Guo, Zixin; Wang, Tzu-Jui Julius; Laaksonen, Jorma

Post-Attention Modulator for Dense Video Captioning

Published in:
Proceedings of the 26th International Conference on Pattern Recognition (ICPR)

DOI:
10.1109/ICPR56361.2022.9956260

Published: 01/01/2022

Document Version
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Post-Attention Modulator for Dense Video Captioning

Zixin Guo, Tzu-Jui Julius Wang, Jorma Laaksonen
Department of Computer Science, School of Science, Aalto University, Finland
{zixin.guo, tzu-jui.wang, jorma.laaksonen}@aalto.fi

***Abstract*—Dense video captioning (VC) aims at generating a paragraph-long description for events in video segments. Borrowing from the success in language modeling, Transformer-based models for VC have been shown effective also in modeling cross-domain video-text representations with cross-attention (Xatt). Despite Xatt's effectiveness, the queries and outputs of attention, which are from different domains, tend to be weakly related. In this paper, we argue that the weak relatedness, or domain discrepancy, could impede a model from learning meaningful cross-domain representations. Hence, we propose a simple yet effective Post-Attention Modulator (PAM) that post-processes Xatt's outputs to narrow the discrepancy. Specifically, PAM modulates and enhances the average similarity over Xatt's queries and outputs. The modulated similarities are then utilized as a weighting basis to interpolate PAM's outputs. In our experiments, PAM was applied to two strong VC baselines, VTransformer and MART, with two different video features on the well-known VC benchmark datasets ActivityNet Captions and YouCookII. According to the results, the proposed PAM brings consistent improvements in, e.g., CIDEr-D at most to 14.5%, as well as other metrics, BLEU and METEOR, considered.**

I. Introduction

Dense video captioning (VC) [1], [2] aims at automatically generating a human understandable text paragraph that describes the contents in the given video frames. Along with other vision and language (VL) models for various VL tasks, such as visual captioning [3], [4], [5], visual question answering (VQA) [6] and VL retrieval [7], models for VC are supposed to be capable of learning to align and reason from different modalities.

Inspired by the great improvement in attention modules [8], [9] introduced for resolving natural language understanding tasks, Transformer networks composed of self-attention (Satt) along with cross-attention (Xatt) layers have been proposed for several VC tasks [5], [10], [11]. These two attention mechanisms have been shown effective in modeling cross-modal representations. A typical way to utilize Xatt for VC is to take the textual representations as *queries* and video features as *keys* and *values* [5]. Conventionally, also a shortcut path is created from the textual queries to Xatt's outputs, which are then normalized with layer normalization [8].

Though the efficacy of Xatt has been demonstrated, the queries and keys lie in different domains, i.e. the queries are from language and the keys are from vision, inheriting large domain discrepancy in, e.g., Xatt's queries and outputs. For example, it could be seen from Fig. 1 that Xatt's queries and outputs tend to be weakly related in the first and second layers of VTransformer [5] and MART [11]. In line with [12], which shows that the relatedness between the queries and the attentive outputs can affect the captioning results, we argue that the weakly-relatedness could hinder the model from learning meaningful cross-domain representations.

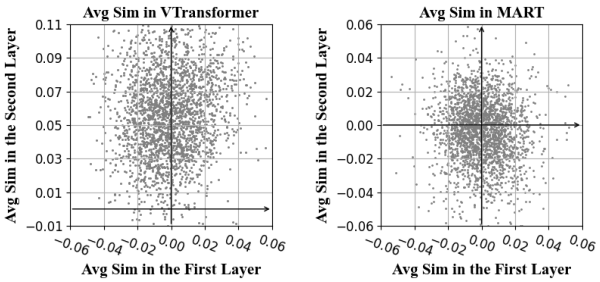

Fig. 1: The average cosine similarity between Xatt's queries and outputs of samples in VTransformer (left) and MART (right) on ActivityNet Captions.

We thereby propose Post-Attention Modulator (PAM) which enhances the similarities over Xatt's queries and outputs. PAM is placed after Xatt and refines Xatt's outputs along with Xatt's queries. Specifically, with their similarities used as the weighting basis, Xatt's queries and outputs are combined to interpolate PAM's outputs. The core of PAM comes with a similarity modulation scheme that modulates the similarities between Xatt's queries and outputs. We propose three different means realized by different loss functions with the similar aim.

PAM, when applied to two strong VC baselines VTransformer and MART, shows its effectiveness quantitatively and qualitatively on on two popular benchmark datasets: ActivityNet Captions [1] and YouCookII [2]. Better captioning results are similarly obtained with different types of pre-extracted features, such as appearance/motion features [5] and COOT video-text feature [13]. Our main contributions are summarized as follows:

1. We propose PAM along with three different modulation schemes. PAM is placed after Xatt for dense VC to refine Xatt's outputs where the refined outputs share greater similarity with Xatt's queries.

2. We assess PAM on two strong baselines on two well-known benchmark VC datasets with two pre-extracted video features. The empirical results demonstrate the effectiveness of PAM and some differences between the modulation schemes.

3. We provide qualitative analysis on how the similarities

between Xatt's queries and outputs correlate with the captioning accuracy.

II. Related Work

A wide range of Transformer-based [5], [11], [10] dense VC models have been proposed. Notably, Zhou et al. [5] proposed VTransformer which follows the architectures proposed in [8] for predicting sentences in the dense VC task. The model gains fluent continuity and coherence in sentence-level predictions while accurately capturing concrete events in segments[1]. However, while connecting the generated sentences into a paragraph, repetitions of words and erroneous logical order can usually be found because the model does not consider relations between the segments in the video. As such, the model's applicability is restricted due to context fragmentation, i.e. modelling each segment independently without interacting with its surrounding context.

