The Supplementary Material

Yang Zhao, Zhou Zhao*, Zhu Zhang, Zhijie Lin
College of Computer Science, Zhejiang University
{awalk, zhaozhou, zhangzhu, linzhijie}@zju.edu.cn

1. Formula Derivation, Proof and Clarification

Derivation of Formula (12) First, we claim that $P(A_{i,h_i}) = 1$ is true for $\forall i \in [1, 2n_v - 1]$. It's intuitive because $a(i, h_i)$ always points to the root node and any prediction always falls into the interval corresponding to it.

Besides, we can easily point out that

$$P(A_{i,k}A_{i,k+1}\cdots A_{i,h_i}) = P(A_{i,k}) \tag{1}$$

It's quite trivial because $A_{i,k+\Delta k}$ $(\Delta k \geq 0)$ always occurs when $A_{i,k}$ occurs.

Afterwards, we try to expand the item $P(\tau_s \in \boldsymbol{\tau}_i)$ and perform a formula simplification.

$$P(\tau_s \in \boldsymbol{\tau}_i) = P(A_{i,0}) = P(A_{i,0}A_{i,1}\cdots A_{i,h_i}) \tag{2}$$

Therefore,

$$\begin{aligned} P(\tau_s \in \boldsymbol{\tau}_i) &= P(A_{i,h_i}) \prod_{j=0}^{h_i-1} P(A_{i,j}|A_{i,j+1}\cdots A_{i,h_i}) \\ &= \prod_{j=0}^{h_i-1} P(A_{i,j}|A_{i,j+1})P(A_{i,h_i}) \\ &= \prod_{j=0}^{h_i-1} P(A_{i,j}|A_{i,j+1}) \end{aligned} \tag{3}$$

And the final formula can be worked out through cascaded decision navigation procedure.

Mathematical induction on Formula (15) In order to verify that the cumulative results for decision navigation on each level conform to the definition of probability distribution, we need to prove $\sum_{i=2^h}^{2^{h+1}-1} \Phi_i = 1$ is true for $\forall h \in [0, H-1]$.

For $h = 0$, we have

$$\sum_{i=2^0}^{2^{0+1}-1} \Phi_i = \Phi_1 = 1, \tag{4}$$

Let us assume $\sum_{i=2^h}^{2^{h+1}-1} \Phi_i = 1$ is true for $h = k-1$. Hence, $\sum_{i=2^{k-1}}^{2^k-1} \Phi_i = 1$. And the next step is to prove $\sum_{i=2^k}^{2^{k+1}-1} \Phi_i = 1$ also holds. We have

$$\begin{aligned} \sum_{i=2^k}^{2^{k+1}-1} \Phi_i &= \sum_{i=2^{k-1}}^{2^k-1} (\Phi_{2i} + \Phi_{2i+1}) \\ &= \sum_{i=2^{k-1}}^{2^k-1} (\phi_i\Phi_i + (1-\phi_i)\Phi_i) \\ &= \sum_{i=2^{k-1}}^{2^k-1} \Phi_i = 1 \end{aligned} \tag{5}$$

Therefore, the cumulative results for decision navigation on each level conform to the definition of probability distribution, and we don't need to perform any extra normalization for them.

Furthur explanation for signal decomposition The process of signal decomposition can be regarded as an approximation of wavelet or fourier transformation. Although the basis function on each level is learnt by a multi-layer perceptron and doesn't have the properties held by most wavelet or fourier basis functions, we can still consider it as a composition of standard basis functions whose frequencies are equal to or less than the sampling frequency. Therefore, we actually conduct a rough transformation or decomposition by specifying the sampling frequency and coefficient of functions at each level manually and generate a proper composition of various basis functions via learnable parameters.

2. More Details for Experiment Analysis

In this section, we will further elaborate on the failure case in ActivityNet Caption [2] dataset and discuss different cases in Charades-STA [1] and TACoS [3] datasets to

*Zhou Zhao is the corresponding author.

conduct a comprehensive qualitative evaluation. Moreover, some analysis for the number of parameter and the selection of hyper-parameter is also included in this part.

Analysis on failure case in ActivityNet Caption dataset After being flipped or rotated, the patterns (including appearances and motions) of the plausible actions (i.e. *rope traverse* and *declined pull up*) are quite similar with that of the real one (i.e. *push up*), which can be observed in Figure 1. In this situation, our model misidentifies these different actions as variants of *pushing up* under the change of perspective.

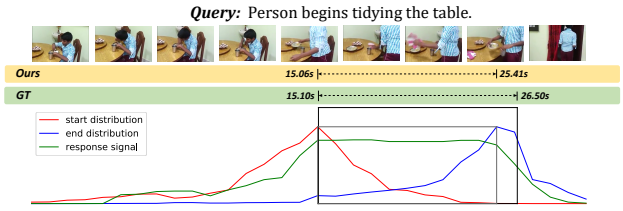
(a) push up (b) rope traverse (c) declined pull up

Figure 1. Similar appearance of different actions after flipping and rotation.

