URBAN ROBOTICS LAB

What I got from ICCV 2019

Lab Seminar (2019.11.12)

Yunsoo Kim

Ji Lin, et al. TSM: Temporal Shift Module for Efficient Video Understanding
Internationcal Conference on Computer Vision (ICCV). 2019.

Yu Wu, et al. Dual Attention Matching for Audio-Visual Event Localization
Internationcal Conference on Computer Vision (ICCV). 2019.

Intro_TSM (Temporal Shift Module)

Dealing with the videos

2D CNN

- Low cost
- Individual frames cannot well model the temporal information

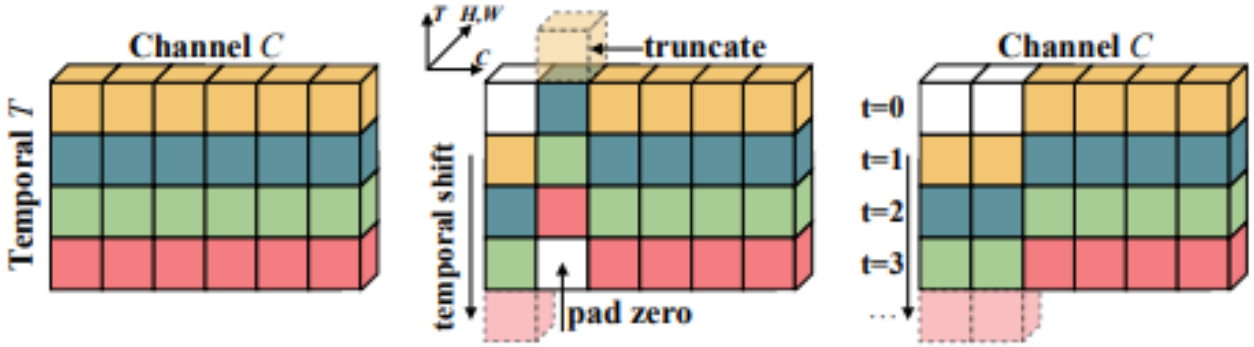
3D CNN

- Can jointly learn spatial and temporal features
- High cost

→ How can we learn temporal features with low cost?

Main Idea_TSM

Improved 2D CNN



(a) The original tensor without shift.

(b) Offline temporal shift (bi-direction).

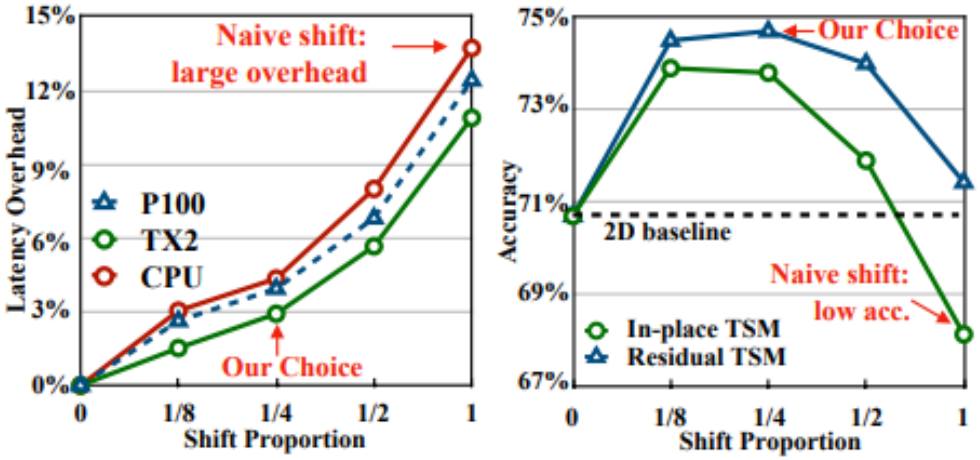
(c) Online temporal shift (uni-direction).

Shift Operation in TSM

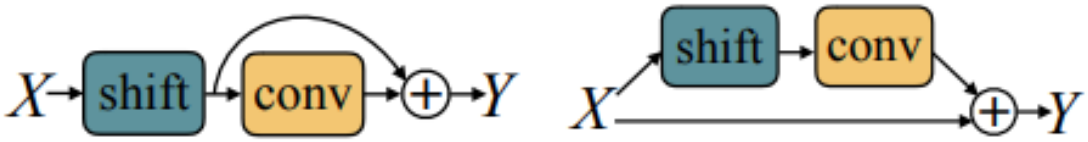
Two techniques

Shift small portion of the channels for efficient temporal fusion
→ Cuts down the data movement cost

Insert residual branch to preserve the current frame
→ Can get spatial feature learning too



(a) Overhead *vs*. proportion. (b) Residual *vs*. in-place.



