**Self-supervised Spatiotemporal Representation Learning by Exploiting Video Continuity**

Hanwen Liang\(^1\), Niamul Quader \(^1\), Zhixiang Chi \(^1\), Lizhe Chen \(^1\), Peng Dai\(^1\), Juwei Lu\(^1\), Yang Wang\(^{1,2}\),

\(^1\) Huawei Noah’s Ark Lab \(^2\) University of Manitoba, Canada

{hanwen.liang, niamul.quader1, zhixiang.chi, lizhe.chen, peng.dai, juwei.lu, yang.wang3}@huawei.com

---

**Abstract**

Recent self-supervised video representation learning methods have found significant success by exploring essential properties of videos, e.g., speed, temporal order, etc. This work exploits an essential yet under-explored property of videos, the video continuity, to obtain supervision signals for self-supervised representation learning. Specifically, we formulate three novel continuity-related pretext tasks, i.e., continuity justification, discontinuity localization, and missing section approximation, that jointly supervise a shared backbone for video representation learning. This self-supervision approach, termed as Continuity Perception Network (CPNet), solves the three tasks altogether and encourages the backbone network to learn local and long-ranged motion and context representations. It outperforms prior arts on multiple downstream tasks, such as action recognition, video retrieval, and action localization. Additionally, the video continuity can be complementary to other coarse-grained video properties for representation learning, and integrating the proposed pretext task to prior arts can yield much performance gains.

---

**1 Introduction**

Self-supervised video representation learning has recently received great attention owing to its success in learning informative spatiotemporal features from unlabeled videos. These methods commonly take inspiration from human’s visual understanding system and devise various pretext tasks rooted in certain video attributes, e.g., speed or playback rate (Benaim et al. 2020; Wang, Jiao, and Liu 2020; Chen et al. 2021; Yao et al. 2020), arrow of time (Wei et al. 2018), motion and appearance statistics (Wang et al. 2019) etc. However, these attributes over the input video clips are temporally invariant and coarse-grained. For example, speediness is mostly constant for a given clip instance. This limits the methods’ potential in extensively exploring the fine-grained features of videos (Wang et al. 2021a). To learn both coarse- and fine-grained features within a self-supervision framework, in this work, we exploit an essential yet under-explored property of videos, namely, “video continuity”.

**Video continuity** suggests that objects are represented as the same persisting individuals over time and motion across consecutive frames (Yi et al. 2008). Our choice of using video continuity for designing a self-supervision strategy is motivated by the research findings in cognition sciences (Spelke et al. 1995; Yi et al. 2008). They claim that temporal continuity is essential for a correct and persisting understanding of the visual environment. In fact, based on years of visual experience, human beings can easily detect discontinuity in videos, if any. Furthermore, humans are often capable of inferring the high-level semantics associated with the missing section at the discontinuous point. For example, in Fig. 1, after we manually cut off some portions from the long-jump video, one can easily notice the discontinuity between “takeoff” and “landing” and infer the “action-in-the-air” of the athlete. We hypothesize that enabling the neural networks to master this exercise of detecting discontinuity and estimating the high-level semantics of the missing sections will empower the model to obtain high-quality spatiotemporal representations of videos. This hypothesis is motivated by the following observations. Effective video embedding requires learning both short- and long-ranged features of videos. The features could be temporally-rich motion patterns and spatially-rich context information, both of which are complementary to each other (Huang et al. 2021; Wang, Jiao, and Liu 2020). Solving the continuity-aware tasks requires the model to learn those features comprehensively. Fig. 2(a) gives an illustration of the continuity-aware tasks used in this work. First, identifying **whether the clip is continuous or not**, i.e., continuity justification, requires a global or long-term view of the motion consistency across the clip. Inferring a clip to be discontinuous based on a local perception of motion...
irregularity is insufficient (e.g. a continuous running video could have a local motion irregularity due to the sudden acceleration by the runner). Second, finding where discontinuity occurs, i.e. discontinuity localization, necessitates a local fine-grained grasp of a dramatic motion change along the video stream. Third, estimating what is missing semantically, i.e. missing section approximation, requires model to have a high-level understanding of both the motion patterns and the context information in the neighbouring segments.