A popular approach is to account for history information from video frames [14], [15]. Dai et al. [16] proposed Transformer-XL with a segment-level recurrence mechanism in which predictions of the sequences are dependent on the previous language segments. Motivated by this, Lei et al. [11] proposed a memory-augmented recurrent Transformer (MART) to summarize the previous video segments into the memory states which provides richer historic context to predict the next sentence. Notable improvements have been achieved, however, the model may be misled to generate incorrect captions by the attentive outputs that can be weakly related to the queries in the attention mechanism [12]. Attention on Attention (AoA) is then proposed for image captioning to filter out less relevant elements to the queries in the attentive outputs. Motivated by [12], we propose a novel Post-Attention Modulator (PAM) to be applied to Transformer-based models for dense VC problem. PAM refines Xatt's outputs with its queries by looking at the enhanced similarities between Xatt's queries and outputs. When applied to VTransformer, PAM enhances video-text relationships; when applied to MART, it jointly models the history context and the current segment more effectively.

III. Methods

Here we describe the proposed PAM and how it is used with the two selected baselines, i.e. VTransformer [5] and MART [11]. Given ordered video clips $[C_1, \cdots, C_T]$, either VTransformer or MART generates a paragraph consisted of sentences $[L_1, \cdots, L_T]$ describing the content of those clips. Note that VTransformer generates orderless sentences which are to be rearranged to produce the final paragraph; while MART, which predicts the next sentence conditioned on the previous video and sentence segments, always generates ordered sentences. In what follows, we introduce VTransformer and MART along which we highlight where PAM comes into play.

[1]A "segment" here, if not defined separately, refers to a video segment consisting of multiple frames.

A. Background

1) Vanilla Transformer: Vanilla Transformer (VTransformer) [5] is an extension of the Transformer [8] for the dense VC task. The scaled dot-product attention, formulated in Eq. (1), generates the weighted sum of the *value* matrix $V \in R^{s_k \times d}$ via calculating similarities between the *query* and *key* matrices, $Q \in R^{s_q \times d}$ and $K \in R^{s_k \times d}$, respectively:

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d}})V, \tag{1}$$

where s_k and s_q are the sequence lengths, and d is the hidden dimension. The multi-head attention, or $\text{MultiHead}(Q, K, V)$ [8], is usually used instead to enable modeling the representations in different subspaces. As shown in Fig. 2, the VTransformer is composed of several cascaded encoder and decoder layers. Self-attention (Satt), where the queries, keys and values are constructed from the same input features, is utilized in these layers. Cross-attention (Xatt), where the queries are from a different domain than the keys and values, is utilized in the decoder layers. Specifically, the encoder takes video as input and calculates correlations between video frames via Satt. In the decoder layer, masked Satt is operated over the outputs from the previous decoder layer. Masking is applied on each word's representation to prevent leaking the information to the network on the subsequent words to be modeled. Afterwards, Xatt between visual and textual modalities takes the masked Satt's outputs to construct queries, and outputs from the same level of encoder layers to construct keys and values. What follows is the feed-forward layer utilized for a more fine-grained representation. Moreover, the skip connection and layer normalization [17] are added to prevent the output from degeneration and to stabilize the hidden state dynamics, respectively [18].

2) Memory Augmented Recurrent Transformer: Different from Transformer and VTransformer [8], [5] that adopt separate encoder and decoder networks, Memory Augmented Recurrent Transformer (MART) [11] shares encoders and decoders, as shown in Fig. 2. A memory updater is proposed to encode the history information of video and caption segments in a recurrent way. Specifically, in the l^{th} decoder layer, the memory updater at step t takes its history value $M^l_{t-1} \in \mathbb{R}^d$ from the previous step $t-1$ and the hidden states $Y^l_t \in \mathbb{R}^{s_h \times d}$ from the feed-forward layer as inputs, and generates a new output M^l_t for the next step. Operations in the memory updater are formulated as:

$$M^l_t = (1 - \bar{Z}^l_t) \odot C^l_t + \bar{Z}^l_t \odot M^l_{t-1} \tag{2}$$

$$C^l_t = \tanh(W^l_{mc} M^l_{t-1} + W^l_{sc} S^l_t + b^l_c) \tag{3}$$

$$\bar{Z}^l_t = \text{sigmoid}(W^l_{mz} M^l_{t-1} + W^l_{sz} S^l_t + b^l_z) \tag{4}$$

$$S^l_t = \text{MultiHead}(M^l_{t-1}, Y^l_t, Y^l_t), \tag{5}$$

where $W^l_{mc}, W^l_{sc}, W^l_{mz}, W^l_{sz} \in \mathbb{R}^{d \times d}$ are the learnable weights, $b^l_c, b^l_z \in \mathbb{R}^d$ are biases, and $\odot$ is the element-wise product. Motivated by LSTM [19] and GRU [20], the update gate $\bar{Z}^l_t \in \mathbb{R}^d$ gauges history information's contribution.

B. Post-Attention Modulator (PAM)

1) PAM's Motivation: Xatt produces cross-modal representations via correlating queries and keys. However, large discrepancies may exist between Xatt's outputs and queries which are constructed across domains. This can be seen in their weak similarities in Fig. 1. Therefore, the residual shortcut path from the queries to Xatt's outputs can confuse the predictions. Thus, we propose Post-Attention Modulator (PAM) that better correlates Xatt's outputs with its queries to reduce the domain discrepancy. We introduce and study variants of PAM with three different modulation schemes.