(a) In-place TSM. (b) Residual TSM.

Figure 3. Residual shift is better than in-place shift. In-place shift happens before a convolution layer (or a residual block). Residual shift fuses temporal information inside a residual branch.

Structure of TSM

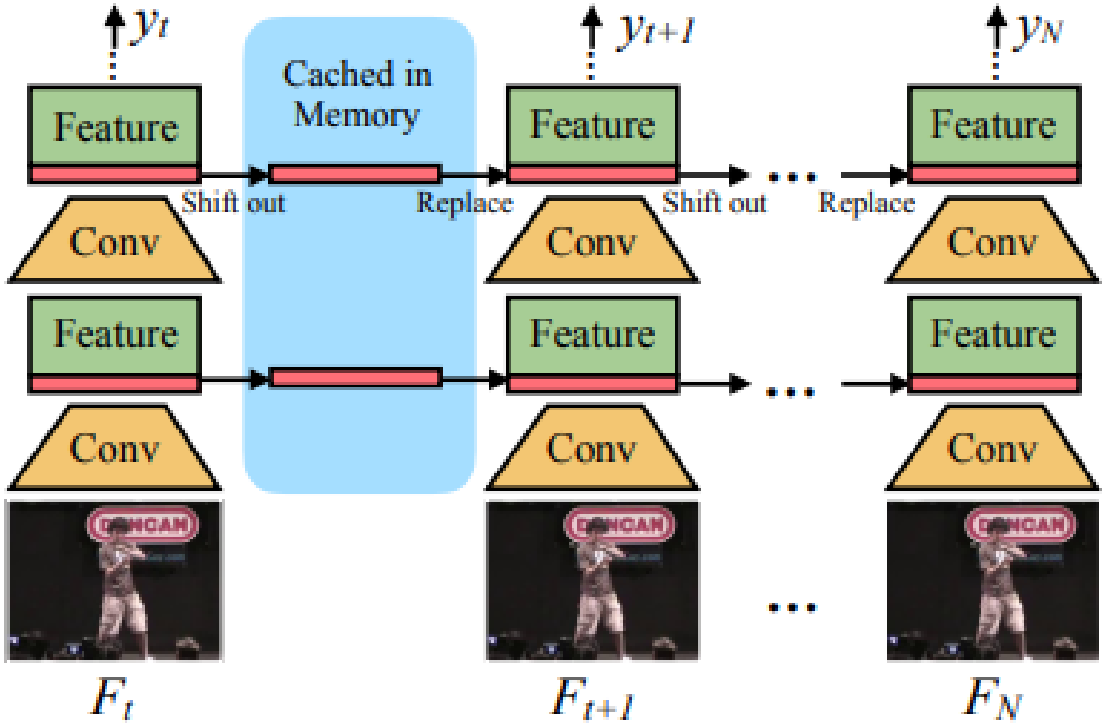


Figure 4. Uni-directional TSM for online video recognition.

Results of TSM

Table 2. Comparing TSM against other methods on Something-Something dataset (center crop, 1 clip/video unless otherwise specified).

Model	Backbone	#Frame	FLOPs/Video	#Param.	Val Top-1	Val Top-5	Test Top-1
TSN [58]	BNInception	8	16G	10.7M	19.5	-	-
TSN (our impl.)	ResNet-50	8	33G	24.3M	19.7	46.6	-
TRN-Multiscale [58]	BNInception	8	16G	18.3M	34.4	-	33.6
TRN-Multiscale (our impl.)	ResNet-50	8	33G	31.8M	38.9	68.1	-
Two-stream $TRN_{RGB+Flow}$ [58]	BNInception	8+8	-	36.6M	42.0	-	40.7
ECO [61]	BNIncep+3D Res18	8	32G	47.5M	39.6	-	-
ECO [61]	BNIncep+3D Res18	16	64G	47.5M	41.4	-	-
$ECO_{En}Lite$ [61]	BNIncep+3D Res18	92	267G	150M	46.4	-	42.3
$ECO_{En}Lite_{RGB+Flow}$ [61]	BNIncep+3D Res18	92+92	-	300M	49.5	-	43.9
I3D from [50]	3D ResNet-50	32×2clip	153G[1]×2	28.0M	41.6	72.2	-
Non-local I3D from [50]	3D ResNet-50	32×2clip	168G[1]×2	35.3M	44.4	76.0	-
Non-local I3D + GCN [50]	3D ResNet-50+GCN	32×2clip	303G[2]×2	62.2M[2]	46.1	76.8	45.0
TSM	ResNet-50	8	33G	24.3M	45.6	74.2	-
TSM	ResNet-50	16	65G	24.3M	47.2	77.1	46.0
TSM_{En}	ResNet-50	24	98G	48.6M	49.7	78.5	-
$TSM_{RGB+Flow}$	ResNet-50	16+16	-	48.6M	**52.6**	**81.9**	**50.7**