Following this thread, we propose a Continuity Perception Network (CPNet) solving the novel continuity-aware pretext tasks in Fig. 2(a), to learn effective spatiotemporal representations in a self-supervised manner. We assume there is none or few shot transition in source videos and the discontinuity in the clips refers to the break-point manually created within the same scene (shot). Specifically, given the continuous and discontinuous clips, CPNet is trained to finish two discriminative tasks of continuity justification and discontinuity localization, which drive the model to perceive the global and local motion patterns of the video sequence. For the task of missing content estimation, instead of explicitly reconstruct reconstruction in RGB space, we formulate it as a contrastive learning task and estimate in the feature space. As shown in Fig. 1, since the discontinuous clip encircles its inner missing section in the source video, their motions are more similar to each other than that of two temporally further disjoint clips, even from the same video. We first use a triplet loss (Schroff, Kalenichenko, and Philbin 2015) to pull the features of the discontinuous clip and its inner missing section closer than a disjoint continuous clip from the same video. Further, based on the observation that clips from the same video have similar appearance compared to those from different videos, we use an additional context-based contrastive loss (Wang, Jiao, and Liu 2020; Chen et al. 2021) as a regularizer to pull features of clips from the same video together. This contrastive learning scheme will promote the features of the discontinuous clip to approximate that of its inner missing section, and encourage the model to learn both fine-grained motion change and context information in the video. The CPNet learns video representations by jointly solving these three continuity-aware pretext tasks.

We carry out extensive experiments and demonstrate the superiority of CPNet in learning more effective video representations. CPNet outperforms prior arts on multiple downstream tasks including action recognition, video retrieval and action localization. Also, the discontinuity localization task is shown to be the most effective pretext task in CPNet, and incorporating it into other typical self-supervised learning methods can bring significant performance gains.

Our major contributions are summarized as follows:

- To the best of our knowledge, this is the first work that explicitly exploits video continuity to obtain supervision signals for self-supervised video representation learning.
- We propose CPNet to solve the novel continuity-aware pretext tasks and promote the model to learn coarse- and fine-grained motion and context features of the videos.
- We conduct comprehensive ablation studies and experiments on multiple downstream tasks to validate the utilities of the proposed CPNet - these include the SOTA performances on action recognition and video retrieval tasks and evidence of complementary nature to other self-supervised video representation learning methods.

## 2 Related Work

In this section, we discuss two groups of recent advances in self-supervised video representation learning: pretext task-based and contrastive learning-based methods. We also discuss some video analysis techniques that are related to our proposed continuity-aware pretext tasks.

### Pretext task-based self-supervision

Pretext tasks usually exploit diverse video properties to obtain supervision signals from unlabeled videos. These tasks operate multiple transformations on source videos for model to recognize and have shown to be effective in self-supervised representation learning (Wang et al. 2021b). Examples include identifying temporal order of shuffled clips or frames (Lee et al. 2017; Xu et al. 2019; Suzuki et al. 2018), predicting video’s playback rate (Benaim et al. 2020; Wang, Jiao, and Liu 2020; Chen et al. 2021) or motion and appearance statistics (Wang et al. 2019), identifying the rotation angle of video clips (Jing et al. 2018) or solving spatiotemporal jigsaw puzzles (Ahsan, Madhok, and Essa 2019; Kim, Cho, and Kweon 2019), etc. In this work, we focus on an essential yet less-touched video property, the video continuity. In some sense, the transformations involved in past works also learn the spatiotemporal continuity implicitly, but they never explicitly use this property for obtaining supervision signals. Besides, the transformations in past works are mostly applied at the whole clip or video-level and provide coarse-grained labels for supervision, whereas the proposed method manipulates at a local temporal point (i.e. where the discontinuity occurs). This distinction encourages our method to exploit frame-wise temporal coherence and capture more fine-grained motion changes. Also, our method uses RGB data only and saves the computation cost for the acquisition of cross-modality data or complex hand-crafted visual priors, e.g. dense trajectories (Wang et al. 2021a).

### Contrastive learning-based self-supervision

Most contrastive learning-based methods build upon the instance discrimination objective and encourage the model to learn some temporal invariance of video instances (Han, Xie, and Zisserman 2019, 2020a; Wang, Jiao, and Liu 2020; Han, Xie, and Zisserman 2020b; Tao, Wang, and Yamasaki; Yang et al. 2020; Yang, Mirmehdi, and Burghardt 2020). They treat the clips from the same video as positives and those from different videos as negatives. For example, RSPNet (Chen et al. 2021) designs an A-VID task to learn appearance features of videos with instance discrimination. Considering that videos have both temporally variant and invariant properties (Dave et al. 2021), CVRL (Qian et al. 2021) relaxes the invariant constraint and develops a weighted temporal sampler to avoid excessive sampling of distant clips. COCLR (Han, Xie, and Zisserman 2020b) uses cross-modal mining to obtain positive samples across video instances. Our work integrates both the distinctiveness across different videos and the temporal variance within the same video. The contrastive
learning strategy that takes the discontinuous outer samples and the inner samples as positive pairs, saves the efforts to carefully design complex clip sampler and reduces the computation cost from multiple modalities.

