

## 6 Supplementary Material

### 6.1 Multi-headed Attention

**Scaled dot-product attention** The notion of multi-headed attention is based on the idea of *scaled dot-product attention* which is defined as follows

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V, \quad (13)$$

where  $\sqrt{d}$  is a scaling factor designed to keep the Softmax gradients in a sufficient range,  $Q, K, V$  are sequences of *queries*, *keys*, and *values*, Softmax is applied row-wise.

**Attention with Many Heads** The concept of multiple heads was introduced in [45] to allow a model to learn  $H$  distinct representation sub-spaces at each position while preserving the same computation efficiency. An attention *head* is usually presented as (13) with parametrized inputs

$$\text{head}_h(q, k, v) = \text{Attention}(qW_h^q, kW_h^k, vW_h^v), \quad h \in [1, H] \quad (14)$$

where  $q \in \mathbb{R}^{T_q \times D_q}$ ,  $k \in \mathbb{R}^{T_k \times D_k}$ ,  $v \in \mathbb{R}^{T_k \times D_k}$  and  $W_h^* \in \mathbb{R}^{D_* \times D_{\text{in}}}$ . Note that the inputs  $k$  and  $v$  are expected to have the same dimension ( $T_k \times D_k$ ) while  $q$  might have a different one. The weights  $W_h^*$  are mapping the corresponding inputs into an internal space  $D_{\text{in}} = \frac{D_q}{H}$  such that  $D_q$  is a multiple of  $H$ . The mapping into  $D_{\text{in}}$  space allows the attention to be calculated between the features which originally were of distinct dimensions ( $D_q \neq D_k$ ). The multi-headed attention is, then, defined as the concatenation of  $H$  attention heads mapped back to sub-space of queries ( $D_q$ ) with  $W^{\text{out}} \in \mathbb{R}^{H \cdot D_{\text{in}} \times D_q}$

$$\text{MultiHeadAttention}(q, k, v) = [\text{head}_1(q, k, v), \text{head}_2(q, k, v), \dots, \text{head}_H(q, k, v)]W^{\text{out}}. \quad (15)$$

### 6.2 Feature Extraction

Both audio and visual features are pre-calculated before training. The audio features are extracted with the VGGish network [15], which was pre-trained on *AudioSet* [12]. More specifically, the VGGish model processes 0.96 seconds long segments. The audio segments, in turn, are represented as log mel-scaled spectrograms of size  $96 \times 64$  which are obtained via *Short-time Fourier Transform*. The STFT utilizes a 25 ms *Hann* window with 15 ms step applied to the 16 kHz mono audio track. The pre-classification layer of VGGish outputs a 128-d embedding for each spectrogram. Therefore, the audio track of an  $i^{\text{th}}$  video in the dataset is represented with a sequence of 128-d features of length  $T_a^i$ , each feature in the stack represents 0.96 seconds of the original audio track.

To extract features from the visual stream, we employ the I3D network [4] pre-trained on *Kinetics* dataset. Specifically, I3D inputs 64 RGB and 64 optical flow frames of size  $224^2$  extracted at 25 fps. Similar to [17], we extract the flow frames using *PWCNet* [42]. Both sets of frames are, first, resized such that  $\min(\text{Height}, \text{Width}) = 256$ , and, then, the central region of size  $224^2$  is cropped. After, both stacks of frames are passed through the corresponding streams of I3D. It outputs a 1024-d representation for RGB and flow 64-frame stacks from the second-to-the-last layer. Following the authors of I3D, we sum the representations from both streams. It results in a single 1024-d representation for every

stack of 64 frames. Therefore, the visual track of  $i$ th video is represented with a sequence of 1024-d features of length  $T_v^i$  where every features spans 2.56 seconds (64 frames) of the original video.

The tokens (or, roughly, words) from captions are embedded with *Global Vector* (GloVe) representations pre-trained on the *Common Crawl* dataset (2.2M vocabulary) [32]. The pre-trained model is represented as a lookup table which maps a token to a 300-d embedding. If a token is missing in the vocabulary, an average vector among all vocabulary words is returned. Therefore, each previous token of a caption is represented with a 300-d vector.

