VLTinT: Visual-Linguistic Transformer-in-Transformer for Coherent Video Paragraph Captioning

Kashu Yamazaki[*1], **Khoa Vo**[*1], **Quang Sang Truong**[1], **Bhiksha Raj**[2, 3], **Ngan Le**[1]

[1] AICV Lab, University of Arkansas, Fayetteville, Arkansas, USA
[2] Carnegie Mellon University, Pittsburgh, Pennsylvania, USA
[3] Mohammed bin Zayed University of AI
{ kyamazak, khoavoho, sangt, thile}@uark.edu, bhiksha@cs.cmu.edu

Abstract

Video Paragraph Captioning aims to generate a multi-sentence description of an untrimmed video with multiple temporal event locations in a coherent storytelling. Following the human perception process, where the scene is effectively understood by decomposing it into visual (e.g. human, animal) and non-visual components (e.g. action, relations) under the mutual influence of vision and language, we first propose a visual-linguistic (VL) feature. In the proposed VL feature, the scene is modeled by three modalities including (i) a global visual environment; (ii) local visual main agents; (iii) linguistic scene elements. We then introduce an autoregressive *Transformer-in-Transformer (TinT)* to simultaneously capture the semantic coherence of intra- and inter-event contents within a video. Finally, we present a new *VL contrastive loss function* to guarantee the learnt embedding features are consistent with the captions semantics. Comprehensive experiments and extensive ablation studies on the ActivityNet Captions and YouCookII datasets show that the proposed Visual-Linguistic Transformer-in-Transform (VLTinT) outperforms previous state-of-the-art methods in terms of accuracy and diversity. The source code is made publicly available at: https://github.com/UARK-AICV/VLTinT.

Introduction

Video captioning is the task of automatically generating a caption for a video. An important branch of video captioning is dense video captioning (DVC) (Krishna et al. 2017), which requires generating a list of temporal event proposals and the associated sentence description of each event to form a coherent paragraph description of a video. As a simplified version of DVC, video paragraph captioning (VPC) (Park et al. 2019) focuses on generating better paragraph captions given a set of event segments in a video, which eases the requirement of event proposal generation. In general, a VPC model consists of two main components: an encoder to represent each event segment as a feature; and a decoder to generate captions while maintaining the consistency within each event and the coherence among all sentences of the generated paragraph.

Videos contain rich semantic knowledge of multiple modalities, e.g., vision, text, speech, and non-speech audio. Understanding a video involves multiple factors such as a single human actor, group of human actors, non-human actor, and phenomenon (Vo et al. 2021b; Vo-Ho et al. 2021; Hutchinson and Gadepally 2021; Vo et al. 2023). Examples of non-human actors and phenomena performing action include dog chasing, car running, and cloud floating. The existing VPC approaches (Zhou et al. 2018; Dai et al. 2019; Lei et al. 2020) employ CNN-based networks as a black-box to encode the video feature, which could overlook the contributions of various modalities in the semantic contents of a video. We observe that human perception involves the *interaction of vision and language* and propose *VL Encoder* to resolve this above challenge. Our VL Encoder is based on two observations:

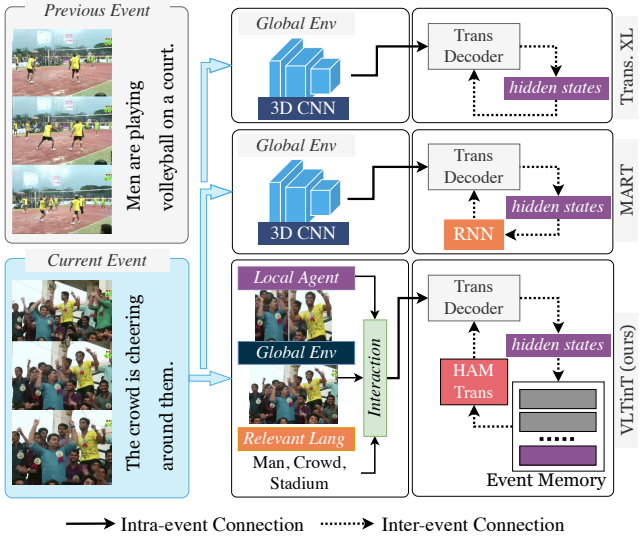

Figure 1: A high-level comparison between our VLTinT and recent SOTA VPC methods. In the encoder, both Transformer-XL (Dai et al. 2019) and MART (Lei et al. 2020) encode visual features by applying 3D CNN-based backbone network whereas our VLTinT encodes visual-linguistic feature by (i) global visual environment, (ii) local visual main agents, (iii) linguistic scene elements, and a fusion mechanism. In the decoder, Transformer-XL uses recurrence to address context fragmentation, MART uses a highly summarized memory to remember history information whereas we propose to utilize a transformer to model the contextual dependencies at both intra- and inter-levels.

[*]These authors contributed equally.

language influences the basic perceptual process, affecting performance on tasks that might seem to be wholly perceptual in nature (Lupyan et al. 2020); and video content is effectively understood by the combination of agents/actors and the surrounding environment (Vo-Ho et al. 2021; Vo et al. 2021b,a, 2022, 2023).

Our VL Encoder consists of four modalities: (i) global visual environment representing the overall surrounding scene, (ii) local visual main agents representing the human agents committing events, and (iii) linguistic scene elements captioning descriptive details of both visual and non-visual elements, and (iv) a fusion module modeling the interaction of those features and combine them into a unified representation. Besides, to only focus on the main agents who actually contribute to the event as well as the most relevant scene elements of the event, we make use of a Hybrid Attention Mechanism (HAM) following (Vo et al. 2021a, 2022, 2023).

In VPC, each event is described by one sentence, and they all should logically follow each other. Thus, two kinds of dependencies have to be modeled in VPC, i.e., intra- and inter-event dependencies. In the early days of development, RNN-based models were applied to build the caption generator to model intra-event coherency (Xiong, Dai, and Lin 2018; Park et al. 2019). Recently, Transformer-based models have proven to be more effective in generating captions (Dai et al. 2019; Lei et al. 2020; Ging et al. 2020; Yamazaki et al. 2022). However, in (Zhou et al. 2018), each event is decoded independently and inter-event coherency is not taken into account. This limitation is later addressed as a context fragmentation (Dai et al. 2019) and RNN-based memory (Lei et al. 2020; Ging et al. 2020; Yamazaki et al. 2022). However, none of the existing work leverages the success of the transformer in modeling inter-event coherence. To pose this challenge, we propose a novel *Transformer-in-Transformer architecture (TinT Decoder)*. To the best of our knowledge, TinT Decoder is the first fully transformer network for VPC, which simultaneously models both intra- and inter-event in an end-to-end framework. The network comparison between our VLTinT and the existing SOTA VPC approaches is in Fig. 1. Furthermore, most prior VPC work makes use of maximum likelihood estimation (MLE) loss to train the model. However, MLE loss does not guarantee that the learnt event embedding features intimately represent the groundtruth captions. Thus, we introduce a novel *VL contrastive loss*, to maintain the learning of both visual and linguistic semantics during training without adding additional computational costs. The VL Encoder along with TinT Decoder comprises a novel method, termed *Visual-Linguistic Transformer-in-Transformer (VLTinT)*. The main contributions of this paper are summarized as follows:

- A novel VL Encoder, which represents the video content by separately modeling (i) global visual feature, (ii) local visual main agents, and (iii) linguistic scene elements; and their interactions.
- A novel TinT Decoder to simultaneously model intra- and inter-event dependencies in an end-to-end fashion producing a coherent paragraph.
- A novel VL contrastive loss function to better align both visual and linguistic information.