TSM does not only significantly improve the 2D baseline but also outperform state-of-the-art methods, which heavily rely on 3D convolutions

Table 1. Our method consistently outperforms 2D counterparts on multiple datasets at zero extra computation (protocol: ResNet-50 8f input, 10 clips for Kinetics, 2 for others, full-resolution).

	Dataset	Model	Acc1	Acc5	Δ Acc1
Less Temporal	Kinetics	TSN	70.6	89.2	+3.5
		Ours	**74.1**	**91.2**	
	UCF101	TSN	91.7	99.2	+4.2
		Ours	**95.9**	**99.7**	
	HMDB51	TSN	64.7	89.9	+8.8
		Ours	**73.5**	**94.3**	
More Temporal	Something V1	TSN	20.5	47.5	+28.0
		Ours	**47.3**	**76.2**	
	Something V2	TSN	30.4	61.0	+31.3
		Ours	**61.7**	**87.4**	
	Jester	TSN	83.9	99.6	+11.7
		Ours	**97.0**	**99.9**	

Intro_DAM (Dual Attention Matching)

Dealing with the audio-visual matching

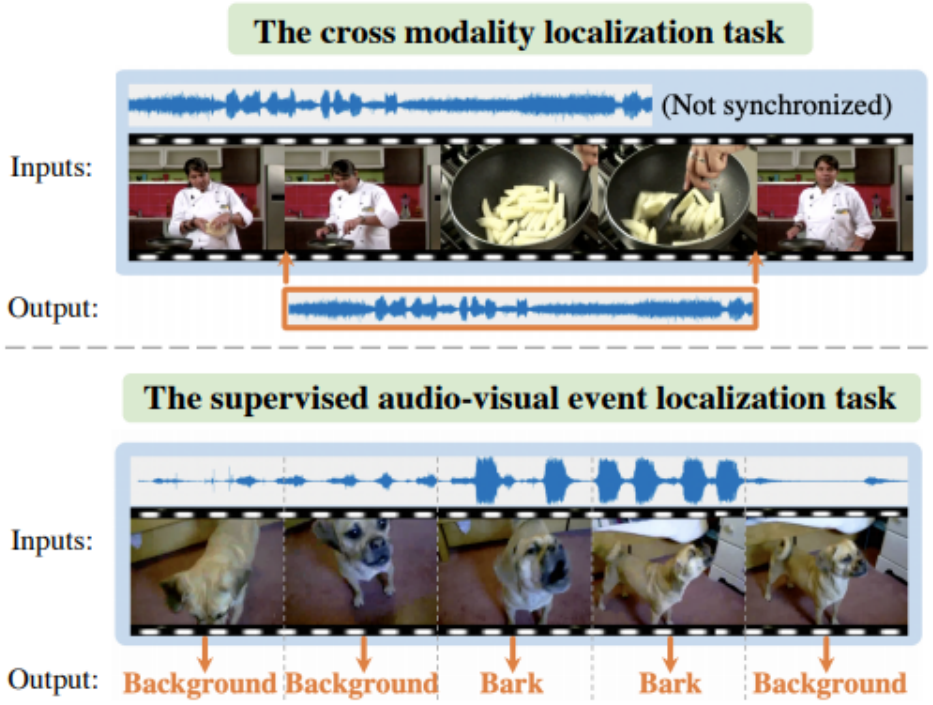


Figure 1. Examples of the audio-visual event localization problem. It includes two tasks, *i.e.*, the *cross-modality localization* (CML) task and the *supervised audio-visual event localization* (SEL) task. The CML task (the upper one in the figure) is to localize the event boundary in one modality given an input event signal in the other modality. The SEL task (the lower one) is to predict the event category (including background) of each input audio-visual segment. The orange color in the figure indicates the output of each task.