2) PAM's Formulation: As seen in Eq. (1), Xatt's outputs are generated by positively weighing the *values* that reside in a modality different from the queries. Enforcing Xatt's queries and its outputs to be more correlated, e.g. sharing larger cosine similarity, could help better align the representations of the two modalities. To achieve this, we propose PAM to extend Xatt in Transformer to absorb the relevant information of the queries into its outputs. PAM takes Xatt's outputs X and its queries Y as inputs and generates Z as:

$$Z = g(X, Y), \tag{6}$$

where $g(\cdot, \cdot)$ is the combination function. Note that X, Y, Z represent different quantities in VTransformer and MART as will be described later in this section, hence, we intentionally specify their dimensionalities only later. PAM linearly weights $\mathbf{x}_i$ and $\mathbf{y}_i$, the i-th row vectors of X and Y, respectively, based on their cosine similarity $\alpha(\mathbf{x}_i, \mathbf{y}_i)$, to generate the output $\mathbf{z}_i$, the i-th row vector of Z. Formally,

$$\mathbf{z}_i = g(\mathbf{x}_i, \mathbf{y}_i) = (1 - \alpha(\mathbf{x}_i, \mathbf{y}_i))\, \mathbf{x}_i + \alpha(\mathbf{x}_i, \mathbf{y}_i)\, \mathbf{y}_i, \tag{7}$$

$$\alpha(\mathbf{x}_i, \mathbf{y}_i) = \frac{\mathbf{x}_i \cdot {\mathbf{y}_i}^T}{\|\mathbf{x}_i\|\, \|\mathbf{y}_i\|}. \tag{8}$$

$\mathbf{z}_i$ is interpolated and extrapolated when $\mathbf{x}_i$ and $\mathbf{y}_i$ are positively and negatively related, respectively, as shown in the vector plot in Fig. 2. When the similarity between $\mathbf{x}_i$ and $\mathbf{y}_i$ is weak, i.e. $\alpha(\mathbf{x}_i, \mathbf{y}_i) \approx 0$, as it initially inclines so (shown in Fig. 1), the PAM's outputs are close to Xatt's outputs, i.e. $\mathbf{z}_i \approx \mathbf{x}_i$. As such, PAM takes no effect and its outputs reduce to those from either VTransformer or MART.

C. Modulating $\alpha(\boldsymbol{x}_i, \boldsymbol{y}_i)$ in PAM

Recall that PAM is proposed for enhancing the correlation between Xatt's outputs and queries. To ensure this, one has to prevent $\mathbf{z}_i \approx \mathbf{x}_i$ in Eq. (7), otherwise, PAM is ineffective. We propose three different loss functions to modulate $\alpha(\mathbf{x}_i, \mathbf{y}_i)$.

1) Direct Modulation (DM): DM directly increases $\alpha(\mathbf{x}_i, \mathbf{y}_i)$ via:

$$\mathcal{L}_{DM} = \frac{1}{s_h} \sum_{l=1}^{L} \sum_{i}^{s_h} -\alpha(\mathbf{x}_i^l, \mathbf{y}_i^l), \tag{9}$$

where L is the number of encoder and decoder layers, and s_h is the number of row vectors in X, Y and Z. $\mathcal{L}_{DM}$ explicitly pulls Xatt's queries and outputs closer.

2) Query Modulation (QM): DM may make $\mathbf{z}_i$ too close to $\mathbf{y}_i$ if $\alpha(\mathbf{x}_i, \mathbf{y}_i)$ is enlarged too much. Hence, as the second approach, we propose a triplet loss $\mathcal{L}_{QM}$ to minimize the hinged difference between two similarity measurements, i.e. the similarity between PAM's output Z and Xatt's queries Y, and that between Xatt's outputs X and the same queries. Formally,

$$\mathcal{L}_{QM} = \frac{1}{s_h} \sum_{l=1}^{L} \sum_{i}^{s_h} \max(b_l + \alpha(\mathbf{x}_i^l, \mathbf{y}_i^l) - \alpha(\mathbf{z}_i^l, \mathbf{y}_i^l),\, 0), \tag{10}$$

where b_l is a positive (possibly layer-specific) scalar. $\mathcal{L}_{QM}$ increases $\alpha(\mathbf{z}_i^l, \mathbf{y}_i^l)$ as it encourages $\alpha(\mathbf{z}_i^l, \mathbf{y}_i^l) > \alpha(\mathbf{x}_i^l, \mathbf{y}_i^l) + b_l$. In other words, $\alpha(\mathbf{x}_i, \mathbf{y}_i)$, which weighs $\mathbf{y}_i$ in Eq. (7), emphasizes more on $\mathbf{y}_i$ to pull $\mathbf{z}_i$ and $\mathbf{y}_i$ closer. In addition, $\alpha(\mathbf{x}_i, \mathbf{y}_i)$ will be encouraged to be positive, otherwise, $\alpha(\mathbf{x}_i^l, \mathbf{y}_i^l) - \alpha(\mathbf{z}_i^l, \mathbf{y}_i^l) > 0$ (see Fig. 2). Furthermore, $\alpha(\mathbf{x}_i, \mathbf{y}_i)$ would not be enhanced to its extreme, i.e. $\alpha(\mathbf{x}_i, \mathbf{y}_i) \approx 1$, otherwise, it leads to $\mathbf{z}_i \approx \mathbf{y}_i$ and so $\alpha(\mathbf{z}_i^l, \mathbf{y}_i^l) \not> \alpha(\mathbf{x}_i^l, \mathbf{y}_i^l) + b_l$.

