ActBERT: Learning Global-Local Video-Text Representations

Linchao Zhu[1,2] and Yi Yang[2*]
[1] Baidu Research [2] ReLER, University of Technology Sydney
{linchao.zhu,yi.yang}@uts.edu.au

Abstract

In this paper, we introduce ActBERT for self-supervised learning of joint video-text representations from unlabeled data. First, we leverage global action information to catalyze mutual interactions between linguistic texts and local regional objects. It uncovers global and local visual clues from paired video sequences and text descriptions for detailed visual and text relation modeling. Second, we introduce a TaNgled Transformer block (TNT) to encode three sources of information, i.e.*, global actions, local regional objects, and linguistic descriptions. Global-local correspondences are discovered via judicious clues extraction from contextual information. It enforces the joint video-text representation to be aware of fine-grained objects as well as global human intention. We validate the generalization capability of ActBERT on downstream video-and-language tasks,* i.e.*, text-video clip retrieval, video captioning, video question answering, action segmentation, and action step localization. ActBERT significantly outperforms the state-of-the-art, demonstrating its superiority in video-text representation learning.*

1. Introduction

While supervised learning has been successful in a variety of computer vision tasks [17, 9, 38, 29], self-supervised representation learning from unlabeled data has attracted increasing attention in recent years [4, 27]. In self-supervised learning, a model is first pre-trained on a large amount of unlabeled data with a surrogate loss. The fine-tuning process further helps the pre-trained model to be specialized in downstream tasks. Recently, there has been rapid progress in self-supervised representation learning for texts [7, 45], where the Bidirectional Encoder Representations from Transformers (BERT) model [7] generalizes remarkably to many natural language tasks, *e.g.*, question answering [2].

Motivated by BERT's success in self-supervised training, we aim to learn an analogous model for video and text joint modeling. We exploit video-text relations based on narrated instructional videos, where the aligned texts are detected by off-the-shelf automatic speech recognition (ASR) models. These instructional videos serve as natural sources for video-text relationship studies. First, they are vastly available and freely accessible on YouTube and other platforms [26, 33]. Second, the visual frames are aligned with the instructional narrations. The text narrations not only cover the objects in the scene explicitly but identify the salient action in the video clip.

To generalize BERT to video-and-language tasks, Sun *et al*. [33] extended the BERT model by learning from quantized video frame features. The original BERT takes discrete elements as inputs and predicts the corresponding tokens as the output. In contrast, visual features are distributed representations with real value, while the real-value features cannot be directly categorized into discrete labels for "visual token" prediction. Sun *et al*. [33] discretized visual features into visual words via clustering. These visual tokens can be directly passed to the original BERT model. However, detailed local information, *e.g.*, interacting objects, human actions would be possibly lost during clustering. It prevents the model from uncovering fine-grained relations between video and text. In this paper, we propose ActBERT to learn a joint video-text representation that uncovers global and local visual clues from paired video sequences and text descriptions. Both the global and the local visual signals interact with the semantic stream mutually. ActBERT leverages profound contextual information and exploits fine-grained relations for video-text joint modeling.

First, ActBERT incorporates global actions, local regional objects and text descriptions in a joint framework. Actions, *e.g.*, "cut", "rotate", "slice", are essential to various video-related downstream tasks. The recognition of human actions can demonstrate the model's capacity in motion understanding and complex human intention reasoning. It could be beneficial to explicitly model human actions during model pre-training. Long-term action sequences furthermore offer temporal dependencies about an instruc-

[*]This work was done when Linchao Zhu visited Baidu Research. Yi Yang is the corresponding author.

tional task. Though action clues are important, they are largely ignored in previous self-supervised video-text training [33, 26], where actions are treated identically to objects. To model human actions, we first extract verbs from the text descriptions and construct an action classification dataset from the original dataset. Then, a 3D convolution network is trained to predict the action labels. The features from the optimized network are used as the action embedding. In this way, clip-level actions are represented, and the corresponding action label is inserted. Besides global action information, we incorporate local regional information to provide fine-grained visual cues [21, 34, 32, 19, 5]. Object regions provide detailed visual clues about the whole scene, including the regional object feature, the position of the object. The language model can benefit from the regional information for better language-and-visual alignment.

Second, we introduce a TaNgled Transformer block (TNT) to encode features from three sources, *i.e*., global actions, local regional objects, and linguistic tokens. Previous studies [21, 34] consider two modalities when designing the new transformer layers, *i.e*., fine-grained object information from image and natural language. Lu *et al*. [21] introduced a co-attentional transformer layer, where the key-value pairs from one modality are passed to the other modality's attention block to act as the new key-value pairs. However, in our scenario, there are three sources of inputs. The two sources, *i.e*., local regional features and linguistic texts, offer detailed descriptions of the occurring event in the clip. The other global action feature provides the human intention in time-series as well as a straightforward clue for contextual inferring. We design a new tangled transformer block for cross-modality feature learning from three sources. To enhance the interactions between two visual cues and linguistic features, we use a separate transformer block [40] to encode each modality. The mutual cross-modal communication is later enhanced with two additional multi-head attention blocks. The action feature catalyzes mutual interactions. With the guidance from the action features, we inject visual information to the linguistic transformer, and incorporate linguistic information to the visual transformers. The tangled transformer dynamically selects judicious cues its context to facilitate the target prediction.