Related video analysis techniques. Shot boundary detection (SBD) is a task crucial for many video understanding problems. It aims to detect transition and boundaries between consecutive shots (Abdulhussain et al. 2018). SBD is similar to our discontinuity localization task in detecting temporal discontinuity. The difference is that SBD defines discontinuity at shot-level and tries to grasp the semantic transition between scenes. In our proposal, we define discontinuity at frame-level within the same scene and allure the model to obtain more sensitive motion dynamics. If needed, SBD techniques, e.g. PySceneDetect tool, can be easily used for pre-processing in our method to obtain source videos with none or few shot transitions. Video inpainting is another technique related to our pretext tasks, which aims to remove objects or restore missing or tainted regions present in a video sequence (Moran 2009). The similarity between our method and video inpainting lies in explicitly removing and restoring video content based on visual coherence and consistency. Differently, video inpainting creates deficiency at spatial dimension and reconstructs the missing areas in RGB space, while our task creates at temporal dimension and implicitly estimate the missing content in feature space.

3 Method

3.1 Overview

Our continuity-perception self-supervision strategy tries to solve multiple continuity-aware pretext tasks altogether to learn effective spatiotemporal video representations. Let $V = \{v_i\}_{i=1}^N$ be an unannotated video set containing $N$ videos. For a clip $c_i$ sampled from video $v_i$, our method aims to learn an encoder $F_{\theta_i}$ parameterized by $\theta_i$ that maps $c_i$ to continuity-aware rich feature $f_i$. We define feature $f_i$ to be rich in continuity-related information if it can be easily used for answering the following – (1) is $c_i$ continuous or not? (2) if $c_i$ is not continuous, where is the discontinuous point? (3) if $c_i$ is not continuous, can $f_i$ serve as a good estimation for the feature representation of the missing section in $c_i$?

Fig. 2(b) gives an overview of the proposed CPNet, which is a three-branch architecture with all branches sharing the same backbone encoder $F_{\theta_j}$. Given three non-overlapping clips from video $v_i$ – a continuous clip $c_{i,c}$, a discontinuous clip $c_{i,d}$ and its inner missing section $c_{i,m}$, the three branches respectively solve one pretext task and operate jointly to optimize the backbone. The first branch is used to classify continuous or discontinuous. The second branch is used to localize the break-point in $c_{i,d}$. The final branch tries to learn appropriate feature representation of the discontinuous clip $c_{i,d}$ that can be a good approximation to the feature representation of $c_{i,m}$ with a contrastive learning scheme.

3.2 Continuity Perception Task Preparation

Data and supervision acquisition. To perform the proposed continuity-aware pretext task, we assume there is only one break-point in timeline to form the discontinuous clip. We use the $l_n$ to denote the length of the discontinuous clip and $l_m$ to denote that of its missing section. Given a video $v_i$, we first sample an initial clip with length $l_n + l_m$, and then uniformly sample a start-breaking index $j_i \in [1, l_n - 1]$ (index starting from 0). From the $j_i^{th}$ frame, we extract a clip with length $l_m$ from the initial clip to form the missing section $c_{i,m}$. The remaining front and back parts are concatenated to form a discontinuous clip $c_{i,d}$ of length $l_n$. We constrain $j_i$ to be within the range $[1, l_n - 1]$, so that the missing section generation coincides with the creation of a discontinuous clip. We formulate the discontinuity localization as a $(l_n - 1)$-class classification problem with $j_i$ as the label. The continuous clip, $c_{i,c}$, is formed by randomly sampling a non-overlapping clip with length $l_n$ from a same video.
Model structure. We use a typical 3D-ConvNet as the backbone $F_{\theta_j}$ shared by the three pretext tasks. Each of the tasks respectively incorporates a small projection network at head, denoted as $\theta_j$, $L_q$, and $E_q$, to process video embeddings $f_i$ from backbone. All the projection heads have the spatial-temporal average pooling layers, which ensure that the deep feature embeddings of the inputs have the same dimension even when the temporal lengths (i.e. $l_n$ and $l_m$) may be different. The continuity justification and discontinuity localization tasks use fully-connected layers at the end of project heads ($\theta_j$ and $L_q$) for classification. The classification section approximation task uses $E_q$ to embed $f_{i,c}$, $f_{i,d}$ and $f_{i,m}$ to lower-dimensional features $e_{i,c}$, $e_{i,d}$ and $e_{i,m}$.