Related Works

Dense Video Captioning

In general, video captioning can be divided into either single sentence (Pasunuru and Bansal 2017; Wang et al. 2019) for short videos or multiple sentences (Wang et al. 2021b) for long and untrimmed videos. DVC belongs to the second category and it has emerged as a multitask problem that combines event localization and event captioning to generate an informative caption for such videos. DVC can be implemented by visual feature only (Krishna et al. 2017; Li et al. 2018; Zhou et al. 2018; Mun et al. 2019; Deng et al. 2021) or multimodal features such as audio (Rahman, Xu, and Sigal 2019), speech (Shi et al. 2019; Iashin and Rahtu 2020), and both (Iashin and Rahtu 2020). Our VPC method shares a common setup with DVC with the multimodal feature. Our feature is encoded using both vision and language modalities to better extract contextual scene representation.

Video Paragraph Captioning

(Zhou et al. 2018) first introduced the transformer to the VPC task known as Vanilla Transformer, where each event is decoded individually without knowing the coherence between sentences. To address this limitation, (Lei et al. 2020) modified the Transformer-XL (Dai et al. 2019) and proposed MART. MART decodes the caption to learn word-level dependencies by a transformer while modeling the paragraph coherence based on GRU (Chung et al. 2014). Different from the existing VPC methods which utilize pre-trained backbone networks to extract feature, (Yamazaki et al. 2022) inherits the merits of both vision and language models and proposed VLCap. However, all previous works are RNN-based and limited in capturing long-range dependencies as well as suffers from the problem of gradient vanishing (Pascanu, Mikolov, and Bengio 2013). In this work, we leverage a transformer to simultaneously model the long-range dependencies between words (i.e., intra-event) and sentences (i.e., inter-event).

Transformer Models

Transformer (Vaswani et al. 2017) and Vision Transformer (ViT) (Dosovitskiy et al. 2021) have recently attracted significant interest in the research community. ViT applies a pure transformer to the visual recognition task by treating the image as a composition of 16×16 local patches. (Han et al. 2021) presents TNT to further divide them into smaller 4×4 patches. Specifically, TNT is built with a non-autoregressive inner and outer transformer, which deal with local sub-patches and sequence of local sub-patches, respectively. NesT (Zhang et al. 2022) proposes an alternative approach to model local and global information by nesting the canonical transformers hierarchically and connecting them with a proposed aggregation function. To model temporal coherency of intra- and inter-event, we propose a novel TinT. In our TinT, the outer transformer is designed as an autoregressive structure to model inter-event coherency whereas the inner transformer handles intra-event coherency.

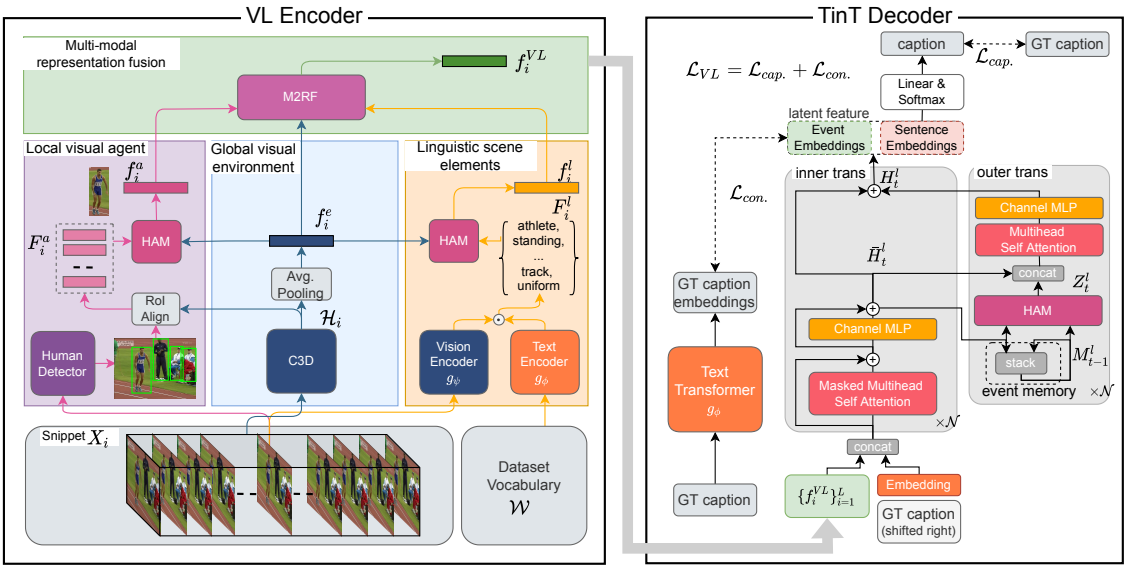

Figure 2: Overall network architecture of our proposed VLTinT, which contains two modules, i.e., VL Encoder and TinT Decoder. (Left) VL Encoder: given a snippet X_i, the VL Encoder simultaneously extracts local visual features from main agents, global visual features from the environment, and linguistic relevant scene elements; and models interaction between those three modalities through our M2RF module. (Right) TinT Decoder: the canonical transformer encoder is extended by an autoregressive outer transformer that can selectively access the 1^{st} to $t-1^{th}$ hidden states, which are stored in the event memory, at the t^{th} event captioning step.

Proposed VLTinT

Our VLTinT consists of two main modules corresponding to VL Encoder and TinT Decoder. The VL Encoder aims to extract VL representation of each event and the TinT Decoder aims to generate a caption of each event while simultaneously modeling intra- and inter-event coherency. Both modules are trained in an end-to-end fashion by our proposed VL loss. The over architecture is shown in Fig. 2.

Problem Setup

In VPC, we are given an untrimmed video $\mathcal{V} = \{v_i\}_{i=1}^{|\mathcal{V}|}$, where $|\mathcal{V}|$ is the number of frames, and a list of its important events $\mathcal{E} = \{e_i = (e_i^b, e_i^e)\}_{i=1}^{|\mathcal{E}|}$, where $|\mathcal{E}|$ is the number of events within a video and an event e_i is defined by a pair of beginning and ending timestamps (e_i^b, e_i^e). Our objective is to generate a coherent paragraph that matches the ground truth paragraph $\mathcal{P} = \{\mathbf{s}_i\}_{i=1}^{|\mathcal{E}|}$ that describes the whole video $\mathcal{V}$. In this setup, i^{th} sentence $\mathbf{s} = \{s_1 \dots s_N\}$ that consists of N words is the description of its corresponding event e_i.