3) Key Modulation (KM): In our third variant KM, Xatt's queries $\mathbf{y}_i^l$ are replaced with keys K_j in Eq. (10):

$$\mathcal{L}_{KM} = \frac{1}{s_h s_k} \sum_{l=1}^{L} \sum_{i}^{s_h} \sum_{j}^{s_k} \max(b_l + \alpha(\mathbf{x}_i^l, K_j^l) - \alpha(\mathbf{z}_i^l, K_j^l),\, 0), \tag{11}$$

where K_j^l denotes the j^{th} row vector of the key matrix K^l of s_k rows. KM implicitly enhances $\alpha(\mathbf{x}_i^l, \mathbf{y}_i^l)$ by differentiating $\mathbf{x}_i$ and $\mathbf{z}_i$ via keys K_j^l as what it is realized by $\mathcal{L}_{KM}$. One can see from Eq. (7) that, in order to differentiate $\mathbf{x}_i$ and $\mathbf{z}_i$, more $\mathbf{y}_i$ should be included in $\mathbf{z}_i$, thus, calling for a larger $\alpha(\mathbf{x}_i, \mathbf{y}_i)$. b_l is in Eq. (11) to gauge the similarity difference which will be studied in Section IV-C.

The final training loss $\mathcal{L}_{train}$ is composed of the cross entropy loss $\mathcal{L}_{xe}$ over the generated and ground-truth word tokens and the modulation loss variants $\mathcal{L}_{cos} \in \{\mathcal{L}_{DM}, \mathcal{L}_{QM}, \mathcal{L}_{KM}\}$:

$$\mathcal{L}_{train} = \mathcal{L}_{xe} + \gamma_{cos}\, \mathcal{L}_{cos}, \tag{12}$$

where γ_{cos} is a non-negative hyperparameter whose value we will examine in Section IV.

D. Applying PAM in Transformers

1) PAM in VTransformer: PAM can be easily compatible with, but not limited to, VTransformer and MART. In Vtransformer, given video embeddings $V \in \mathbb{R}^{s_v \times d}$ and text embeddings $T \in \mathbb{R}^{s_t \times d}$, respectively for the encoder and decoder, the l^{th} encoder and decoder layers generate their self-attentive representations $\bar{V}^l$ and Y^l, respectively. s_v and s_t are the lengths of the video and text sequences, respectively. d is the dimension of the representations. Xatt in l^{th} decoder layer outputs X^l by attending to $\bar{V}^l$ and Y^l with the former being embedded as keys and values and the latter as queries, as shown in Fig. 2. PAM generates the refined representations Z^l based on Xatt's outputs X^l and queries Y^l.

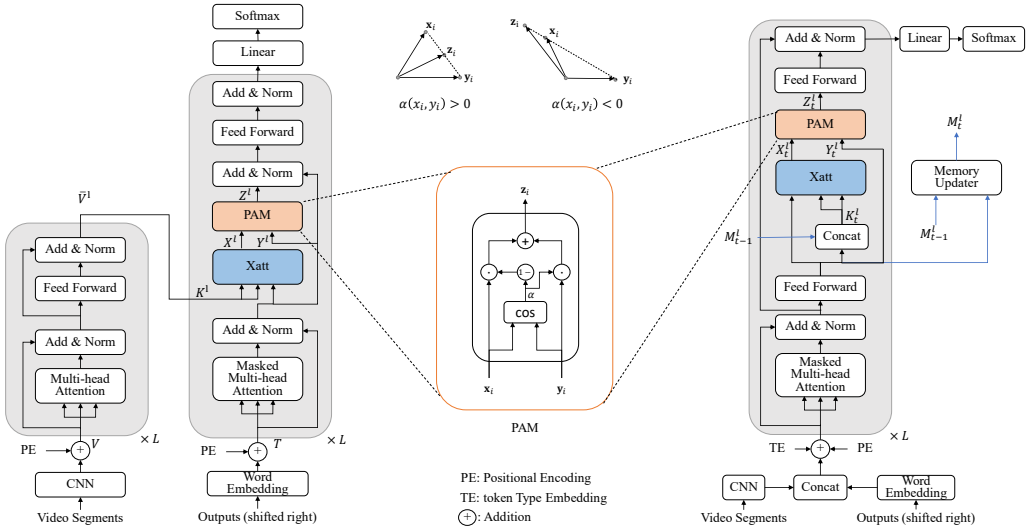

Fig. 2: The architectures of VTransformer and MART with the proposed Post-Attention Modulator (PAM). $\mathbf{x}_i, \mathbf{y}_i$, and $\mathbf{z}_i$ denote single row vectors in matrices X, Y, and Z, and scalar α is short for $\alpha(\mathbf{x}_i, \mathbf{y}_i)$ in Eq. (7). X and Y represent Xatt's outputs and queries, respectively. In the original VTransformer and MART, inputs Z^l are obtained directly from X^l without Y^l.

2) PAM in MART: MART takes the joint video-text embedding $H_t = [V_t, T_t] \in \mathbb{R}^{(s_v+s_t)\times d}$ as inputs at step t, where $[\cdot,\cdot]$ denotes the row-wise concatenation of two matrices. Xatt in l^{th} layer outputs X_t^l given the queries Y_t^l, i.e. the self-attentive video-text representations, and the keys and values being formed by $[Y_t^l, M_{t-1}^l]$, i.e. the concatenation of the queries and the previous memory state. Likewise, PAM outputs $Z_t^l = g(X_t^l, Y_t^l)$ as in VTransformer.

