# GOCA: Guided Online Cluster Assignment for Self-Supervised Video Representation Learning

Huseyin Coskun$^{\star 1,2}$, Alireza Zareian$^{1}$, Joshua L. Moore$^{1}$,
Federico Tombari$^{2,3}$, and Chen Wang$^{1}$

$^{1}$ Snap Inc.
$^{2}$ TU Munich, Germany
$^{3}$ Google
hcoskun@snap.com

**Abstract.** Clustering is a ubiquitous tool in unsupervised learning. Most of the existing self-supervised representation learning methods typically cluster samples based on visually dominant features. While this works well for image-based self-supervision, it often fails for videos, which require understanding motion rather than focusing on background. Using optical flow as complementary information to RGB can alleviate this problem. However, we observe that a naïve combination of the two views does not provide meaningful gains. In this paper, we propose a principled way to combine two views. Specifically, we propose a novel clustering strategy where we use the initial cluster assignment of each view as prior to guide the final cluster assignment of the other view. This idea will enforce similar cluster structures for both views, and the formed clusters will be semantically abstract and robust to noisy inputs coming from each individual view. Additionally, we propose a novel regularization strategy to address the feature collapse problem, which is common in cluster-based self-supervised learning methods. Our extensive evaluation shows the effectiveness of our learned representations on downstream tasks, e.g., video retrieval and action recognition. Specifically, we outperform the state of the art by 7% on UCF and 4% on HMDB for video retrieval, and 5% on UCF and 6% on HMDB for video classification. [1]

**Keywords:** Clustering, Self-Supervised Learning, Action Recognition

## 1 Introduction

The pursuit of understanding human activities in videos is a fundamental problem in computer vision. Representation learning methods with supervised training strategies showed promising results on various tasks, such as action understanding [12,20,28,43,54], action detection and localization [13,95], and action proposal [32,56]. It is fair to say that large-scale, manually labeled video datasets such as Kinetics [45], AVA [35], and Epic [22] substantially contributed to that success. In spite of those promising results, these algorithms can usually only recognise activities if they have access to a semantically labelled dataset. The cost and challenges of collecting large-scale, manually labelled videos hinder further improvements in activity understanding. On the other hand,

[⋆] Work done during internship at Snap Inc.

[1] Code available at https://github.com/Seleucia/goca

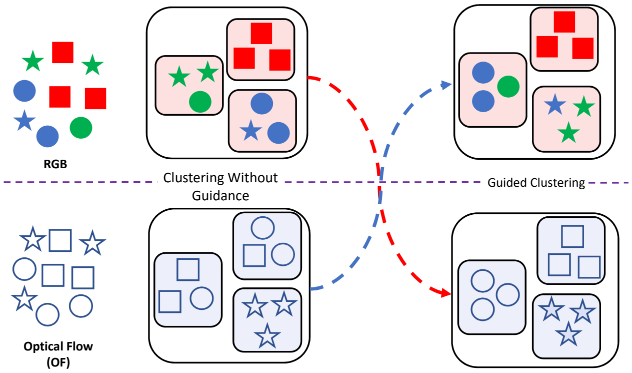

Fig. 1: An abstract illustration of the proposed idea. The filled and empty shapes represent *RGB* and *OF* views respectively. Each shape represents a different type of activity, while colors represent intra-class variations. In this toy example, the initial clustering is wrong for both views. *RGB* fails because it clusters based on color (e.g. irrelevant background information), while *OF* fails since it confuses square with circle (due to the lower resolution of details). By combining the two types of information, clustering can be achieved correctly on both views.

the internet is a virtually unlimited source of unlabeled videos (e.g. YouTube). Therefore, designing a representation learning strategy that does not rely on manual labelling is fundamentally important.

Self-supervised learning (SSL) aims to address this issue, by designing pretext tasks that only rely on input, and training networks to solve those tasks. Recent advances in SSL for image understanding [11,15,34,36,38,70,79,96] have achieved excellent performance in various downstream tasks. Motivated by this success, several papers brought these ideas to the video domain [23,29,68,76]. Although these methods show promising results to some extent, they rely solely on an *RGB* stream. As demonstrated by [4,6,37], this is not sufficient to learn a strong temporal representation.

