	R@5	R@10	Median R
Single-Stream				
HERO	16.8	43.4	57.7	–
ClipBERT	22.0	46.8	59.9	6
DeCEMBERT	17.5	44.3	58.6	9
VLM	28.1	55.5	67.4	4
Multi-Stream				
ActBERT	8.6	23.4	33.1	36
UniVL	20.6	49.1	62.9	6

Table 5
Performance of Video Captioning on YouCookII. B, M, R, C are abbreviations of BLUE, METEOR, ROUGE, Cider.

Methods	B-3	B-4	M	R	C
Single-Stream					
ViedoBERT	6.80	4.04	11.01	27.50	0.49
DeCEMBERT	–	11.92	20.01	40.22	0.58
VLM	17.78	12.27	18.22	41.51	1.39
Multi-Stream					
CBT	–	5.12	12.97	30.44	0.44
ActBERT	8.66	5.41	13.30	30.56	0.65
UniVL	16.46	11.17	17.57	40.09	1.27

4.1. Pre-training dataset

Since transformers lack some inductive biases as CNNs, it requires large scale of datasets for pre-training. Consequently, the quality, quantity and diversity of dataset has significant influence on the general performance of transformers. For the problem of quantity, the most commonly used dataset for pre-training so far is Howto100M (Miech et al., 2019), which contains over 100M video-sentence pairs. Experiments on (Miech et al., 2019) prove that increasing the amount of training data improves the performance of variable evaluated tasks. For the problem of quality, since large scale of manual video annotations are expensive, the corresponding captions of videos are usually generated from ASR automatically (Miech et al., 2019; Sun et al., 2019), which inevitably introduces mistakes and misalignments to captions for corresponding video content. DeCEMBERT (Tang et al., 2021) has mitigated these problems by adding extra inputs (dense video captions (Johnson et al., 2016)) and adjusting the training objective. For the problem of diversity, pre-training dataset used in VideoBERT (Sun et al., 2019) focuses on cooking domain. Videos of Alivol-10M (Lei et al., 2021a) come from E-commerce website. Videos of Howto100M (Miech et al., 2019) are crawled from YouTube. These pre-training datasets are mainly from a single domain and inevitably have domain gaps with various downstream datasets, which has been demonstrated to be harmful to the performance of pre-trained models. On this topic, Zhou et al., (2021) proved that pre-training on a considerably small subset of domain-focused data can effectively close the source-target domain gap and achieve significant performance gain. Similar conclusion can be found in HERO (Li et al., 2020) that domain gap of finetuning and pre-training can not be eliminated by data volume. In conclusion, although a lot of explorations have been done, there is still a long way to go in order to improve the quantity, quality and diversity of pre-training datasets.

4.2. Video-language transformer designs

Existing works mostly follow the paradigm from NLP domain and make adjustments to adapt to Video-Language processing, which includes using multi-stream structure to meet the needs of multimodal input, designing reordering proxy tasks to exploit the sequential structure of videos, and adding audio modality as supplementary information. Although the results of these applications are quite encouraging, current methods require further intuition to better match the Video-Language tasks. Firstly, how to deal with the visual backbone properly remains unsolved. Existing works either apply a independently trained visual backbone to extract video features (Xie et al., 2018) or train a model that includes 2D CNN backbone in the end-to-end manner (Lei et al., 2021b). The first type of approach not only leads to domain gap

between feature extraction and pre-training, but also hinders model improvement due to the loss of fine-grained visual information. The other type of approach tends to lose the temporal information in the video. Secondly, standard evaluation of Video-Language pre-training is an urgent need for sustainable development in this field. So far, different models are evaluated on different downstream tasks/datasets with different detailed settings, which is unfair to compare their performance. A unified benchmark is needed to evaluate different pre-training methods, such as GLUE (Wang et al., 2018) in NLP. VALUE (Li et al., 2021) has proposed an evaluation benchmark that covers 11 datasets over 3 popular tasks including retrieval, caption and videoQA, but it has not yet been popularized.

