Hierarchically Decoupled Spatial-Temporal Contrast for Self-supervised Video Representation Learning

Zehua Zhang    David Crandall
Indiana University Bloomington
{zehzhang, djcran}@indiana.edu

Abstract

We present a novel technique for self-supervised video representation learning by: (a) decoupling the learning objective into two contrastive subtasks respectively emphasizing spatial and temporal features, and (b) performing it hierarchically to encourage multi-scale understanding. Motivated by their effectiveness in supervised learning, we first introduce spatial-temporal feature learning decoupling and hierarchical learning to the context of unsupervised video learning. We show by experiments that augmentations can be manipulated as regularization to guide the network to learn desired semantics in contrastive learning, and we propose a way for the model to separately capture spatial and temporal features at multiple scales. We also introduce an approach to overcome the problem of divergent levels of instance invariance at different hierarchies by modeling the invariance as loss weights for objective re-weighting. Experiments on downstream action recognition benchmarks on UCF101 and HMDB51 show that our proposed Hierarchically Decoupled Spatial-Temporal Contrast (HDC) makes substantial improvements over directly learning spatial-temporal features as a whole and achieves competitive performance when compared with other state-of-the-art unsupervised methods. Code will be made available.

1. Introduction

As a solution to the growing need for large-scale labeled data for training complex neural network models [6, 20, 35, 56, 58], unsupervised representation learning aims to learn good feature embeddings from data without annotations. Using the learned representations as initialization, downstream tasks only need to be fine-tuned on a relatively small labeled dataset in order to yield reasonable performance. Much recent progress in unsupervised image representation learning [3, 19, 21, 25, 44, 46, 57, 73, 78] is driven by contrastive learning. While they solve the same pretext task of instance-level variant matching, these methods differ in how they obtain variant embeddings of the same instance, e.g., using augmentations [3, 25, 73, 78], future representations [46], or momentum features [19]. By optimizing a contrastive loss [14], they are essentially maximizing intra-instance embedding similarity and minimizing inter-instance embedding similarity, which leads to a spread-out approach for decoupling spatial-temporal feature learning.
Compared with the success of contrastive learning for images, most state-of-the-art unsupervised video representation learning work relies on context-based proxy tasks such as time arrow classification [72] or clip order prediction [74]. While the results of some recent methods [15, 16, 33, 55] suggest the potential of contrastive learning for videos, they work in a “one for all” manner by solving a single contrastive learning task with the activation maps from the final layer, expecting the model to capture all the features through the learning procedure. As a result, the learned model may lack a general understanding of spatial and temporal semantics, and instead just memorize spatial-temporal combinations.

In this paper, we argue that good video representations should be able to capture spatial and temporal features in a more general form at multiple scales, and thus it is helpful to decompose the overall goal of learning spatial-temporal features into hierarchical subtasks, respectively emphasizing spatial and temporal features. To this end, we present Hierarchically Decoupled Spatial-Temporal Contrast (HDC), in which we decouple the learning objective into separate subtasks of Spatial Contrast and Temporal Contrast and perform the learning hierarchically.

Neural networks are notorious for learning shortcuts to “cheat” [11, 72]: if there is an easy way to solve the problem, the network will hardly try to find a more complex solution. We make use of this property for the decoupling and direct the network to learn different features by manipulating augmentations. In particular, in the Spatial Contrast subtask, we deliberately provide a shortcut by creating augmented variants with only spatial augmentations (random spatial cropping, color jittering, etc.). As the timestamps of the query clip and its augmented copy are the same, it is possible to solve the matching task based merely on consistency of spatial semantics, and thus the network will try to “cheat” by focusing more on spatial features. In Temporal Contrast learning, we randomly select a new clip from the video of the query clip (random temporal cropping) before applying spatial augmentations in order to obtain a variant whose spatial semantics are as different from the query clip as possible. Since spatial context may vary dramatically after applying the temporal and spatial transformations, the model is prevented from “cheating” through spatial similarity and encouraged to rely more on similarity of temporal semantics to solve the pretext task. Fig. 1 shows nearest neighbor retrieval results demonstrating the effectiveness of our approach for the decoupling, and suggests the potential of directing the network to learn desired features through manipulating augmentations as regularization.

In order to capture multi-scale features, we further perform Spatial Contrast and Temporal Contrast learning hierarchically by optimizing towards a compound loss. During hierarchical learning, we model the significance of instance-wise consistency in a given layer with different weights, because features from different layers do not share the same level of invariance against augmentations [79].

