A Discussion on Experimental Results

Two closest works to our TeG setup are CVRL [48] and ρMoco/BYOL/SwAV [18], where a deep 3D-ResNet-50 is adopted as the backbone and a contrastive loss is used as the objective. However, the code and models of [18] are not available at the time of submission. We tried to reproduce their models but found it difficult and expensive to match all their results, mainly due to the huge computational costs required with its proposed multi-clip sampling strategy on Kinetics. Furthermore, there are four variants of their models (SimCLR, MoCo, BYOL, SwAV) with different multi-clip settings ($\rho = 1, 2, 3, 4$), making it even harder to reproduce. Since CVRL [48] is a concurrent work of [18] with a similar objective and open-sourced, we choose to follow CVRL's settings on architecture and hyper-parameters. Tab. 4 shows the detailed comparison of 3D-ResNet-50 backbones used in two works. Our primary goal here is to investigate the difference between temporally persistent and fine-grained features with strong baselines, not solely competing for best numbers.

For experiments, we try our best to incorporate the results from [18] and [48] as much as possible towards a fair comparison. We list three factors in Tab. 3 (pre-train epoch, frame and FLOPs) and offer more discussions here.

The recent state-of-the-art self-supervised learning methods on the image domain [9, 13, 24, 28] heavily rely on the accuracy to verify the effectiveness of proposed algorithm. Due to the difficulty of direct comparison between different methods, training time and training computation are hardly listed in an explicit way. As in our case, we try to incorporate these factors into comparison.

Firstly, for pre-train epoch, we argue that it is not directly comparable between different methods. For example, [18] finds that increasing pre-train epoch from 400 to 800 epochs, leads to identical performance for ρMoCo, and detrimental for ρBYOL and ρSimCLR. And ρSwAV is the strongest performer for short training of 50 epochs. A few potential reasons affecting the convergence speed could be: 1) using a large memory bank (ρMoCo), 2) using a momentum encoder (ρMoCo, ρBYOL), 3) using negative examples (ρSimCLR, ρMoCo).

Secondly, we also list the pre-train frames as a reference since it directly affects the pre-train FLOPs.

Lastly, for the pre-train FLOPs, while [18] provides the FLOPs for a single clip when they benchmark on different backbones from S3D to R3D, we argue it is important to report the FLOPs of all clips when compare on the same backbone like R3D. For example, the performance of ρBYOL($\rho = 1$) achieves 60.6% on Kinetics-400, while ρBYOL($\rho = 4$) achieves 68.9%. For these two models, the FLOPs of a single clip remain the same, while ρBYOL($\rho = 4$) have $4\times$ FLOPs when we calculate the total FLOPs of all clips. Thus we consider the total FLOPs as a better metric for reflecting the actual computation for a method.

For results in Tab. 3, we achieve a competitive performance (67.8%) with a moderate computational cost (136G). And we want to re-emphasize that our primary goal here is to investigate the difference between temporally persistent and fine-grained features with strong baselines (65%+ for both TeG-PS and TeG-FG, and we observe a 2.8% difference with is large on K400), not solely competing for best numbers.

B Additional Implementation Details

B.1 VidSitu

VidSitu [52] contains 23.6k training, 1.3k validation and 1.3k test videos. Since the test set is held out for a challenge, we benchmark on the validation set. We download the videos with 720×1280 resolution and 30 frame-per-second, using the script provided by the authors. During training, we apply random cropping with the area ratio set as (0.3, 1.0) and then resize frames to 224×224 as in [48].

B.2 Kinetics-GEBD

Kinetics-GEBD [56] annotates 20k out of 240k training videos and all 20k validation videos from Kinetics-400. Each video is annotated by 5 sets of event boundaries. The multi-labeling does not affect our self-supervised pre-training stage since no labels are used.

For generating training clips, we adopt the practice from the dataset authors by selecting the annotation entry with the highest F1 consistency score with other entries. The annotation is in the format of timestamps and we choose the closest frame to a ground truth timestamp as an event boundary. We adopt a 32-frame long sliding window with a stride of 3-frame. The $16th$ frame of the sliding window is considered as the center frame, and the window would get a positive label when the time difference between the center frame and ground truth is less than 0.15 second.