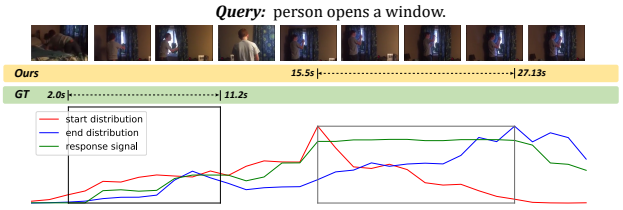
Qualitative analysis on Charades-STA dataset As shown in Figure 2, the success case indicates that our model can not only capture the right subject and object but also correctly identify the target action and relationship in the video. However, in the failure case, where the boy sequentially performs the action of *opening the curtain and window*, *picking up the phone* and *shooting out of the window*, our model fails to distinguish between *opening the window* and *shooting out of the window*. Looking into the original video, we can find it's also difficult for human to distinguish them at the given resolution directly and the action of *picking up the phone* actually gives us a great hint to infer the successive behavior. This observation suggests that high-resolution video or high-quality feature will help to boost the performance. Besides, the ability of common sense reasoning may be beneficial as well.

Qualitative analysis on TACoS dataset The success case and failure case for TACoS dataset are also illustrated in Figure 3. In this dataset, the scenes in different video segments are almost the same and various actions are quite similar, which results in only slight differences between adjacent frames. The success case demonstrates that our model can handle this problem well. But when it comes to the failure case, we can find that our model fails to capture the significant phrase *the other*, thus mistaking the process for *peeling the first kiwi* as the target result.

Hyper-parameter selection for the frame number Considering that the average frame number of these three

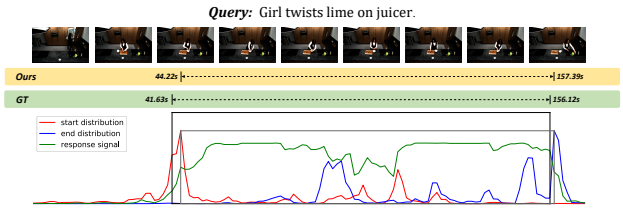


(a) Success Case



(b) Failure Case

Figure 2. Qualitative examples on the Charades-STA dataset.



(a) Success Case

(b) Failure Case

Figure 3. Qualitative examples on the TACoS dataset.

datasets varies greatly, we adopt different settings of n_v according to the characteristics of these datasets, so as to balance the target frame numbers and resampling scale. To make it clearer, we first denote the frame number of original video as n_o, and the resampling scale is given as $\frac{n_v}{n_o}$. Next we divide the original video into n_v segments using the following n_v+1 endpoints $\{0, [\frac{n_o}{n_v}], [\frac{2n_o}{n_v}], \dots [\frac{n_o(n_v-1)}{n_v}], n_o\}$, where $[x]$ represents the integer closest to x. Then we generate a new frame sequence of length n_v by applying average pooling within each segment. The prediction result can be mapped back by multiplying $\frac{n_o}{n_v}$ to align with the original video. Figure 4 shows the impact of different frame numbers on the model performance.

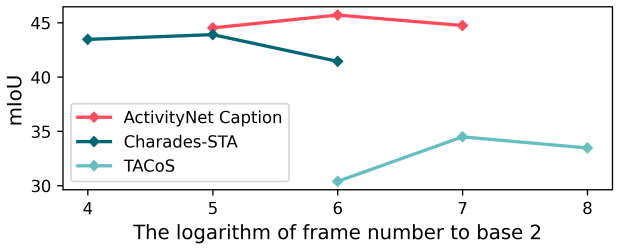


Figure 4. Impact of different frame numbers n_v on all three datasets.

Analysis for the number of parameter The number of parameter for *2D-TAN*[4] and different variants of our model are demonstrated in Table 1. Due to the parameter-sharing mechanism used in most components, every time the frame number doubles, we just need to add an extra group of navigator and decomposer, which is a multi-layer perceptron in essence. Therefore, the number of parameter in our model keeps almost unchanged as the length of video increases.

Table 1. the number of parameter for different models

Model	CPN			2D-TAN
# of frames	32	64	128	64
# of parameter	10.38M	10.50M	10.63M	91.59M

References

[1] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. Tall: Temporal activity localization via language query. In *Proceedings of the IEEE international conference on computer vision*, pages 5267–5275, 2017. 1

[2] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *Proceedings of the IEEE international conference on computer vision*, pages 706–715, 2017. 1

[3] Michaela Regneri, Marcus Rohrbach, Dominikus Wetzel, Stefan Thater, Bernt Schiele, and Manfred Pinkal. Grounding action descriptions in videos. *Transactions of the Association for Computational Linguistics*, 1:25–36, 2013. 1

[4] Songyang Zhang, Houwen Peng, Jianlong Fu, and Jiebo Luo. Learning 2d temporal adjacent networks for moment localization with natural language. *arXiv preprint arXiv:1912.03590*, 2019. 3