Furthermore, we design four surrogate tasks to train ActBERT, *i.e*., masked language modeling with global and local visual cues, masked action classification, masked object classification and cross-modal matching. The pre-trained ActBERT is transferred to five video-related downstream tasks, *i.e*., video captioning, action segmentation, text-video clip retrieval, action step localization, and video question answering. We quantitatively show ActBERT achieves the state-of-the-art performance with a clear margin.

2. Related Work

Video and language. There are many existing video-and-language tasks to evaluate the model's capacities in joint video-text representation learning, e.g., video question answering [36, 10, 18, 54], video captioning [46, 52], text-video retrieval [47, 41, 25], video grounding [50]. In video and language modeling, it can be difficult to learn relations between ordered video frames and their corresponding descriptions, where video temporal information and the interactions between multiple objects spatio-temporally requires to be incorporated. The dominant approach for multi-modal modeling is to leverage Recurrent Neural Networks (RNNs) and their variants, *e.g*., Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), to model sequence relations, *e.g*., [28, 53]. Zhou *et al*. [52] leveraged masked transformers in both the encoder and the decoder for dense video captioning. Most of these works are conducted on well-annotated datasets where the descriptions are manually generated, requiring considerable human interference. There are other works to learn video representations from limited annotated data [55]. The video data is a natural source to learn cross-modal representations. The text descriptions are automatically generated by off-the-shelf automatic speech recognition (ASR) models. This is more scalable and general to the model's deployment in real-world applications. In this paper, we focus on learning joint video-text representation in a self-supervised way.

Cross-modal pre-training. In the past year, many works extended BERT to model cross-modal data [21, 32, 34, 5, 19, 33]. The recent BERT model for video-text modeling [33] introduces visual words for video frames encoding, where local regional information is largely ignored. The synchronized video-audio signal is also a good test-bed for cross-modal representation learning [3, 15]. However, they leveraged low-level audio signals and only considered the synchronization nature of video data. In this work, we focus on video-text joint representation learning. Our ActBERT leverages multi-source information and achieves remarkable performance in many downstream video-text tasks.

Instructional videos. Learning from instructional videos is challenging due to its data complexity across various tasks [6, 1, 51, 26]. These videos are collected from many domains, *e.g*., cooking, sports, gardening. Many works also regard the transcriptions generated from instructional videos as a source of supervision [1, 51, 26]. However, we employ ActBERT to explicitly model human actions, local regions in a unified framework. We improve [26] with more specific relation modeling between videos and their description. We quantitatively demonstrated that ActBERT is more suitable for unsupervised video-text modeling.

3. Model Architecture

3.1. Preliminary

We first illustrate the original BERT [7] model. BERT [7] pre-trains a language model on large corpora in an unsupervised way. The pre-trained model is found to be flexible and beneficial to a variety of downstream tasks, *e.g*., question answering [2].

In BERT [7], the input entities are processed by a multi-layer bidirectional transformer [40]. The embeddings of each input are processed with stacked self-attention layers to aggregate contextual features. The attention weights are adaptively generated. The output features contain contextual information about the original input sequence. In self-attention, the generated features are irrelevant to input sequence order, and it enables the output representation to be permutation-invariant. The output representation is not affected when the input sequence is shuffled. A position embedding is commonly applied to each input entity for the incorporation of sequential order clues.

In the original BERT, Devlin *et al*. introduced two tasks for pre-training. In the task of masked language modeling (MLM), a portion of input words are randomly masked out. These masked-out words are replaced by a special token "[MASK]". The task is to predict the masked words based on the observations from the contextual contents. The contextual contents are unmasked elements that provide useful relevant cues for the prediction of the masked word.

The other task, *i.e*., Next Sentence Prediction (NSP), models order information between two sentences. Two sentences are sampled from a document, and NSP aims to identify if the second sentence is adjacent to the first sentence with the correct order. The two sentences are concatenated via a token "[SEP]", so that the models can be aware of the inputs being separated sentences. The prediction is made upon the output features of the first token "[CLS]". This is a binary classification problem, and a simple sigmoid classifier is used. A prediction of "1" indicates the sentences are consecutive, and the second sentence is right after the first sentence.

3.2. ActBERT

3.2.1 Input Embeddings

There are four types of input elements in ActBERT. They are actions, image regions, linguistic descriptions and special tokens. Special tokens are used to distinguish different inputs.

Each input sequence starts with a special token "[CLS]" and ends with another token "[SEP]". We put the linguistic descriptions after "[CLS]". There are the action inputs followed by local regional features. We denote the action features as $a_1, \ldots, a_L$, the frame region features as $r_1, \ldots, r_M$. The sequential text descriptions is denoted as $w_1, \ldots, w_N$. The whole sequence is denoted as $\{[\text{CLS}], w_1, \ldots, w_N, [\text{SEP}], a_1, \ldots, a_L, [\text{SEP}], r_1, \ldots, r_M, [\text{SEP}]\}$. "[SEP]" is also inserted between different sentences. We also insert "[SEP]" between regions that are from different clips, which can help the model to identify the clip boundaries. For each input step, the final embedding feature consists of four different embeddings. The embeddings are position embedding, segment embedding, token embedding, visual feature embedding. We added a few new tokens to distinguish action features and regional object features. The visual embedding is introduced to extract visual and action information. These embeddings are added to be the final feature of ActBERT. We explain them in detail as follows.

Position embedding. Following [7], we incorporate a learnable position embedding to every input in the sequence. Since self-attention does not consider order information, position encoding offers a flexible way to embed a sequence when the sequence order matters. For the actions in different clips, the position embeddings are different as the video clips are ordered. For the regions extracted from the same frame, we use the same position embedding. To distinguish regions from the same frame, we consider spatial position embedding for different spatial positions. The details will be described in "Visual (action) embedding".