In summary, our contributions are as follows:

- We demonstrate the effectiveness of spatial-temporal feature learning decoupling and hierarchical learning in the context of unsupervised learning for the first time.
- We show that the network can be guided to learn desired features in contrastive learning by manipulating augmentations as regularization, and introduce a new way for the network to separately capture spatial and temporal semantics in self-supervised video representation learning.
- We propose an approach to improve hierarchical contrastive learning by modeling the divergent levels of invariance in different layers as different loss weights.
- By optimizing a novel compound loss, our Hierarchically Decoupled Spatial-Temporal Contrast (HDC) outperforms other unsupervised methods and sets a new state-of-the-art on downstream tasks of action recognition on UCF101 and HMDB51.

2. Related Work

Unsupervised Video Representation Learning was originally based on input reconstruction [23, 24, 37, 38, 50, 62], while more recent methods derive implicit pseudo-labels from the unlabeled data to use as self-supervision signals for the corresponding pretext task [49, 70]. For example, several models use chronological order of video frames to define proxy tasks such as frame order prediction [39] or verification [13, 45], clip order prediction [74], and time arrow classification [72]. Other pretext tasks, such as spatial-temporal jigsaw [32], future prediction [15, 40, 42, 54, 63, 64], temporal correspondence estimation [28, 29, 70, 71], audio-video clustering [2], video colorization [65], motion and appearance statistics prediction [67], and loss distillation across multiple tasks [47] have also been explored.

Contrastive Learning [14] is very effective for unsupervised representation learning for images [3, 19, 21, 25, 44, 46, 57, 73, 78]. These methods try to learn a feature space in which variants of the same sample are close together while variants from different samples are far apart, and mainly differ in how they create the variants. For example, Oord et al. [46] predict the future in the latent space as a variant of the real embeddings of the future. In the video domain, Han et al. [15] predict dense feature maps of future clips and match them with corresponding real embeddings from other distractions. This idea is further extended to a memory-augmented version [16] for improvement. Sun et al. [55] proposed to use bidirectional transformers for multimodal contrastive learning from text and videos. A cycle-
contrastive loss inspired by CycleGAN [82] is presented by Kong et al. [33] to use the relationship between videos and frames. Recasens et al. [51] and Wang et al. [66] both explore a new pretext task of matching short-term view to long-term view of the same video. Tschannen et al. [60] also use video-induced invariance to formulate a pretext task based on contrastive learning. Despite its motivation, interestingly, it aims to learn image representations instead of video embeddings. There is more discussion in Sec. 3.4.

3. Our Method

To explore the potential of self-supervised video representation learning merely from RGB clips, we formulate a novel pretext task of Hierarchically Decoupled Spatial-Temporal Contrast (HDC) in which we maximize intra-instance representation similarity and minimize inter-instance representation similarity spatially (Fig. 2), temporally (Fig. 2), and hierarchically (Fig. 3).

3.1. Decoupled Contrast

Motivated by the observation in supervised learning that factoring 3D filters into separate spatial and temporal components yields significant gains [59], we propose to decouple the overall objective of unsupervised spatial-temporal learning into separate subtasks and provide regularizations to guide them to emphasize spatial and temporal features, respectively (Fig. 2).

Spatial Contrast is designed to focus on learning spatial representations. Neural networks are notorious for learning shortcuts to “cheat” [11, 72]. While previous work tries to avoid all cheating [11, 32, 39, 72], we intentionally make use of this property. In Spatial Contrast, the network is provided with a shortcut by augmenting clips with only spatial transformations. This allows the model to “cheat” by using only spatial features to capture intra-instance similarity and inter-instance difference. In other words, the network will not bother with temporal features because capturing spatial similarity is already enough for solving the matching task.

In particular, as shown in Fig. 2, given a batch of clips $X = [x_1, x_2, ..., x_B]$, each from a different video, we augment each clip $i$ with augmentations $\varphi_{o,i}$ and $\varphi_{s,i}$ sampled from families of candidate spatial transformations $\Phi_0$ and $\Phi_s$ to obtain corresponding variants $u_{o,i} = \varphi_{o,i}(x_i)$, $\varphi_{o,1}(\cdot) \sim \Phi_0$ and $u_{s,i} = \varphi_{s,i}(x_i)$, $\varphi_{s,1}(\cdot) \sim \Phi_s$, which are only spatially-augmented.

