Temporal Saliency Query Network for Efficient Video Recognition (Supplementary)

Table 1. Supplementary Material Overview.

Section	Content
A.	Implementation Details.
B.	Practical Inference Speed.
C.	Additional Ablation study.
D.	Additional Qualitative Analysis.
E.	Additional Visualization of TSQ Embeddings.

A Implementation Details

Here we provide some implementation details of TSQNet. We use MobileNetv2 and EfficientNet-B0 as the video encoder in VQM and object recognizer in TQM, respectively. For video encoder in TQM, we use the ImageNet pre-trained model and finetuned it on target datasets *e.g.,* ActivityNet, *etc.,* for 10 epochs. And for Object recognizer, we directly use the officially released ImageNet model to extract object score of the ImageNet 1000 classes. We use positional embedding on frame sequence in transformer decoder to model temporal order information.

Next, we introduce how we obtain visual prototype based representation for visual TSQ embeddings initialization. First we apply a classifier to get the classification results for each frame. Then we select the top m percent of frames which can correctly predict the ground truth video category, which are then averaged to obtain the representation of each video. Finally, we pool all the video representations of each category to get the prototype representation of each category. We use $m = 30$ for all experiments in this paper.

Finally, we describe in detail how to fuse the VQM and TQM salient scores into the final saliency measurement. Suppose we have the VQM salient scores $S^v \in \mathbb{R}^T$ and TQM salient scores $S^t \in \mathbb{R}^T$ of one video. We join the top saliency score frames from two modalities to get final K salient frames. Specifically, the number of selected frames from two modalities are determined by $\lambda_v K$ and $\lambda_t K$, respectively, where $\lambda_v + \lambda_t = 1$. For example of selecting 5 frames from 16 frames with $\lambda_v = 0.6$ situation, we select top $5 \times 0.6 = 3$ frames from VQM and top $5 - 3 = 2$ ones are from TQM. And if there exists duplication, which results in a final result of less than 5 frames, the selection will be deferred in the VQM according to the descending order of S^v until meeting the 5-frame budget. We use $\lambda_v = 0.6$ and $\lambda_t = 0.4$ in experiments.

B Practical Inference Speed

To further verify the practical efficiency of our method, we compare the inference speed with two state-of-the-art methods FrameExit [1] and AdaFocus [2] on ActivityNet. FrameExit [1] reduce computation cost by early stopping in temporal sequential prediction. AdaFocus [2] suppose that the existing methods are spatially redundant, so it only selects salient areas to classify for each frames. We test the speed of two methods by running the official code released by the authors. We evaluate the inference speed of all methods on a NVIDIA 3090 GPU with Intel(R) Xeon(R) CPU E5-2650 v3 @ 2.30GHz CPU. Results in two different settings with batch size of 1 and 32 are reported. Note that FrameExit [1] exits from recognition at different time for different videos, so it cannot inference in batch setting, which we only report the latency with batch size = 1 here. Experimental results in Table 2 show that our method not only saves much theoretical computation complexity but also achieves the fastest actual inference speed (**121.1** video/s) on both single-sample and batch setting.

Table 2. Comparisons of practical inference speed with state-of-the-art methods on ActivityNet.

Method	mAP (%)	FLOPs (G)	Throughput(bs=1) (videos/s)	Throughput(bs=32) (videos/s)
AdaFocus [2]	75.0	26.6	5.5	73.8
FrameExit [1]	76.1	**26.1**	9.8	-
Ours	**76.5**	**26.1**	**17.7**	**121.1**

C Additional Ablation Study

In this section, more ablation experiments are conducted to supplement the main paper. ResNet-101 is utilized for the recognition network as the same as in ablation studies of the main paper.

