Supplementary Materials

In this supplementary material, we provide additional details about the experimental settings, codes, cross-modal encoder architecture, differentiable Top-K operator, rationality of T2W attention, computation cost and more detailed visualization results.

A. Architecture Details

Here we describe the implementation details of our asymmetric cross-modal encoder, which contains a stack of layers. As shown in Figure A1, each layer contains a W2P attention block and a T2W attention block. Both blocks perform a series of operations such as multi-head attention, normalization (norm) and multi-layer perception (MLP). Specifically, given video tokens $\boldsymbol{V}^l$ and text tokens $\boldsymbol{X}^l$ as inputs, W2P and T2W attentions respectively output $\boldsymbol{V}^{l+1}$ and $\boldsymbol{X}^{l+1}$, which are used as the next layer's inputs.

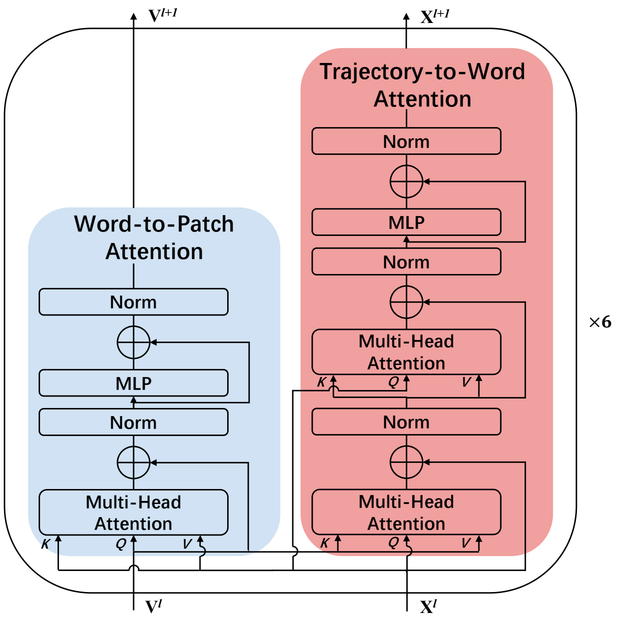


Figure A1. Detailed architecture of Cross-modal encoder.

B. Differentiable Top-K

Here we describe the implementation details of differentiable frame-selector module. Given the importance scores $\boldsymbol{s}$ generated from the scorer network (cf. Section 3.3 in the main text), we select the indices of K highest scores, denoting as $\boldsymbol{M} \in \mathbb{R}^{T\times K}$, where each column in $\boldsymbol{M}$ is a one-hot (T) dimensional indicator. We keep Top-K most relevant frames by:

$$\hat{\boldsymbol{V}} = \boldsymbol{M}^T \boldsymbol{V}. \tag{a}$$

To learn the parameters of the scorer network using an end-to-end training without introducing any auxiliary losses, we resort to the perturbed maximum method [1] to construct a differentiable Top-K operator. In particular, selecting Top-K frames is equavalent to solving a linear program of the follwing form:

$$\underset{\boldsymbol{M}\in\mathcal{C}}{\operatorname{argmax}} \langle \boldsymbol{M}, \boldsymbol{s} \rangle. \tag{b}$$

where $\boldsymbol{M}$ is the optimization variable and $\mathcal{C}$ is the convex polytope constrain set. We follow [1] to calculate forward and backward operations to solve Eq. (b)

C. Rationality of T2W

Theoretically, we illustrate why T2W emphasizes more continuous frames than P2W. To simplify, we adopt single-head attention for subsequent computations. Assume a video contains $T \times N$ tokens, where T and N denote the frame number and token number per frame, and $\boldsymbol{v}_t^n$ denotes the n-th token in the t-th frame. Given the query word, P2W calculates attention weights α over all $N \times T$ features: $\boldsymbol{z}_{P2W} = \sum_{t=1}^{T} \sum_{n=1}^{N} \alpha_{N\times(t-1)+n} \boldsymbol{v}_t^n$ where $\sum_{k=1}^{N\times T} \alpha_k = 1$, while T2W respectively uses Eq. (3)& (4) to calculate two sets of attention weights γ and β: $\boldsymbol{z}_{T2W} = \sum_{t=1}^{T} \beta_t (\sum_{n=1}^{N} \gamma_t^n \boldsymbol{v}_t^n)$ where $\sum_{k=1}^{T} \beta_k = 1$ and $\forall t \in 1,..,T, \sum_{n=1}^{N} \gamma_t^n = 1$. When certain frames exhibit more salient patterns corresponding to the query word, P2W's α will be larger in these frames, yet smaller α in others due to $\sum_{k=1}^{N\times T} \alpha_k = 1$. Consequently, P2W tends to focus on these episodic frames with salient patterns. However, T2W's γ for different frames are calculated independently, preventing one frame's salient pattern from obscuring others. Thus Eq. (4) in T2W has more opportunities to attend to each continuous frame. As a result, we claim that T2W will not act as P2W which tends to focus only on certain episodic frames. Quantitatively, we calculate the entropy of the attention weights for each frame to check which method attends to more frames. For T2W and P2W, the attention weight of the t-th frame are respectively represented as β_t and $\sum_{n=1}^{N} \alpha_{N\times(t-1)+n}$. We use the MSRVTT test set and sample 8 frames per video, which means the entropy upper bound is $log8 = 3$. The average entropy for T2W and P2W is 2.94 and 2.39 respectively. The higher entropy for T2W demonstrates that it can attend to more frames compared to P2W. These discussions will be added in revision.