Our primary goal is to train an encoder so that the similarity between feature embeddings of $u_{o,i}$ and $u_{s,j}$ is maximized when $i = j$, and minimized otherwise. Let $f(\cdot)$ denote the encoder. The embeddings of $u_{o,i}$ and $v_{s,j}$ are $v_{o,i} = f(u_{o,i})$ and $v_{s,j} = f(u_{s,j})$, where $v_{o,i}$ and $v_{s,j}$ are both vector embeddings, e.g., obtained through global pooling. By measuring similarity with cosine distance $\text{sim}(v_{o,i}, v_{s,j}) = (v_{o,i} \cdot v_{s,j})/(\|v_{o,i}\| \cdot \|v_{s,j}\|)$, our goal can be achieved by optimizing a contrastive loss called InfoNCE [46],

$$L_s = -\sum_{i=1}^{B} \log \frac{\exp(\text{sim}(v_{o,i}, v_{s,i})/\tau)}{\sum_{j=1}^{B} \exp(\text{sim}(v_{o,i}, v_{s,j})/\tau)},$$

(1)

where $\tau$ is a temperature controlling the concentration of the feature embedding distribution [22] (usually around 0.1). Eq. 1 can be viewed as a cross entropy loss between a pseudo prediction $S \in R^{B \times B}$ and a pseudo label $S \in R^{B \times B}$. $S$ is an identity matrix and $S$ is a similarity matrix where $S_{ij}$ measures the similarity of one clip variant $u_{o,i}$ to another variant $u_{s,j}$ in the feature space,

$$S_{ij} = \frac{\exp(\text{sim}(v_{o,i}, v_{s,j})/\tau)}{\sum_{m=1}^{B} \exp(\text{sim}(v_{o,i}, v_{s,m})/\tau)}.$$  

(2)

In this spirit, the Spatial Contrast subtask is a self-supervised multi-way classification problem in which we want to match one clip variant to another when they come from the same video and distinguish it from variants coming from different videos.

Temporal Contrast emphasizes learning temporal representations. To do this, we need to prevent the network from cheating through spatial feature similarity: we want variants whose spatial context varies dramatically from the original clip, but whose temporal context remains nearly the same and thus is essential for capturing instance-level invariance. To do this, we add temporal augmentations of random temporal cropping before applying spatial transformations to produce another temporally-augmented variant. Note that spatial augmentations are crucial here to further alter spatial context in order to prevent cheating, and we show its importance in Section 4.1.

Specifically, to produce a temporally-augmented variant $u_{t,j}$ for each clip $x_j$, random temporal cropping $\Gamma(\cdot)$ is first applied, followed by spatial augmentations $\varphi_{t,j}$ sampled from another family of candidate spatial transformations $\Phi_t$, after which we have $u_{t,i} = \varphi_{t,j}(\Gamma(x_j))$, $\varphi_{t,1}(\cdot) \sim \Phi_t$. Temporal Contrast is modeled between $u_{o,i}$ and $u_{t,j}$ using a similar technique as in Spatial Contrast. Then we minimize,

$$L_t = -\sum_{i=1}^{B} \log \frac{\exp(\text{sim}(v_{o,i}, v_{t,i})/\tau)}{\sum_{j=1}^{B} \exp(\text{sim}(v_{o,i}, v_{t,j})/\tau)},$$

(3)

As with Spatial Contrast learning, the Temporal Contrast subtask is also a self-supervised multi-way classification problem of matching clip variants of the same video.

3.2. Hierarchical Contrast

Inspired by the effectiveness of multi-scale features in supervised learning [26, 81], we introduce hierarchical learning to unsupervised learning by conducting Decoupled Contrast learning hierarchically. As illustrated in Fig. 3,
Figure 2: A simple example of how our Decoupled Contrast works between two sample videos. In Spatial Contrast (SC), two sets of spatial transformations, \( \phi_{s,1} \) and \( \phi_{s,2} \), are sampled from families of candidate spatial transformations \( \Phi_s \), and applied to the same clip \( x_i \) to obtain variants \( u_{o,i} \) and \( u_{s,i} \) for SC learning. In Temporal Contrast (TC), we first extract another clip \( x_i' \) from the same video as \( x_i \) by random temporal cropping \( \Gamma(\cdot) \), and then apply \( \phi_{t,i} \) sampled from families of candidate spatial transformations \( \Phi_t \) to obtain a variant \( u_{t,i} \), which will be used with \( u_{o,i} \) for TC learning.

Feature maps from different layers or blocks of the encoder \( f(\cdot) \) are collected and pooled to produce multi-scale vector embeddings. Two pooling strategies are applied to fully leverage instance-level consistency among different layers: (1) Temporal Contrast uses 3D global average pooling along both temporal and spatial dimensions, and (2) Spatial Contrast performs 2D global average pooling only along the spatial dimension. Therefore, for each scale, we obtain one vector representation for each clip variant in Hierarchical Temporal Contrast learning, but may get more than one vector for each variant in Hierarchical Spatial Contrast learning, depending on the length of input clip and the temporal downsample factor of the encoder for a certain layer. Our motivation is that because the timestamps in Spatial Contrast are the same for two clip variants from the same video, the spatial instance-level invariance should exist not only for the whole clip, but also for corresponding sub-clips.