Visual-Linguistic (VL) Encoder

Our VL Encoder is responsible for comprehensively representing each snippet X_i of an event into a representative feature to compose a sequence of snippet features for the decoder. Given an event $e = (e^b, e^e)$ and its corresponding video frames $\mathcal{V}_e = \{v_i | e^b \leq i \leq e^e\}$, we follow the standard settings from existing works (Zhou et al. 2018; Lei et al. 2020; Song, Chen, and Jin 2021) and divide $\mathcal{V}_e$ into a sequence of δ-frame *snippets* $\{X_i\}_{i=1}^{L}$. Each snippet X_i consists of δ consecutive frames and $\mathcal{V}_e$ has a total of $L = \lceil \frac{|\mathcal{V}_e|}{\delta} \rceil$ snippets.

The VL Encoder module encodes each snippet X_i to a VL representation f_i^{VL} as shown in Fig.2 (left). Therefore, video segment $\mathcal{V}_e$ is encoded into VL representation $\{f_i^{VL}\}_{i=1}^{L}$.

The VL Encoder first models a video with the three modalities, (i) global visual environment (ii) local visual main agents (iii) linguistic relevant scene elements, and then fuses them into one representation based on the interactions between them. Given a snippet X_i, it is encoded into these three modalities, corresponding to f_i^e, f_i^a and f_i^l, respectively. The final feature f_i^{VL} representing the interaction is extracted by fusing f_i^e, f_i^a and f_i^l through our Multi-modal Representation Fusion (M2RF) module as follows:

(i) Global Visual Environment:

This modality provides the visual semantic information from the entire spatial scene of input snippet X_i. To obtain such target, we adopt a backbone 3D-CNN network (Ji et al. 2013) to X_i to extract feature map $\mathcal{H}_i$ at the last convolutional block of the network. Then, we obtain the global environment visual feature $f_i^e \in \mathbb{R}^{d_{\text{emb}}}$ by processing $\mathcal{H}_i$ with an average pooling operation to reduce the entire spatial dimension followed by channel MLP. The procedure is summarized as follows:

$$f_i^e = \text{MLP}_{\theta_e}(\text{Avg.Pooling}(\mathcal{H}_i)) \quad (1)$$

(ii) Local Visual Main Agents:

This modality provides the visual features of the main human agents, who actually contribute to the formation of the event being described. Even though most of the events are associated with agents, not all agents committing movements are related to the main content of the event segment. Using a similar assumption as in (Vo-Ho et al. 2021; Vo et al. 2021b), we apply a human detector to the center frame of X_i to

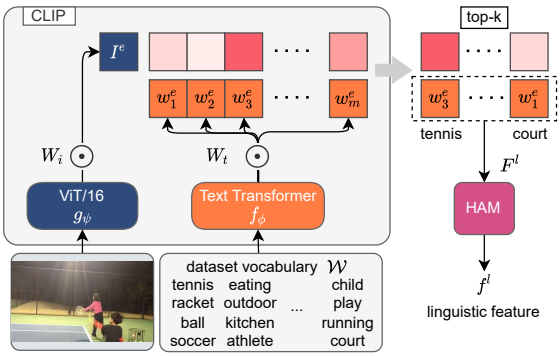

Figure 3: Illustration of relevant scene elements extraction process where ViT/16 and Text Transformer are the pre-trained models from CLIP (Radford et al. 2021).

obtain the bounding boxes of all human agents. Afterward, we align each of the detected bounding boxes $\mathcal{B}_i$ onto the feature map $\mathcal{H}_i$, which is obtained by the previous modality, using RoIAlign (He et al. 2017). Then, features overlapped by each agent bounding box are averagely pooled into a single feature vector to represent visual information of the agent inside that box. Finally, we obtain a set of local agent visual features $F_i^a \in \mathbb{R}^{N_a \times d_a}$, where N_a and d_a are the number of detected agents and agent embedding dimension, respectively. Finally, we apply HAM (detailed later) to adaptively select an arbitrary number of main agents from N_a detected agents and extract their mutual relationships to form a unified agent-aware visual feature $f_i^a \in \mathbb{R}^{d_{\text{emb}}}$ as follows:

$$f_i^a = \text{HAM}(\text{MLP}_{\theta_a}(F_i^a), f_i^e) \tag{2}$$

(iii) Linguistic Relevant Scene Elements:

This modality provides additional contextual details of the scene. While the two former modalities capture visual information of spatial appearances and temporal motions, their features may overlook some of the scene components because of the spacial reduction in the pooling operation from the feature map $\mathcal{H}_i$. Furthermore, non-visual features could hardly be captured by a normal vision backbone model. Recent studies (Patashnik et al. 2021; Yang, Zhang, and Zou 2022) have shown the extreme zero-shot capability of Contrastive Language-Image Pre-training (CLIP) where the model can estimate the semantic similarity between a set of words and an image. Trained on large-scale text-image pairs, CLIP can correlate not only the visual words but also the non-visual words to the given image. We thus leverage CLIP as a linguistic feature extractor to obtain top k scene elements (i.e., k texts) that are highly correlated with the middle frame of the input snippet X_i. Specifically, we construct a vocabulary $\mathcal{W} = \{w_1, \dots w_m\}$ based on the groundtruth captions of our training dataset. Each vocabulary $w_i \in \mathcal{W}$ is encoded by a transformer network f_ϕ into a text feature f_i^w. Let W_t be a text projection matrix pre-trained by CLIP, the embedding text vocabulary is computed as

$$w^e = W_t \cdot f_\phi(\mathcal{W}) = W_t \cdot f^w \text{where } f^w = \{f_i^w\}_{i=1}^m. \tag{3}$$

Let W_i be an image projection matrix pre-trained by CLIP, the center frame I of the input snippet X_i is first encoded by a pre-trained ViT g_ψ to extract visual feature f^I, and then embedded by W_i as below:

$$I^e = W_i \cdot g_\psi(I) = W_i \cdot f^I \tag{4}$$

The pairwise cosine similarities between embedded I^e and w^e are then computed. Top k similarity scores are chosen as linguistic categorical concept features $F_i^l \in \mathbb{R}^{k \times d_l}$. This feature is also subjected to the HAM module to select only the most relevant representative linguistic features and merge them into a single representation $f_i^l \in \mathbb{R}^{d_{\text{emb}}}$ as follows:

$$f_i^l = \text{HAM}(\text{MLP}_{\theta_l}(F_i^l), f_i^e) \tag{5}$$

The flowchart of extracting f_i^l is illustrated in Fig.3.

(iv) Multi-modal Representation Fusion (M2RF):

This component aims to fuse features from the three modalities. While concatenation or summation are the two common fusion mechanisms, they treat all modalities equally. To better model the impact of each individual modality, we propose M2RF as a function g_γ, which takes the features f_i^e, f_i^a, and f_i^l as its input. We extract the inter-feature relationships by utilizing a self-attention (SA) layer (Vaswani et al. 2017) followed by a mean operation. The final representation $f_i^{VL} \in \mathbb{R}^{d_{\text{emb}}}$ of a given snippet X_i is defined as follows:

$$f_i^{VL} = g_\gamma([f_i^e; f_i^a; f_i^l]) = \text{mean}(\text{SA}([f_i^e; f_i^a; f_i^l])) \tag{6a}$$

where $[;]$ represents the concatenation of features in a new dimension, where self-attention is applied on the new dimension and reduced by the mean operation to account for permutation invariance.