Motivation_DAM

Dealing with the audio-visual matching

Previous Methods

- First divide a video sequence into short segments
- Extract visual and acoustic features
- Minimize distances between two features

→ Weak for long-duration task

: Look the global temporal co-occurrences
(Correlation in a long duration between the visual and audio modalities)

Methods_DAM

- Looks into a longer video duration to model the whole event better, while also attaining local temporal information by a global cross-check mechanism

Preliminaries

S = (S^A, S^V) : the sequence of each modality

Temporal length of the sequence S is N seconds
→ Split into N non-overlapping segments $\{s_t^A, s_t^V\}_{t=1}^{N}$

There is an event-relevance label $y_t \in \{0, 1\}$
,which $y_t = 1$ means audio and visual content contains the event

Define event-relevant region
$$T_E = \{t | y_t = 1, 1 \leq t \leq N\}$$

Also, extract features in the segment level f_t^A and f_t^V

Dual Attention Matching module

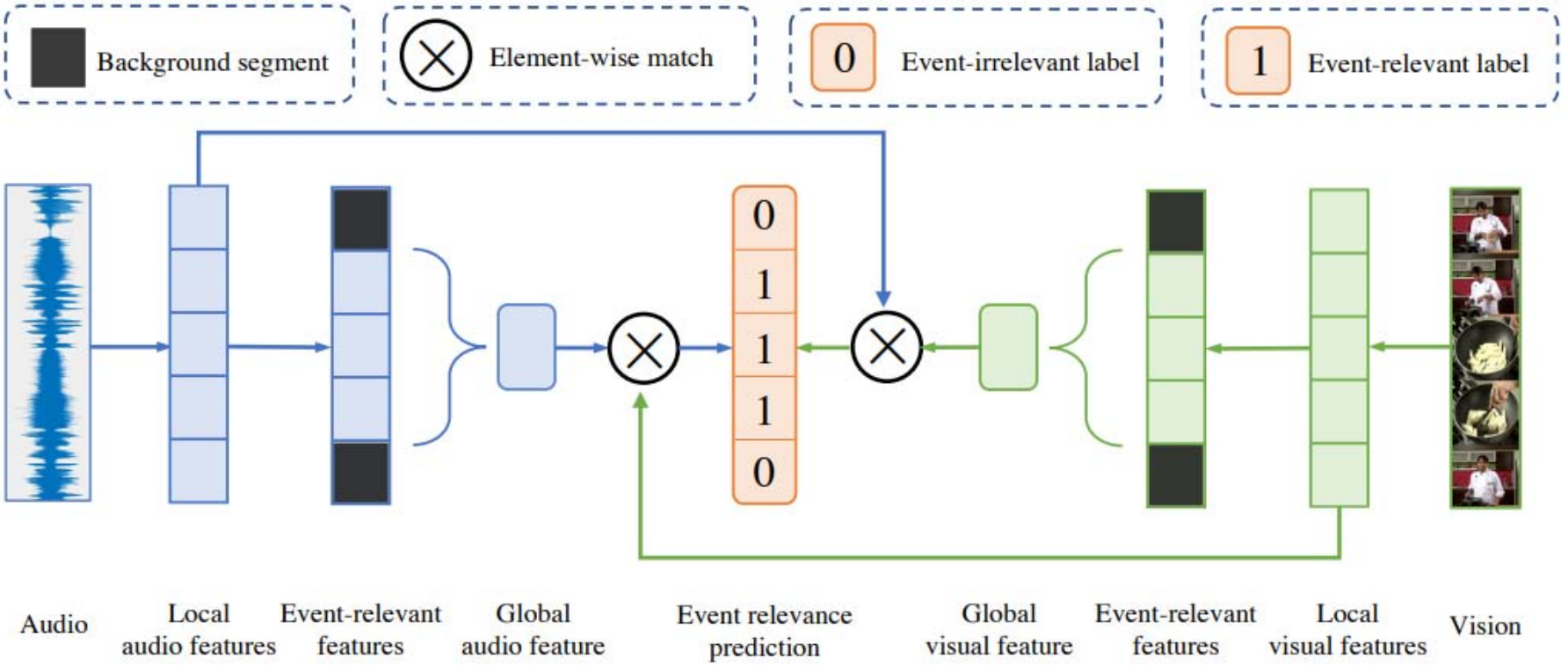


Figure 2. The proposed dual attention matching (DAM) module. DAM looks into a longer video duration to better model the high-level event information, while also attaining local temporal information by a global cross-check mechanism. DAM is optimized by finding which segments in the other are relevant to the event. We first extract the local features for each input segment and gather the features only in the event-relevant region. Then the self-attention is conducted on these local features to obtain a global event feature in this modality. To localize the event temporally, we check each local segment by calculating the dot product between the global feature (from this modality) and local feature (from the other modality). The dot product result should be 1 for those event segments and 0 for the background segments.