As shown in Fig. 3, for Hierarchical Spatial Contrast learning at scale \( k \), we apply 2D global average pooling and obtain vector embeddings of \( v_{o,i}^{k,n} \), \( n = 1, 2, \ldots, N \) for clip variant \( u_{o,i} \) and vector embeddings of \( v_{s,j}^{k,n} \), \( n = 1, 2, \ldots, N \) for variant \( u_{s,j} \), where \( N \) is the size of the temporal dimension of the feature maps. The loss at scale \( k \) becomes,

\[
L^k = -\frac{1}{N} \sum_{n=1}^{N} \sum_{i=1}^{B} \log \frac{\exp\left(\text{sim}(v_{o,i}^{k,n}, v_{s,j}^{k,n})/\tau\right)}{\sum_{j=1}^{B} \exp(\text{sim}(v_{o,i}^{k,n}, v_{s,j}^{k,n})/\tau)},
\]

for Spatial Contrast, and

\[
L^k = -\frac{1}{N} \sum_{n=1}^{N} \sum_{i=1}^{B} \log \frac{\exp(\text{sim}(v_{o,i}^{k,n}, v_{t,i}^{k,n})/\tau)}{\sum_{j=1}^{B} \exp(\text{sim}(v_{o,i}^{k,n}, v_{t,j}^{k,n})/\tau)},
\]

while the loss for Hierarchical Temporal Contrast remains almost the same for scale \( k \),

3.3. Self-supervised Learning with HDC

We consider 3 different backbones, C3D [58], 3D-ResNet18 [18, 20] and R(2+1)D-10 [59], as our encoder \( f(\cdot) \) for fair comparison with existing methods. All fully-connected layers are removed (last 3 layers of C3D, and last layer of 3D-ResNet18 and R(2+1)D-10). Batch normalization layers [27] in 3D-ResNet18 and R(2+1)D-10 are replaced with instance normalization layers [61] to prevent...
Random Augmentation Random Augmentation Random Augmentation

Hierarchical Temporal Contrast

Hierarchical Spatial Contrast

Feature maps \(N' \times M' \times M' \times C'\)

Feature vectors \(N' \times C'\)

Feature vectors \(N'' \times C''\)

Feature vectors \(N'' \times C''\)

Feature maps \(N \times M \times M \times C\)

Feature maps \(N \times C\)

Feature maps \(N'' \times M'' \times M'' \times C''\)

Maximize Similarity

Maximize Similarity

Maximize Similarity

Maximize Similarity

S-Avg Pooling

G-Avg Pooling

S-Avg Pooling

S-Avg Pooling

G-Avg Pooling

G-Avg Pooling

Random Augmentation

Random Augmentation

Random Augmentation

Video

Clip \(x\)

Clip \(x'\)

Figure 3: An example illustrating Hierarchical Contrast. Here we only show learning among variants from the same video and thus the network is maximizing similarity between all corresponding pairs. Otherwise if variants are from different videos, the network will minimize similarity. Feature maps at multiple scales are 2D global average pooled along the spatial dimension (S-Avg Pooling) in Hierarchical Spatial Contrast learning, and 3D global average pooled along the temporal and spatial dimensions (G-Avg Pooling) in Hierarchical Temporal Contrast learning. The channel dimension is omitted for clarity.

Predictions) The families of spatial augmentations \(\Phi_o, \Phi_s,\) and \(\Phi_t\) contain the same set of transformations of random spatial cropping, scale jittering, horizontal flipping, color jittering, and channel replication [39]. We use \(\tau = 0.07\) for computing the contrastive loss.

We perform self-supervised training on Kinetics-400 [31], which has 400 action classes and over 400 videos per class. We resize frames, preserving aspect ratio, so that the shorter side is 128 pixels. Each mini-batch contains 128 clips from 128 videos and each clip consists of 16 randomly-cropped continuous frames of shape \(128 \times 128 \times 3\). After spatial and temporal augmentations, three sets of clip variants are obtained, each of shape \(128 \times 16 \times 112 \times 112 \times 3\). Before being fed to the encoder, these clips are rescaled following [6] so pixel values are between -1 and 1.

Our model is implemented with Tensorflow [1] and Keras [9]. We use stochastic gradient descent with learning rate 0.1, momentum 0.9, decay 0.0001, and \(L2\) regularizer \(5e^{-5}\). HDC is trained as a whole towards minimizing a novel compound contrastive loss, namely HD-NCE,

\[
L = \sum_k (\alpha_k \cdot L^k_s + \beta_k \cdot L^k_t),
\]

where \(L^k_s\) and \(L^k_t\) are in Eqs. 4 and 5, and \(k = 3, 4, 5\) as we use features from blocks 3, 4, 5 of the encoder.