IV. Experiments

We evaluate PAM by applying it to two baseline models with two pre-extracted features on two benchmark datasets, ActivityNet Captions [1] and YouCookII [2].

A. Experimental Setup

1) Datasets: Table I summarizes the statistics of ActivityNet Captions [1] and YouCookII [2]. Each video is composed of several segments and each segment is described by a sentence. We assess the models on the *val* split on YouCookII. On ActivityNet Captions, we use the models with the highest CIDEr-D score [21] on the *ae-val* split and evaluate them on the *ae-test* split.

TABLE I: Statistics of the datasets. Num and Avg denote the number of videos and the average number of segments per video, respectively.

Dataset	Split	Num	Avg	Dataset	Split	Num	Avg
ActivityNet Captions	*train*	10,009	3.65	YouCookII	*train*	1333	7.7
	val	4,917					
	ae-val	2,460			*val*	457	
	ae-test	2,457					

2) Video Features: For fair comparison, we adopt the same pre-extracted and pre-processed video features, appearance and motion features (App+Mot) [11] and COOT video-clip features [13], as the baselines.

3) Evaluation Metrics: Following [11], [13], we consider the automated captioning metrics: BLEU (B) [22], METEOR (M) [23], CIDEr-D (C) [21], and Repetition (R) [14]. BLEU, METEOR and CIDEr-D measure the similarity between the candidate and reference sentences, while R score measures the amount of redundant information in the generated paragraph. As suggested in [11], we set $n = 4$ for the n-grams considered in BLEU and R, which are labeled as B@4 and R@4, respectively, throughout this section.

4) Implementation Details: All architectures have two encoder and decoder layers, i.e. $L = 2$, with hidden layers of size $d = 768$ and 10% dropout probability. All the Satt and Xatt layers have twelve attention heads. Adam [24] is used with initial learning rate $\eta = 1e-4$, β_1 value 0.9, β_2 value 0.999, $\mathcal{L}2$ weight decay value 0.01. The batch size is 16 segments. Models are trained from scratch for at most 50 epochs and CIDEr-D is monitored for early stopping for 10 epochs. As in [11], [13], greedy decoding with maximum 22 steps is applied while generating captions. We set the loss weight $\gamma_{cos} = 1$ in Eq. (12). We provide ablation studies on b_l and γ_{cos} in Section IV-C.

B. Results

1) Baselines: We compare our proposed PAM method against the baselines as follows:

Transformer-XL [16]: Transformer-XL models correlations between language segments with recurrence. It is adapted for video paragraph captioning tasks in [11]. Two variants are experimented: Transformer-XL and that with Recurrent Gradient (Transformer-XLRG). They differ in whether allowing gradient flow between different recurrent steps.

AoANet [12]: Originally proposed for image captioning, AoANet, in which another attention is added on Xatt for filtering out irrelevant elements in Xatt's outputs to its queries, is adapted here for the dense VC task.

VTransformer [5]: VTransformer generates unordered sentences, which are then re-organized to make a paragraph, given video segments. We use the same model also adpoted in [11].

MART [11]: MART devices a memory module to augment the model's representations with information from the history video and text segments. The sentences in the paragraph are recurrently generated in the same order as the video segments.

2) Quantitative Results: Table II shows the results on YouCookII *val* and ActivityNet Captions *ae-test* splits.

TABLE II: Captioning results on the YouCookII *val*, ActivityNet Captions *ae-test* split. Results of models with an asterisk (*) are reported in [11], [13]. Results with a dagger (†) are our reproduction. Bold and underlined indicate the best and the second best results, respectively.

Model	YouCookII *val* b_l	B@4	M	C	R@4↓	ActivityNet Captions *ae-test* b_l	B@4	M	C	R@4↓
Feature: App+Mot										
Transformer-XL* [11]	–	6.56	14.76	26.35	6.30	–	10.25	14.91	21.71	8.79
Transformer-XLRG* [11]	–	6.63	14.74	25.93	6.03	–	10.07	14.58	20.34	9.37
AoANet† [12]	–	7.54	15.88	32.02	3.38	–	9.85	16.21	22.15	7.32
VTransformer* [11]	–	7.62	15.65	32.26	7.83	–	9.31	15.54	21.33	7.45
VT+PAM DM	–	7.56	**15.94**	34.85	**3.35**	–	9.84	15.65	22.95	**5.85**
VT+PAM QM	0.30	**8.11**	15.66	35.32	5.94	0.25	**10.01**	**15.98**	**23.20**	8.39
VT+PAM KM	0.55	7.61	15.71	**36.94**	4.13	0.55	10.00	15.62	23.07	6.38
MART* [11]	–	8.00	15.90	35.74	4.39	–	9.78	15.57	22.16	**5.44**
MART+PAM DM	–	7.77	15.91	36.00	3.63	–	10.26	15.80	23.10	6.30
MART+PAM QM	0.20	**8.12**	**16.00**	36.86	4.98	0.05	10.27	15.77	23.53	6.11
MART+PAM KM	0.55	8.00	**16.00**	**38.10**	**2.76**	0.55	**10.41**	**15.81**	**23.90**	6.47
Feature: COOT										
Transformer-XL* [13]	–	–	–	–	–	–	10.57	14.76	22.04	15.85
Transformer-XLRG†	–	–	–	–	–	–	10.62	14.85	23.99	12.46
VTransformer* [13]	–	11.09	19.34	54.67	**4.57**	–	10.47	**15.76**	25.90	19.14
VT+PAM DM	–	11.05	19.34	57.05	7.80	–	10.54	15.68	26.45	18.58
VT+PAM QM	0.30	**11.42**	**19.86**	57.68	6.08	0.25	10.60	15.70	26.58	17.99
VT+PAM KM	0.60	11.19	19.50	**58.20**	6.76	0.55	**10.68**	15.75	**26.86**	**17.58**
MART* [13]	–	11.30	19.85	57.24	6.69	–	10.85	15.99	28.19	6.64
MART+PAM DM	–	11.60	19.71	57.97	6.55	–	11.11	15.84	28.33	8.86
MART+PAM QM	0.20	**11.85**	19.82	**60.83**	6.77	0.20	11.18	15.99	29.10	**6.57**
MART+PAM KM	0.55	11.68	**20.02**	58.34	**5.30**	0.55	**11.31**	**16.00**	**29.77**	7.16