Johansson's classical psychology work [44] shows that humans can recognize activities by only watching a few bright dots depicting the movement of the main body joints. This intuitive work motivated researchers to use optical flow as a representation of motion for activity understanding [12,30,71,72], and they have achieved significant improvements over *RGB*-only models in the supervised learning literature. Inspired by this success, many recent SSL works [33,37,80] have explored using optical flow (*OF*) to advance SSL beyond *RGB*-only baselines. Han *et al.* (CoCLR) [37] used *OF* to retrieve positive samples for the infoNCA [63], which led to significant improvements. Nevertheless, CoCLR did not utilize *OF* for training the backbone and hence may not have realized the full potential of learning motion representations. VICC [80] adopted the online cluster assignment [11] to videos by considering *OF* as another view of *RGB*, and minimized the distance between *RGB* and *OF* features during online clustering. This method obtained SOTA results on various datasets. However, enforcing similarity between *RGB* and *OF* features can be detrimental, especially if one information source is noisy, as is usually the case for OF due to camera motion. Furthermore, these models [37,80] require a complicated training strategy that successively updates one model while freezing the parameters of the other model, which prevents end-to-end training.

In this paper, we introduce the guided online cluster assignment algorithm (GOCA) to address the aforementioned problems. Specifically, for a given video with *RGB* and *OF* representations, we first compute initial cluster assignments for only using *RGB* or *OF* separately, and then we use these assignments as priors for each other to compute a final assignment that is guided by both views, as illustrated in Fig. 1. After we obtain our final assignment, we train a backbone network by minimizing a cross entropy loss between the final cluster assignment of different augmentations of the same video (as used in  [11]). The proposed idea has several benefits compared to the state of the art [11,37,80]. First, it constructs more robust clusters during training due to prior information, which is particularly important when one information source is noisy. Second, allowing *RGB* and *OF* to share information by means of sharing cluster assignment encourages the two views to form similar cluster structure, which leads to more semantically abstract representations. Third, both *RGB* and *OF* backbones are trained jointly and information flows both ways during training, which is beneficial for both backbones due to the complementary nature of these views. Fourth, compared to the CoCLR method [37], *OF* is utilized more explicitly in our formulation, which leads to stronger spatio-temporal representations. Finally, the proposed approach circumvents complicated training strategies [37,33,80] and allows simple and end to end training.

We also propose a novel prototype regularization method to address the feature collapse problem, where all features are mapped to a single point. This is a common problem in SSL and was partially addressed in SwAV [11] via equipartition constraint. However, it requires careful tuning[1] of the parameter $\lambda$ (See Eq. (5)) of the Sinkhorn algorithm, which makes it hard used in practice [9,16,17,69]. We address this problem by constructing cluster prototypes which are maximally distant from each other. We achieve this by locating the $N$ prototypes in the $\Phi$ dimensional space such that they divide the space equally. Despite the simplicity of the proposed idea, it yields consistent performance improvements.

To sum up, our contributions are: 1) we introduce a novel Guided Online Cluster Assignment (GOCA) algorithm that aims to learn stronger spatio-temporal representations by utilizing the complementary information of *RGB* and *OF*; 2) we mathematically prove that guidance based clustering can be achieved efficiently with the Sinkhorn algorithm; 3) we propose a prototype regularization strategy that addresses the common feature collapse problem; 4) we perform extensive evaluations of our method using two backbones (S3D [85] and R(2+1)D [81]) on four different evaluation regimes (See Sec. 4). The proposed model outperforms the state of the art in almost all experiments. Furthermore, we present ablation studies to show the effect of each contribution and the key parameters of our method.

## 2 Related Works

[1] https://github.com/facebookresearch/swav#common-issues

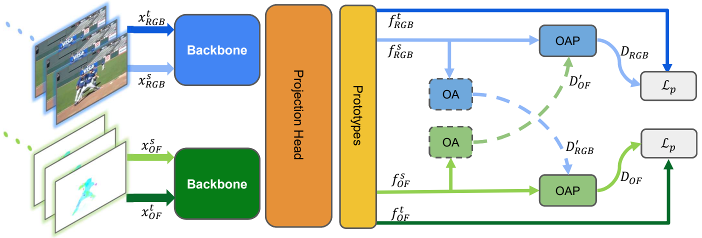

Fig. 2: An overview of the proposed GOCA algorithm. The *RGB* and Optical Flow (*OF*) views have the same backbone architectures and share the projection head and prototypes. Optimal assignment (OA) and optimal assignment with prior (OAP) are defined in Eq. (6) and Eq. (7), respectively. $\mathcal{L}_p$ is defined for a batch of videos in Eq. (13)

