Supplementary Material For ReAct

1 Encoder in Detail

To be self-contained, we provide the detailed structure of the encoder. As Fig. 2 shows, for the input video feature $F \in \mathbb{R}^{T \times D}$, a local offset position and attention weight will be predicted with two fully-connected layers, respectively. For each time step, feature are then sampled according to the K offsets with linear interpolation. The sampled features are weighted by the attention weights and summed up to produce the updated frame feature for the corresponding time step.

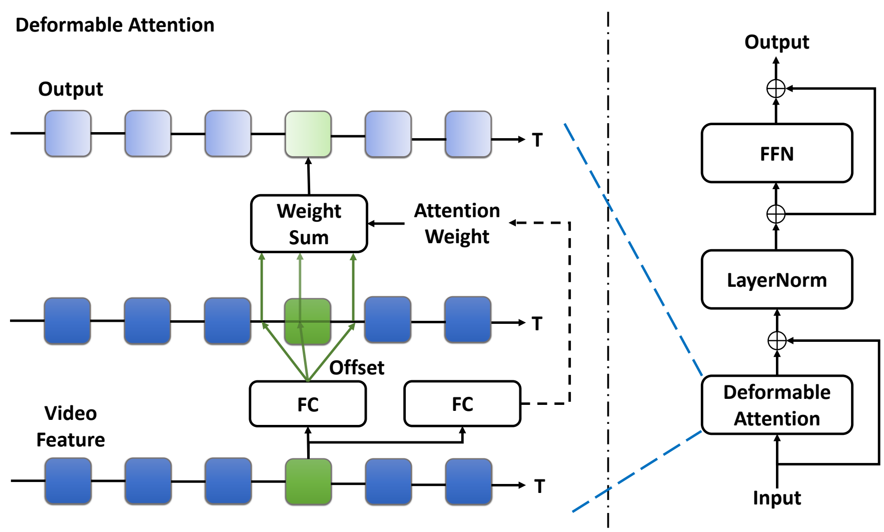

Fig. 1. Illustration of the encoder.

2 Encoder in Detail

To be self-contained, we provide the detailed structure of the encoder. As Fig. 2 shows, for the input video feature $F \in \mathbb{R}^{T \times D}$, a local offset position and attention weight will be predicted with two fully-connected layers, respectively. For each time step, feature are then sampled according to the K offsets with linear interpolation. The sampled features are weighted by the attention weights and summed up to produce the updated frame feature for the corresponding time step.

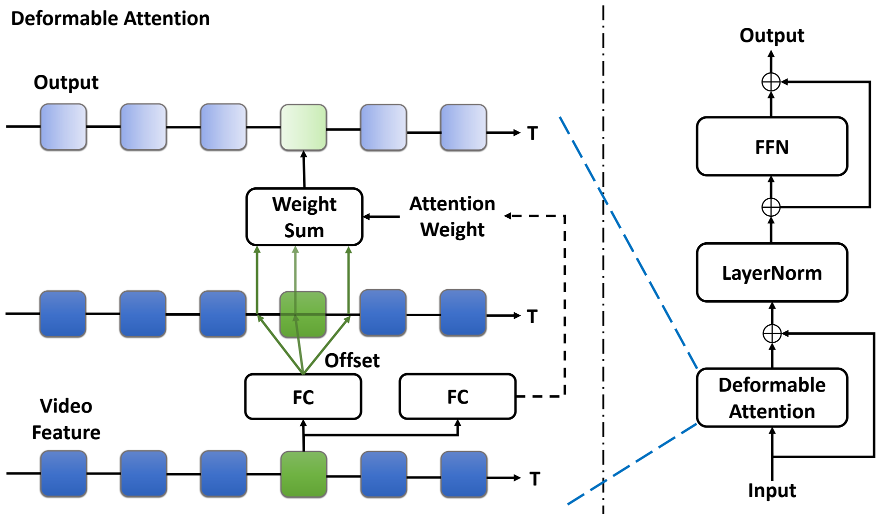

Fig. 2. Illustration of the encoder.

3 Decoder in Detail

To help understand our method better, we introduce the decoder in detail. There are two attention modules in the decoder: the proposed relational attention module and a cross-attention module.

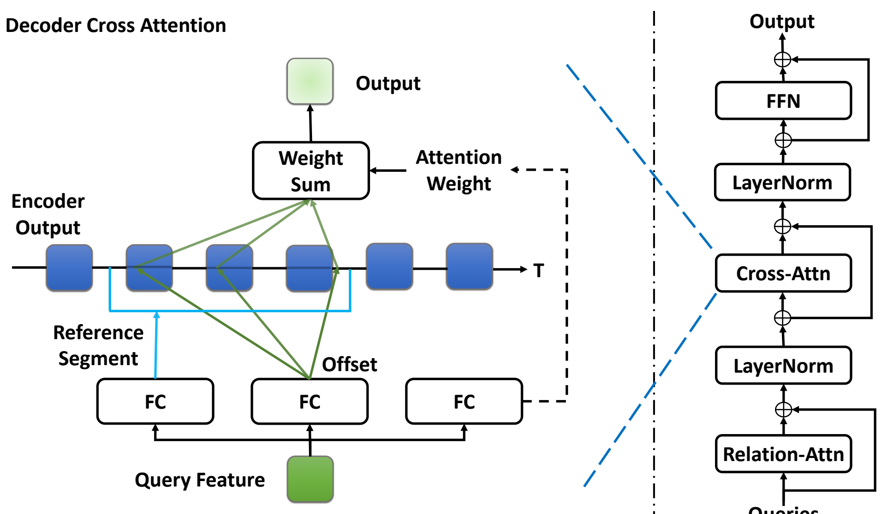

Fig. 3. Illustration of the Deformable Cross Attention module.