3.3 Continuity Perception Self-supervision

Continuity justification. For this task, we adopt the same-batch training strategy (Benaim et al. 2020) during training, where each batch contains both continuous and discontinuous clips from each video sample. The motivation is that, as the positives and negatives are from the same videos, the model will rely on perceiving global motion consistency of the input clips rather than context information or any other artificial cues to solve the task. We use the cross-entropy loss ($L_{CE}$) for the optimization. Assume that there are $K$ video samples in one batch, the continuity justification loss $L_J$ is,

$$L_J = \frac{1}{K} \sum_{i=1}^{K} \left( L_{CE}(\theta_j(F_{\theta_j}(c_{i,d}))) + L_{CE}(\theta_j(F_{\theta_j}(c_{i,c}))) \right) \tag{1}$$

Discontinuity localization. In this task, the label for the discontinuity localization in $c_{i,d}$ is more fine-grained compared to the binary label in the continuity justification task. It drives the network to perceive more sensitive and fine-grained motion changes and complements the representation learning of the shared backbone $F_{\theta_j}$. We also use cross-entropy loss for optimization. The above two pretext tasks encourage the backbone to learn both coarse- and fine-grained motion patterns. The discontinuity localization loss $L_L$ is formulated as,

$$L_L = \frac{1}{K} \sum_{i=1}^{K} \left( L_{CE}(\theta_j(F_{\theta_j}(c_{i,d}))) + L_{CE}(\theta_j(F_{\theta_j}(c_{i,c}))) \right) \tag{2}$$

Missing section approximation. To further drive $F_{\theta_j}$ to estimate the content in the missing section, we design the following contrastive learning mechanism to obtain an appropriate feature of a discontinuous clip, which can be a good approximation to the feature of its missing section. Since the missing section $c_{i,m}$ is surrounded by discontinuous clip $c_{i,d}$, they are temporally connected and supposed to contain more similar motions compared to a further clip even from the same video. We first takes $c_{i,d}$ as the anchor, $c_{i,m}$ as the positive and $c_{i,c}$ as the negative in a triplet loss (first term in Eq.3) (Schroff, Kalenichenko, and Philbin 2015), to learn the motion variance within the same video. Further, considering that $c_{i,d}$ and $c_{i,c}$ are from the same video and they have more similar context compared to those from different videos, we propose an additional constraint to regularize the feature representations of $c_{i,d}$ and $c_{i,c}$ not to be too far away. We adopt the vanilla context-based contrastive learning (Wang, Jiao, and Liu 2020) that takes $c_{i,d}$ as the anchor, $c_{i,c}$ as the positive and clips from different videos $\{v_j\}_{j \neq j}$ as the negatives in a contrastive loss (second term in Eq.3) (Chen et al. 2020). Overall, our contrastive continuity learning loss $L_E$ is formulated as:

$$L_E = \frac{1}{K} \sum_{i=1}^{K} \left( \omega \times \max(0, \gamma - (p_i^+ - p_i^-)) - (1 - \omega)\log \frac{q_i^+}{\sum_{j=1 \neq i}^{K} q_i^{-}} \right) \tag{3}$$

where $p_i^+ = sim(e_{i,d}, e_{i,m})$ and $p_i^- = sim(e_{i,d}, e_{i,c})$ are similarities between positive and negative pairs of the triplet loss, $q_i^+ = \exp(sim(e_{i,d}, e_{i,c}) / \tau)$ is the single positive pair and $q_i^{-} = \exp(sim(e_{i,d}, e_{j,c}) / \tau) + \exp(sim(e_{i,d}, e_{j,c}) / \tau)$ are similarities between negative pairs in the contrastive loss, $\tau$ is a temperature value which affects the concentration level of feature distributions. We use cosine similarity for $sim(\cdot, \cdot)$, $\omega \in [0,1]$ is a hyperparameter to balance the relative contribution of the triplet loss and the contrastive loss. Increasing $\omega$ will promote the model to focus more on within-video motion variance to separate the feature representations of $c_{i,d}$ from $c_{i,c}$ (compared to that of $c_{i,m}$), while decreasing $\omega$ will encourage the model to focus more on cross-video context difference to discriminate video instances. The combination of the two objectives further pushes the features of $c_{i,d}$ and $c_{i,m}$ to be close to each other, i.e. the feature of the discontinuous clip becomes a good approximation to the feature of its missing pair. Overall, the model is motivated to learn the context features, including background and object appearance, and fine-grained motion changes in the process.