C.1 Ablation of class-specific classifier

We show the illustrative examples of the combinations of the attention structure and the class-specific classifier under the situation of 200 class and 1280 feature dimensions in Figure 1. "CA + CA" represents the class-agnostic attention structure (with 1 query) combined with the class-agnostic classifier (with a single FC). "CS + CA", i.e. our TSQNet, the class-specific attention structure (with C queries) combined with the class-agnostic classifier (with a single FC). "CS + CS" represents the class-specific attention structure (with C queries)

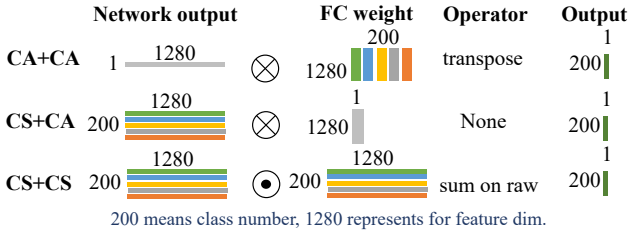

Fig. 1. Illustrative examples of three combinations between the attention structure and the classifier of TSQNet, i.e. "CS+CA", "CA+CA" and "CS+CS".

combined with class-agnostic classifier (with C FCs). It is interesting that the performance of "CS+CA" (68.3) is much lower than that of "CA+CA" (73.3), which seems like a more naive baseline than "CS+CA". When using the class-specific attention structure to obtain feature with shape of 200×1280, the FC classifier (200×1280) must have a one-to-one correspondence with each class, *i.e.,* "CS+CS" (74.3), to achieve good results. If using one 1×1280 FC, *i.e.,* "CS+CA", to process all classes with the same parameters, discrimination power are insufficient and accuracy will decrease dramatically, which will be even lower than "CA+CA".

C.2 Ablation study of α and β

First, we explore the appropriate values for α and β, *i.e.,* the ratios of $\mathcal{L}_{t\to v}$ and $\mathcal{L}_{v\to t}$ in Table 3 and Table 4, respectively. We first fix $\alpha = 0$ to find the best β. As shown in Table 3, as β increases, the performance of both TQM and TSQNet rises up to a maximum at $\beta = 0.6$ and then falls down. The performance of VQM remains unchanged, which demonstrates $\mathcal{L}_{t\to v}$ mainly benefit TQM in interactions. Then we fix $\beta = 0.6$ to explore the impacts of α. As presented in Table **??**, the performance shows similar trend and the best results of TQM, VQM and TSQNet are achieved when α and β both equal to 0.6, which implies $\mathcal{L}_{v\to t}$ benefits both TQM and VQM in interactions. After $\beta = 0.6$, the performance of VQM breaks down, for prohibitively large β hinders the convergence of the VQM.

C.3 Detailed Ablation study for transformer decoder structures

We further ablate the structure of the standard transformer decoder, *viz.,* self-attention, number of layers and heads. Typical transformer decoder contains a self-attention layer on the top of query matrix and multiple cross-attention layers with multi-head structure. In TSQNet, we use a quite brief version of transformer decoder, containing a single-head cross-attention layer without self-attention layers, to realize TSQ layer. Next we discuss the effectiveness of this design.

Table 3. Ablation study of β when fixing $\alpha = 0$.

β	TSQNet	TQM	VQM
0.0	74.9	72.0	74.6
0.2	74.9	72.3	74.6
0.4	75.0	72.5	74.6
0.6	**75.1**	**72.7**	74.6
0.8	74.8	72.6	74.6
1.0	74.7	72.6	74.6

Table 4. Ablation study of α when fixing $\beta = 0.6$.

α	TSQNet	TQM	VQM
0.0	75.1	72.7	74.6
0.2	75.0	72.8	74.6
0.4	75.1	72.8	74.7
0.6	**75.3**	**73.1**	**74.8**
0.8	71.2	72.5	67.5

Impact of Self-attention layer. On one hand, self-attention layer on queries make each TSQ embedding interact with each other, which may cause the class-specific information to mix with each other and deviates the class-specific nature of TSQ embeddings. On the other hand, self-attention layers bring in extra computation complexity of $O(C^2)$, where C is the number of categories. As shown in Table 5, adding self-attention layer presents lower performance, which demonstrates that modelling relations between TSQ embeddings of categories can not produce better saliency measuring results.

Table 5. Ablation study of the usage of self-attention.