In the following, we elaborate on the deformable cross-attention module. As Fig. 3 showed, reference segment, offset position, and attention weights are predicted by three fully-connect layers, based on which the network samples sparse features to update the query feature at each decoder layer. There are two main differences in the deformable attention module between the encoder and decoder. First, the inputs and outputs are different. The input of the cross-attention in the decoder is the queries, while the input of the encoder is video features. The second difference is the reference segment. In the encoder, temporal offsets for each frame are sampled only around that frame. Whereas for the cross-attention module, an additional reference segment length is predicted for each query feature, and the offsets are normalizes such that the sampled frames are always in the segment.

4 Architecture and Training Detail

For THUMOS14, following [5], we use the TSN network [3] pre-trained on Kinetics [1] to extract features, which are then down-sampled every five frames. Each video feature is cropped in sequence with a window size 256, and two adjacent windows will have 192 overlapped features with a stride rate of 0.25. In the training phase, ground-truth cut by windows over 75% duration will be kept, and all empty windows without any ground-truth are removed. Finally, all ground-truth coordinates are re-normalized to the window coordinate system. we set $L_q = 40$, $L_E = 2$, $L_D = 4$ for the number of queries, encoder layer and decoder layer, respectively. Each deformable attention module will sample 4 temporal offsets for computing the attention. The hidden layer dimension of the feedforward network is set to 1024, and the other hidden feature dimension in the intermediate of the network is all set to 256. The pair-wise IoU threshold τ and feature similarity threshold γ in ACE module are set to 0.5 and 0.2, respectively. For ActivityNet, the pre-trained TSN network by Xiong *et al.* [4] is adopted to extract features. Then each video feature downsamples every 16 frames, and the resultant feature will be rescaled to 100 snippets using linear interpolation. We only do video-level detection instead of window-level. We set the $L_q = 60$, $L_E = 3$, $L_D = 4$. We sample 4 temporal offsets for the deformable module. The dimension of hidden features is set to 256, and we set the pair-wise IoU threshold τ and feature similarity threshold γ to 0.9 and -0.2, respectively. Following previous works [5, 7, 8, 6], we combined the Untrimmed-Net video-level classification results [2] with our classification score.

5 Visualization of the Classification Loss

To further demonstrate the effect of ACE-*dec* loss, we compute the classification loss for the Activitynet-1.3 test set. As Fig. 4 shows, compared to the Focal Loss, the ACE-*dec* loss improves not only the convergence speed but also the accuracy.

References

1. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al.: The kinetics human action video dataset. arXiv preprint arXiv:1705.06950 (2017)

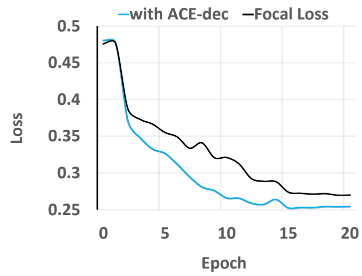

Fig. 4. Visualization of the test classification loss. We record the testing loss with or without ACE-*dec* loss during training

2. Wang, L., Xiong, Y., Lin, D., Van Gool, L.: Untrimmednets for weakly supervised action recognition and detection. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 4325–4334 (2017)
3. Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Van Gool, L.: Temporal segment networks for action recognition in videos. IEEE transactions on pattern analysis and machine intelligence **41**(11), 2740–2755 (2018)
4. Xiong, Y., Wang, L., Wang, Z., Zhang, B., Song, H., Li, W., Lin, D., Qiao, Y., Van Gool, L., Tang, X.: Cuhk & ethz & siat submission to activitynet challenge 2016. arXiv preprint arXiv:1608.00797 (2016)
5. Xu, M., Zhao, C., Rojas, D.S., Thabet, A., Ghanem, B.: G-tad: Sub-graph localization for temporal action detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10156–10165 (2020)
6. Yang, L., Peng, H., Zhang, D., Fu, J., Han, J.: Revisiting anchor mechanisms for temporal action localization. IEEE Transactions on Image Processing **29**, 8535–8548 (2020)
7. Zeng, R., Huang, W., Tan, M., Rong, Y., Zhao, P., Huang, J., Gan, C.: Graph convolutional networks for temporal action localization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7094–7103 (2019)
8. Zhao, P., Xie, L., Ju, C., Zhang, Y., Wang, Y., Tian, Q.: Bottom-up temporal action localization with mutual regularization. In: European Conference on Computer Vision. pp. 539–555. Springer (2020)