Segment embedding. We consider multiple video clips for long-term video context modeling. Each video clip or video segment has a corresponding segment embedding. The elements, *i.e*., action inputs, regional object inputs, linguistic descriptions, have the same segment embedding in the same video clip.

Token embedding. Each word is embedded with WordPiece embeddings [42] with a 30,000 vocabulary. In addition to the special tokens mentioned above ("[CLS]", "[MASK]", "[SEP]"), we introduce "[ACT]" and "[REGION]" to represent the action features and the region features extracted from video frames, respectively. Note that all action inputs have the identical token embedding, which reveals the modality of the inputs.

Visual (action) embedding. We now explain the visual (action) embedding in details. We first illustrate the procedure to obtain the action embedding. For each video clip, we extract verbs from its corresponding descriptions. For simplicity, we remove clips that do not have any verbs. We then build a vocabulary from all the extracted verbs. After verb vocabulary construction, each video clip has one or multiple category labels. We train a 3D convolutional neural network on this constructed dataset. The inputs to the 3D network is a tensor that contains an additional temporal dimension. We leverage a softmax classifier on top of the convolutional neural network. For clips with multiple labels, we normalize the one-hot label with ℓ_1-norm, where

the scores for all labels are summed to be 1. After the model is trained, we extract the features after global average pooling as the **action features**. This feature can well represent the actions that occurred in the video clip.

To obtain regional object features, we extract bounding boxes and the corresponding visual features from a pre-trained object detection network. Similar to Lu *et al*. [21], we utilized pre-trained Faster R-CNN network [29] to extract the categorical distribution under the COCO vocabulary [20]. The image region features offer detailed visual information for visual and text relation modeling. For each region, the visual feature embeddings are the feature vectors before the output layer in the pre-trained network. Following [21], we incorporate spatial position embeddings to represent region locations with a 5-D vector. This vector consists of four box coordinates and the fraction of the region area. Specifically, we denote the vector as $(\frac{x_1}{W}, \frac{y_1}{H}, \frac{x_2}{W}, \frac{y_2}{H}, \frac{(x_2-x_1)*(y_2-y_1)}{W*H})$, where W is the frame width, H is the frame height, and (x_1, y_1) and (x_2, y_2) are the top-left and bottom-right coordinates, respectively.

This vector is then embedded to match the dimension of the visual feature. The final regional object feature is the summation of the spatial position embedding and the object detection feature.

3.2.2 Tangled Transformer

We design a TaNgled Transformer (TNT) to better encode three sources of information, *i.e*., action features, regional object features and linguistic features.

Instead of using only one transformer that treats the visual and text features equally, our tangled transformer consists of three transformers. The three transformers take three sources of features, respectively. To enhance the interactions between visual and linguistic features, we propose to inject visual information to the linguistic transformer and incorporate linguistic information to the visual transformers. With cross-modal interactions, the tangled transformer can dynamically select judicious cues for target prediction.

We denote the intermediate representations at transformer block l as $h^l = \{(h^l_{w_0}, \ldots, h^l_{w_N}), (h^l_{a_0}, \ldots, h^l_{a_L}), (h^l_{r_0}, \ldots, h^l_{r_M})\}$. For simplicity, we denote $h^l_w = \{h^l_{w_0}, \ldots, h^l_{w_N}\}$, $h^l_a = \{h^l_{a_0}, \ldots, h^l_{a_L})\}$, and $h^l_r = \{h^l_{r_0}, \ldots, h^l_{r_M})\}$, which are processed by w-transfomer, a-transformer, and r-transformer, respectively (Figure 1). Besides the standard multi-head attention encoding features from the same modality, we leverage the other two multi-head attention blocks to enhance mutual interactions between the transformer blocks. Specifically, we utilize h^l_a to catalyze mutual interactions. We denote the multi-head attention as $output = Multihead(Q, K, V)$, where Q is the query, K is the key, V is the value. The details of multi-head

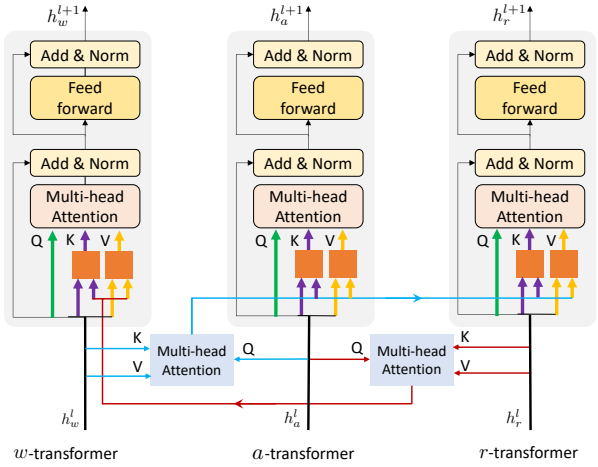

Figure 1: **Our tangled transformer** takes three sources of information as inputs, which enhances the interactions between linguistic features and visual features.

attention can be found in [40]. We use h^l_a as a query to attend judicious cues from h^l_w and h^l_r:

$$c_w = Multihead(W^1_q h^l_a, W^w_k h^l_w, W^w_v h^l_w), \quad (1)$$

$$c_r = Multihead(W^2_q h^l_a, W^r_k h^l_r, W^r_v h^l_r), \quad (2)$$

where W^*_* are learnable weights. c_w is the blended feature from linguistic representations, while c_r is the guided feature from regional object representation. We then generate a new key-value pair from c_w using a linear layer. This generated key-value pair is stacked with the key-value pairs from the original a-transformer and r-transformer. Similarly, we generate a new key-value pair from c_r, which is stacked with key-value pair in w-transformer. With this form tangled transformer, visual and linguistic features are further associated.