Multi-task joint optimization. Our continuity perception self-supervision strategy trains the shared backbone $F_{\theta_j}$ to generate feature representations that are reliable for each of the above pretext tasks. We achieve this by jointly optimizing our network with the multi-loss function,

$$L = w_1 \times L_J + w_2 \times L_L + w_3 \times L_E \tag{4}$$

where $w_1$, $w_2$, $w_3 \in [0,1]$ are the individual weights on the losses. The joint optimization promotes the model to learn continuity-aware rich features that embed the global-local motion patterns and the context information of the video.

4 Experiments and Results

4.1 Implementation Details

Datasets and architecture. We use the following benchmark datasets to evaluate the efficacy of CPNet, i.e. UCF101 (Soomro, Zamir, and Shah 2012), HMDB51 (Kuehne et al. 2011), Kinetics400 (abbr. K400) (Kay et al. 2017), Diving48 (Li, Li, and Vasconcelos 2018) and ActivityNet-v1.3 (Caba Heilbron et al. 2015). For UCF101 and HMDB51, We use the training/testing split 1 for fair comparison to prior works. ActivityNet-v1.3 is a benchmark dataset for action temporal localization task.

We use four well-known 3D-ConvNets as the backbone encoder, i.e. C3D (Tran et al. 2015), R3D-18(R3D) (Hara,

**Self-supervised pretraining.** In model pretraining stage, we use the training set of UCF101 or K400 without any annotations. Stochastic gradient descent (SGD) is used for optimization with an initial learning rate of 0.01. For UCF101 (K400), the model is pretrained with a batch size of 32 (64) for 200 (40) epochs, and the learning rate is decayed by 0.1 at the 100th and 150th (20th and 30th) epoch when the loss plateaus. We let $\omega=0.5$ in (3) and $w_1/w_2=1.0, w_3=0.1$ in (4). For all self-supervised pretraining and downstream tasks, we set the length of input video clip $l_m$ as 16 with a resolution of $112 \times 112$. The length of the missing section ($l_m$) during pretraining is determined in ablation study. When pretraining with UCF101(K400), we use 25-fps (15-fps) source frames for both pretrained and evaluated datasets. More discussion about the choice of fps is in Supp. file. Common augmentations are applied on input video clips, including color jittering, random scaling, cropping and horizontal flipping.

**Downstream tasks.** We use several downstream tasks to evaluate the effectiveness of our method. (1) **Action recognition:** We append an adaptive spatial-temporal average pooling layer and a fully-connected linear after the pretrained backbone for action recognition. The model is finetuned end-to-end on the training set of UCF101, HMDB51 or Diving48 for 200 epochs with a batch size of 16. We optimize with SGD with an initial learning rate of 0.01, which is decayed by 0.1 at the 80th and 160th epoch when the loss plateaus. Following the common evaluation protocol (Wang, Jiao, and Liu 2020; Wang et al. 2021b), we perform center cropping and average the scores on 10 uniformly sampled clips on validation splits of these datasets. Top-1 accuracy is used as the evaluation metric. (2) **Video retrieval:** We pretrain the backbone on the training split of UCF101 and add an adaptive spatial-temporal average pooling layer at top to obtain features of video clips. The features of the test set are used to query the k-nearest neighbor videos from the training set. Cosine distance is utilized as the similarity metric. When the class of a query video appears in the classes of k-nearest training videos, it is considered to be a correct prediction. We evaluate our method on UCF101 and HMDB51 and use recall at top-k (R@k) as evaluation metric. (3) **Action temporal localization:** To show the efficacy of our method in learning more fine-grained motion features, we also perform the task of action localization on the ActivityNet-v1.3 dataset. This task aims to generate action proposals that cover groundtruth instances with high recall and temporal overlap. We pretrain model on the training set of ActivityNet-v1.3 with the same settings as we pretrain on UCF101, and extract features of both training and validation sets with pretrained backbone. We adopt the popular BMN (Lin et al. 2019) framework for action localization. BMN is trained on the extracted features of the training set and then evaluated on the validation set. We adopt Average Recall (AR) and calculate AR under different Average Number of proposals (AN) as AR@AN as evaluation metric. We also calculate the Area under the AR vs. AN curve (AUC) with AN varying from 0 to 100.