Transformer-in-Transformer (TinT) Decoder

Inspired by the recent transformer-based vision-language models (Chen et al. 2020b; Lei et al. 2020), we adopt the unified encoder-decoder transformer structure as a foundation for the caption generator, i.e., an inner transformer. The inner transformer's input is described as following. In this setup, video features $\mathcal{F}^{VL}$ is formed by concatenating all f_i^{VL} obtained by applying VL Encoder into each snippet X_i, i.e., $\mathcal{F}^{VL} = \{f_i^{VL}\}_{i=1}^L \in \mathbb{R}^{L \times d_{\text{emb}}}$. Textual tokens $\mathcal{F}^{text}$ is encoded by a pre-trained text transformer g_ϕ from CLIP and a MLP layer, i.e., $\mathcal{F}^{text} = MLP(g_\phi(\text{Shifted GT text})) \in \mathbb{R}^{N \times d_{\text{emb}}}$, where N is the sequence length of the text tokens. Following (Lei et al. 2020), learnable token type embeddings $\mathcal{F}^{type} \in \mathbb{R}^{(L+N) \times d_{\text{emb}}}$ are introduced to inform the location of the video and the caption representations. $\mathcal{F}^{type}$ is initialized as 0/1 vectors, i.e., video as 0 and text as 1. For the t^{th} event, an intermediate hidden states $\bar{H}_t^l \in \mathbb{R}^{(L+N) \times d_{\text{emb}}}$ is computed in Eq. 7b as canonical inner transformer encoder, where $\tilde{H}_t^l$ is the internal states after Masked Multihead Self Attention (MSA).

$$H_t^0 = [\mathcal{F}^{VL}; \mathcal{F}^{text}] + \mathcal{F}^{type} \in \mathbb{R}^{(L+N) \times d_{\text{emb}}} \tag{7a}$$

$$\bar{H}_t^l = \text{MLP}(\tilde{H}_t^l) + \tilde{H}_t^l, \tilde{H}_t^l = \text{MSA}(H_t^l) + H_t^l \tag{7b}$$

While the inner transformer can effectively model intra-event coherency, it cannot handle the contextual relationship of inter-event. To address this limitation, we introduce an

autoregressive outer transformer. The outer transformer selectively utilizes the activations of the inner transformer from the previous time steps for generating a coherent paragraph. Specifically, we take advantage of HAM to select only the most relevant hidden states of all previous events stored in event memory with respect to the current one. The outer transformer process is formulated below:

$$M_t^l = [M_{t-1}^l; \bar{H}_t^l] \tag{8a}$$
$$Z_t^l = \text{HAM}(M_{t-1}^l, \bar{H}_t^l) \tag{8b}$$
$$H_t^l = \text{MLP}(g_\gamma([\bar{H}_t^l; Z_t^l])) + \bar{H}_t^l \tag{8c}$$

For the t^{th} event, an intermediate hidden states $\bar{H}_t^l$ is stacked to the event memory $M_t^l \in \mathbb{R}^{t\times(L+N)\times d_{\text{emb}}}$, where $M_0^l = \varnothing$ as in Eq. 8a. Eq. 8b computes the context Z_t^l from the previous states of the event memory and the current intermediate hidden states $\bar{H}_t^l$ using HAM. Finally, in Eq. 8c, the context is integrated with the intermediate hidden states $\bar{H}_t^l$ using g_γ, which was introduced in Eq. 6a, and the hidden states are updated via residual connection. After the last layer, video token positions in $H_t^{\mathcal{N}}$ are ignored, and only the text token positions are fed to a feed-forward layer followed by softmax to predict a caption for the t^{th} event.

Hybrid Attention Mechanism (HAM)

HAM inherits the merits of both hard attention (Patro and Namboodiri 2018) and the self-attention (Vaswani et al. 2017) to select a rational number of representative features out of a set of input features and to extract mutual relationships among the sub-set of selected features, respectively, and fuse them into a unified representation. HAM was introduced by (Vo et al. 2021a) and it then has been successfully applied in to video analysis i.e. action localization (Vo et al. 2022, 2023). Fig. 4 visualizes the workflow of HAM, which is formulated as follows:

$$\mathcal{H}_{\text{in}} = \mathcal{F}_{\text{in}} \oplus f_{\text{ref}} \tag{9a}$$
$$\mathcal{C} = \text{softmax}(||\mathcal{H}_{\text{in}}||_2) \tag{9b}$$
$$\mathcal{M} = \mathcal{C} > \frac{1}{N_{in}} \tag{9c}$$
$$f_{\text{out}} = g_\gamma(\mathcal{F}_{\text{in}} \odot \mathcal{M}) \tag{9d}$$

where $\mathcal{F}_{\text{in}} \in \mathbb{R}^{N_{\text{in}}\times d_{\text{in}}}$ and $f_{\text{ref}} \in \mathbb{R}^{d_{\text{in}}}$ are a set of input features and a reference feature, respectively, where N_{in} is the total number of input features and d_{in} is the embedding dimension of input and reference features. HAM takes $\mathcal{F}_{\text{in}}$ and f_{ref} as inputs and compute the most relevant feature f_{out} as its output.

Visual-Linguistic (VL) Contrastive Loss

Typically, the existing VPC methods exploit the MLE loss to train their models. The MLE loss serves the objective of increasing the likelihood of predicted captions to be matched with the groundtruths. However, it is unable to address the question of how well the learnt event embedding features represent the groundtruth captions. To this end, we leverage the recent advantages of contrastive learning (Wu et al. 2018; Chen et al. 2020a) and propose $\mathcal{L}_{con}$ to pull all snippets of the same event and push snippets of different events. Our VL Loss consists of two terms corresponding to captioning loss ($\mathcal{L}_{cap.}$) and a contrastive contextual loss ($\mathcal{L}_{con.}$). While $\mathcal{L}_{cap.}$ aims to decode captions that match with groundtruths, $\mathcal{L}_{con.}$ guarantees the learnt latent features are close to the semantic information encoded in the groundtruth captions.

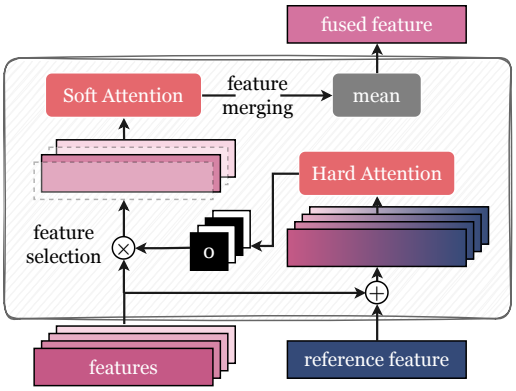

Figure 4: Illustration of HAM. HAM is capable of selecting and representing an arbitrary number of representative features from the input features $\mathcal{F}_{in}$ with a guidance from reference feature f_{ref}.