Global Feature

$$\phi^A(S^A) = \texttt{mean}(\texttt{self-att}(F_E^A)), \quad (3) \qquad \phi^V(S^V) = \texttt{mean}(\texttt{self-att}(F_E^V)), \quad (4)$$

F_E is the features in event region T_E

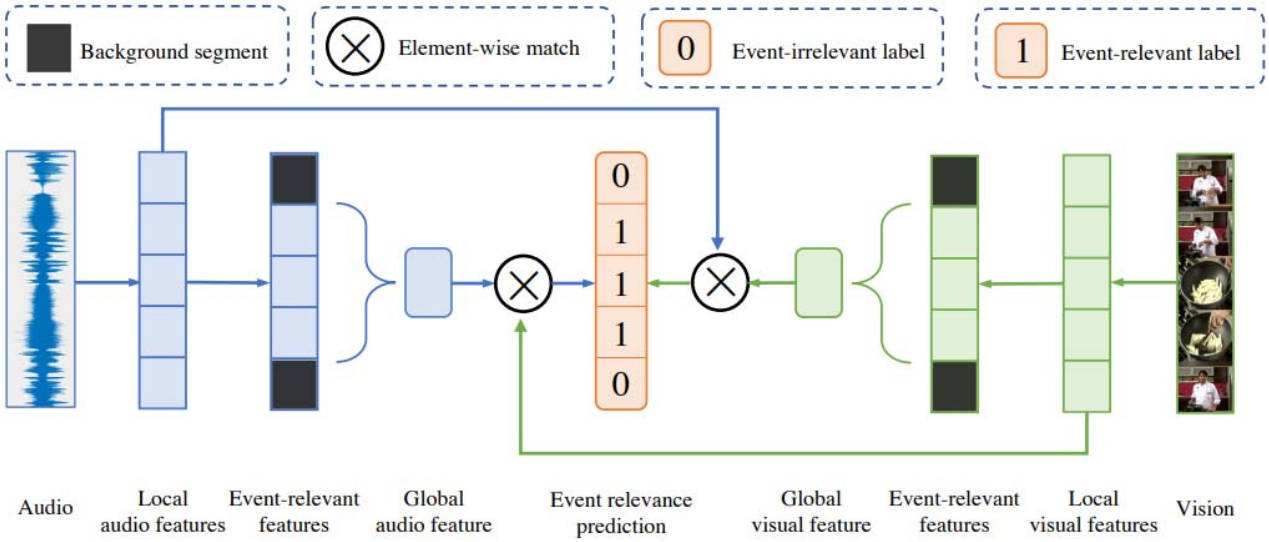


Figure 2. The proposed dual attention matching (DAM) module. DAM looks into a longer video duration to better model the high-level event information, while also attaining local temporal information by a global cross-check mechanism. DAM is optimized by finding which segments in the other are relevant to the event. We first extract the local features for each input segment and gather the features only in the event-relevant region. Then the self-attention is conducted on these local features to obtain a global event feature in this modality. To localize the event temporally, we check each local segment by calculating the dot product between the global feature (from this modality) and local feature (from the other modality). The dot product result should be 1 for those event segments and 0 for the background segments.