D. Fine-tuning Setups

Here we describe the implementation details for fine-tuning the pre-trained model. All downstream tasks receive input frames of resolution 224×224. During fine-tuning, we randomly select N_v frames from the video. We use the same RandomAugment, AdamW optimizer, weight decay. The default settings for fine-tuning on each dataset are in Table A2. During inference, we do not use augmentation and sample uniformly.

E. Codes

Our main code is in "TW-BERT/src/modeling/TW_BERT_model.py", which is provided in the supplementary materials. We perform the W2P and T2W attention (cf. Section 3.2 in the main text) in function "cross-modal_encoder", which is implemented in the file "TW-BERT/src/modeling/cross_modal_bert/cross_models.py". Both two attention operation are implemented in the file "TW-BERT/src/modeling/cross_modal_bert/cross_attention.py". In the file "TW-BERT/src/modeling/timesformer/vit_all.py", we implement the Hierarchical Frame-Selector (cf. Section 3.3 in the main text) in class "VisualFrameSelection", and the differentiable Top-K operator in class "PerturbedTopKFuntion" and "PredictorLG".

F. Computation Cost

Table A1 shows FLOPs/parameters/runtime of diverse settings corresponding to Table 2&3, with FLOPs and inference time measured by 1 and 100 samples, respectively. We assert that HFS lessens computational demands since it filters input frames, retaining only relevant ones for the cross-modal encoder. Thus, valuable temporal knowledge is preserved while using fewer frames. For example, we additionally implement F@32 for QA, whose accuracy is 48.6, F@32-24-16 gets 48.5 while uses less computation burdens due to less input frames. Also, comparing F@16 to F@32-24-16, there's a modest increase in training demands: 4M parameters, 1.26× FLOPS, and 1.21× inference time.

Table A1. Computation cost comparison with different settings.

Layers	FLOPs(G)	Params(M)	Time(s)	**Frame**	FLOPs(G)	Params(M)	Time(s)
				F@8	325.0	232	11.0
				F@20-14-8	445.7	236	12.4
[2,4]	569.5	236	14.0	**F@12**	430.0	232	12.9
[3,6]	602.4	236	14.9	**F@24-18-12**	534.3	236	14.2
[4,8]	635.4	236	15.6	**F@16**	555.0	232	13.8
[5,10]	668.4	236	16.2	**F@32-24-16**	701.3	236	16.7
[6,12]	701.3	236	16.7	**F@32**	955.0	232	22.0

G. Qualitative Results

In Figure A2, we show more qualitative results of the heat maps of the attention weights of **TW-BERT** from five downstream datasets. **TW-BERT** selects the most relevant frames according to the whole text, then it appears to implicitly form a trajectory across time for a given query word to avoid over-exploiting the trivial spatial contexts.

H. Limitation

Despite the effectiveness of TW-BERT across a wide range of downstream video-language tasks, our model still has limitation: we provide a novel perspective to consider the videos that are composed of moving object trajectories. Our method benefits from temporal knowledge in the video and thus if a downstream task contains a few temporal contexts, the effect of T2W attention (cf. Section 3.2 in the main text) may not be obvious.

References

[1] Quentin Berthet, Mathieu Blondel, Olivier Teboul, Marco Cuturi, Jean-Philippe Vert, and Francis Bach. Learning with differentiable pertubed optimizers. *Advances in neural information processing systems*, 33:9508–9519, 2020. 1

[2] David Chen and William B Dolan. Collecting highly parallel data for paraphrase evaluation. *meeting of the association for computational linguistics*, 2011. 3

[3] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. In *ICCV*, page 5804–5813, 2017. 3

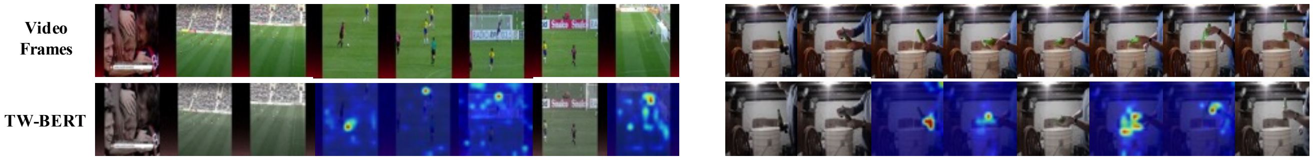
[4] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *ICCV*, page 706–715, 2017. 3