Assessing PAM. One can firstly observe that PAM DM obtains consistent increments on VTransformer and MART while PAM QM and KM bring further improvements in most of the cases. For instance, PAM QM and KM improve VTransformer by $8.1\% - 14.5\%$ on C scores with App+Mot and $2.6\% - 6.4\%$ with COOT features across benchmark datasets. On a stronger baseline MART, C scores are improved by $3.1\% - 7.8\%$ with App+Mot and $1.9\% - 6.2\%$ with COOT. It is worth noting that larger improvements from adding PAM on VTransformer than on MART can be seen in the most cases, showing PAM's capability on improving originally less performant models. Of all the modulation schemes, PAM KM often delivers the best scores. Putting these improvements aside, PAM, nevertheless, occasionally seems to generate more repetitive words as indicated by slightly larger R scores.

Effect on Features: App+Mot vs. COOT. VTransformer and MART with COOT embeddings generate more accurate captions than with appearance and motion feature in all cases. Models with PAM similarly improve both baselines in B@4, M and C with both features.

3) Significance of Differences: We examine the differences in the C scores brought by the models' random initializations with the box plots showing in Fig. 3. The PAM variants' effectiveness is verified by their higher lower bounds on the C scores compared to the upper bounds of the VTransformer and MART baselines (bl), respectively.

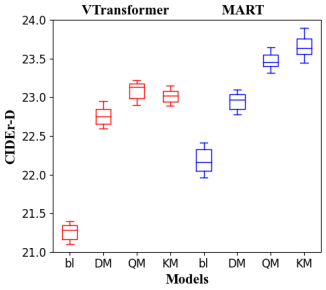

Fig. 3: Effect of models' seven random initialization on ActivityNet Captions *ae-test* split with the App+Mot feature.

C. Ablation Studies

We study how b_l in Eqs. (10-11) and the loss weight γ_{cos} in Eq. (12) affect the model behavior and performance.

1) Effect of $\alpha(\mathbf{x}_i, \mathbf{y}_i)$ on C scores: We examine our main argument – enhancing the correlation between Xatt's queries and outputs benefits a model's predictions – by analyzing $\alpha(\mathbf{x}_i, \mathbf{y}_i)$ against the individual C scores. Fig. 4 shows the average similarities, each of which is calculated over $\alpha(\mathbf{x}_i, \mathbf{y}_i)$ for all input pairs $(\mathbf{x}_i, \mathbf{y}_i)$ in each video in ActivityNet Captions *ae-test* split. The average $\alpha(\mathbf{x}_i, \mathbf{y}_i)$ of the baselines either completely without PAM or without the modulation scheme in PAM are small and spread around zero with smaller C scores. PAM with the modulation schemes appear to improve the C scores with the increased similarities between Xatt's queries and outputs. When adopting $\mathcal{L}_{DM}$, $\alpha(\mathbf{x}_i, \mathbf{y}_i)$ is pushed to the greatest value it could achieve, leading to the improved accuracy. This observation supports our argument, however, to the extreme case, if a Xatt's output $\mathbf{x}_i$ and a query $\mathbf{y}_i$ are perfectly aligned, then either of them becomes redundant. As such, PAM with QM and KM obtain better accuracy by not overly enhancing $\alpha(\mathbf{x}_i, \mathbf{y}_i)$.

2) Effect of b_l on $\alpha(\boldsymbol{x}_i, \boldsymbol{y}_i)$: Here we study how b_l modulates the similarities, whose values are shown against b_l in Fig. 5. One can see that first increasing the b_l value from zero leads to larger similarities. However, once b_l goes beyond

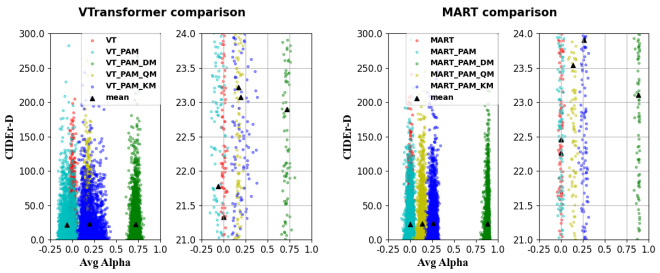

Fig. 4: The average similarities α in Eq. (7) of the first decoder layer versus the C scores of samples in ActivityNet Captions *ae-test* split on VTransformer- (left) and MART-based (right) PAM models. In each panel, the right figure provides a close-up look to the left. The black upper triangles denote the average C scores and similarities α. "PAM" denotes PAM without using $\mathcal{L}_{cos}$.