<table>
<thead>
<tr>
<th>Method</th>
<th>Length of $l_m$</th>
<th>Action recognition</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Random Init</td>
<td>8</td>
<td>70.1 37.5 62.7</td>
<td></td>
</tr>
<tr>
<td>CJ</td>
<td>4</td>
<td>70.9 38.1 65.4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>79.2 40.1 63.5</td>
<td></td>
</tr>
<tr>
<td>DL</td>
<td>4</td>
<td>73.1 38.1 66.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>76.1 39.1 67.0</td>
<td></td>
</tr>
<tr>
<td>MSA</td>
<td>4</td>
<td>72.1 40.1 68.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>63.1 39.1 66.0</td>
<td></td>
</tr>
<tr>
<td>CPNet</td>
<td>4</td>
<td>80.1 40.1 72.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>79.3 40.1 71.6</td>
<td></td>
</tr>
</tbody>
</table>

**4.2 Ablation Study**

To determine the optimal length of the missing section ($l_m$) and analyze the positive effects brought by each pretext task, we conduct ablation studies with R(2+1)D and evaluate with action recognition task on UCF101, HMDB51 and Diving48. In this section, we use 90% of the training split of UCF101 for pretraining. During evaluation, for each dataset, 90% of the training set is used for finetuning (the same 90% pretrained for UCF101) and the rest 10% is used for testing.

**Sensitivity to the length of missing section.** The length of the missing section ($l_m$) is an important setting in CPNet that affects the model’s sensitivity to video continuity. A small value of $l_m$ may make the discontinuity too difficult to detect and the pretext task too ambiguous to solve, whereas a large value may make the task too easy and reduce the necessity for model to learn high-quality discriminative representations. We test the sensitivity of each pretext task to $l_m$ by pretraining with only one of the tasks activated. As shown in Table 1, $l_m=8$ gives better results compared to the lower $l_m=4$ and the higher $l_m=16$ for continuity justification and discontinuity localization, though $l_m=4$ gives comparable results for missing section approximation. Similar observations are found in (Wang, Jiao, and Liu 2020) that the pretext tasks cannot be too simple or too ambiguous to get good performance. For CPNet with all three tasks, $l_m=8$ gives the best result, so we use $l_m=8$ for experiments in the rest of the paper. We also show the pretraining performance of the model on the pretext tasks in Sec.C in the Supp. file.

**Effectiveness of pretext tasks.** From the 2nd to 4th sections in Table 1, we can see that each pretext task can
brings improvement over the random initialization. Notably, the discontinuity localization task is the most effective one among others. Since the discontinuous location is sampled uniformly along the timeline of the input, this task promotes the model to densely capture the motion changes and learn fine-grained representations corresponding to each timestamps of the input. Also, Fig. 2 shows that this task is more efficient as it requires processing only one clip from each video sample. The light-weight property and effective behavior of this task shows its superiority over many competing pretext task-based and contrastive learning-based approaches. While other proposed tasks are not as potent as the discontinuity localization, they are complementary to each other and provide the best when all of them (i.e. CPNet) are used together (the 5th, 6th sections in Table 1). We conjecture that, this joint improvement comes from the complementary local and long-ranged motion and contextual feature learning pathways of the individual tasks. We also note that the missing section approximation is similar to a generic context-based contrastive learning without the triplet loss. Removing triplet loss from CPNet degrades performances (CPNet- in Table 1), suggesting that our missing section approximation strategy in CPNet works better than a generic context-based contrastive learning.

### 4.3 Evaluation of Self-supervised Representation

In this section, we evaluate the representation capability of our self-supervision strategy in multiple downstream tasks.