Note that our tangled transformer is different from the co-attentional transformer block in [21] in several ways. First, the co-attentional transformer block simply passes the keys and values from one modality to the other modality's attention block, without further pre-processing. Second, [21] treats the two modalities equally, while our tangled block utilizes a global cue to guide the selection of local hints from linguistic and visual features. Third, the keys and values from different modalities replace the origin key-values in [21], while our tangled transformer stacks the key-value with the original one. In this way, both the linguistic and visual features are incorporated during transformer encoding.

3.2.3 ActBERT Training

We introduce four tasks for ActBERT pre-training. Our framework is presented in Figure 2. We naturally extend

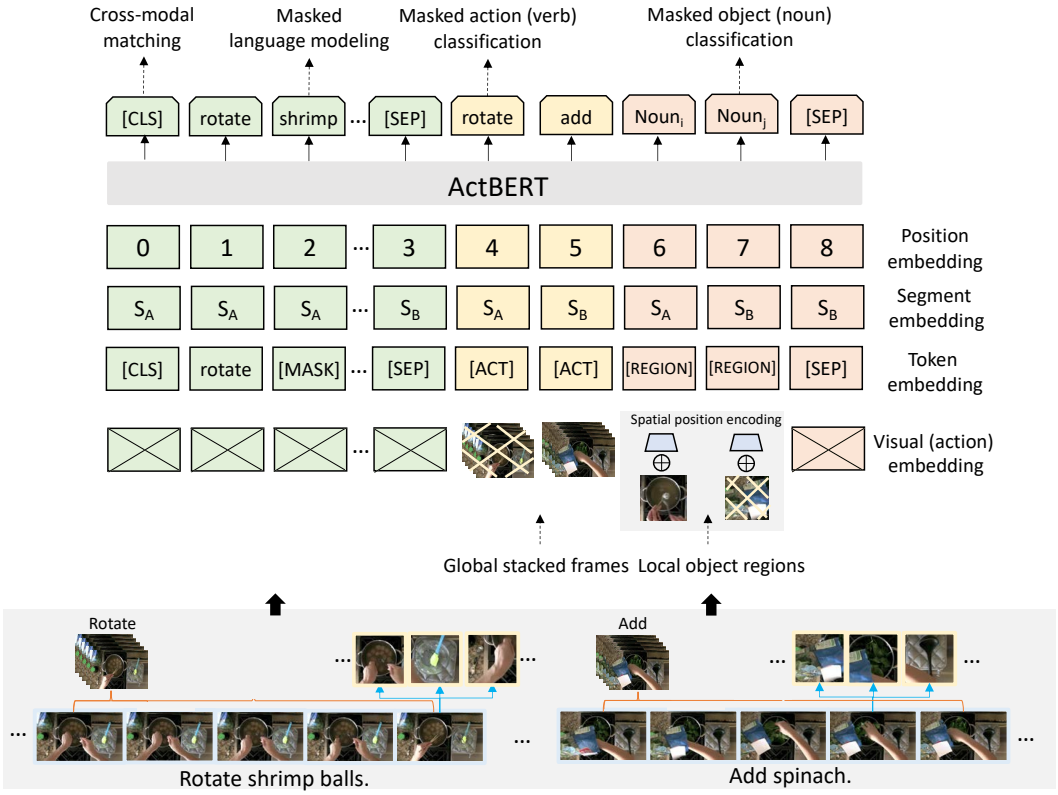

Figure 2: **Our ActBERT framework**. We incorporate three sources of information during pre-training, *i.e.*, global actions, local regional objects, and text descriptions. The yellow grid indicates that the action or the region object is masked out.

the Masked Language Modeling in our cross-modal setting. There are some existing extensions for image and language pre-training [21, 33], and video and language pre-training [33]. Compared to [33], we explicitly model actions and regional information in a unified framework.

Masked Language Modeling with Global and Local Visual Cues. We extend the Masked Language Modeling (MLM) task in BERT to our setting. We leverage visual cues from local regional objects and global actions to uncover the relationships between visual and linguistic entities. As described in Section 3.1, each word in the input sentence is randomly masked with a fixed probability. The task forces the model to learn from contextual descriptions, and at the same time, extract relevant visual features to facilitate prediction. When a verb word is masked out, the model should exploit the action features for a more accurate prediction. When a description of an object is masked out, local regional features can provide more contextual information. Thus, the strong model needs to align visual and linguistic inputs locally and globally. The output feature is then appended with a softmax classifier over the whole linguistic vocabulary.

Masked Action Classification. Similarly, in Masked Action Classification, the action features are masked out. The task is to predict the masked action label based on linguistic features and object features. Explicit action prediction can be beneficial in two perspectives. First, action sequential cues can be exploited in the long-term. For example, for a video with action sequences of "get into", "rotate", "add", this task can better exploit the temporal order information regarding performing this instructional assignment. Second, the regional objects and linguistic texts are leveraged for better cross-modality modeling. Note that in Masked Action Classification, the goal is to predict the categorical label of the masked-out action feature. This task can enhance the action recognition capability of the pre-trained model, which can be further generalized to many downstream tasks, *e.g.*, video question answering.