Another promising direction is to improve the generalization ability of pre-trained models. As a collection of multiple modalities, a video contains more than semantic information. For example, ActBERT (Zhu and Yang, 2020) uses fine-grained object regions of videos, VATT (Akbari et al., 2021) explores the inner relationship of frame sequence, audio and sentence. We believe that there exist more clues that can mined from videos, such as scene information, character information. How to make use of these information and transfer to more downstream tasks are promising future directions. On the basis of multiple input, video analysis should not be limited to analysis of general semantics. Tasks related to image, audio, and text modalities are expected to be covered by a comprehensive model. What's more, we notice that existing works mostly focus on the domains of activity, films, and TV shows. Other domains such as medical field, surveillance recordings have a lot of potential applications as well.

4.3. *Transformer efficiency*

A well-known concern of transformer is the efficiency problems of quadratic time and memory complexity, which hinders transformer's scalability in practice. As mentioned in (Tay et al., 2020), model efficiency refers to both memory footprint and computation cost. For the memory efficient transformers, Lee et al. (Lee et al. (2021) use weight sharing across layers and modalities to reduce overall model size. Similar idea is originated from Universal Transformers (Dehghani et al., 2019) and Albert (Lan et al., 2020). For the computation efficient transformers, Michel et al., (2019) remove some heads at test time without impacting performance. Prasanna et al. (2020) also reduce the computation cost by pruning and decomposing original transformer structure.

In summary, studies of efficient transformers are mainly from NLP domain, which focus more on handling longer sequence input. Video naturally conveys more information than pure text. Video-Language processing requires deeper model structure, larger parameters and thus has higher requirements for hardware and computation.

5. Conclusion

Pre-training has become a popular approach in NLP and has been further applied in vision tasks. Comparing to other vision-language pre-training works, less pre-training works reported in the Video-Language area. We therefore conduct a comprehensive overview of pre-training methods for Video-Language processing in this paper. This survey first reviews the background knowledge related to transformer, then summarizes the pre-training and finetuning process of Video-Language learning by introducing the common proxy tasks and downstream tasks respectively. Furthermore, we describe commonly used video datasets according to their scale, annotation type etc. We also summarize the state of the art transformer models for Video-Language learning, highlight their key strength and compare their down-stream performance. Finally, we conclude the paper with discussions of the current challenges and possible future research directions.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (No. 62072462) and Large-Scale Pre-Training Program 468 of Beijing Academy of Artificial Intelligence (BAAI).