Methods	mAP (%)
w/ self-atten	74.2
w/o self-atten	**74.4**

Table 6. Ablation study of Transformer Decoder layers and heads.

Methods	mAP (%)
1 layer 8 head	73.7
2 layer 1 head	73.6
1 layer 1 head	**74.4**

Number of layers and heads. In TSQNet, the number of cross-attention layers and heads are both one. We present ablation experiments of more layers with more heads in Table 6. It is shown that both the increase of number of layers and heads make the mAP drop. For multiple cross-attention layers, the performance drop may attribute to lower discrepancy between queries in intermediate layers, which makes attention weights lack discrimination power between categories. For multi-head structure, the worse results may result from attention dimension splitting operation when calculating the similarity between query matrix and key matrix, which produces separate local similarities for multiple groups in feature dimension rather than the holistic similarity of the feature dimension.

D Additional Qualitative Analysis

Figure 2 and Figure 3 show more qualitative results of TSQNet on ActivityNet and FCVID. For each dataset, we selected six examples, first three of which

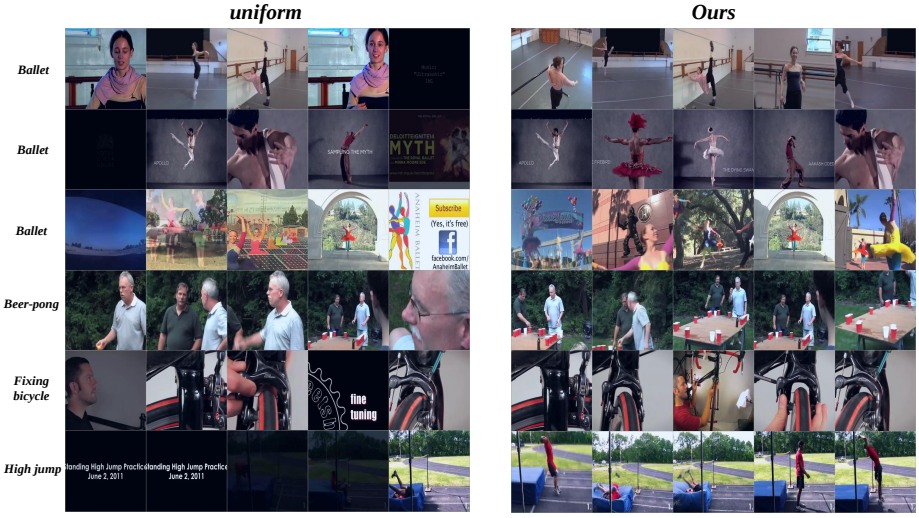

Fig. 2. Qualitative Analysis on ActivityNet.

belongs to the same category and the last three belongs to different categories. In Figure 2, we can see that our approach samples significantly more salient frames than the uniform baseline. Similar in Figure 3, uniform baseline selects many irrelevant frames, whereas our method selects more theme-related frames.

E Additional Visualization of TSQ Embeddings

In this section, we provide the complete t-SNE visualization for TSQ embeddings of both the VQM and TQM on ActivityNet to supplement the local zooms visualization in Section 4.5 of the main paper. Specifically, we visualize the start and end states of training for the two modules in two different initialization fashions, *i.e.,* random and proposed initialization, respectively. For VQM, we compare the random initialization with the visual common appearance feature initialization. For TQM, we compare the random initialization with the class name Bert embedding feature initialization.

TSQ embeddings of TQM. Figure 4 shows the visualization of TSQ embeddings of TQM with class name Bert embedding feature initialization **before** training. Figure 5 shows the visualization of TSQ embeddings of TQM with class name Bert embedding feature initialization **after** training. Figure 6 shows the visualization of TSQ embeddings of TQM with random initialization **before** training. Figure 7 shows the visualization of TSQ embeddings of TQM with random initialization **after** training.