Therefore, the bi-modal encoder's in and out dimensions are  $d_a = 128$  and  $d_v = 1024$  for audio and visual streams while the decoder inputs and outputs  $d_c = 300$ .

### 6.3 Implementation Details

The batch of size 32 and 16 were used during training of captioning and proposal generation modules, respectively. To form a batch, in the captioning module, the features are padded up to the longest sequence in the batch. For the proposal generator, the features are extracted from entire videos and padded up to 300 for visual and 800 for audio to form a batch. These number were selected to cover all possible lengths of the features in the training set. The padding is masked out as it is done for the next caption tokens in the decoder (see Sec. 3.3). Each head in the bi-modal multi-headed generator predicts  $\Psi_a = 48$  and  $\Psi_v = 128$  anchors for audio and visual modalities. We used the following lists each of size  $K_a = K_v = 10$  for kernel sizes (given in cell-coordinates):  $[5, 13, 23, 35, 51, 69, 91, 121, 161, 211]$  for audio and  $[1, 5, 9, 13, 19, 25, 35, 45, 61, 79]$  for visual modalities which are determined by K-Means algorithm. The size of both intermediate layers in proposal generation head is 512. Note that  $\frac{128}{48} = \frac{800}{300} = \frac{2.56}{0.96}$  which preserves the balance between predictions from both modalities (see Sec. 3.2).

Since the modality features might have a different size, we also need to map them into an internal space inside of the bi-modal attention modules ( $D_{in}$ ), see Eq. (14) for more details. We select the internal space to be of size  $D_{in} = 1024$ . Both the encoder and decoder of the bi-modal transformer have  $N = 2$  layers and  $H = 4$  heads in each of the multi-headed attention modules. The caption vocabulary size and, hence, the generator's output dimension is 10 172. We use  $\gamma = 0.7$  in the label smoothing and the probability of dropout  $p = 0.1$ . The localization and objectness loss coefficients are 1, and the noobjectness coefficient is 100. *Adam* optimizer with default hyper-parameters [18] and learning rate  $5 \cdot 10^{-5}$  is used to train both caption and proposal generator. The hyper-parameters are selected on the validation set.

We highlight that the whole process of training both parts of the model was designed to keep a unified training procedure avoiding using different techniques such as *reduce-on-plateau*, *weight decay*, different learning rate, optimizer when training the proposal generator, sometimes, favoring elegance at the cost of performance. We encourage others to try different combinations when training both stages to achieve better results.

The captioning module was trained until METEOR on the validation set has not improved for 30 epochs while the proposal generator is trained for 70 epochs at most. In our experiments, the training of the final captioning module reaches the peak performance at 26<sup>th</sup> epoch while the proposal generator achieves the highest F1-score on the validation set at 17<sup>th</sup> epoch. We select the proposals on the epoch with the highest metric and caption them with the best captioning model. The training of the captioning module until the best performance takes 10 hours and 3.5 hours to train the proposal generator on *one* Nvidia GeForce RTX 2080Ti. We use PyTorch [30] as our primary library for the implementation.

	Validation	B@3	B@4	M	Recall	Prec.	F1
Wang <i>et al.</i> [49]	Full	2.27	1.13	6.10	57.60	44.80	50.40
	As ours	2.29	1.15	6.14	57.86	44.88	50.55
Zhou <i>et al.</i> [62]	Full	2.91	1.44	6.91	86.33	38.57	53.31
	As ours	2.92	1.45	6.92	86.33	38.55	53.30
Ours	Full	3.50	1.72	7.69	73.22	43.97	54.95
	As ours	3.84	1.88	8.44	80.31	48.23	60.27

Table 4: The performance of other methods on the filtered ActivityNet Captions validation set for videos which are no longer available (around 91 % (as ours)). The results are reported in the learned proposal setting. As expected, the performance of other models remains at the same level while ours gains the missing 9 %. Metrics are BLEU3–4, METEOR, recall, precision, and F1-measure.

Caption Module	Proposal Generator	B@3	B@4	M	Recall	Prec.	F1
Wang <i>et al.</i> [49]	[49]	2.29	1.15	6.14	57.86	44.88	50.55
Ours	[49]	2.87	1.41	7.03	57.86	44.88	50.55
Ours	Ours	3.84	1.88	8.44	80.31	48.23	60.27

Table 5: The comparison of the captioning performance between our model and [49] on the learned proposals provided in [49]. The results are reported on the filtered ActivityNet Caption validation datasets.