References

Akbari, H., Yuan, L., Qian, R., Chuang, W.H., Chang, S.F., Cui, Y., Gong, B., 2021. Vatt: Transformers for Multimodal Self-Supervised Learning from Raw Video, Audio and Text arXiv:2104.11178.
Alberti, C., Ling, J., Collins, M., Reitter, D., 2019. Fusion of Detected Objects in Text for Visual Question Answering. CoRR abs/1908.05054.
Arnab, A., Dehghani, M., Heigold, G., Sun, C., Lucic, M., Schmid, C., 2021. Vivit: A Video Vision Transformer. ICCV.
Belinkov, Y., Durrani, N., Dalvi, F., Sajjad, H., Glass, J.R., 2017. In: Barzilay, R., Kan, M. (Eds.), What Do Neural Machine Translation Models Learn about Morphology?. ACL.
Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D., 2020. Language Models Are Few-Shot Learners. NeurIPS.
Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S., 2020. End-to-end Object Detection with Transformers. ECCV.
Chen, X., Fang, H., Lin, T.Y., Vedantam, R., Gupta, S., Dollar, P., Zitnick, C.L., 2015. Microsoft Coco Captions: Data Collection and Evaluation Server.
Chen, S., Yao, T., Jiang, Y., 2019. In: Kraus, S. (Ed.), Deep Learning for Video Captioning: A Review. IJCAI.
Chen, T., Kornblith, S., Norouzi, M., Hinton, G.E., 2020. A Simple Framework for Contrastive Learning of Visual Representations. ICML.
Chung, J., Gülçehre, Ç., Cho, K., Bengio, Y., 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. CoRR abs/1412.3555.
Dai, W., Dai, C., Qu, S., Li, J., Das, S., 2017. Very Deep Convolutional Neural Networks for Raw Waveforms. ICASSP, IEEE, pp. 421–425.
Das, P., Xu, C., Doell, R.F., Corso, J.J., 2013. A Thousand Frames in Just a Few Words: Lingual Description of Videos through Latent Topics and Sparse Object Stitching. CVPR.
Dehghani, M., Gouws, S., Vinyals, O., Uszkoreit, J., Kaiser, L., 2019. Universal Transformers. ICLR.
Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L., 2009. Imagenet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255.
Devlin, J., Chang, M., Lee, K., Toutanova, K., 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. CoRR.
Ding, L., Xu, C., 2018. Weakly-supervised Action Segmentation with Iterative Soft Boundary Assignment. CVPR.
Dolan, W.B., Brockett, C., 2005. Automatically Constructing a Corpus of Sentential Paraphrases. IWP@IJCNLP.
Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N., 2021. An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale. ICLR, OpenReview.net.
Feichtenhofer, C., Fan, H., Malik, J., He, K., 2019. Slowfast Networks for Video Recognition. ICCV, IEEE.
Gehring, J., Auli, M., Grangier, D., Yarats, D., Dauphin, Y.N., 2017. Convolutional Sequence to Sequence Learning. ICML.
Gemmeke, J.F., Ellis, D.P.W., Freedman, D., Jansen, A., Lawrence, W., Moore, R.C., Plakal, M., Ritter, M., 2017. Audio Set: an Ontology and Human-Labeled Dataset for Audio Events. ICASSP.
Gu, C., Sun, C., Ross, D.A., Vondrick, C., Pantofaru, C., Li, Y., Vijayanarasimhan, S., Toderici, G., Ricco, S., Sukthankar, R., Schmid, C., Malik, J., 2018. AVA: A Video Dataset of Spatio-Temporally Localized Atomic Visual Actions. CVPR.
He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep Residual Learning for Image Recognition. CVPR.
He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.B., 2020. Momentum Contrast for Unsupervised Visual Representation Learning. CVPR.
Jaiswal, A., Babu, A.R., Zadeh, M.Z., Banerjee, D., Makedon, F., 2020. A Survey on Contrastive Self-Supervised Learning. CoRR abs/2011.00362.
Jang, Y., Song, Y., Yu, Y., Kim, Y., Kim, G., 2017. TGIF-QA: toward Spatio-Temporal Reasoning in Visual Question Answering. CVPR.
Johnson, J., Karpathy, A., Fei-Fei, L., 2016. Densecap: Fully Convolutional Localization Networks for Dense Captioning. CVPR.
Józefowicz, R., Vinyals, O., Schuster, M., Shazeer, N., Wu, Y., 2016. Exploring the Limits of Language Modeling. CoRR abs/1602.02410.

Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., Suleyman, M., Zisserman, A., 2017. The Kinetics Human Action Video Dataset.
Krishna, R., Hata, K., Ren, F., Fei-Fei, L., Niebles, J.C., 2017a. Dense-captioning Events in Videos. ICCV.
Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L., Shamma, D.A., Bernstein, M.S., Fei-Fei, L., 2017b. Visual genome: connecting language and vision using crowdsourced dense image annotations. Int. J. Comput. Vis.
Kuehne, H., Jhuang, H., Garrote, E., Poggio, T.A., Serre, T., 2011. HMDB: A Large Video Database for Human Motion Recognition. ICCV.
Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R., 2020. ALBERT: A Lite BERT for Self-Supervised Learning of Language Representations. ICLR.
Lecun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. Proc. IEEE 86, 2278–2324. https://doi.org/10.1109/5.726791.
Lee, K., Chen, X., Hua, G., Hu, H., He, X., 2018. Stacked Cross Attention for Image-Text Matching. ECCV.
Lee, S., Yu, Y., Kim, G., Breuel, T.M., Kautz, J., Song, Y., 2021. Parameter Efficient Multimodal Transformers for Video Representation Learning. ICLR.
Lei, J., Yu, L., Bansal, M., Berg, T.L., 2018. TVQA: Localized, Compositional Video Question Answering. EMNLP.
Lei, C., Luo, S., Liu, Y., He, W., Wang, J., Wang, G., Tang, H., Miao, C., Li, H., 2021a. Understanding Chinese Video and Language via Contrastive Multimodal Pre-training arXiv:2104.09411.
Lei, J., Li, L., Zhou, L., Gan, Z., Berg, T.L., Bansal, M., Liu, J., 2021b. Less Is More: Clipbert for Video-And-Language Learning via Sparse Sampling. CVPR.
Li, Y., Song, Y., Cao, L., Tetreault, J.R., Goldberg, L., Jaimes, A., Luo, J., 2016. TGIF: A New Dataset and Benchmark on Animated GIF Description. CVPR.
Li, L., Chen, Y., Cheng, Y., Gan, Z., Yu, L., Liu, J., 2020. HERO: Hierarchical Encoder for Video+language Omni-Representation Pre-training. EMNLP.
Li, L., Lei, J., Gan, Z., Yu, L., Chen, Y.C., Pillai, R., Cheng, Y., Zhou, L., Wang, X.E., Wang, W.Y., Berg, T.L., Bansal, M., Liu, J., Wang, L., Liu, Z., 2021. Value: A Multi-Task Benchmark for Video-And-Language Understanding Evaluation arXiv: 2106.04632.
Lu, J., Batra, D., Parikh, D., Lee, S., 2019. Vilbert: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-And-Language Tasks. NeurIPS.
Luo, H., Ji, L., Shi, B., Huang, H., Duan, N., Li, T., Li, J., Bharti, T., Zhou, M., 2020. Univl: A Unified Video and Language Pre-training Model for Multimodal Understanding and Generation arXiv:2002.06353.
Michel, P., Levy, O., Neubig, G., 2019. Are Sixteen Heads Really Better than One? NeurIPS.
Miech, A., Zhukov, D., Alayrac, J., Tapaswi, M., Laptev, I., Sivic, J., 2019. Howto100m: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips. ICCV.
Miech, A., Alayrac, J., Smaira, L., Laptev, I., Sivic, J., Zisserman, A., 2020. End-to-end Learning of Visual Representations from Uncurated Instructional Videos. CVPR.
Pan, Y., Li, Y., Luo, J., Xu, J., Yao, T., Mei, T., 2020. Auto-captions on GIF: A Large-Scale Video-Sentence Dataset for Vision-Language Pre-training arXiv:2007.02375.
Prasanna, S., Rogers, A., Rumshisky, A., 2020. When BERT Plays the Lottery, All Tickets Are Winning. EMNLP.
Radford, A., Narasimhan, K., 2018. Improving Language Understanding by Generative Pre-training.
Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., 2019. Language Models Are Unsupervised Multitask Learners.
Rae, W., Potapenko, A., Jayakumar, S.M., Hillier, C., Lillicrap, T.P., 2020. Compressive Transformers for Long-Range Sequence Modelling. ICLR.
Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P., 2016. Squad: 100, 000+ Questions for Machine Comprehension of Text. EMNLP.
Ren, S., He, K., Girshick, R.B., Sun, J., 2015. Faster R-CNN: towards Real-Time Object Detection with Region Proposal Networks. NeurIPS.
Rohrbach, M., Amin, S., Andriluka, M., Schiele, B., 2012. A Database for Fine Grained Activity Detection of Cooking Activities. CVPR.
Sigurdsson, G.A., Varol, G., Wang, X., Farhadi, A., Laptev, I., Gupta, A., 2016. Hollywood in Homes: Crowdsourcing Data Collection for Activity Understanding. ECCV.
Soomro, K., Zamir, A.R., Shah, M., 2012. Ucf101: A Dataset of 101 Human Actions Classes from Videos in the Wild arXiv:1212.0402.
Sun, C., Shrivastava, A., Singh, S., Gupta, A., 2017. Revisiting Unreasonable Effectiveness of Data in Deep Learning Era. ICCV.