Event-relevant prediction

$$p_t^A = \sigma(\phi^V(S^V) \cdot f_t^A),$$
$$p_t^V = \sigma(\phi^A(S^A) \cdot f_t^V),$$
$$p_t = \frac{1}{2}(p_t^A + p_t^V).$$

Learn $p_t = 1$ for relevant region
$p_t = 0$ for non-relevant region

Results

Method	A2V	V2A	Average
DCCA [2]	34.1	34.8	34.5
AVDLN [30]	35.6	44.8	40.2
Ours	**47.1 ± 1.6**	**48.5 ± 1.4**	**47.8 ± 1.5**

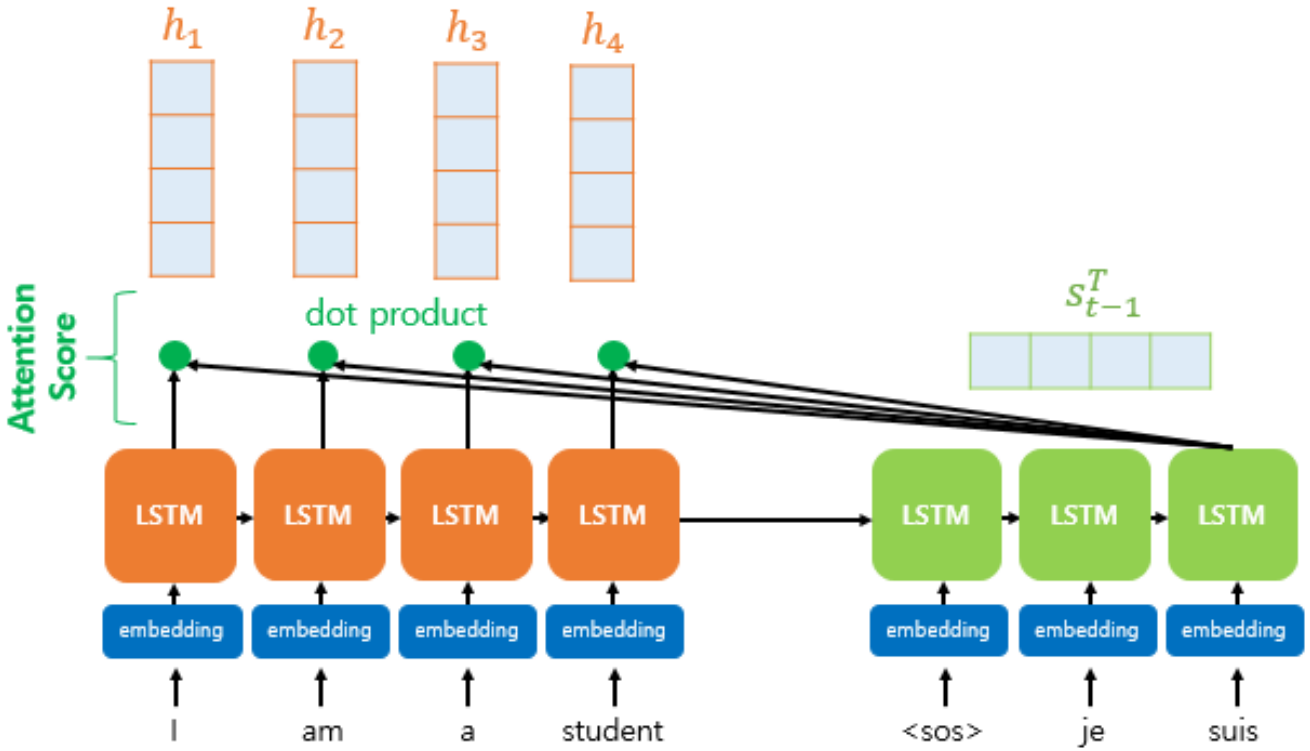
Table 1. Comparisons with the state-of-the-art methods on the cross-modality localization task. A2V: visual localization from audio sequence query; V2A: audio localization from visual sequence query. "Average" indicates the averaged score of two tasks. We report the mean and standard deviation of three runs to reduce randomness.

Method	Accuracy (%)
ED-TCN [18]	46.9
Audio (pre-trained VGG-like [14])	59.5
Visual (pre-trained VGG-19 [28])	55.3
Audio-visual [30]	71.4
AVSDN* [19]	72.6
Audio-visual+Att [30]	72.7
Ours	**74.5 ± 0.6**

Table 2. Comparisons with the state-of-the-art methods in the supervised audio-visual event localization task on the AVE dataset. * indicates the reproduced performance using the same pre-trained VGG-19 feature for a fair comparison. We report the mean and standard deviation of three runs to reduce randomness.

