In this supplementary, we present additional quantitative and qualitative results of our proposed few-shot (FS) action recognition framework, STRM. The quantitative results are discussed in Sec. A1 followed by qualitative analysis in Sec. A2.

### A1. Additional Quantitative Results

#### Impact of joint spatio-temporal enrichment:
Tab. A1 shows the impact of replacing our patch-level enrichment (PLE) and frame-level enrichment (FLE) sub-modules in the proposed STRM framework with a joint spatio-temporal (Jnt-ST) enrichment sub-module on the SSv2 [1] dataset. The performance of Baseline TRM is also shown for comparison. Jointly enriching all the spatio-temporal patches across the frames, as in Jnt-ST, does improve the performance over the baseline but with a 50% increase in FLOPs due to computing attention over all the spatio-temporal patches in a video. Although using two layers of Jnt-SA gains over the single layer variant, it requires twice the number of FLOPs than Baseline TRM. Our proposed approach of enriching patches locally with in a frame and then enriching the frames globally in a video requires only ~ 4% additional FLOPs over the baseline and obtains superior performance. This shows the importance of proposed enrichment mechanism in our STRM framework.

#### Table A1. Impact of replacing our PLE and FLE sub-modules with joint spatio-temporal self-attention sub-module on SSv2.

<table>
<thead>
<tr>
<th>Baseline TRM</th>
<th>Jnt-ST(1)</th>
<th>Jnt-ST(2)</th>
<th>Ours:STRM</th>
</tr>
</thead>
<tbody>
<tr>
<td>62.1</td>
<td>64.7</td>
<td>65.8</td>
<td>68.1</td>
</tr>
</tbody>
</table>

#### Impact of varying the enrichment mechanism:
We present the impact of varying the enrichment mechanisms in our PLE and FLE sub-modules in Tab. A2 on the SSv2 dataset. It is worth mentioning that irrespective of the enrichment mechanism employed, integrating PLE and FLE sub-modules enhances the feature discriminability, leading to improved performance over Baseline TRM. However, we observe that employing an MLP-mixer [4] for enriching patches locally with in a frame (PLE) or employing self-attention [5] for enriching frames globally across frames in a video (FLE) results in sub-optimal performance. This is because self-attention enriches the tokens locally in a pairwise and sample-dependent manner and is likely to be less suited for enriching the frames at a global level. Similarly, the MLP-mixer is sample-agnostic and enriches the tokens globally through a persistent relationship memory while being less suitable for enriching the patches at a local level.

Thereby, employing self-attention for local patch-level enrichment and simultaneously an MLP-mixer for global frame-level enrichment achieves the best performance and achieves an absolute gain of 6.0% over baseline. These results emphasize the efficacy of enhancing spatio-temporal
features by integrating local (sample-dependent) patch-level and global (sample-agnostic) frame-level enrichment along with a query-class similarity classifier in our STRM for the task of FS action recognition.

**Impact of varying $\lambda$:** Fig. A1 shows the FS action recognition performance comparison for different values of $\lambda$, which is the weight factor for the query-class similarity classification loss in the proposed STRM framework. Setting $\lambda$ high (> 0.4) is likely to decrease the importance of the modeling temporal relationships between query and support actions in the TRM module during training and consequently leads to a drop in performance. Furthermore, we observe that employing this intermediate layer classification loss with a low weight (around 0.1) improves the performance and achieves the best results of 68.1% accuracy for FS action recognition on the SSv2 dataset.

**Class-wise performance gains:** Fig. A2 shows the class-wise gains obtained by the proposed STRM framework over Baseline TRM on the SSv2 dataset. We observe that our STRM achieves gains above 10% for classes such as Dropping something next to something, Showing something next to something, etc. Out of 24 action classes in the test set, our STRM achieves performance gains on 21 classes. These results show that enriching the features by encoding the spatio-temporal contexts aids in improving the feature discriminability, leading to improved FS action recognition performance.