During fine-tuning, we sample a clip of 16 frames with a stride of 2 inside each window, and feed it into the video encoder. No temporal augmentation is applied. Instead of global average pooling, we conduct two separate average pooling before and after the center frame. We then concatenate the two features and perform binary classification.

For prediction, we use the same sliding window and stride as in training. If a window is classified as positive, we use the timestamp of the center frame as the detected boundary. We would merge consecutive positive predictions into a single prediction by averaging their predicted timestamps.

Please refer to the original paper [56] and the challenge evaluation code [55] for details on how to deal with multiple ground truths and calculate the final F1 score.

C Visualization of Feature Similarity

We provide a visualization of feature similarity to demonstrate the difference between temporally persistent and fine-grained features. Concretely, for every video in VidSitu validation set, we sample the center clip of each event and feed them into the trained video encoder to get their feature vectors. We then calculate the cosine similarities between all pairs of features, forming a 5×5 similarity matrix. We extract learned features from two video encoders **TeG-PS** and **TeG-FG**. Randomly selected examples are shown in Fig. 6. From these examples, we can see that temporally persistent features would generally produce higher similarity scores compared with temporally fine-grained features. We also draw a further observation that temporally fine-grained features are robust when the temporal content within the video changes very little (see the example in row 5, column 1 of Fig. 6).

Stage	Network	Output size $T \times S^2$
data	-	16×224^2
conv_1	$\mathbf{5}\times 7^2$, 64 stride $\mathbf{2}\times 2^2$	8×112^2
pool_1	1×3^2 max stride 1×2^2	8×56^2
conv_2	$\begin{bmatrix} 1\times 1^2, 64 \\ 1\times 3^2, 64 \\ 1\times 1^2, 256 \end{bmatrix} \times 3$	8×56^2
conv_3	$\begin{bmatrix} 1\times 1^2, 128 \\ 1\times 3^2, 128 \\ 1\times 1^2, 512 \end{bmatrix} \times 4$	8×28^2
conv_4	$\begin{bmatrix} 3\times 1^2, 256 \\ 1\times 3^2, 256 \\ 1\times 1^2, 1024 \end{bmatrix} \times 6$	8×14^2
conv_5	$\begin{bmatrix} 3\times 1^2, 512 \\ 1\times 3^2, 512 \\ 1\times 1^2, 2048 \end{bmatrix} \times 3$	8×7^2
global average pooling		1×1^2

(a) R3D architecture in CVRL [48]

Stage	Network	Output size $T \times S^2$
data	-	8×224^2
conv_1	$\mathbf{1}\times 7^2$, 64 stride $\mathbf{1}\times 2^2$	8×112^2
pool_1	1×3^2 max stride 1×2^2	8×56^2
conv_2	$\begin{bmatrix} 1\times 1^2, 64 \\ 1\times 3^2, 64 \\ 1\times 1^2, 256 \end{bmatrix} \times 3$	8×56^2
conv_3	$\begin{bmatrix} 1\times 1^2, 128 \\ 1\times 3^2, 128 \\ 1\times 1^2, 512 \end{bmatrix} \times 4$	8×28^2
conv_4	$\begin{bmatrix} 3\times 1^2, 256 \\ 1\times 3^2, 256 \\ 1\times 1^2, 1024 \end{bmatrix} \times 6$	8×14^2
conv_5	$\begin{bmatrix} 3\times 1^2, 512 \\ 1\times 3^2, 512 \\ 1\times 1^2, 2048 \end{bmatrix} \times 3$	8×7^2
global average pooling		1×1^2

(b) R3D architecture in [18]

Table 4: **Detailed comparison of 3D-ResNet-50 backbones.** The only difference is in conv_1 (highlighted in **bold**). Network parameters: (a)31.8M, (b)31.8M; network FLOPs: (a)45.6G, (b)41.7G; We follow the setting of (a) since it is already open-sourced. We can reproduce the results with architecture (a) and its corresponding data augmentation as well as training hyper-parameters.