## 6.4 More Ablation Studies

### 6.4.1 Why Do You Exclude Videos from the Validation Set?

In our experimentation, we exclude videos which are no longer available on YouTube (9 %) from the ground truth validation datasets as it would be unfair to compare our model to the methods which could make a prediction based on the video content while our model gets zero scores on a missing video. Therefore, we evaluate the predictions made by other models [49, 62] on the same validation set as we have. We selected only these two methods as they made either a code or evaluation results publicly available.

In other words, we hypothesise that the performance of other methods will not change after excluding videos from both predictions and ground truth while the performance of our method will be higher by around 9 % (a portion of the missing videos). The results of the comparison are shown in Tab. 4 and, indeed, imply that the performance of other methods remains on the same level (less than 2 % change). We remind a reader that the compared methods were trained on the full training dataset while ours was trained on only 91 %.

### 6.4.2 Might Your Model Improve Results of Other Methods?

Since [49] have not made the results publicly available for captioning ground truth (see Tab. 1), we cannot compare it with our model directly. To this end, we apply our final captioning model on the generated proposals from [49] to eliminate the effect caused by

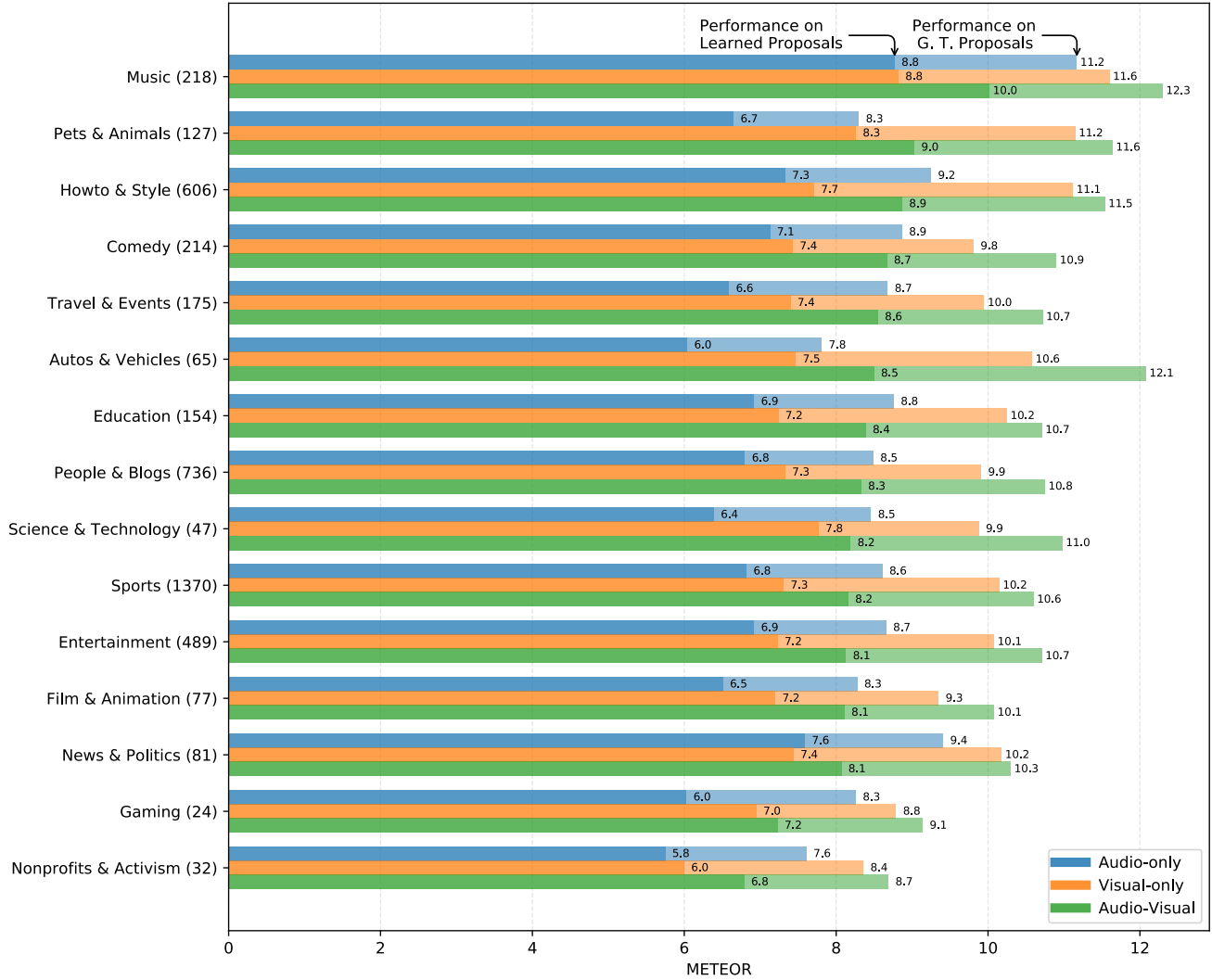


Figure 4: The performance comparison between different modalities (Audio-only, Visual-only, and Bi-modal) in two settings (ground truth and learned proposals) across different YouTube video categories. The video categories are sorted according to the performance of the Audio-Visual model in the learned proposal setup. The number of videos in a category is shown in brackets. ActivityNet Captions validation subset is used for the comparison.

different proposal generator modules. The results of the comparison are reported on the filtered ActivityNet Caption validation datasets (see Sec. 6.4.1) and shown in Tab. 5. The results suggest that our model has a better captioning performance on this set of metrics.