TSQ embeddings of VQM. Figure 8 shows the visualization of TSQ embeddings of VQM with common appearance feature initialization **before** training. Figure 9 shows the visualization of TSQ embeddings of VQM with common

Fig. 3. Qualitative Analysis on FCVID.

appearance feature initialization **after** training. Figure 10 shows the visualization of TSQ embeddings of VQM with random initialization **before** training. Figure 11 shows the visualization of TSQ embeddings of VQM with random initialization **after** training.

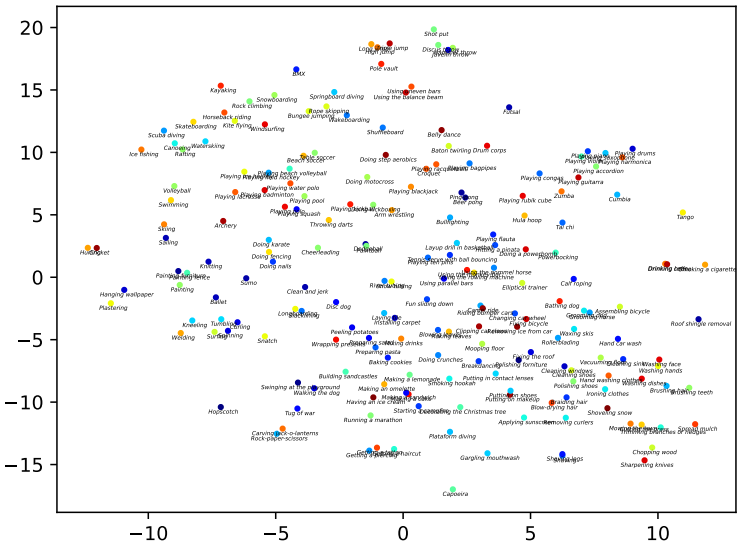

Fig. 4. TSQ embeddings of TQM with class name Bert initialization before training.

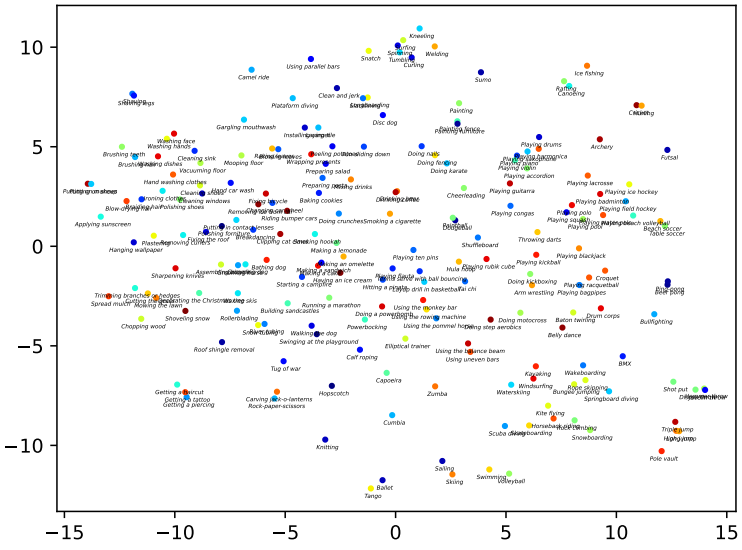

Fig. 5. TSQ embeddings of TQM with class name Bert initialization after training.

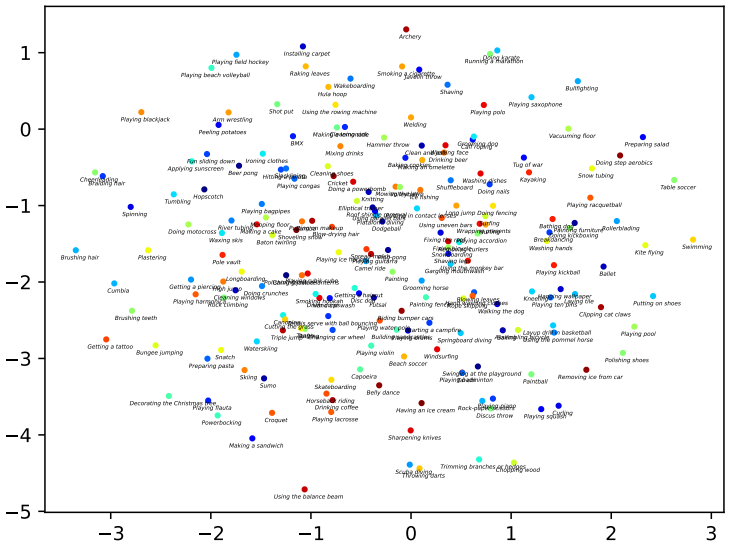

Fig. 6. TSQ embeddings of TQM with random initialization before training.