Significance of instance-wise consistency. \(\alpha_k\) and \(\beta_k\) represent the significance of instance-wise consistency in each layer, essentially weighting how much the corresponding subtask will contribute to our main goal. We simply tested

1. \(\alpha_3 = \beta_3 = \alpha_4 = \beta_4 = \alpha_5 = \beta_5 = 1.0\), and
2. \(\alpha_3 = \beta_3 = \alpha_4 = \beta_4 = \alpha_5 = \beta_5 = 0.0001\).
\[ \alpha_3 = \beta_3 = 0.25, \alpha_4 = \beta_4 = 0.5 \text{ and } \alpha_5 = \beta_5 = 1.0, \] and the second set worked better (see Section 4.1).

### 3.4. Difference from Recent Work

Some work [75,76] also explores hierarchical contrastive learning in videos. However, while HDC matches variants obtained with carefully designed augmentations, [75] brings closer features from the slow and fast streams of a Slowfast network [12], and [76] predicts future motion patterns. Moreover, our work introduces spatial-temporal learning decoupling, which is another novel component.

Another similar method is CVRL [48], which is similar in spirit to the variant of our model trained with only Temporal Contrast incorporated in the last layer. Our HDC decouples the learning into separate subtasks emphasizing spatial and temporal features and further performs the learning hierarchically. As later experiments (Sec. 4.1 and Tab. 1) show, HDC achieves significant improvement over the CVRL-alike variant (the second row in Tab. 1).

### 4. Experiments

We follow a common protocol [45] to evaluate the effectiveness of our HDC by using the learned representations as initialization and fine-tuning on the downstream task of action recognition on UCF101 [52] and HMDB51 [36]. UCF101 [52] consists of 13,320 videos and 101 classes of human action. It has three train/test splits with a split ratio of about 7:3. HMDB51 [36] is another widely-used action recognition dataset containing 6,766 videos and 51 classes. It also has three splits with a similar split ratio as UCF101.

In our ablation studies, if not explicitly mentioned, we report Top-1 accuracy on only UCF101 split 1. When comparing our method with other state-of-the-art, results averaged on three splits of UCF101 and HMDB51 are reported.

In fine-tuning, we use the same network (C3D, 3D-ResNet18, or R(2+1)D-10) as we did in self-supervised learning. A dropout layer [53] of ratio 0.5 is added after global average pooling, followed by a single fully-connected layer and softmax activation for classification.

### 4.1. Ablation Studies

We conduct ablation experiments to analyze our design choices.

#### Decoupled Contrast.

We first evaluate the effectiveness of decomposing the video representation learning task into subtasks of Spatial and Temporal Contrast and performing joint learning with them in Tab. 1. The first three rows present the results with only Spatial Contrast learning, with only Temporal Contrast learning, and with both Spatial Contrast and Temporal Contrast learning. Hierarchical Contrast learning is not involved and the learning is based on the features of the last (5th) block. We see that, first, Temporal Contrast achieves slightly better performance than Spatial Contrast, perhaps because temporal semantics are more important in video learning. Then, by incorporating both Spatial and Temporal Contrast, performance improves significantly. This indicates that Spatial Contrast and Temporal Contrast learn complementary features, and validates our hypothesis that applying different augmentations as a way of regularization guides the network to learn different features. Furthermore, the nearest neighbor retrieval results in Fig. 1 and Fig. 4 show that Spatial Contrast and Temporal Contrast focus on spatial and temporal features respectively.

We note that Spatial Contrast in and of itself can be viewed as directly migrating the basic contrastive learning model from the image to video domain, while Temporal Contrast learning adapts to this new domain by further applying temporal augmentations to create the sample variant. Thus the results suggest that traditional contrastive learning models should be further adapted to learn a good embedding space for video, and we present one possible solution.

<table>
<thead>
<tr>
<th>SC</th>
<th>TC</th>
<th>( \alpha_k )</th>
<th>( \beta_k )</th>
<th>Top-1 Acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>-</td>
<td>1</td>
<td>-</td>
<td>61.3</td>
</tr>
<tr>
<td>-</td>
<td>5</td>
<td>-</td>
<td>1</td>
<td>62.7</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>65.5</td>
</tr>
<tr>
<td>4,5</td>
<td>4,5</td>
<td>1,1</td>
<td>1,1</td>
<td>66.9</td>
</tr>
<tr>
<td>4,5</td>
<td>4,5</td>
<td>0.5,1</td>
<td>0.5,1</td>
<td>67.8</td>
</tr>
<tr>
<td>3,4,5</td>
<td>3,4,5</td>
<td>1,1,1</td>
<td>1,1,1</td>
<td>66.8</td>
</tr>
</tbody>
</table>