### 6.4.3 What is the Impact of Audio and Visual Cues across Different Video Types?

Following [17], we inspect if the final model’s performance consistently improves across different types of videos. To form a list of video types, we retrieve a YouTube video category for each video in the validation dataset. The YouTube category is annotated by the author when they upload a video to the service. YouTubeAPI [57] was used to retrieve the categories automatically. Since there was a time gap between downloading videos and their categories, 67 were no longer available. Such videos were removed from the comparison. Also, we removed one video category with less than 20 videos.

Fig. 4 shows the performance comparison between bi-modal (final), audio-only, and visual-only models across different video categories in two settings: captioning ground truth

<b>2<sup>nd</sup> Encoder's Sub-layer</b>	<b>B3</b>	<b>B4</b>	<b>M</b>
Self-Attention	3.60	1.74	8.14
Bi-modal Attention	3.84	1.88	8.44

Table 6: The effect of replacing the bi-modal attention with a self-attention module in encoder layers. The comparison is shown on validation subsets of ActivityNet Captions in the learned proposal setting. The metrics are BLEU3–4 and METEOR.

and learned proposals. The results suggest the consistent gain in performance when both modalities are used compared to the uni-modal models. This pattern holds across both settings and all categories. In addition, it appears that the visual modality provides more cues to the model than the audio modality in nearly all cases. Moreover, the dataset seems to be biased to “Sports” and “People & Blogs” videos, which hold almost half of the dataset. Yet, the results show no evidence of over-fitting to these categories. Among all categories, “Music” appears to be the “easiest” one, which might be explained by a small variety of ways to describe the content of this kind. Meanwhile, the models perform the worst on “Gaming” and “Nonprofits & Activism” categories, which might occur because of the lack of such videos in the dataset.