Masked Object Classification. In Masked Object Classification, the regional object features are randomly masked out. We follow [21] to predict a distribution over fixed vocabulary for the masked-out image region. The target distribution of the masked-out region is calculated as the softmax activation that is extracted by forwarding the region to the same pre-trained detection model in the feature extraction stage. The KL divergence between the two distributions is minimized.

Cross-modal matching. Similar to the Next Sentence Pre-

diction (NSP) task, we apply a linear layer on top of the output of the first token "[CLS]". It is followed by a sigmoid classifier, indicating the relevance score of the linguistic sentences and the visual features. If the score is high, it shows that the text well-describes the video clips. The model is optimized via a binary cross-entropy loss. To train this cross-modal matching task, we sample negative video-text pairs from the unlabeled dataset. We follow [26] for sampling positive pairs and negative pairs.

4. Experiments

In this section, we evaluate ActBERT in multiple downstream video-and-language tasks. We quantitatively evaluate the generalization capability of ActBERT on five challenging tasks, *i.e.*, text-video clip retrieval, video captioning, video question answering, action segmentation, and action step localization.

4.1. ActBERT implementation details

HowTo100M. We pre-train ActBERT on the HowTo100M dataset [26]. The HowTo100M dataset is constructed by querying YouTube API. The top 200 search results are kept. This dataset covers a total of 23,611 tasks, *e.g.*, maintenance and repair, animal rescue, food preparation. This dataset is biased towards actions, where the verbs like "go", "make", "come" being the most frequent. The nouns are also distributed in a long-tailed way, where objects like "water", "cup" are ranked top. Each video has a corresponding narration that is extracted from video subtitles. As the association between video clips and texts are not manually annotated, the video-text connection can sometimes be weak. There are cases of noisy correspondences, where the actors sometimes talk about unrelated things. Though noisy, we found pre-training on HowTo100M can still significantly improve the performance of downstream tasks.

Pre-training details. To construct video-text inputs for ActBERT pre-training, we sample video clips from the HowTo100M dataset. Instead of only using one clip for video-text joint training, we leverage multiple adjacent clips to cover a longer context. This enables ActBERT to model relations in different segments. We sample 10 adjacent video clips, and the temporal-aligned linguistic tokens are extracted to form a video-text pair.

To obtain the local regional features, we use Faster R-CNN pre-trained on the Visual Genome [16] dataset following [21]. The backbone is ResNet-101 [9]. We use the frame rate of 1 FPS to extract the regional features. Each region feature is RoI-pooled from the convolutional feature from that region. We set the detection confidence threshold as 0.4, and each frame contains at most five boxes. Transformer and co-attentional transformer blocks in the visual stream have hidden state size of 1024 and 8 attention heads.

To obtain the action features, we first construct an action classification dataset. We sample frames at 8 FPS. For each clip, we extract the verb from its text descriptions. Then, we train a ResNet-3D [39] network with a softmax classification loss. We initialized the weights of the ResNet-3D model from a pre-trained model on Kinetics [12]. The Kinetics dataset covers 400 actions from YouTube videos. The 3D convolutional network converges faster using when it is pre-trained on Kinetics. The input clip length to ResNet-3D is 32. The clip covers a 4-second video duration. The spatial shape of the input frame is 224×224. The initial learning rate is set to 0.001. The batch size is 16. We decay the learning rate by 0.1 at iteration 100,000, and the total number of training iterations is 1,000,000. We keep other training settings unchanged following [39]. During feature extraction, we sample the central clip, and each frame is central cropped. We use the feature after global average pooling as the clip representation.

During ActBERT pre-training, 15% of input features are randomly masked out. ActBERT has 12 layers of transformer blocks. Each transformer block has a hidden unit size of 768. We initialize the linguistic transformer with the BERT model pre-trained on the BookCorpus [56] and English Wikipedia. The other two transformers are randomly initialized. The network is optimized by Adam optimizer. We set the learning rate to be 10^{-5}. We trained the model for five epochs due to the large-scale data.

4.2. Results on video-and-text tasks

We evaluate ActBERT on five downstream tasks, *i.e.*, action step localization, action segmentation, text-video clip retrieval, video captioning, and video question answering. We evaluate the five tasks on CrossTask [57], COIN [35], YouCook2 [51], and MSR-VTT [44]. Videos from the test sets of these datasets are removed during pre-training on HowTo100M.

4.2.1 Datasets

CrossTask: We evaluate action step localization on the CrossTask [57] dataset. CrossTask [57] contains 83 tasks and 4.7k videos related to cooking, car maintenance, crafting, etc. We use the recall metric described in [57], which is defined by the number of step assignments that fall into the ground-truth interval, divided by the total number of steps in the video. **COIN:** We evaluate the action segmentation task on the recent COIN [35] dataset. COIN [35] contains 180 tasks and 11,827 videos. This dataset consists of 46,354 annotated segments. The videos are collected from YouTube. **YouCook2:** We evaluate text-video clip retrieval and video captioning on YouCook2. YouCook2 is a cooking video dataset collected from YouTube, covering a large variety of cooking styles, methods, ingredients and cookwares [51].