[5] Anna Rohrbach, Marcus Rohrbach, Niket Tandon, and Bernt Schiele. A dataset for movie description. In *CVPR*, page 3202–3212, 2015. 3

[6] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msrvtt: A large video description dataset for bridging video and language. In *CVPR*, pages 5288–5296, 2016. 3

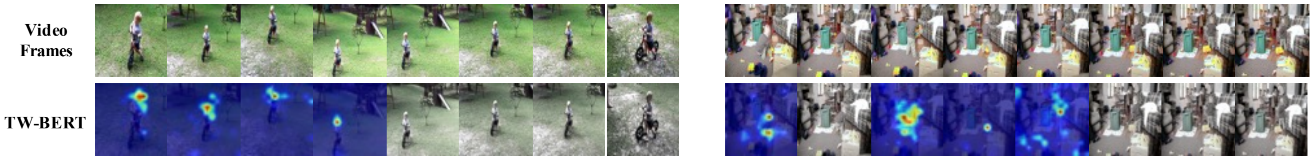
Table A2. Fine-tuning hyper-parameters of different tasks.

Config	**MSRVTT-Ret**	**DiDeMo-Ret**	**LSMDC-Ret**	**ActivityNet-Ret**	**MSRVTT-QA**	**MSVD-QA**
learning rate	2.5e-5	4e-5	4e-5	4e-5	5e-5	5e-5
learning rate schedule	linear decay	linear decay	linear decay	linear decay	linear decay	linear decay
weight decay	1e-3	1e-3	1e-3	1e-3	1e-3	1e-3
batch size	64	96	96	80	96	96
train epochs	5	10	10	10	10	15
warmup ratio	0.1	0.1	0.1	0.1	0.1	0.1
max text length	40	50	40	50	40	40
selection layer	[6,12]	[6,12]	[6,12]	[6,12]	[6,12]	[6,12]
frame number	20-14-8	20-14-8	20-14-8	20-14-8	32-24-16	32-24-16



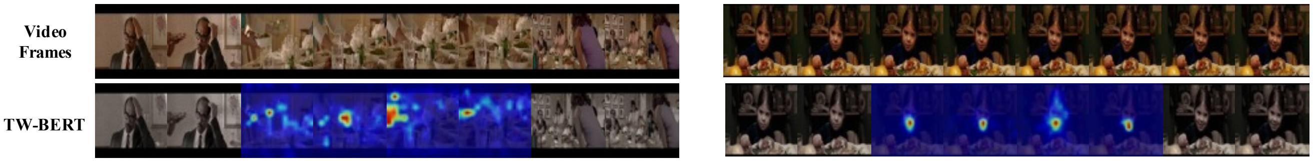
(a) "The player scores a goal"

(b) "A man pouring green glass bottle into a white bucket"



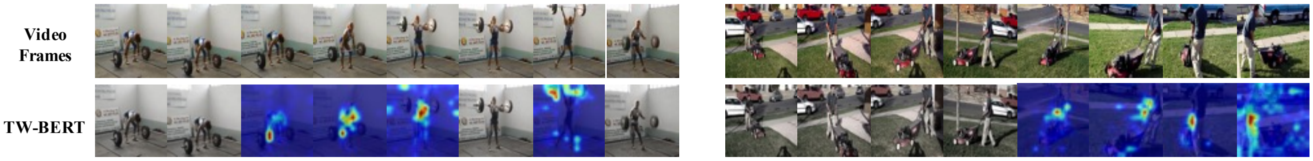
(c) "Boy rides bike in background"

(d) "Little boy stands up and starts walking with his toy"



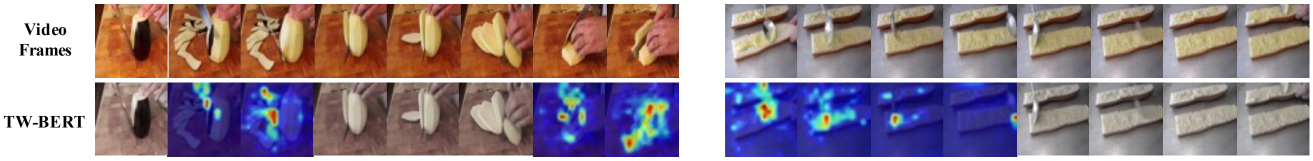
(e) "Someone brings food to the table"

(f) "Liberty Bell smiles slowly"



(g) "A female lifter lifts a barbell over her head"

(h) "A man is pushing a lawn mower across a lawn"



(i) "A man cutting up a fruit"

(j) "Spreading butter on a roll"

Figure A2. Visualizations of the attention maps from cross-modal encoder. Sample (a) and (b) are from MSRVTT [6], (c) and (d) are from DiDeMo [3], Sample (e) and (f) are from LSMDC [5], (g) and (h) are from ActivityNet [4], (i) and (j) are from MSVD [2] dataset. **TW-BERT** first discards irrelevant frames (the ones without blue backgrounds), then attends to the patches related to given query word by Trajectory-to-Word attention.