Training from scratch: - - 43.5

<table>
<thead>
<tr>
<th>Top-1 Acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>3,4,5</td>
</tr>
</tbody>
</table>

Table 1: Ablation study of decoupled contrast and hierarchical contrast. We use 3D-ResNet18 as the backbone, pretrain on Kinetics-400 and report top-1 accuracy on UCF101 split 1. SC and TC indicate the scale at which we perform Spatial Contrast (SC) or Temporal Contrast (TC) self-supervised learning – i.e., the index of the block we take features from. \( \alpha_k \) and \( \beta_k \) are defined in Eq. 6 (listed in order of scale). Crops are taken for each clip [69]. The class score for each video is obtained by averaging over all crops and clips.
Hierarchical Contrast. Comparing rows 3 with 4 or 6 of Tab. 1, we observe that Hierarchical Contrast learning at more scales yields better performance. This suggests that instance-level invariance widely exists for mid-level and high-level features from previous layers, and can be used to capture multi-scale semantics. However, rows 4 and 6 show that simply adding more scales does not consistently bring improvement. We argue that this is because instance-level invariance may be weaker for mid-level features, and adding more scales while giving them the same weight of significance will distract the network and harm the learning. We discuss this below.

Significance of Instance-level Invariance at Different Scales. Zeiler et al. [79] showed that early layers of neural networks learn low-level features which change a lot due to augmentations. As we perform Hierarchical Contrast learning at more scales, those mid-level features may not share the same level of invariance against augmentations as the last layer’s features. We use $\alpha_k$ and $\beta_k$ to model the significance of instance-level invariance at different scales. We conducted experiments with different values of $\alpha_k$ and $\beta_k$, and the results (row 4 vs row 5, row 6 vs the last row in Tab. 1) show that smaller weights for lower levels of the hierarchy yields better performance than assigning 1’s. This suggests that significance decreases at lower layers, which is consistent with [79]. We did not exhaustively tune $\alpha_k$ and $\beta_k$, so better performance is likely possible through tuning.

Spatial augmentation ablutions. Tab. 2 shows ablation results of using different spatial augmentations. We find that channel replication is crucial for the model to learn good features, perhaps because it is a non-linear projection of RGB channels which can effectively prevent the network from learning trivial solutions based on color distribution [39]. As another way to prevent such trivial solutions, color jittering uses a linear function and thus is less effective. However, when neither color jittering nor channel replication is used, the accuracy drops greatly, indicating the network may suffer from the trivial solution.

We note that, as shown in [7], there are other spatial augmentations which can further improve the performance of contrastive learning. However, the purpose of this paper is not to exhaustively explore effects of different augmentations. By applying a set of simple augmentations, we show the generalizability of our HDC.

Spatial augmentations in Temporal Contrast. We verify the importance of applying spatial augmentations after temporal random cropping in Temporal Contrast. We use our full model of HDC with C3D as the backbone, pretrain on Kinetics-400, and report average accuracy on 3 splits of UCF101 and HMDB51. It achieves 72.3% on UCF101 and 39.3% on HMDB51 when spatial augmentations are used in Temporal Contrast versus only 68.9% on UCF101 and 38.0% on HMDB51 when spatial augmentations are not used. This supports our design choice of having spatial augmentations in Temporal Contrast to obtain variants whose spatial context varies as much as possible.

### Table 2: Ablation study of different spatial augmentations.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>α</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>Random Initialization</td>
<td>Shuffle &amp; Learn</td>
<td>Büchler et al.</td>
</tr>
<tr>
<td>β</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>Random Initialization</td>
<td>Shuffle &amp; Learn</td>
<td>Büchler et al.</td>
</tr>
</tbody>
</table>

Our full model (HDC) 69.0

### Table 3: Top-1 accuracy averaged on 3 splits of UCF101 and HMDB51. Parentheses show accuracy obtained with our own implementation. HDC with † is pretrained by decreasing learning rate more slowly.