*Self-supervised representation learning.* The success of image-based SSL methods [25,26,38,48,62,96,70,79,94] has inspired their application in the video domain, resulting in several recently proposed methods in that domain. Early approaches for video representation learning [1,24,75,82] typically relied on the idea of predicting the future frames given the past frames, which requires carefully training a deep generative model. Other works focus on designing proxy tasks to exploit temporal information. In a pioneering work, Misra *et al.* [60] designed a pretext task that predicts whether the order of video frames is correct or wrong. Follow-up methods achieved better performance by designing new pretext tasks such as predicting clip order [31,49,60,86], pace [8,18,83,88], or the arrow of time [66,84]. To achieve better generalization, aside from designing novel pretext tasks, some recent works focus on using instance-based contrastive learning approaches [23,29,39,52,64,68]. Since videos often contain multi-modal data, many recent works used audio [2,3,4,5,6,46,65] or language [2,3,59,77] as complementary information to improve results. Unlike those works, we focus on learning representations purely on visual input, namely *RGB* and optical flow (*OF*). Note that we compute *OF* from *RGB* videos in an unsupervised way.

*Clustering based self-supervised learning.* Early works in this direction adopted clustering to representation learning by simply applying clustering to obtain pseudo-labels [4,10,87]. The common approach has two alternating steps, as it performs clustering offline, rather than concurrently with training as described in Sec. 3.1. One of the main limitations of that approach is that such naive implementations may form degenerate clusters where all samples are clustered to the same point. Asano *et al.* [6,7] tackle this problem by adding an equipartition constraint on the number of samples per cluster and converting the pseudo-label generation step into an optimal assignment problem. They solve the resulting problem using the Sinkhorn-Knopp algorithm [21]. Although the equipartition constraint achieves promising results, it is still inefficient due to the offline clustering. Recently, Caron *et al.* [11] (closest work to ours) eliminated offline label generation via an online clustering algorithm which allows them to train end-to-end on large-scale datasets. Although it works well for images, we observe that adapting this idea to videos is not as effective as with images. As seen in [4,6] and our results, RGB alone is not sufficient to form representative clusters for videos. We tackle this problem by using *OF* as complementary information to guide clustering.

# 3 Method

Fig. 2 shows an overview of the proposed method. For a given video with *RGB* and *OF* views, we first compute two different augmentations for each view, and then pass them through the *RGB* and *OF* backbones, followed by a shared projection head and a shared prototype layer in order to compute cluster assignments.

## 3.1 Preliminaries

State-of-the-art clustering based SSL methods (SwAV [11], VICC [80], Sela [90], Selavi [7]) are built on top of the Cuturi *et. al.* [21] formulation. The Cuturi *et. al.* [21] formulation is used to find optimal assignments from samples to cluster centers (or prototypes) under constraints.

Formally, consider a given minibatch of $M$ videos $X = \{x_1, ..., x_M\}$, and $N$ prototypes $P = \{p_1, ..., p_N\}$ represented by trainable vectors. The idea of [11] is to compute two random augmentations $X^t$ and $X^s$, and compute their feature vectors $F^t = \{f^t_1, ..., f^t_i\}$ and $F^s = \{f^s_1, ..., f^s_i\}$ using an encoder network $\theta$. Next, optimal assignments $D^t = \{d^t_1, ..., d^t_M\}$ and $D^s = \{d^s_1, ..., d^s_M\}$ are computed from features $F^t$ and $F^s$ to prototypes $P$, as described in the following paragraph. $d^t_i \in D^t$ and $d^s_i \in D^s$ vectors represent assignment values from $f^s_i$ and $f^t_i$ to prototypes, respectively. Finally, $\theta$ is trained with the following loss function:

$$\mathcal{L}(F^t, F^s) = \textstyle\sum_i^M \Big( l(d^t_i, g^s_i) + l(d^s_i, g^t_i) \Big), \tag{1}$$

where

$$l(d^t, g^s) = -\sum_n^N d^t_n log(g^s_n), \;\; g^s_n = \frac{exp((f^s)^\top p_n)}{\sum_{n'} exp((f^s)^\top p_{n'})} \tag{2}$$

This loss function minimizes the distance of two different augmentations by comparing them according to their assignment. Ideally, the two augmentations should be assigned to the same prototype, due to identical semantic content.