#### 6.4.4 What Happens if the Bi-Modal Attention Block is Replaced by Uni-modal Self-Attention?

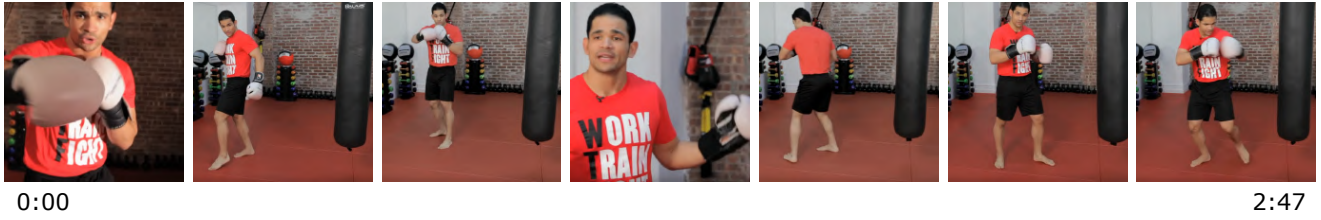
In this ablation study, we would like to estimate the influence of the bi-modal attention blocks on the model performance (see middle blocks of Encoder and Decoder layers in Fig. 2). Yet, we can ablate only the encoder as the bi-modal attention is essential for the decoder since it inputs two streams. One solution would be to fuse the outputs of the encoder. This would, in turn, allow us to replace two bi-modal attention blocks in the decoder with one. However, it is not possible since the temporal spans of the encoder’s output streams, in general, are distinct ( $A^v \in \mathbb{R}^{T_a \times d_a}$  and  $V^a \in \mathbb{R}^{T_v \times d_v}$ ). Therefore, in this setting, each encoder layer has two pairs of self-attention blocks which preserves the final model’s number of parameters.

The results are presented in Tab. 6. We observed a substantial decline in performance among all metrics when the bi-modal attention block is replaced by self-attention. This further suggests the importance of the proposed approach. Besides, the results of the model with self-attention in encoder layers are still much stronger than any V- and A-only models (see Tab 3), which proves the importance of an audio-visual approach to the task.

## 6.5 Qualitative Analysis

Fig. 5 provides the qualitative analysis of the final captioning model compared to ground truth captions. Additionally, we provide captions produced by uni-modal captioning models (audio- and visual-only). The results show that the caption, produced by a bi-modal captioning module as well as the audio-only model managed to grasp the concept of talking when captioning the largest segment (the top one) while video-only model neglects it. The video, by itself, consists of an explanation of how to do a martial art movement and highly verbose. Therefore, even though the ground truth does not mention that the person talks during the video, the predictions of our final model are not entirely erroneous. However, the colour of the man’s shirt is incorrectly guessed, which might be explained by the presence of the black punching bag on the screen. Finally, the caption produced by the visual-only model also makes sense.





**GT:** A man in a padded gym room demonstrates how to kick box with a pair of gloves and a hanging punching bag

**Ours (Audio):** A man is standing in a room talking to the camera

**Ours (Visual):** A man is standing in a room, swinging his arms and legs around

**Ours (Audio Visual):** A man in a black shirt is standing in a room talking

**GT:** A barefoot man in a red t-shirt and wearing boxing gloves stands in a mirror walled gym room, on a padded floor and demonstrates how to jab, all while talking to the camera.

**Ours (Audio):** A man is seen speaking to the camera and leads into him holding a bow and speaking to the camera

**Ours (Visual):** The man then begins to move on the floor and begins to move on the machine

**Ours (Audio Visual):** He then demonstrates how to do a karate moves

**GT:** The man then demonstrates foot work and guard and block boxing techniques

**Ours (Audio):** A man is seen speaking to the camera and leads into him speaking to the camera

**Ours (Visual):** He continues to demonstrate several moves as he moves around the room and ends by hitting his head

**Ours (Audio Visual):** He is standing in a gym talking to the camera

**GT:** The man then incorporates the use of the punching bag to demonstrate block, jab and foot work moves on the bag

**Ours (Audio):** The man then begins to talk to the camera and leads into him speaking to the camera

**Ours (Visual):** He continues to hit the bag while several others watch him and ends by hitting the ball

**Ours (Audio Visual):** The man continues to play with the bag and leads into him hitting the bag

Figure 5: The results of the qualitative analysis for a video from ActivityNet Caption validation dataset. The predictions of our bi-modal model are compared to the uni-modal model predictions and ground truth (GT) annotations. The video shows a man who explains how to do a martial art movement—the YouTube video id [EIibo7aTpys](#).

Moreover, if we consider the results of the audio-only model, we may notice that it mostly gets the signal of “talking” and exploits it in a prediction. Indeed, it might be challenging even for a non-English human to understand what the video is about given only the audio track. We also notice that captions provided by an annotator are significantly more detailed compared to the predictions of the captioning model, which are somewhat more general. This is the issue which needs more attention in future research as it seems to be a problem for any dense video captioning system.