Sun, C., Myers, A., Vondrick, C., Murphy, K., Schmid, C., 2019. Videobert: A Joint Model for Video and Language Representation Learning. ICCV.
Sun, C., Baradel, F., Murphy, K., Schmid, C., 2020. Learning Video Representations Using Contrastive Bidirectional Transformer. ECCV.
Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A., 2017. Inception-v4, Inception-Resnet and the Impact of Residual Connections on Learning. AAAI.
Tan, H., Bansal, M., 2019. LXMERT: Learning Cross-Modality Encoder Representations from Transformers. CoRR abs/1908.07490.
Tang, Y., Ding, D., Rao, Y., Zheng, Y., Zhang, D., Zhao, L., Lu, J., Zhou, J., 2019. COIN: A Large-Scale Dataset for Comprehensive Instructional Video Analysis. CVPR.
Tang, Z., Lei, J., Bansal, M., 2021. Decembert: Learning from Noisy Instructional Videos via Dense Captions and Entropy Minimization. NAACL-HLT.
Tapaswi, M., Zhu, Y., Stiefelhagen, R., Torralba, A., Urtasun, R., Fidler, S., 2016. Movieqa: Understanding Stories in Movies through Question-Answering. CVPR.
Tay, Y., Dehghani, M., Bahri, D., Metzler, D., 2020. Efficient Transformers: A Survey arXiv:2009.06732.
Tsai, Y.H., Bai, S., Liang, P.P., Kolter, J.Z., Morency, L., Salakhutdinov, R., 2019. Multimodal Transformer for Unaligned Multimodal Language Sequences. ACL.
Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I., 2017. Attention is all you need. In: Advances in Neural Information Processing Systems.
Vinyals, O., Toshev, A., Bengio, S., Erhan, D., 2015. Show and Tell: A Neural Image Caption Generator. CVPR.
Wang, M., Ming, Y., Liu, Q., Yin, J., 2017. Image-based video retrieval using deep feature. In: 2017 IEEE International Conference on Smart Computing. SMARTCOMP, pp. 1–6. https://doi.org/10.1109/SMARTCOMP.2017.7947017.
Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S.R., 2018. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. EMNLP.
Warstadt, A., Singh, A., Bowman, S.R., 2018. Neural Network Acceptability Judgments arXiv preprint arXiv:1805.12471.
WL, T., 1953. “cloze procedure”: a new tool for measuring readability. Journal. Q.
Wu, Y., Schuster, M., Chen, Z., Le, Q.V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, L., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M., Dean, J., 2016. Google’s Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. CoRR abs/1609.08144.
Xie, S., Sun, C., Huang, J., Tu, Z., Murphy, K., 2018. Rethinking Spatiotemporal Feature Learning for Video Understanding.
Xu, J., Mei, T., Yao, T., Rui, Y., 2016. MSR-VTT: A Large Video Description Dataset for Bridging Video and Language. CVPR.
Xu, H., Ghosh, G., Huang, P., Arora, P., Aminzadeh, M., Feichtenhofer, C., Metze, F., Zettlemoyer, L., 2021. VLM: Task-Agnostic Video-Language Model Pre-training for Video Understanding. ACL.
Yang, L., Tang, K.D., Yang, J., Li, L., 2017. Dense Captioning with Joint Inference and Visual Context. CVPR.
Yu, Y., Kim, J., Kim, G., 2018. A Joint Sequence Fusion Model for Video Question Answering and Retrieval. ECCV.
Zadeh, A., Chen, M., Poria, S., Cambria, E., Morency, L., 2017. Tensor Fusion Network for Multimodal Sentiment Analysis. EMNLP.
Zellers, R., Bisk, Y., Schwartz, R., Choi, Y., 2018. SWAG: A Large-Scale Adversarial Dataset for Grounded Commonsense Inference. EMNLP.
Zhou, L., Xu, C., Corso, J.J., 2018a. In: McIlraith, S.A., Weinberger, K.Q. (Eds.), Towards Automatic Learning of Procedures from Web Instructional Videos. AAAI.
Zhou, L., Zhou, Y., Corso, J.J., Socher, R., Xiong, C., 2018b. End-to-end Dense Video Captioning with Masked Transformer. CVPR.
Zhou, L., Liu, J., Cheng, Y., Gan, Z., Zhang, L., 2021. Cupid: Adaptive Curation of Pre-training Data for Video-And-Language Representation Learning.
Zhu, L., Yang, Y., 2020. Actbert: learning global-local video-text representations. In: CVPR.
Zhu, Y., Kiros, R., Zemel, R.S., Salakhutdinov, R., Urtasun, R., Torralba, A., Fidler, S., 2015. Aligning Books and Movies: towards Story-like Visual Explanations by Watching Movies and Reading Books. ICCV.
Zhu, Y., Li, X., Liu, C., Zolfaghari, M., Xiong, Y., Wu, C., Zhang, Z., Tighe, J., Manmatha, R., Li, M., 2020. A Comprehensive Study of Deep Video Action Recognition arXiv:2012.06567.
Zhukov, D., Alayrac, J., Cinbis, R.G., Fouhey, D.F., Laptev, I., Sivic, J., 2019. Cross-task Weakly Supervised Learning from Instructional Videos. CVPR.