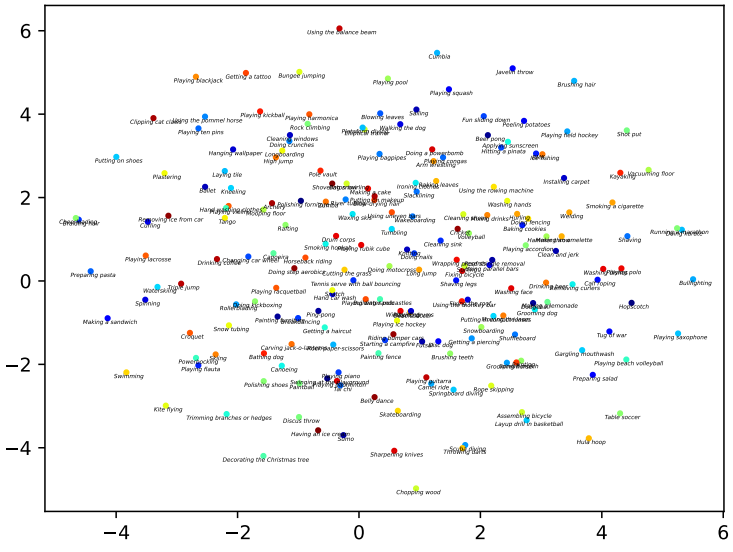

Fig. 7. TSQ embeddings of TQM with random initialization after training.

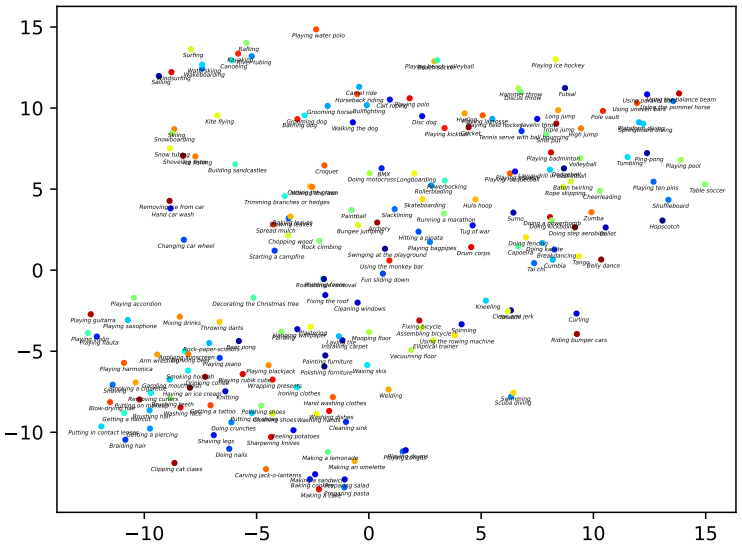

Fig. 8. TSQ embeddings of VQM with appearance initialization before training.