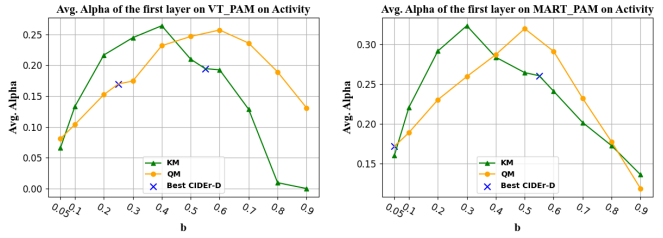

Fig. 5: The average α in Eq. (7) of the first decoder layer versus the choice of b_l on VTransformer+PAM (left) and MART+PAM (right) on ActivityNet Captions *ae-test* split.

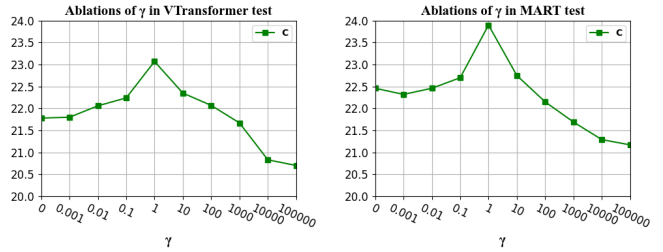

Fig. 6: Effects of γ_{cos} on $\mathcal{L}_{KM}$ on VTransformer (left) and MART (right) C score in ActivityNet-Captions *ae-test* split.

a certain value, the penalty forced by it in either $\mathcal{L}_{QM}$ or $\mathcal{L}_{KM}$ starts to decrease the similarities. This phenomenon could be explained as follows: with a large b_l, minimizing $\alpha(\mathbf{x}_i, \mathbf{y}_i) - \alpha(\mathbf{z}_i, \mathbf{y}_i)$ in Eq. (10) tends to minimize $\alpha(\mathbf{x}_i, \mathbf{y}_i)$ and maximize $\alpha(\mathbf{z}_i, \mathbf{y}_i)$, individually, leading to a lower $\alpha(\mathbf{x}_i, \mathbf{y}_i)$.

3) Effect of b_l on model accuracy: Table III shows the results of using PAM with varing values for b_l. It can be seen that setting a large value of b_l, e.g. 0.7, in both PAM QM and KM hurts the accuracy. We owe this to the fact that a large value of b_l inclines to reduce the similarities between Xatt's queries and outputs, as discussed in Section IV-C2. Empirically, the best results are obtained with a moderately large value of b_l in both PAM QM and PAM KM.

TABLE III: Captioning results with different b_l values on the ActivityNet Captions *ae-test* split.

Model	Modulation	b_l	B@4	M	C	R@4↓	Modulation	b_l	B@4	M	C	R@4↓
VT+PAM	–	–	9.42	15.50	21.78	8.06	–	–	–	–	–	–
VT+PAM	QM	0.1	9.88	15.79	22.88	**6.97**	KM	0.1	9.78	15.61	22.84	7.03
VT+PAM	QM	0.3	**9.96**	**15.85**	**23.09**	8.42	KM	0.3	9.79	15.65	22.89	6.81
VT+PAM	QM	0.5	9.82	15.72	22.95	7.98	KM	0.5	**9.85**	**15.72**	**22.94**	6.92
VT+PAM	QM	0.7	9.79	15.64	22.86	7.72	KM	0.7	9.76	15.59	22.86	**6.54**
MART+PAM	–	–	9.80	15.60	22.46	7.23	–	–	–	–	–	–
MART+PAM	QM	0.1	**10.23**	**15.79**	**23.42**	6.29	KM	0.1	10.22	15.71	23.27	6.82
MART+PAM	QM	0.3	10.17	15.74	23.25	6.25	KM	0.3	10.29	15.66	23.18	6.71
MART+PAM	QM	0.5	10.19	15.68	23.22	6.18	KM	0.5	**10.31**	**15.78**	**23.72**	**6.58**
MART+PAM	QM	0.7	10.18	15.74	23.33	**6.02**	KM	0.7	10.23	15.69	23.45	6.64

4) Effects of γ_{cos}: We study how the weight γ_{cos} in the loss $\mathcal{L}_{train}$ of Eq. (12) affects the model accuracy. In Fig. 6, we observe the different values of γ_{cos} influence the models' C scores and the peaked values are obtained with $\gamma_{cos} = 1$.

D. Qualitative Results

Some sample captions generated by the models are shown in Fig. 7. Compared with the two baselines, PAM captures the key activities with more precise descriptions. For example, PAM specifically identifies that a man is performing a martial arts moves and routines while the other two baselines are only to describe the basic movements involved, such as moving and kicking legs. Moreover, in the second example, VTransformer+PAM KM captures that the woman holds the stick and uses the balls to play the game while VTransformer does not recognize the stick. MART+PAMs are better than MART by being more descriptive on the actions (holds a stick and hits the ball) that the woman is performing.

Ground truth: A man is seen bowing before a large group of people and performing a martial arts routine on a large stage. The man continues moving his arms and legs around and ends with him bowing to the audience.
VTransformer: A man is seen standing in a circle and begins moving himself around and kicking his legs around. The man continues to spin around while the camera captures his movements.
VTransformer+PAM QM: A man is seen walking around a large stage and leads into a man performing various martial arts moves. The man continues to dance around while the camera captures his movements.
VTransformer+PAM KM: A man is seen walking into a large circle and leads into a man performing a martial arts routine. The man performs a series of kicks and throws in the air.
MART: A man is seen standing in a large circle and begins performing a routine in front of a large crowd. The man continues moving around while the camera captures his movements.
MART+PAM QM: A man is seen standing in a large circle and begins performing a martial arts moves. The man continues moving around and ends with him jumping off the side.
MART+PAM KM: A man is seen standing in a circle and begins performing a martial arts routine. The man continues moving around the area while the camera captures his movements.