Captioning Loss $\mathcal{L}_{cap.}$: Kullback–Leibler (KL) divergence is commonly utilized to minimize the divergence between empirical distribution $p(\mathbf{s}|\mathcal{V}_e)$ and predicted distribution $p_\theta(\mathbf{s}|\mathcal{V}_e)$ for a video segment $\mathcal{V}_e$. However, this objective easily makes the captioning model overfit high-frequency tokens and phrases, which results in repetitive phrases. In order to enhance the smoothness of the predicted sentence, a regularization term τ is introduced to the training objective with hyper-parameter λ as:

$$\theta^* = \underset{\theta}{\text{argmin}}\, \mathbb{E}_{\mathbf{s}\sim p(\mathbf{s})}\left[\log\left(\frac{p(\mathbf{s})}{p_\theta(\mathbf{s})}\right) + \lambda\tau(\mathbf{s})\right] \tag{10}$$

The second term τ imposes a token-level high-frequency penalties as (Song, Chen, and Jin 2021). Based on the observation that the model tends to generate words that have been generated before, we penalize the previously appeared words in the regularization term:

$$\tau(\mathbf{s}) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c\in\{s|s_{<i}\}} \log\left(1 - p_\theta(c|s_{<i}, \mathcal{E})\right) \tag{11}$$

where c is the candidate word at n to be penalized.
Our $\mathcal{L}_{cap.}$ is defined as follows:

$$\mathcal{L}_{cap.} = -\frac{1}{N}\sum_{i=1}^{N}(\log p_\theta(s_i|s_{<i}, \mathcal{V}_e)) + \lambda\tau(\mathbf{s}) \tag{12}$$

where θ is the model parameters, $s_{1:N}$ is the target ground truth sequence.

Contrastive Contextual Loss $\mathcal{L}_{con.}$: We propose $\mathcal{L}_{con.}$ to optimize the latent feature of the input event to be highly

Methods	Venue	Input	B4 ↑	M ↑	C ↑	R ↑	Div2 ↑	R4 ↓
Vanilla Trans. (Zhou et al. 2018)	CVPR	Res200/Flow	9.75	15.64	22.16	28.90†	77.40†	7.79
AdvInf (Park et al. 2019)	CVPR	C3D/Object	10.04	16.60	20.97	–	–	5.76
GVD (Zhou et al. 2019)	CVPR	Res200/Flow/Object	11.04	15.71	21.95	–	–	8.76
Trans.-XL (Dai et al. 2019)	ACL	Res200/Flow	10.39	15.09	21.67	30.18†	75.96†	8.54
Trans.-XLRG (Lei et al. 2020)	ACL	Res200/Flow	10.17	14.77	20.40	–	–	8.85
MART (Lei et al. 2020)	ACL	Res200/Flow	10.33	15.68	23.42	30.32†	75.71†	5.18
PDVC (Wang et al. 2021a)	ICCV	C3D/Flow	11.80	15.93	27.27	–	–	–
VLTinT (ours)	–	C3D/Ling	**14.93**	**18.16**	**33.07**	**36.86**	**77.72**	**4.87**

Table 1: Performance comparison of VLTinT with other SOTA models on ActivityNet Captions *ae-val*. † denotes results by us.

Methods	Venue	Input	B4 ↑	M ↑	C ↑	R ↑	Div2 ↑	R4 ↓
Vanilla Trans. (Zhou et al. 2018)	CVPR	Res200/Flow	9.31	15.54	21.33	28.98†	77.29†	7.45
Trans.-XL (Dai et al. 2019)	ACL	Res200/Flow	10.25	14.91	21.71	30.25†	76.17†	8.79
Trans.-XLRG (Lei et al. 2020)	ACL	Res200/Flow	10.07	14.58	20.34	–	–	9.37
MART (Lei et al. 2020)	ACL	Res200/Flow	9.78	15.57	22.16	30.85†	75.69†	5.44
MART$^{\text{COOT}}$ (Ging et al. 2020)	NIPS	COOT	10.85	15.99	28.19	–	–	6.64
Memory Trans. (Song, Chen, and Jin 2021)	CVPR	I3D	11.74	15.64	26.55	–	**83.95**	**2.75**
VLTinT (ours)	–	C3D/Ling	**14.50**	**17.97**	**31.13**	**36.56**	77.72	4.75

Table 2: Performance comparison of VLTinT with other SOTA models on ActivityNet Captions *ae-test*. † denotes results by us.

correlated with its groundtruth description. This loss function implicitly encourages our VLTinT to learn better representations of the events and enhance its overall performance without extra computational cost.

Specifically, $\mathcal{L}_{con.}$ processes the entire mini-batch of training examples $\mathcal{B} = \{(\mathcal{V}_b, \mathbf{s}_b)\}_{b=1}^{|\mathcal{B}|}$, where $\mathcal{V}_b$ is a set of snippets within the same event and $\mathbf{s}_b$ is its corresponding groundtruth description sentence. On the one hand, video snippets in $\mathcal{V}_b$ are processed through our proposed VLTinT to obtain the event embeddings, which corresponds to the video token position $\mathcal{F}_b^{\mathcal{N}} \in \mathbb{R}^{L \times d_{\text{emb}}}$ of the final hidden state $H_b^{\mathcal{N}}$. On the other hand, we process each groundtruth caption sentence $\mathbf{s}_b$ through the transformer g_ϕ of CLIP (Radford et al. 2021) to obtain a representation feature $f_b^{\mathcal{T}} \in \mathbb{R}^{d_{\text{emb}}}$. Then, $\mathcal{L}_{con.}$ processes $\mathcal{F}_b^{\mathcal{N}}$ and $f_b^{\mathcal{T}}$ as follows:

$$\mathcal{L}_{con.} = -\sum_{b_1=1}^{|\mathcal{B}|}\sum_{b_2=1}^{|\mathcal{B}|}[\mathbb{1}_{b_1=b_2}\log(e^{\rho}(f_{b_1}^{\mathcal{N}}\cdot f_{b_2}^{\mathcal{T}})) + (1-\mathbb{1}_{b_1=b_2})(1-\log(e^{\rho}(f_{b_1}^{\mathcal{N}}\cdot f_{b_2}^{\mathcal{T}})))] \quad (13a)$$

where $f_b^{\mathcal{N}} = \text{mean}(\mathcal{F}_b^{\mathcal{N}})$. $\mathbb{1}_{b_1=b_2}$ returns 1 when samples come from the same event, i.e., $b_1 = b_2$ and 0 when samples come from the different events i.e., $b_1 \neq b_2$. ρ is a learnable temperature parameter initialized as $\log(1/0.07)$, to prevent scaling of the dot product values and stabilize the training.

Finally, our proposed VL contrastive loss $\mathcal{L}_{VL}$ is defined as:

$$\mathcal{L}_{VL} = \mathcal{L}_{cap.} + \mathcal{L}_{con.} \quad (14)$$

Experiments

Datasets and Metrics

We benchmark VLTinT on two popular datasets, ActivityNet Captions (Krishna et al. 2017) and YouCookII (Zhou, Xu, and Corso 2018). ActivityNet Captions consists of 10,009 training videos and 4,917 validation videos. We follow the previous work (Lei et al. 2020) to split the original validation set into two subsets: *ae-val* with 2,460 videos for validation and *ae-test* with 2,457 videos for testing. YouCookII contains 1,333 training and 457 validation videos. We report our results on the validation sets. We evaluate the performance on four standard metrics, i.e., BLEU@4 (B@4) (Papineni et al. 2002), METEOR (M) (Denkowski and Lavie 2014), CIDEr (C) (Vedantam, Zitnick, and Parikh 2015), ROUGE (R) (Lin 2004). Whereas to benchmark the diversity of generated captions, we use two diversity metrics, including 2-gram diversity (Div@2) (Shetty et al. 2017) and 4-gram repetition (R@4) (Xiong, Dai, and Lin 2018).