<table>
<thead>
<tr>
<th>Method</th>
<th>Architecture</th>
<th>HMDB51</th>
<th>UCF101</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random Initialization [45]</td>
<td>CaffeNet</td>
<td>13.3</td>
<td>38.6</td>
</tr>
<tr>
<td>Shuffle &amp; Learn [45]</td>
<td>CaffeNet</td>
<td>18.1</td>
<td>50.2</td>
</tr>
<tr>
<td>Büchler et al. [5]</td>
<td>CaffeNet</td>
<td>25.0</td>
<td>58.6</td>
</tr>
<tr>
<td>Random Initialization [39]</td>
<td>VGG-M-2048</td>
<td>18.3</td>
<td>51.1</td>
</tr>
<tr>
<td>OPN [39]</td>
<td>VGG-M-2048</td>
<td>23.8</td>
<td>59.8</td>
</tr>
<tr>
<td>Random Initialization [74]</td>
<td>R3D-18</td>
<td>21.5</td>
<td>54.4</td>
</tr>
<tr>
<td>Clip Order [74]</td>
<td>R3D-18</td>
<td>29.5</td>
<td>64.9</td>
</tr>
<tr>
<td>VCP [41]</td>
<td>R3D-18</td>
<td>31.5</td>
<td>66.0</td>
</tr>
<tr>
<td>Random Initialization [33]</td>
<td>R3D-18+1</td>
<td>19.4</td>
<td>44.7</td>
</tr>
<tr>
<td>Clip Order [33]</td>
<td>R3D-18+1</td>
<td>37.8</td>
<td>69.4</td>
</tr>
<tr>
<td>Random Initialization [15]</td>
<td>2D3D-ResNet18</td>
<td>17.1</td>
<td>46.5</td>
</tr>
<tr>
<td>DPC [15]</td>
<td>2D3D-ResNet18</td>
<td>34.5</td>
<td>68.2</td>
</tr>
<tr>
<td>Random Initialization [4]</td>
<td>I3D</td>
<td>29.6</td>
<td>47.9</td>
</tr>
<tr>
<td>SpeedNet [4]</td>
<td>I3D</td>
<td>43.7</td>
<td>66.7</td>
</tr>
<tr>
<td>Random Initialization [18]</td>
<td>3D-ResNet18</td>
<td>17.1 (19.1)</td>
<td>42.4 (43.8)</td>
</tr>
<tr>
<td>3D-ST-Puzzle [32]</td>
<td>3D-ResNet18</td>
<td>33.7</td>
<td>65.8</td>
</tr>
<tr>
<td>Our method (HDC)</td>
<td>3D-ResNet18</td>
<td>38.4</td>
<td>69.8</td>
</tr>
<tr>
<td>Our method (HDC†)</td>
<td>3D-ResNet18</td>
<td>38.1</td>
<td>68.5</td>
</tr>
<tr>
<td>Random Initialization [74]</td>
<td>C3D</td>
<td>23.2 (22.5)</td>
<td>61.6 (51.4)</td>
</tr>
<tr>
<td>Motion &amp; Appearance [67]</td>
<td>C3D</td>
<td>33.4</td>
<td>61.2</td>
</tr>
<tr>
<td>Clip Order [74]</td>
<td>C3D</td>
<td>28.4</td>
<td>65.6</td>
</tr>
<tr>
<td>VCP [41]</td>
<td>C3D</td>
<td>32.5</td>
<td>68.5</td>
</tr>
<tr>
<td>Cho et al. [8]</td>
<td>C3D</td>
<td>34.3</td>
<td>70.4</td>
</tr>
<tr>
<td>Our method (HDC)</td>
<td>C3D</td>
<td>39.5</td>
<td>73.1</td>
</tr>
<tr>
<td>Our method (HDC†)</td>
<td>C3D</td>
<td>39.3</td>
<td>72.3</td>
</tr>
<tr>
<td>Random Initialization [74]</td>
<td>R(2+1)D-10</td>
<td>22.0</td>
<td>56.2</td>
</tr>
<tr>
<td>Clip Order [74]</td>
<td>R(2+1)D-10</td>
<td>30.9</td>
<td>72.4</td>
</tr>
<tr>
<td>VCP [41]</td>
<td>R(2+1)D-10</td>
<td>32.2</td>
<td>66.3</td>
</tr>
<tr>
<td>Cho et al. [8]</td>
<td>R(2+1)D-10</td>
<td>36.8</td>
<td>74.8</td>
</tr>
<tr>
<td>PacePrediction [68]</td>
<td>R(2+1)D-10</td>
<td>36.6</td>
<td>77.1</td>
</tr>
<tr>
<td>Our method (HDC)</td>
<td>R(2+1)D-10</td>
<td>40.0</td>
<td>76.8</td>
</tr>
<tr>
<td>Our method (HDC†)</td>
<td>R(2+1)D-10</td>
<td>39.8</td>
<td>76.2</td>
</tr>
</tbody>
</table>
Figure 4: Sample results of nearest neighbor retrieval on UCF101 split 1, showing the top-1 retrieved clip from our full model (HDC), and Spatial Contrast (SC), Temporal Contrast (TC), and Joint Spatial-Temporal Contrast (SC + TC) models. Below each clip is its action category, with green underlined text indicating correct retrievals and red text indicating incorrect. Our proposed HDC achieved better results because of the ability to capture both spatial and temporal features at multiple scales.

