X-Shot results

In the main paper, we introduced Temporal-Realational CrossTransformers (TRX) for few-shot action recognition. They are designed specifically for \( K \) \( > 1 \) problems where TRX is able to match sub-sequences from the query against sub-sequences from multiple support set videos.

Table 1 in the main paper shows results on the standard 5-way 5-shot benchmarks on Kinetics [3], Something-Something V2 (SSv2) [4], HMDB51 [5] and UCF101 [6]. For completeness we also provide 1-, 2-, 3-, 4- and 5-shot results for TRX with \( \Omega = \{1\} \) (i.e. frame-to-frame comparisons) and \( \Omega = \{2, 3\} \) (i.e. pair and triplet comparisons) on the large-scale datasets Kinetics and SSv2. These are in Table 1 in this supplementary, where we also list results from all other works which provide these scores.

For 1-shot, in Kinetics, TRX performs similarly to recent few-shot action-recognition methods [8, 1, 7], but these are all outperformed by OTAM [2]. OTAM works by finding a strict alignment between the query and single support set video per class. It does not scale as well as TRX when \( K > 1 \), shown by TRX performing better on the 5-shot benchmark. This is because TRX is able to match query sub-sequences against similar sub-sequences in the support set, and importantly ignore sub-sequences (or whole videos) which are not as useful. Compared to the strict alignment in OTAM [2], where the full video is considered in the alignment, TRX can exploit several sub-sequences from the same video, ignoring any distractors. Despite not being as well suited to 1-shot problems, on SSv2 TRX performs similarly to OTAM. 2-shot TRX even outperforms 5-shot OTAM. Table 1 again highlights the importance of tuples, shown in the main paper, where TRX with \( \Omega = \{2, 3\} \) consistently outperforms \( \Omega = \{1\} \).

Figure 5 in the main paper shows how TRX scales on SSv2 compared to CMN [8, 9], which also provides X-shot results (\( 1 \leq X \leq 5 \)). The equivalent graph for Kinetics is shown in Fig. 1 here. This confirms TRX scales better as the shot increases. There is less of a difference between TRX with \( \Omega = \{1\} \) and \( \Omega = \{2, 3\} \), as Kinetics requires less temporal knowledge to discriminate between the classes than SSv2 (ablated in Sec. 4.3.1 and 4.3.2 in the main paper).

The impact of positional encoding

TRX adds positional encodings to the individual frame representations before concatenating them into tuples. Table 2 shows that adding positional encodings improves SSv2 for both single frames and higher-order tuples (by +0.3\% and +0.6\% respectively). For Kinetics, performance stays the same as single frames and improves slightly with tuples (+0.4\%) for the proposed model. Overall, positional encoding improves the results marginally for TRX.

References

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>Kinetics</th>
<th>SSv2†</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>85.2</td>
<td>53.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>85.2</td>
<td>53.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>85.5</td>
<td>58.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>85.9</td>
<td>59.1</td>
</tr>
</tbody>
</table>

Table 2: The importance of incorporating positional encoding for single frames and the proposed model \(\Omega=\{2,3\}\).