Ground truth: A woman is seen speaking to the camera on a red carpet while holding sticks in her hands. The woman then begins hitting the balls around an area playing the game of crochet. The woman continues hitting the balls and ends by speaking to the camera.
VTransformer: A woman is seen speaking to the camera while holding up various objects and leads into her cutting a ball, A close up of a table is shown followed by a woman speaking to the camera and holding up various. The woman then demonstrates how to use the mop and bucket as well as her hand.
VTransformer+PAM QM: A woman is seen speaking to the camera while holding up a large ball and leads into her playing a. A woman is seen standing in a room holding a stick and speaking to the camera. She then shows how to use the balls to hit the ball.
VTransformer+PAM KM: A woman is seen speaking to the camera and leads into her holding a ball. A woman is seen speaking to the camera and leads into her holding up a stick. She then demonstrates how to use the balls to play the game.
MART: A woman is seen walking into frame and leads into a person playing a game of bag. The person begins playing the game of curling. The person continues to play and ends with text across the screen.
MART+PAM QM: A woman is seen speaking to the camera while holding up various objects in her hands. She then uses a stick to hit the ball around the area. She then holds up a rag and shows off the tools to the camera.
MART+PAM KM: A woman is seen speaking to the camera while holding up a stick and leads into her holding up a. She then puts the balls into the bucket and begins hitting the ball around. She then puts the ball down and hits the ball around the area.

Fig. 7: Results with different models for ActivityNet videos BRuansCVV3U (top) and 9Xrw-WOipSI (bottom). Some key activities in texts are highlighted in red.

V. Conclusion

This work focused on the dense video captioning (VC) problem and provided a study on the weak correlation between the queries and outputs of multi-modal cross-attention in Transformer. We proposed a novel Post-Attention Modulator (PAM) with three different modulations, with which the captioning accuracy was improved due to the enhanced correlation. Experimental results demonstrated that PAM is capable of predicting better captions than the strong VTransformer and MART baselines on the ActivityNet Captions and YouCookII datasets. As a future direction, we aim to study how the repetitions in the generated captions could be further reduced.

ACKNOWLEDGMENT

This work has been supported by the Academy of Finland in projects 317388, 329268 and 345791. We also acknowledge the computational resources provided by both the Aalto Science-IT project and CSC – IT Center for Science, Finland.

REFERENCES

[1] R. Krishna, K. Hata, F. Ren, L. Fei-Fei, and J. Carlos Niebles, "Dense-captioning events in videos," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 706–715.

[2] L. Zhou, C. Xu, and J. Corso, "Towards automatic learning of procedures from web instructional videos," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[3] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3156–3164.

[4] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, "Bottom-up and top-down attention for image captioning and visual question answering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6077–6086.

[5] L. Zhou, Y. Zhou, J. J. Corso, R. Socher, and C. Xiong, "End-to-end dense video captioning with masked transformer," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8739–8748.

[6] Z. Yu, J. Yu, Y. Cui, D. Tao, and Q. Tian, "Deep modular co-attention networks for visual question answering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 6281–6290.

[7] B. Zhang, H. Hu, and F. Sha, "Cross-modal and hierarchical modeling of video and text," in *Proceedings of the European conference on computer vision*, 2018, pp. 374–390.

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

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 4171–4186.

[10] V. Iashin and E. Rahtu, "Multi-modal dense video captioning," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2020, pp. 958–959.

[11] J. Lei, L. Wang, Y. Shen, D. Yu, T. Berg, and M. Bansal, "MART: Memory-augmented recurrent transformer for coherent video paragraph captioning," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 2603–2614.

[12] L. Huang, W. Wang, J. Chen, and X.-Y. Wei, "Attention on attention for image captioning," in *Proceedings of the IEEE international conference on computer vision*, 2019, pp. 4634–4643.

[13] S. Ging, M. Zolfaghari, H. Pirsiavash, and T. Brox, "COOT: Cooperative hierarchical transformer for video-text representation learning," in *Advances in neural information processing systems*, 2020.

[14] Y. Xiong, B. Dai, and D. Lin, "Move forward and tell: A progressive generator of video descriptions," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 468–483.

[15] S. Gella, M. Lewis, and M. Rohrbach, "A dataset for telling the stories of social media videos," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 968–974.

[16] Z. Dai, Z. Yang, Y. Yang, J. G. Carbonell, Q. Le, and R. Salakhutdinov, "Transformer-XL: Attentive language models beyond a fixed-length context," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 2978–2988.

[17] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[18] Y. Dong, J.-B. Cordonnier, and A. Loukas, "Attention is not all you need: Pure attention loses rank doubly exponentially with depth," *arXiv preprint arXiv:2103.03404*, 2021.

[19] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[20] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder–decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1724–1734.

[21] R. Vedantam, C. Lawrence Zitnick, and D. Parikh, "Cider: Consensus-based image description evaluation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 4566–4575.

[22] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002, pp. 311–318.

[23] A. Lavie and A. Agarwal, "METEOR: An automatic metric for mt evaluation with high levels of correlation with human judgments," in *Proceedings of the second workshop on statistical machine translation*, 2007, pp. 228–231.

[24] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR (Poster)*, 2015.