Implementation Details

To extract visual features of the environment, we use C3D (Ji et al. 2013) pre-trained on Kinetics-400 (Kay et al. 2017) as the backbone network. The agent visual feature is extracted by Faster-RCNN (Ren et al. 2015) that is pre-trained on the COCO dataset (Lin et al. 2014). To extract the linguistic scene element features, we employ CLIP (Radford et al. 2021) ViT-B/16 model made publically available by OpenAI. We set the hidden size to 768, the number of transformer layers to 3, and the number of attention heads to 12. Adam optimizer was used to train VLTinT with an initial learning rate of 1e-4, $\beta_1 = 0.9$, $\beta_2 = 0.999$, L_2 weight decay of 0.01, and learning rate warmup over the first 5 epochs. During the training, we

Methods	Venue	Input	B@4 ↑	M ↑	C ↑	R ↑	R@4 ↓
Vanilla Trans.(Zhou et al. 2018)	CVPR	Res200/Flow	4.38	11.55	38.00	–	–
MART (Lei et al. 2020)	ACL	Res200/Flow	8.00	15.90	35.74	–	4.39
MART^{COOT} (Ging et al. 2020)	NIPS	COOT	**9.44**	**18.17**	46.06	–	6.30
VLTinT (ours)	–	C3D/Ling	9.40	17.94	**48.70**	**34.55**	**4.29**

Table 3: Performance comparison of VLTinT with other SOTA models on YouCookII validation set.

			at-test split						*ae-val* split					
Env.	Agt.	Ling.	B@4 ↑	M ↑	C ↑	R ↑	Div@2↑	R@4 ↓	B@4 ↑	M ↑	C ↑	R ↑	Div@2 ↑	R@4 ↓
√	×	×	13.62	17.41	29.09	35.96	76.14	5.97	14.02	17.58	30.31	36.20	76.11	6.08
×	√	×	11.83	16.22	21.39	33.97	79.20	4.16	12.13	16.57	24.98	34.36	79.18	4.24
×	×	√	13.38	17.69	30.30	35.63	**80.50**	**3.32**	14.00	17.88	31.64	35.95	**80.44**	**3.22**
√	√	×	13.77	17.52	30.05	35.93	77.78	4.69	14.12	17.78	31.15	36.12	78.02	4.56
√	×	√	**14.53**	17.79	30.83	**36.67**	76.47	5.60	14.84	17.97	31.86	36.80	76.41	5.67
√	√	√	14.50	**17.97**	**31.13**	36.56	77.72	4.75	**14.93**	**18.16**	**33.07**	**36.86**	77.72	4.87

Table 4: Ablation study on the contribution of each modality in VL Encoder on ActivityNet Captions dataset. Env., Agt., and Ling. denote the global visual environment, local visual main agents, and linguistic relevant scene elements, respectively.

	B@4↑	M ↑	C ↑	R ↑	R@4 ↓
M RCNN	1.35	9.09	12.29	23.06	18.52
CLIP	**13.38**	**17.69**	**30.30**	**35.63**	**3.32**

Table 5: Performance comparison between two cases trained on TinT network without visual feature: (i) scene elements extracted by Mask R-CNN (M RCNN) (ii) scene elements extracted by CLIP.

use the label smoothing with a value of 0.1 and $\lambda = 0.1$. We ran the experiment on a single NVIDIA RTX 3090 (24GB) GPU.

Qualitative Analysis

Fig.5 shows comparison between VLTinT and Vanilla Transformer (VTrans) (Zhou et al. 2018) and MART (Lei et al. 2020). Overall, VLTinT can generate more descriptive captions with fine-grained details. In particular, we noticed that VTrans and MART are prone to use high-frequency words for their caption, while VLTinT can use expressive but less frequently appearing words, e.g., "A man" vs. "An athletic man" in the example. We attribute this improvement to our VL Encoder, which incorporates relative scene elements. We further observe a caption repetitiveness problem in VTrans and MART, which is handled our proposed TinT Decoder. Notably, with the same action (i.e., run down the track and jump into a sand pit), our VLTinT can tell when the action starts (i.e., begin) and happens (i.e., then). This is thank to the rich spatial information of VL Encoder and strong temporal coherency of TinT Decoder.

Quantitative Analysis

We benchmark and compare VLTinT with the prior SOTA VPC works on both ActivityNet Captions *ae-val*, *ae-test*, and YouCookII as in Tables. 1, 2 and 3, respectively. In those tables, we highlight the **best** and the second-best scores corresponding each metric. Compared to the SOTA approaches

VTrans: A man runs down a track and jumps into a sand pit. The man runs down the track and jumps into a sand pit.

MART: A man is running down a track and jumping into a sand pit. He jumps over a bar and lands in the sand.

VLTinT: An athletic man is seen standing ready and begins running down a track and jumping into a pit. The man then runs down the track and jumps into a sand pit.

GT: An athletic man is seen standing before a track and leads into him running down in a pit of sand. Several more clips are shown of the athletes running down the track and landing into a pit.

Figure 5: Qualitative comparison on ActivityNet Captions *ae-test* split. Red text indicates the captioning mistakes, purple text indicates repetitive patterns, and blue text indicates some distinct expressions.

	use of ling.	inter-event modeling	B@4 ↑	M ↑	C ↑	R ↑
ae-val	×	RNN	11.68	16.79	25.86	33.97
		Trans.	**14.12**	**17.78**	**31.15**	**36.12**
	√	RNN	13.75	17.63	28.01	36.21
		Trans.	**14.93**	**18.16**	**33.07**	**36.86**
ae-test	×	RNN	11.10	15.72	27.67	31.75
		Trans.	**13.77**	**17.52**	**30.05**	**35.93**
	√	RNN	13.45	17.42	29.68	36.09
		Trans.	**14.50**	**17.97**	**31.13**	**36.56**

Table 6: Comparison between RNN and Transformer to model inter-event dependencies in TinT decoder on ActivityNet Captions with C3D (env+agent) is visual feature in the encoder. Linguistic feature (Ling.) is considered as an option.

MART (Lei et al. 2020), MART w/COOT (Ging et al. 2020), and PDVC (Wang et al. 2021a), our VLTinT outperforms with large margins on both accuracy and diversity metrics on ActivityNet Captions. For example on *ae-val* split, accuracy gains 3.13%/1.56%/5.80%6.54% on B@4/M/C/R metrics

whereas diversity increases 0.32% on Div@2 and reduces 0.32% on R@4 compared to the second-best performance. On *ae-test* split, accuracy gains 3.65%/1.98%/2.94%5.71% on B@4/M/C/R metrics whereas diversity increases 0.43% on Div@2 and reduces 0.67% on R@4 compared to the second-best performance. On YouCookII, our performance is the best on C, R, and R@4 metrics with considerable gaps while it achieves compatible performance on B@4 and M metrics.