Thank you

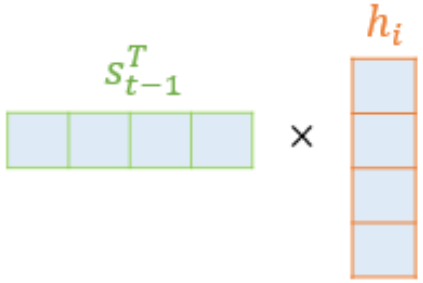
Appendix – Attention Algorithm



$$s_t = f(s_{t-1},\ y_{t-1})$$

$$s_t = f(s_{t-1},\ y_{t-1},\ a_t)$$

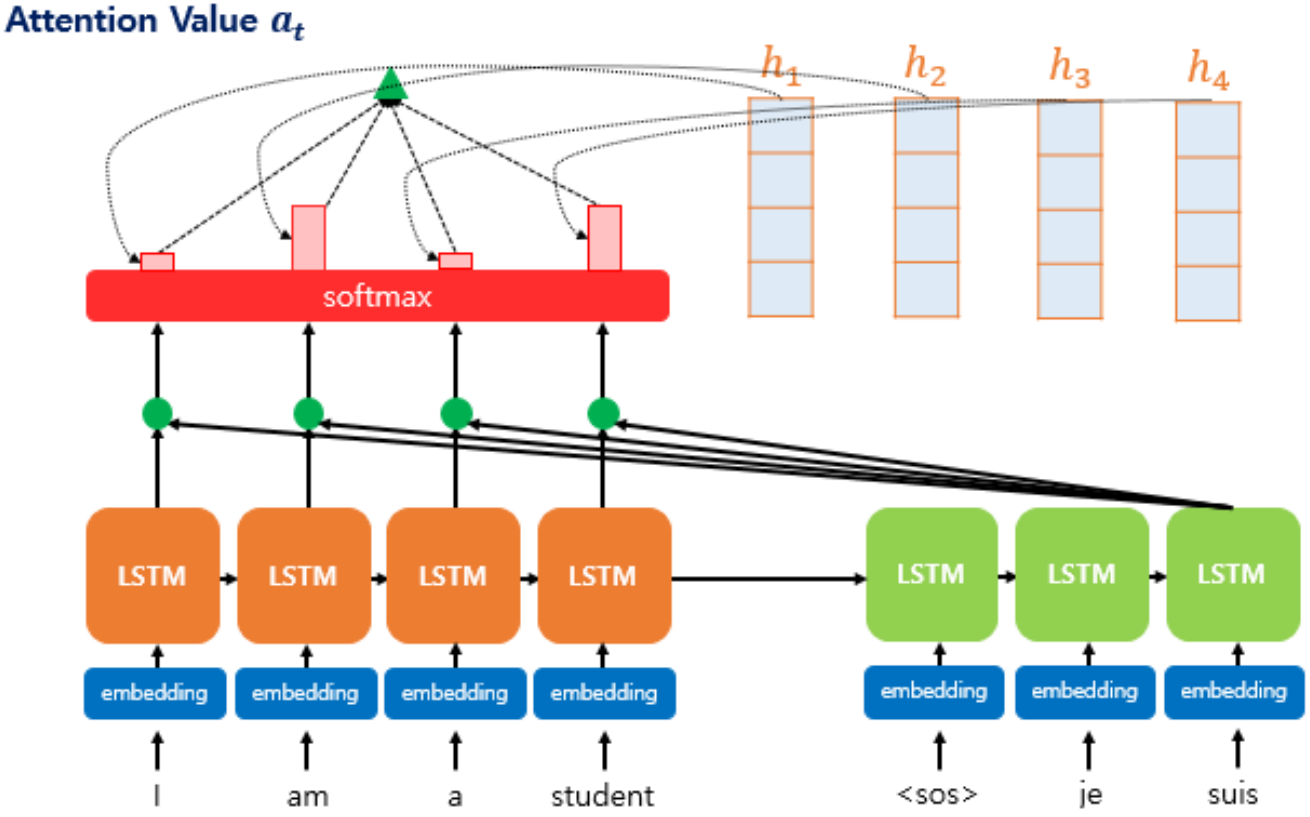
Attention Value a_t



$$score(s_{t-1},\ h_i) = s_{t-1}^T h_i$$

$$e^t = [s_{t-1}^T h_1, \ldots, s_{t-1}^T h_N]$$

Appendix – Attention Algorithm



Attention Value a_t

$$a_t = \sum_{i=1}^{N} \alpha_i^t h_i$$

$$\texttt{att}(q,k,v) = \texttt{Softmax}(\frac{qk^T}{\sqrt{d}}, v),$$

$$\texttt{self-att}(x) = \texttt{att}(W_q x, W_k x, W_v x),$$