Method	BLEU-3	BLEU-4	METEOR	ROUGE-L	CIDEr
Zhou *et al*. [52]	7.53	3.84	11.55	27.44	0.38
S3D [43]	6.12	3.24	9.52	26.09	0.31
VideoBert [33]	6.80	4.04	11.01	27.50	0.49
VideoBert + S3D [33]	7.59	4.33	11.94	28.80	0.55
ActBERT	**8.66**	**5.41**	**13.30**	**30.56**	**0.65**

Table 1: **Video captioning** results on YouCook2. We outperform VideoBERT [33] across all the metrics.

Method	Frame Accuracy (%)
NN-Viterbi [30]	21.17
VGG [31]	25.79
TCFPN-ISBA [8]	34.30
ActBERT w/o region cues	52.10
ActBERT	**56.95**

Table 2: **Action segmentation** results on COIN.

In YouCook2, there are 89 types of recipes and totally 14k clips described with linguistic texts. Following [26], we evaluate the text-video clip retrieval task on the validation clips of YouCook2. **MSR-VTT:** We evaluate text-video clip retrieval and video question answering on MSR-VTT. The MSR-VTT dataset [44] is a general video dataset collected from YouTube with text descriptions. For the video question answering task, we evaluate the multiple-choice VideoQA following [47]. There are 2,990 questions in total for testing. Each test video is associated with a ground-truth caption, a correct answer, and four mismatched descriptions. For text-video clip retrieval, following [47], we use 1,000 pairs text-video for evaluation.

4.2.2 Video captioning

We compare our ActBERT to VideoBERT [33] on the video captioning task. We take the pre-trained action transformer as the video encoder. We follow the setup from [52] that takes the video clips from YouCook2 [51] as input, and a transformer decoder is used to decode videos to captions. We do not use the regional object transformer to fairly compare to [33]. Similar to [33], we cross-validate the hyper-parameters on the training set. We report the standard evaluation metrics for captioning, *i.e*., BLEU, METEOR, and ROUGE, on the validation set. The model is optimized by Adam optimizer for 40k iterations. We set the initial learning rate to 1.0×10^{-3}, and the batch size is 128. The results are shown in Table 1. We outperform VideoBERT [33] across all metrics, achieving a 1.36 improvement on METEOR. It demonstrates that our pre-trained transformer learns a better video representation. It also indicates the effectiveness of ActBERT in modeling video sequences by considering both global and local video cues. Our transformer generalizes better in video captioning.

4.2.3 Action segmentation

The action segmentation task in COIN is to design an action label for a video at the frame-level. To apply ActBERT to action segmentation, we fine-tune ActBERT by adding a linear classifier upon the output features for dense frame labeling. We do not feed the text descriptions during the fine-tuning process. The results are shown in Table 2. The baseline methods are conducted by [35]. Notably, ActBERT significantly outperforms the baselines with more than 20% improvements. It shows that the pre-trained ActBERT can deal with only visual inputs when linguistic descriptions are absent. When we remove the regional information, we observe a performance drop compared to our full model. It shows that detailed local cues are important to the dense frame labeling task.

4.2.4 Action step localization

We evaluate action step localization on CrossTask. To fairly compare to [26], we do not fine-tune on the target dataset. We regard the step action label as the text description and directly feed the text-video pair to ActBERT. We regard the prediction for the first token "[CLS]" as the relevance score of this clip belonging to the label. We choose the action with the max relevance score as the final prediction. The results are shown in Table 3. ActBERT significantly outperforms TVJE [26] with a large margin, *i.e*., the average improvement is 7%. We achieve even better than the supervised baseline. We remove the region cues to have a fair comparison to [26], as [26] does not use object detection features for video and text matching. The results of "ActBERT w/o region cues" also substantially outperform [26], demonstrating the effectiveness of ActBERT pre-training. Our full ActBERT model further improves performance by 4%. This validates that regional information is an important source that provides detailed local object features for text-and-video matching.

4.2.5 Text-video clip retrieval

We evaluate ActBERT on the task of video clip retrieval with natural language queries. Given a linguistic query, it aims to rank the video clips from a gallery video set. We use the following metrics for evaluation [26], *i.e*., Recall@1 (R@1), Recall@5 (R@5), Recall@10 (R@10) and the median rank (Median R). We evaluate ActBERT on YouCook2 and MSR-VTT. We followed [26] to conduct the YouCook2 evaluation. The results are shown in Table 4. ActBERT significantly outperforms TVJE [26] and other baselines.

	Make Kimchi Rice	Pickle Cucumber	Make Banana Ice Cream	Grill Steak	Jack Up Car	Make Jello Shots	Change Tire	Make Lemonade	Add Oil to Car	Make Latte	Build Shelves	Make Taco Salad	Make French Toast	Make Irish Coffee	Make Strawberry Cake	Make Pancakes	Make Meringue	Make Fish Curry	Average
Alayrac *et al*. [1]	15.6	10.6	7.5	14.2	9.3	11.8	17.3	13.1	6.4	12.9	27.2	9.2	15.7	8.6	16.3	13.0	23.2	7.4	13.3
Zhukov *et al*. [57]	13.3	18.0	23.4	23.1	16.9	16.5	30.7	21.6	4.6	19.5	35.3	10.0	32.3	13.8	29.5	37.6	43.0	13.3	22.4
Supervised [57]	19.1	25.3	38.0	37.5	25.7	28.2	**54.3**	25.8	18.3	31.2	47.7	12.0	39.5	23.4	30.9	41.1	**53.4**	17.3	31.6
TVJE [26]	33.5	27.1	36.6	37.9	24.1	35.6	32.7	35.1	30.7	28.5	43.2	19.8	34.7	33.6	40.4	41.6	41.9	27.4	33.6
ActBERT w/o region cues	37.4	29.5	39.0	42.2	29.8	37.5	35.5	37.8	33.2	32.8	48.4	25.2	37.4	35.6	42.4	47.0	46.1	30.4	37.1
ActBERT	**41.8**	**33.6**	**42.7**	**46.8**	**33.4**	**43.0**	**40.8**	**41.8**	**38.3**	**37.4**	**52.5**	**30.1**	**41.2**	**40.4**	**46.1**	**51.0**	**49.7**	**35.1**	**41.4**

Table 3: **Action step localization** results on CrossTask [57].