Ablation Studies

• **Contribution of each modality in VL Encoder:** We examine VLTinT on ActivityNet Captions with different modality settings as given in Table 4. The first three rows show the performance on each individual modality whereas the last three rows show the performance on different combinations. Even though the best performance on overall is obtained by combining all three modalities of both vision (environment and agent) and language (scene elements), the performance on only linguistic feature is promising with notable performance, especially on diversity metrics. This should be included in our future investigation.

• **Effectiveness of linguistic relevant scene elements:** We compare the performance of VLTinT with two cases given in Table 5: (i) scene elements extracted by Mask-RCNN trained on COCO with 80 classes (He et al. 2017) and (ii) scene elements extracted by CLIP. The ablation study shows the effectiveness of the scene elements feature extracted by CLIP over Mask-RCNN. While scene elements consist of human/non-human (e.g., animals, vehicles) and visual/non-visual (e.g., relations, activities) elements, Mask R-CNN can only cover a small portion of them because it was trained on a small number of visual objects/classes, resulting in poor diversity and lower performance on scene understanding compared to CLIP.

• **Robustness of TinT Decoder:** We examine the TinT Decoder with two settings of inter-event modeling, i.e., RNN-based similar to (Lei et al. 2020) and transformer-based (ours). The decoder is also considered with two encoder feature settings, i.e., with and without linguistic features whereas C3D (env+agent) is used as visual features. The result is shown in Table 6. Here we observe the substantial performance gain by modeling inter-event relationships by our autoregressive outer transformer.

• **Effectiveness of VL Loss $\mathcal{L}_{VL}$:** The effectiveness of VL Loss is examined by replacing $\mathcal{L}_{VL}$ with MLE loss, which is a common loss in VPC. The performance of VLTinT on ActivityNet Captions *ae-test* with two loss functions are reported in Table 7.

• **Computational Complexity:** We compare computational complexity vs. accuracy of our VLTinT with SOTA VPC models on the ActivityNet *ae-test* split. We report trainable params (millions), computation (GFLOPs), average inference time (seconds) over 100 random videos, and accuracy metrics in Table 8. In this comparison, we investigate our VLTinT with different settings. Compared to SOTA, our model with only env. has compatible params and inference time with better performance, whereas our model with env. & agent. & lang. gain big margins on accuracy while the complexities remain plausible.

Loss	B@4↑	M ↑	C ↑	R ↑
MLE	13.80	17.72	30.59	36.11
$\mathcal{L}_{VL}$ (ours)	**14.50**	**17.97**	**31.13**	**36.56**

Table 7: Effectiveness of $\mathcal{L}_{VL}$ compared to the standard MLE loss on ActivityNet Captions *ae-test*.

	Computational cost			Accuracy	
Models	Params↓	Comp. ↓	Inf.↓	M↑	C↑
MART	36.25	6.32	0.025	15.57	22.16
Mem Trans	29.69	256.44	0.706	16.10	27.36
E.	36.01	17.69	0.028	17.41	29.09
E./A.	40.37	22.70	0.032	17.52	30.05
E./A./L.	43.40	40.37	0.038	17.97	31.13

Table 8: Computational cost vs. accuracy between VLTinT (E./A./L.) with different settings and SOTA VPC models.

Conclusion

In this work, we have presented VLTinT, a novel model for VPC. The proposed network consists of VL Encoder and TinT Decoder. In VL Encoder, the video feature is extracted by three modalities, i.e., global visual environment, local visual main agents, and linguistic relevant scene elements; and they are fused through M2RF. In TinT Decoder, the intra-event coherency is modeled by the unified inner transformer and inter-event coherency is modeled by the autoregressive outer transformer. Our proposed VLTinT is designed as an end-to-end framework and trained by our proposed VL contrastive loss $\mathcal{L}_{VL}$. Comprehensive experiments and extensive ablation studies on ActivityNet Captions and YouCookII datasets have demonstrated the effectiveness of VLTinT, which outperforms the existing SOTA approaches on both accuracy (B@4, M, C, R) and diversity (Div@2, R@4) metrics.

Future investigations might include further examining linguistic feature in video understanding and exploring the VL Encoder in other video analysis problems.

Acknowledgments

This material is based upon work supported by the National Science Foundation (NSF) under Award No OIA-1946391, NSF 1920920, NSF FAIN-2223793 and NIH 1R01CA277739.

References

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020a. A Simple Framework for Contrastive Learning of Visual Representations. In *Proceedings of the 37th International Conference on Machine Learning*, 1597–1607. PMLR.

Chen, Y.-C.; Li, L.; Yu, L.; El Kholy, A.; Ahmed, F.; Gan, Z.; Cheng, Y.; and Liu, J. 2020b. UNITER: UNiversal Image-

TExt Representation Learning. In Vedaldi, A.; Bischof, H.; Brox, T.; and Frahm, J.-M., eds., *Computer Vision – ECCV 2020*, 104–120. Cham: Springer International Publishing. ISBN 978-3-030-58577-8.

Chung, J.; Gulcehre, C.; Cho, K.; and Bengio, Y. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning, December 2014*.

Dai, Z.; Yang, Z.; Yang, Y.; Carbonell, J.; Le, Q.; and Salakhutdinov, R. 2019. Transformer-XL: Attentive Language Models beyond a Fixed-Length Context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2978–2988. Florence, Italy: Association for Computational Linguistics.

Deng, C.; Chen, S.; Chen, D.; He, Y.; and Wu, Q. 2021. Sketch, Ground, and Refine: Top-Down Dense Video Captioning. In *CVPR*, 234–243.

Denkowski, M.; and Lavie, A. 2014. Meteor Universal: Language Specific Translation Evaluation for Any Target Language. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, 376–380. Baltimore, Maryland, USA: Association for Computational Linguistics.

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

Ging, S.; Zolfaghari, M.; Pirsiavash, H.; and Brox, T. 2020. COOT: Cooperative Hierarchical Transformer for Video-Text Representation Learning. In *Advances on Neural Information Processing Systems (NeurIPS)*.

Han, K.; Xiao, A.; Wu, E.; Guo, J.; XU, C.; and Wang, Y. 2021. Transformer in Transformer. In *Advances in Neural Information Processing Systems*, volume 34, 15908–15919. Curran Associates, Inc.

He, K.; Gkioxari, G.; Dollar, P.; and Girshick, R. 2017. Mask R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Hutchinson, M. S.; and Gadepally, V. N. 2021. Video Action Understanding. *IEEE Access*, 9: 134611–134637.

Iashin, V.; and Rahtu, E. 2020. Multi-Modal Dense Video Captioning. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 958–959.

Ji, S.; Xu, W.; Yang, M.; and Yu, K. 2013. 3D Convolutional Neural Networks for Human Action Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1): 221–231.

Kay, W.; Carreira, J.; Simonyan, K.; Zhang, B.; Hillier, C.; Vijayanarasimhan, S.; Viola, F.; Green, T.; Back, T.; Natsev, P.; Suleyman, M.; and Zisserman, A. 2017. The Kinetics Human Action Video Dataset. *CoRR*, abs/1705.06950.

Krishna, R.; Hata, K.; Ren, F.; Fei-Fei, L.; and Niebles, J. C. 2017. Dense-Captioning Events in Videos. In *International Conference on Computer Vision (ICCV)*.