<table>
<thead>
<tr>
<th>Query Clip</th>
<th>HDC</th>
<th>SC only</th>
<th>TC only</th>
<th>SC + TC</th>
</tr>
</thead>
<tbody>
<tr>
<td>BaseballPitch</td>
<td>BasballPitch</td>
<td>BodyWeightSquats</td>
<td>TaiChi</td>
<td>BaseballPitch</td>
</tr>
<tr>
<td>CuttingInKitchen</td>
<td>CuttingInKitchen</td>
<td>Mixing</td>
<td>Hammering</td>
<td>CuttingInKitchen</td>
</tr>
<tr>
<td>Surfing</td>
<td>Surfing</td>
<td>FrisbeeCatch</td>
<td>Skiing</td>
<td>Surfing</td>
</tr>
</tbody>
</table>

Table 4: Nearest neighbor retrieval results on UCF101.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Top1</th>
<th>Top5</th>
<th>Top10</th>
<th>Top20</th>
<th>Top50</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPN [39]</td>
<td>19.9</td>
<td>28.7</td>
<td>34.0</td>
<td>40.6</td>
<td>51.6</td>
</tr>
<tr>
<td>Büchler et al. [5]</td>
<td>25.7</td>
<td>36.2</td>
<td>42.2</td>
<td>49.2</td>
<td>59.5</td>
</tr>
<tr>
<td>Random Initialized C3D</td>
<td>16.7</td>
<td>27.5</td>
<td>33.7</td>
<td>41.4</td>
<td>53.0</td>
</tr>
<tr>
<td>Clip Order (C3D) [74]</td>
<td>12.5</td>
<td>29.0</td>
<td>39.0</td>
<td>50.6</td>
<td>66.9</td>
</tr>
<tr>
<td>VCP (C3D) [41]</td>
<td>17.3</td>
<td>31.5</td>
<td>42.0</td>
<td>52.6</td>
<td>67.7</td>
</tr>
<tr>
<td>CCL (R3D-18+1) [33]</td>
<td>22.0</td>
<td>39.1</td>
<td>44.6</td>
<td>56.3</td>
<td>70.8</td>
</tr>
<tr>
<td><strong>Our HDC (C3D)</strong></td>
<td><strong>33.9</strong></td>
<td><strong>49.6</strong></td>
<td><strong>55.7</strong></td>
<td><strong>61.6</strong></td>
<td><strong>69.9</strong></td>
</tr>
</tbody>
</table>

Table 5: Nearest neighbor retrieval results on HMDB51.

4.2. Comparison with the State-of-the-art

For fairness, we compare with methods under similar settings. There are other methods that achieve great success. But they adopt much more advanced backbones [10,48,77], require extra preprocessing [16, 17, 72], or needs additional information to prepare the input [2, 34, 43, 55].

We report top-1 accuracy averaged over 3 splits of UCF101 and HMDB51 in Tab. 3. HDC achieves better or comparable performance on both datasets. We note that: 1) PacePrediction [68] shows slightly better results on UCF101, perhaps because its best performance is achieved by further adapting contrastive laerning in addition to pace prediction; 2) SpeedNet [4] benefits from the backbone (29.6% when trained from scratch) and thus has better performance on HMDB51.

By comparing HDC variants using different backbones, we observe that a more advanced backbone always leads to better performance, suggesting that HDC will be able to further benefit from future advances in network architectures. We also find that decreasing the learning rate more slowly is beneficial, perhaps because it allows better optimization.

4.3. Nearest Neighbor Retrieval

We follow [74] and perform nearest neighbor retrieval experiments. As shown in Tables 4 and 5, our method significantly outperforms other methods on both UCF101 and HMDB51. This implies that we learn better features, and explains the good performance on downstream tasks.

As shown in Fig. 1 and Fig. 4, qualitative results further support our idea of manipulating augmentations to guide the network to learn different features and the benefit of hierarchical learning. For example, in the first row of the right column in Fig. 4, HDC succeeds in retrieving a clip of the correct action while the other variants fail: SC focuses more on spatial information of people outdoors, TC pays more attention to the actions of people walking, and ST+TC fails perhaps because it does not learn features at different hierarchies, although it successfully captured both spatial and temporal information of multiple moving people.

5. Conclusion

We considered the problem of unsupervised video representation learning, and introduced Hierarchically Decoupled Spatial-Temporal Contrast (HDC). By decomposing the target into subtasks emphasizing different features and performing learning in a hierarchical manner, HDC is able to capture both rich spatial and temporal semantics at multiple scales. Extensive experiments of action recognition and nearest neighbor retrieval on UCF101 and HMDB51 using 3 different backbones suggest the potential of manipulating augmentations as regularization and demonstrate the state-of-the-art performance of HDC.
References