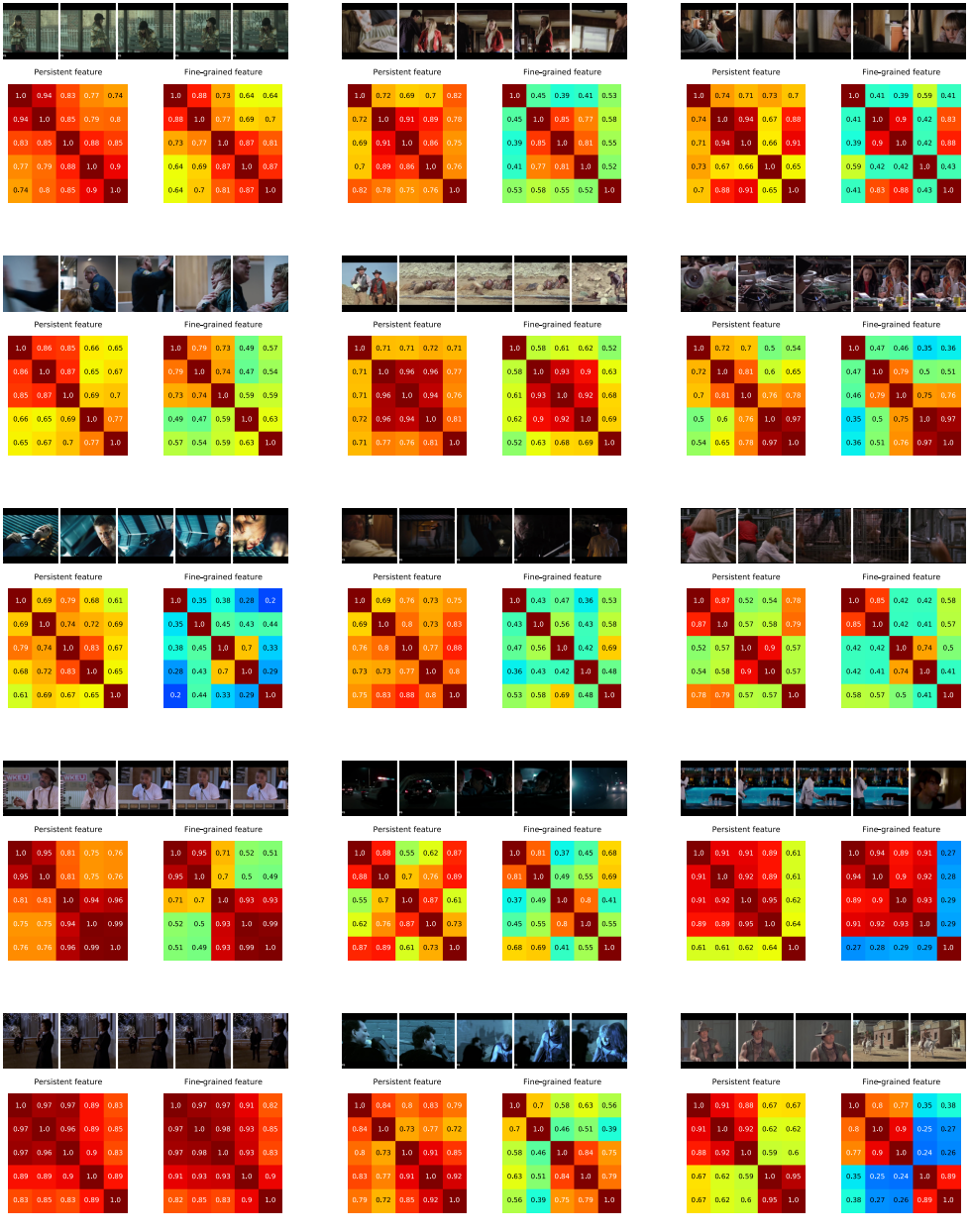

Figure 6: **Random examples** of feature similarity on VidSitu validation videos. In each subfigure, we show the input video (top), the similarity matrices of temporally persistent features (bottom left) and temporally fine-grained features (bottom right).