#### Action recognition.
For action recognition, we pretrain on the training set of UCF101 or K400, and finetune on UCF101, HMDB51 and Diving48 datasets. Table 2 shows that CPNet achieves consistently superior results over all the previous self-supervised methods on both benchmark datasets. With the same backbone, our models pretrained on UCF101 even outperform RSP (Chen et al. 2021) and Video-MoCo (Pan et al. 2021) pretrained on a larger K400 dataset. CPNet also benefits from pretraining on a larger dataset and gets better results. It outperforms RSP (Chen et al. 2021) by 2.7% and 12.5% on UCF101 and HMDB51 with R(2+1)D. Also, Table 3 suggests our method is effective for fine-grained action recognition. CPNet pretrained on Diving48 with I3D outperforms MoCo+BE (Wang et al. 2021b) by 8.8% under the same setting, and by 4.7% even pretrained with K400. Remarkably, CPNet pretrained on UCF101 with C3D outperforms the supervised model pretrained on a larger dataset Sports1M. Based on ablation study in Table 1, we give the credit to the discontinuity localization and missing section approximation tasks, which enforce the model to pay more attention to the fine-grained motion features.

#### Video retrieval.
Table 4 presents the results on video retrieval task with model pretrained on the training split of UCF101. CPNet achieves the highest recall for all K values in UCF101 and HMDB51 with C3D and R(2+1)D. Noticeably, the proposed method outperforms the second-best by 10.0%, 9.0% and 15.0% on UCF101 when K=1 with C3D, R3D and R(2+1)D.

#### Action temporal localization.
Action localization aims to generate temporal boundaries for the action instance in untrimmed videos, which requires a thorough and fine-grained video understanding. We pretrain with C3D using multiple typical self-supervised video representation learnings.
movements over time, e.g., the crawling path of the baby, the rising curve of the barbell and swinging over the pommel horse. The second task concentrates more on the dominant moving object and its fine-grained motion changes. In comparison, the third task of missing section approximation has larger salient regions over important context cues in the video. We find that removing the triplet loss removes salient regions from the fine-grained motion change areas, suggesting that the triplet loss encourages learning of such features.

4.5 Complementary with Existing Methods.

As we mentioned in Sec. 1, the most distinguishing point between video continuity and the video attributes used in prior works is that video continuity is a more fine-grained and temporally variant attribute. To show if this attribute can be complementary to other coarse-grained global attributes, i.e. playback rate, temporal order of clips, we further exploit the utility of the discontinuity localization task by integrating it to prior self-supervised learning approaches, i.e. PacePred (Wang, Jiao, and Liu 2020) and MOCO (He et al. 2020) (All self-implemented. Details in Sec.F of Supp. file). The reason for choosing this pretext task is that it is easy-to-implement and the most effective component of CPNet, and adding it does not bring much computation or memory cost to the base method. Table 6 shows that the integration consistently brings considerable improvement for action recognition tasks with multiple backbones. It suggests that our novel discontinuity localization task can also be a powerful tool used with emerging SOTA self-supervision approaches and that the video continuity attribute can be complementary to other video attributes for spatiotemporal representation learning.

5 Conclusion

This work introduces a novel self-supervised learning framework, CPNet, that explores video continuity property to formulate pretext tasks. These continuity-aware pretext tasks, particularly the discontinuity localization task, are easy-to-implement and effective individually in video representation learning. They can also complement prior self-supervision arts for performance gains. Using these tasks together within CPNet learns fine- and coarse-grained motion and context features, and leads to competitive or SOTA performances on multiple video understanding tasks.

Table 5: Comparison between features from CPNet and other self-supervised approaches for temporal action localization task. Evaluation is performed on validation set of ActivityNet-v1.3 in terms of AR@AN and AUC.

<table>
<thead>
<tr>
<th>Method</th>
<th>AR@1</th>
<th>AR@5</th>
<th>AR@10</th>
<th>AR@100</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random Init</td>
<td>0.2837</td>
<td>0.3612</td>
<td>0.4205</td>
<td>0.6636</td>
<td>55.917</td>
</tr>
<tr>
<td>RSP (Chen et al. 2021)</td>
<td>0.3144</td>
<td>0.4211</td>
<td>0.4879</td>
<td>0.7104</td>
<td>60.989</td>
</tr>
<tr>
<td>PRP (Yao et al. 2020)</td>
<td>0.3077</td>
<td>0.4041</td>
<td>0.4674</td>
<td>0.6997</td>
<td>60.248</td>
</tr>
<tr>
<td>VCOP (Xu et al. 2019)</td>
<td>0.3024</td>
<td>0.3969</td>
<td>0.4586</td>
<td>0.6907</td>
<td>59.451</td>
</tr>
<tr>
<td>TempTrans (Jenni, Mesaritk, and Favaro 2020)</td>
<td>0.3082</td>
<td>0.3999</td>
<td>0.4612</td>
<td>0.6959</td>
<td>59.756</td>
</tr>
<tr>
<td>CPNet</td>
<td>0.3205</td>
<td>0.4273</td>
<td>0.4954</td>
<td>0.7133</td>
<td>62.264</td>
</tr>
</tbody>
</table>

Table 6: Action recognition performances by incorporating video continuity into prior methods. “DL” denotes “discontinuity localization” task. Models are pretrained on UCF101.