Method	Dataset	R@1	R@5	R@10	Median R
HGLMM [14]	YouCook2	4.6	14.3	21.6	75
TVJE [26]	YouCook2	4.2	13.7	21.5	65
TVJE +FT [26]	YouCook2	8.2	24.5	35.3	24
ActBERT	YouCook2	9.6	26.7	38.0	19
C+LSTM+SA [37]	MSR-VTT	4.2	12.9	19.9	55
VSE-LSTM [13]	MSR-VTT	3.8	12.7	17.1	66
SNUVL [48]	MSR-VTT	3.5	15.9	23.8	44
Kaufman *et al*. [11]	MSR-VTT	4.7	16.6	24.1	41
CT-SAN [49]	MSR-VTT	4.4	16.6	22.3	35
JSFusion [47]	MSR-VTT	10.2	31.2	43.2	13
TVJE [26]	MSR-VTT	7.5	21.2	29.6	38
ActBERT	MSR-VTT	8.6	23.4	33.1	36

Table 4: **Text-video clip retrieval** results on YouCook2 and MSR-VTT. "FT" denotes fine-tuning on the training set.

TVJE trains a ranking loss on the HowTo100M dataset. It shows ActBERT is a better pre-training framework for video-text joint representation learning. Notably, our pre-trained model achieves better retrieval performance than the finetuned TVJE model ("TVJE +FT") on YouCook2. It shows the superiority of ActBERT in self-supervised video-text representation learning. In MSR-VTT, ActBERT outperforms TVJE by 1.1% on R@1 when no labeled data is accessed. Note that JSFusion [47] is a supervised method that leverages labeled video and text pairs for training.

4.2.6 Video question answering.

We evaluate ActBERT on the multiple-choice VideoQA task. We fine-tune the pre-trained ActBERT on the MSR-VTT training set. The video-text pairs are fed to ActBERT. We use a linear classifier upon the output feature. We use a small learning rate of 0.0001 and use Adam optimizer for training. At the inference time, we fed each candidate with the video clip to ActBERT. The final choice is made by selecting the candidates with the max matching score. The results are shown in Table 5. We compare to many baselines in this task. Without fancy joint modeling, ActBERT significantly outperforms JSFusion [47] by 3%. It shows ActBERT's strong generalization from a large-scale dataset.

Method	Accuracy
Text-only BLSTM [22]	32.0
Text-only Human [22]	30.2
GoogleNet-2D + C3D [22]	35.7
Merging-LSTM [23]	34.2
SNUVL [48]	38.0
CT-SAN [49]	41.9
LR/RL LSTMs [24]	40.9
JSFusion [47]	45.5
ActBERT	**48.6**

Table 5: **Video question answering (multiple-choices)** results on MSR-VTT.

5. Conclusion

In this paper, we introduce ActBERT for joint video-text modeling in a self-supervised way. We directly model both global and local visual cues for fine-grained visual and linguistic relation learning. ActBERT takes three sources of information as input, *i.e*., global actions, local regional objects, and linguistic descriptions. The novel tangled transformer further enhances the communications between the three sources. Quantitative results on five video-text benchmarks demonstrate the effectiveness of ActBERT. In the future, we will consider evaluating ActBERT on video action recognition and detection. We will also improve ActBERT by designing more powerful modules for video and text modeling.

Acknowledgements. This work is supported by ARC DP200100938.

References

[1] Jean-Baptiste Alayrac, Piotr Bojanowski, Nishant Agrawal, Josef Sivic, Ivan Laptev, and Simon Lacoste-Julien. Unsupervised learning from narrated instruction videos. In *CVPR*, 2016. 2, 8

[2] Chris Alberti, Kenton Lee, and Michael Collins. A bert baseline for the natural questions. *arXiv preprint arXiv:1901.08634*, 2019. 1, 3

[3] Relja Arandjelovic and Andrew Zisserman. Objects that sound. In *ECCV*, 2018. 2

[4] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, 2018. 1

[5] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Learning universal image-text representations. *arXiv preprint arXiv:1909.11740*, 2019. 2

[6] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Scaling egocentric vision: The epic-kitchens dataset. In *ECCV*, 2018. 2

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 1, 3

[8] Li Ding and Chenliang Xu. Weakly-supervised action segmentation with iterative soft boundary assignment. In *CVPR*, pages 6508–6516, 2018. 7

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 6

[10] Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In *CVPR*, 2017. 2

[11] Dotan Kaufman, Gil Levi, Tal Hassner, and Lior Wolf. Temporal tessellation: A unified approach for video analysis. In *ICCV*, 2017. 8

[12] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 6

[13] Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539*, 2014. 8

[14] Benjamin Klein, Guy Lev, Gil Sadeh, and Lior Wolf. Associating neural word embeddings with deep image representations using fisher vectors. In *CVPR*, 2015. 8