Lei, J.; Wang, L.; Shen, Y.; Yu, D.; Berg, T.; and Bansal, M. 2020. MART: Memory-Augmented Recurrent Transformer for Coherent Video Paragraph Captioning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2603–2614. Online: Association for Computational Linguistics.

Li, Y.; Yao, T.; Pan, Y.; Chao, H.; and Mei, T. 2018. Jointly Localizing and Describing Events for Dense Video Captioning. In *CVPR*.

Lin, C.-Y. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, 74–81. Barcelona, Spain: Association for Computational Linguistics.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft COCO: Common Objects in Context. In Fleet, D.; Pajdla, T.; Schiele, B.; and Tuytelaars, T., eds., *Computer Vision – ECCV 2014*, 740–755. Cham: Springer International Publishing. ISBN 978-3-319-10602-1.

Lupyan, G.; Abdel Rahman, R.; Boroditsky, L.; and Clark, A. 2020. Effects of Language on Visual Perception. *Trends in Cognitive Sciences*, 24(11): 930–944.

Mun, J.; Yang, L.; Ren, Z.; Xu, N.; and Han, B. 2019. Streamlined Dense Video Captioning. In *CVPR*.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, 311–318. USA: Association for Computational Linguistics.

Park, J. S.; Rohrbach, M.; Darrell, T.; and Rohrbach, A. 2019. Adversarial Inference for Multi-Sentence Video Description. In *CVPR*, 6591–6601.

Pascanu, R.; Mikolov, T.; and Bengio, Y. 2013. On the difficulty of training recurrent neural networks. In Dasgupta, S.; and McAllester, D., eds., *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, 1310–1318. Atlanta, Georgia, USA: PMLR.

Pasunuru, R.; and Bansal, M. 2017. Multi-Task Video Captioning with Video and Entailment Generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1273–1283. Vancouver, Canada: Association for Computational Linguistics.

Patashnik, O.; Wu, Z.; Shechtman, E.; Cohen-Or, D.; and Lischinski, D. 2021. StyleCLIP: Text-Driven Manipulation of StyleGAN Imagery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2085–2094.

Patro, B.; and Namboodiri, V. P. 2018. Differential Attention for Visual Question Answering. In *CVPR*.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; and Sutskever, I. 2021. Learning Transferable Visual Models From Natural Language Supervision. In Meila, M.; and Zhang, T., eds., *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, 8748–8763. PMLR.

Rahman, T.; Xu, B.; and Sigal, L. 2019. Watch, Listen and Tell: Multi-Modal Weakly Supervised Dense Event Captioning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In Cortes, C.; Lawrence, N.; Lee, D.; Sugiyama, M.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Shetty, R.; Rohrbach, M.; Anne Hendricks, L.; Fritz, M.; and Schiele, B. 2017. Speaking the Same Language: Matching Machine to Human Captions by Adversarial Training. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Shi, B.; Ji, L.; Liang, Y.; Duan, N.; Chen, P.; Niu, Z.; and Zhou, M. 2019. Dense Procedure Captioning in Narrated Instructional Videos. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 6382–6391. Florence, Italy: Association for Computational Linguistics.

Song, Y.; Chen, S.; and Jin, Q. 2021. Towards Diverse Paragraph Captioning for Untrimmed Videos. In *CVPR*, 11245–11254.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L. u.; and Polosukhin, I. 2017. Attention is All you Need. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Vedantam, R.; Zitnick, C. L.; and Parikh, D. 2015. CIDEr: Consensus-based image description evaluation. In *CVPR*, 4566–4575.

Vo, K.; Joo, H.; Yamazaki, K.; Truong, S.; Kitani, K.; Tran, M.; and Le, N. 2021a. AEI: Actors-Environment Interaction with Adaptive Attention for Temporal Action Proposals Generation. In *32nd British Machine Vision Conference 2021, BMVC 2021, Online, November 22-25, 2021*, 111. BMVA Press.

Vo, K.; Truong, S.; Yamazaki, K.; Raj, B.; Tran, M.-T.; and Le, N. 2023. AOE-Net: Entities Interactions Modeling with Adaptive Attention Mechanism for Temporal Action Proposals Generation. *International Journal of Computer Vision*, 131(1): 302–323.

Vo, K.; Yamazaki, K.; Nguyen, P. X.; Nguyen, P.; Luu, K.; and Le, N. 2022. Contextual Explainable Video Representation: Human Perception-based Understanding. arXiv:2212.06206.

Vo, K.; Yamazaki, K.; Truong, S.; Tran, M.-T.; Sugimoto, A.; and Le, N. 2021b. ABN: Agent-Aware Boundary Networks for Temporal Action Proposal Generation. *IEEE Access*, 9: 126431–126445.

Vo-Ho, V.-K.; Le, N.; Kamazaki, K.; Sugimoto, A.; and Tran, M.-T. 2021. Agent-Environment Network for Temporal Action Proposal Generation. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2160–2164.

Wang, T.; Zhang, R.; Lu, Z.; Zheng, F.; Cheng, R.; and Luo, P. 2021a. End-to-End Dense Video Captioning With Parallel Decoding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 6847–6857.

Wang, T.; Zheng, H.; Yu, M.; Tian, Q.; and Hu, H. 2021b. Event-Centric Hierarchical Representation for Dense Video Captioning. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(5): 1890–1900.

Wang, X.; Wu, J.; Chen, J.; Li, L.; Wang, Y.-F.; and Wang, W. Y. 2019. VaTeX: A Large-Scale, High-Quality Multilingual Dataset for Video-and-Language Research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Wu, Z.; Xiong, Y.; Yu, S. X.; and Lin, D. 2018. Unsupervised Feature Learning via Non-Parametric Instance Discrimination. In *CVPR*.

Xiong, Y.; Dai, B.; and Lin, D. 2018. Move Forward and Tell: A Progressive Generator of Video Descriptions. In Ferrari, V.; Hebert, M.; Sminchisescu, C.; and Weiss, Y., eds., *Computer Vision – ECCV 2018*, 489–505. Cham: Springer International Publishing. ISBN 978-3-030-01252-6.

Yamazaki, K.; Truong, S.; Vo, K.; Kidd, M.; Rainwater, C.; Luu, K.; and Le, N. 2022. VLCAP: Vision-Language with Contrastive Learning for Coherent Video Paragraph Captioning. In *2022 IEEE International Conference on Image Processing (ICIP)*, 3656–3661.

Yang, B.; Zhang, T.; and Zou, Y. 2022. CLIP Meets Video Captioning: Concept-Aware Representation Learning Does Matter. arXiv:2111.15162.

Zhang, Z.; Zhang, H.; Zhao, L.; Chen, T.; Arik, S. ; and Pfister, T. 2022. Nested Hierarchical Transformer: Towards Accurate, Data-Efficient and Interpretable Visual Understanding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(3): 3417–3425.

Zhou, L.; Kalantidis, Y.; Chen, X.; Corso, J. J.; and Rohrbach, M. 2019. Grounded Video Description. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhou, L.; Xu, C.; and Corso, J. 2018. Towards Automatic Learning of Procedures From Web Instructional Videos. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Zhou, L.; Zhou, Y.; Corso, J. J.; Socher, R.; and Xiong, C. 2018. End-to-End Dense Video Captioning With Masked Transformer. In *CVPR*.