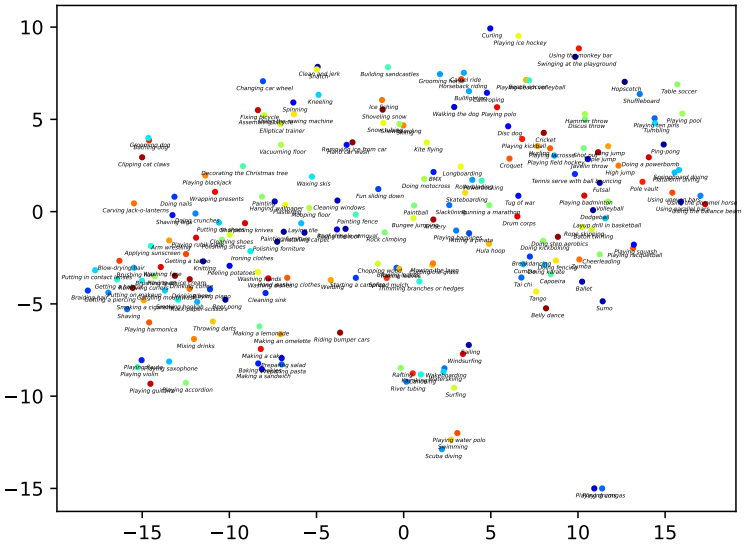

Fig. 9. TSQ embeddings of VQM with common appearance initialization after training.

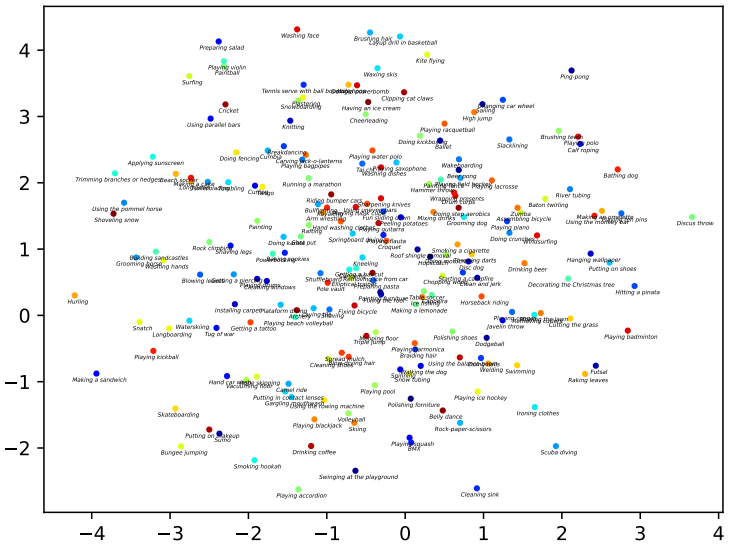

Fig. 10. TSQ embeddings of VQM with random initialization before training.

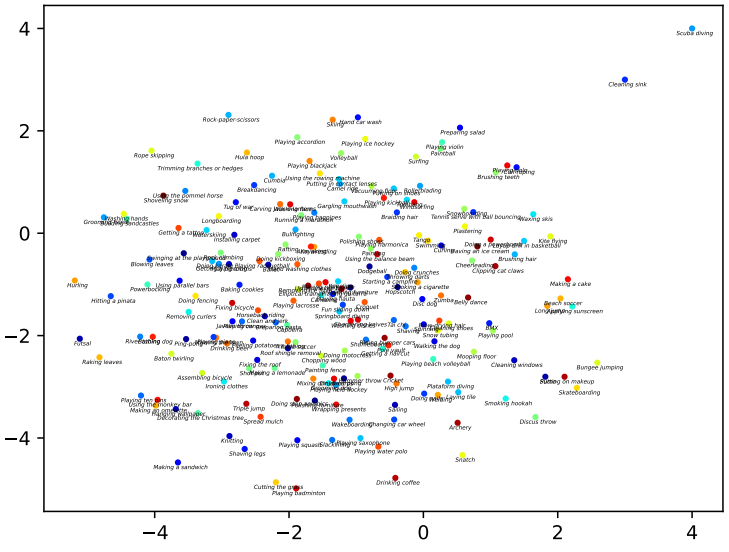

Fig. 11. TSQ embeddings of VQM with random initialization after training.

References

1. Ghodrati, A., Bejnordi, B.E., Habibian, A.: Frameexit: Conditional early exiting for efficient video recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15608–15618 (2021)
2. Wang, Y., Chen, Z., Jiang, H., Song, S., Han, Y., Huang, G.: Adaptive focus for efficient video recognition. arXiv preprint arXiv:2105.03245 (2021)