**A2. Additional Qualitative Results**

Here, we present additional qualitative results w.r.t. tuple matching between query and support actions in Fig. A3 to A7. In each example, a query video is shown on the top along with its ground-truth class name. Three query tuples of cardinality two are shown in red, green and blue. Their corresponding best matches in the support videos (of ground-truth action) obtained by Baseline TRM and our STRM are shown on the left and right, respectively. Generally, we observe that the best matches obtained by Baseline TRM do not encode the same representative features as in the corresponding query tuple. E.g., blue and red tuples in 4th and 5th support videos of Fig. A3, red and blue tuples in 1st and 3rd support videos of Fig. A4. These results show that hand-crafted temporal representations in Baseline TRM are likely to not encode class-specific spatio-temporal context at lower cardinalities. In contrast, our STRM obtains best matches that are highly representative of the corresponding query tuples and also encodes longer temporal variations. E.g., green and blue tuples in 4th and 5th support videos of Fig. A3, blue tuple in 5th support video of Fig. A4. The improved tuple matching between query and support actions in STRM is due to the proposed spatio-temporal feature enrichment, comprising patch-level and frame-level enrichment, which enhances the feature discriminability and the learning of the higher-order temporal representations at lower cardinalities that improves the model flexibility. Furthermore, Fig. A8 shows additional attention map visualizations on four example (novel) classes in the SSv2 dataset. Our STRM is able to emphasize the action-relevant objects in the video reasonably well. E.g., in Fig. A8(a), remote is emphasized in frames 2, 3 and 7. Similarly, while the bag’s position is emphasized in frames 6 and 7, the focus is on the table early on, which is required to reason out the Dropping Something next to Something action in Fig. A8(c). We also observe that fine-grained novel actions with subtle 2D motion differences are harder to classify, e.g., Pretending to put Something behind Something vs. Pretending to put Something underneath Something. In general, our STRM learns to emphasize relevant spatio-temporal features that are discriminative, leading to improved FS action recognition performance.

In summary, these quantitative and qualitative results
Figure A3. **Qualitative comparison between Baseline TRM and our STRM w.r.t. tuple matches.** Three query tuples of cardinality two are shown in red, green and blue for the query video at the top. Their corresponding best matches in the support videos (of ground-truth action) obtained by Baseline TRM and our STRM are shown on the left and right, respectively. The best matches for the blue and red tuples (4th and 5th support videos) in Baseline TRM do not encode the action completely and are less discriminative. We observe that our STRM is able to capture better matches with longer temporal variations (green and blue tuples in 4th and 5th support videos) due to the learned higher order temporal representations. See Sec. A2 for additional details.

Figure A4. **Qualitative comparison between Baseline TRM and our STRM w.r.t. tuple matches.** See Fig. A3 and Sec. A2 for additional details. The best matches for red and blue query tuples obtained by STRM (4th and 5th support videos) are better representatives of the corresponding query tuples, in comparison to the best matches found by Baseline TRM (1st and 3rd support videos).

along with the comprehensive experiments performed (main paper) emphasize the benefits of our proposed spatio-temporal enrichment module in enhancing feature discriminability and model flexibility, leading to improved few-shot action recognition.

**A3. Societal Impact and Future Direction**

Automated understanding of human actions from videos, when deployed responsibly, can be useful in multiple applications. Examples include human-robotic interaction in elderly care facilities, where robots need to process human
actions to effectively assist them. Other applications include behavior analysis for mental-health counseling and athletic rehabilitation. However, if deployed irresponsibly, automated action recognition techniques can be used to retrieve or summarize sensitive clips that breach individual privacy, similar to most computer vision research problems. A likely future work direction, beyond the scope of our current work, is to broaden the few-shot action recognition capability to generalize across varying domains.

A4. Additional Implementation Details

The input videos are rescaled to a height of 256 and $L = 8$ frames are uniformly sampled, as in [3]. Random $224 \times 224$ crops are used as augmentation during training. In contrast,
Figure A7. **Qualitative comparison between Baseline TRM and our STRM w.r.t. tuple matches.** See Fig. A3 and Sec. A2 for additional details. The Baseline TRM fails to obtain support tuples that are representative enough for the query tuples in red and green. Our STRM alleviates this issue and obtains good representative matches (1st and 5th support videos) since it enhances the feature discriminability through patch-level as well as frame-level enrichment and learns higher-order temporal representations.

Figure A8. **Attention map visualization for four example classes in SSv2.** Our STRM learns to emphasize relevant spatio-temporal features that are discriminative, leading to improved FS action recognition performance. For instance, relevant objects for corresponding actions are emphasized: `remote` in frames 2, 3 and 7 in (a), `gems` in frames 4, 6, 7 and 8 in (d), respectively. Similarly, in (c), the focus on the the `table` early on shifts to the `bag`'s position in frames 6 and 7, which is required to reason out the `Dropping Something next to Something` action.

only a centre crop is used during evaluation. We use the PyTorch [2] library to train our STRM framework on four NVIDIA 2080Ti GPUs. Since only a single few-shot task can fit in the memory, the gradients are accumulated and backpropagated once every 16 iterations.

References