<table>
<thead>
<tr>
<th>Backbone</th>
<th>Method</th>
<th>Baseline AR@1</th>
<th>Baseline AR@5</th>
<th>Baseline AR@10</th>
<th>Baseline AR@100</th>
<th>Baseline AUC</th>
<th>UCF101 AR@1</th>
<th>UCF101 AR@5</th>
<th>UCF101 AR@10</th>
<th>UCF101 AR@100</th>
<th>UCF101 AUC</th>
<th>HMDB51 AR@1</th>
<th>HMDB51 AR@5</th>
<th>HMDB51 AR@10</th>
<th>HMDB51 AR@100</th>
<th>HMDB51 AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>R3D</td>
<td>PacePred</td>
<td>76.0</td>
<td>42.8</td>
<td>76.6(+0.6)</td>
<td>44.5(+1.7)</td>
<td></td>
<td>R(2+1)D</td>
<td>76.4</td>
<td>45.0</td>
<td>77.9(+1.3)</td>
<td>48.2(+3.2)</td>
<td></td>
<td>R(2+1)D</td>
<td>78.8</td>
<td>48.2</td>
<td>80.1(+1.3)</td>
</tr>
<tr>
<td>R(2+1)D</td>
<td>VCOP</td>
<td>72.6</td>
<td>41.3</td>
<td>75.4(+2.8)</td>
<td>44.6(+3.3)</td>
<td></td>
<td>R(2+1)D</td>
<td>74.8</td>
<td>45.8</td>
<td>77.0(+2.2)</td>
<td>47.9(+2.1)</td>
<td></td>
<td>R(2+1)D</td>
<td>78.8</td>
<td>48.2</td>
<td>80.1(+1.3)</td>
</tr>
<tr>
<td>C3D</td>
<td>MOCO</td>
<td>72.3</td>
<td>39.8</td>
<td>72.9(+0.6)</td>
<td>41.2(+1.4)</td>
<td></td>
<td>R(2+1)D</td>
<td>71.8</td>
<td>41.8</td>
<td>74.4(+2.6)</td>
<td>46.2(+4.4)</td>
<td></td>
<td>R(2+1)D</td>
<td>75.6</td>
<td>46.0</td>
<td>78.5(+2.9)</td>
</tr>
<tr>
<td>C3D</td>
<td>PacePred</td>
<td>76.4</td>
<td>45.0</td>
<td>77.9(+1.3)</td>
<td>48.2(+3.2)</td>
<td></td>
<td>R(2+1)D</td>
<td>74.8</td>
<td>45.8</td>
<td>77.0(+2.2)</td>
<td>47.9(+2.1)</td>
<td></td>
<td>R(2+1)D</td>
<td>78.8</td>
<td>48.2</td>
<td>80.1(+1.3)</td>
</tr>
<tr>
<td>R3D</td>
<td>VCOP</td>
<td>72.6</td>
<td>41.3</td>
<td>75.4(+2.8)</td>
<td>44.6(+3.3)</td>
<td></td>
<td>R(2+1)D</td>
<td>74.8</td>
<td>45.8</td>
<td>77.0(+2.2)</td>
<td>47.9(+2.1)</td>
<td></td>
<td>R(2+1)D</td>
<td>78.8</td>
<td>48.2</td>
<td>80.1(+1.3)</td>
</tr>
<tr>
<td>C3D</td>
<td>MOCO</td>
<td>72.3</td>
<td>39.8</td>
<td>72.9(+0.6)</td>
<td>41.2(+1.4)</td>
<td></td>
<td>R(2+1)D</td>
<td>71.8</td>
<td>41.8</td>
<td>74.4(+2.6)</td>
<td>46.2(+4.4)</td>
<td></td>
<td>R(2+1)D</td>
<td>75.6</td>
<td>46.0</td>
<td>78.5(+2.9)</td>
</tr>
</tbody>
</table>

Figure 3: Illustration of salient regions for each continuity-aware pretext task.
References