[15] Bruno Korbar, Du Tran, and Lorenzo Torresani. Cooperative learning of audio and video models from self-supervised synchronization. In *NeurIPS*, 2018. 2

[16] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 123(1):32–73, 2017. 6

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012. 1

[18] Jie Lei, Licheng Yu, Mohit Bansal, and Tamara L Berg. Tvqa: Localized, compositional video question answering. *arXiv preprint arXiv:1809.01696*, 2018. 2

[19] Gen Li, Nan Duan, Yuejian Fang, Daxin Jiang, and Ming Zhou. Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. *arXiv preprint arXiv:1908.06066*, 2019. 2

[20] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 4

[21] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*, 2019. 2, 4, 5, 6

[22] Tegan Maharaj, Nicolas Ballas, Anna Rohrbach, Aaron Courville, and Christopher Pal. A dataset and exploration of models for understanding video data through fill-in-the-blank question-answering. In *CVPR*, 2017. 8

[23] Amir Mazaheri, Dong Zhang, and Mubarak Shah. Video fill in the blank with merging lstms. *arXiv preprint arXiv:1610.04062*, 2016. 8

[24] Amir Mazaheri, Dong Zhang, and Mubarak Shah. Video fill in the blank using lr/rl lstms with spatial-temporal attentions. In *ICCV*, 2017. 8

[25] Antoine Miech, Ivan Laptev, and Josef Sivic. Learning a text-video embedding from incomplete and heterogeneous data. *arXiv preprint arXiv:1804.02516*, 2018. 2

[26] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *ICCV*, 2019. 1, 2, 6, 7, 8

[27] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *ECCV*, 2016. 1

[28] Yingwei Pan, Tao Mei, Ting Yao, Houqiang Li, and Yong Rui. Jointly modeling embedding and translation to bridge video and language. In *CVPR*, 2016. 2

[29] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 1, 4

[30] Alexander Richard, Hilde Kuehne, Ahsan Iqbal, and Juergen Gall. Neuralnetwork-viterbi: A framework for weakly supervised video learning. In *CVPR*, 2018. 7

[31] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 7

[32] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. *arXiv preprint arXiv:1908.08530*, 2019. 2

[33] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *ICCV*, 2019. 1, 2, 5, 7

[34] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*, 2019. 2

[35] Yansong Tang, Dajun Ding, Yongming Rao, Yu Zheng, Danyang Zhang, Lili Zhao, Jiwen Lu, and Jie Zhou. Coin: A large-scale dataset for comprehensive instructional video analysis. In *CVPR*, 2019. 6, 7

[36] Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. Movieqa: Understanding stories in movies through question-answering. In *CVPR*, 2016. 2

[37] Atousa Torabi, Niket Tandon, and Leonid Sigal. Learning language-visual embedding for movie understanding with natural-language. *arXiv preprint arXiv:1609.08124*, 2016. 8

[38] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3D convolutional networks. In *ICCV*, 2015. 1

[39] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, 2018. 6

[40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 2, 3, 4

[41] Xin Wang, Jiawei Wu, Da Zhang, Yu Su, and William Yang Wang. Learning to compose topic-aware mixture of experts for zero-shot video captioning. In *AAAI*, 2019. 2

[42] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016. 3

[43] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *ECCV*, 2018. 7

[44] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *CVPR*, 2016. 6, 7

[45] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*, 2019. 1

[46] Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher Pal, Hugo Larochelle, and Aaron Courville. Describing videos by exploiting temporal structure. In *ICCV*, pages 4507–4515, 2015. 2

[47] Youngjae Yu, Jongseok Kim, and Gunhee Kim. A joint sequence fusion model for video question answering and retrieval. In *ECCV*, 2018. 2, 7, 8

[48] Youngjae Yu, Hyungjin Ko, Jongwook Choi, and Gunhee Kim. Video captioning and retrieval models with semantic attention. *arXiv preprint arXiv:1610.02947*, 6(7), 2016. 8

[49] Youngjae Yu, Hyungjin Ko, Jongwook Choi, and Gunhee Kim. End-to-end concept word detection for video captioning, retrieval, and question answering. In *CVPR*, 2017. 8

[50] Luowei Zhou, Yannis Kalantidis, Xinlei Chen, Jason J Corso, and Marcus Rohrbach. Grounded video description. In *CVPR*, 2019. 2

[51] Luowei Zhou, Chenliang Xu, and Jason J Corso. Towards automatic learning of procedures from web instructional videos. In *AAAI*, 2018. 2, 6, 7

[52] Luowei Zhou, Yingbo Zhou, Jason J Corso, Richard Socher, and Caiming Xiong. End-to-end dense video captioning with masked transformer. In *CVPR*, 2018. 2, 7

[53] Linchao Zhu, Zhongwen Xu, and Yi Yang. Bidirectional multirate reconstruction for temporal modeling in videos. In *CVPR*, 2017. 2

[54] Linchao Zhu, Zhongwen Xu, Yi Yang, and Alexander G Hauptmann. Uncovering the temporal context for video question answering. *IJCV*, 124(3):409–421, 2017. 2

[55] Linchao Zhu and Yi Yang. Compound memory networks for few-shot video classification. In *ECCV*, 2018. 2

[56] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *ICCV*, 2015. 6

[57] Dimitri Zhukov, Jean-Baptiste Alayrac, Ramazan Gokberk Cinbis, David Fouhey, Ivan Laptev, and Josef Sivic. Cross-task weakly supervised learning from instructional videos. In *CVPR*, 2019. 6, 8