*Computing optimal assignment:* The optimal assignment (OA) $D$ can be found by solving the following optimization:

$$d_C(F, P) := \min_{D \in U} \langle D, C \rangle, \tag{3}$$

where $C \in R^{M \times N}$ is a distance matrix from a batch of feature vectors to prototypes and $\langle D, C \rangle = tr(C^\top D)$. $U$ represents all possible assignments from our features $F$ to prototypes $P$. Formally:

$$U := \{D \in R_+^{M \times N} \mid D1_N = \psi, D^\top 1_M = \omega\}, \tag{4}$$

where $\psi = \frac{1}{M} * \vec{1}_M$ and $\omega = \frac{1}{N} * \vec{1}_N$, and $\vec{1}_M \in R^M$ and $\vec{1}_N \in R^N$ are ones vectors. The constraint on $U$ ensures all the prototypes are selected. Eq. (3) can be solved with linear programming. Since this is computationally expensive, it is infeasible to carry it out during training for every batch. Cuturi *et al.* [21] address this issue by adding entropy regularization and solving the resulting problem with the Sinkhorn algorithm:

$$d_C(F,P) := \min_{D\in U} \Big(\langle D, C\rangle - \lambda_1 h(D)\Big). \tag{5}$$

where $h$ represents the entropy. The above formulation is a strictly convex and smoothed version of Eq. (3). This allows us to efficiently approximate $D$. Specifically, $D$ has a unique solution for any given $\lambda_1$, which is in the form of: $D^{\lambda_1} = diag(u) K\, diag(v)$ where $K = e^{\lambda_1 C}$, and $u, v$ are non-negative unique vectors. The proof of this statement can be found in [21]. According to the Sinkhorn theorem[27], if $K$ has only positive elements, the unknowns $u$ and $v$ can be determined via the Sinkhorn algorithm.

There are two main problems with this conventional formulation: First, it does not allow utilizing the complementary information, which is particularly important in the video, since it has been shown that using *RGB* alone encourages the model to focus on the background and ignore motion cues [6]. We address this problem by rewriting Eq. (3) such that we can fuse two information sources to obtain the cluster assignment. Second, in the conventional formulation [11,80], the prototypes are randomly initialized and trained with the rest of the model without any restriction, which can lead to a degenerate solution where all prototypes collapse into a single point, and consequently the feature vectors too [9,16,17,69]. To overcome this problem, we propose to use a novel prototype regularizer, which encourages prototypes to be maximally far apart.

### 3.2 Guided Online Cluster Assignment (GOCA)

Our goal is to train backbone networks $\theta_{\text{RGB}}$ and $\theta_{\text{OF}}$ while sharing information between the two views. The single-view approach of [10,11,90] does not allow combining the two information sources. A straightforward way is to concatenate the $F_{\text{RGB}}$ and $F_{\text{OF}}$ features, but we have observed that this approach yields a very small improvement over *RGB*-only training on various downstream tasks (see Experiments 4.1). Another way is to force the similarity of *OF* and *RGB* features as in [80]. Even though this idea obtains better results on downstream tasks, it leads to a loss of information by forcing each view to only maintain mutual information. Hence, it does not reach the maximum potential performance of correctly combining the two views (Tab. 1).

In contrast, we propose a principled way to combine two information sources. We use each information source as a prior to the other. We assign $D'_{\text{OF}}$ and $D'_{\text{RGB}}$ as the initial prototype assignment (computed based on Eq. (5)), and $D_{\text{OF}}$ and $D_{\text{RGB}}$ as the final. To this end, we use the $D'_{\text{OF}}$ as the prior to $D_{\text{RGB}}$, and $D'_{\text{RGB}}$ as the prior to $D_{\text{OF}}$

We implement this idea by modifying Eq. (5) in a way that it takes the prior into account. Our optimal assignment with prior (OAP) optimization problems for $D_{\text{RGB}}$ and $D_{\text{OF}}$ takes this form:

$$d_{C_{\text{RGB}}}(F_{\text{RGB}},P) \quad := \min_{D_{\text{RGB}}\in U} \Big(\langle D_{\text{RGB}}, C_{\text{RGB}}\rangle - \lambda_1 h(D_{\text{RGB}}) + \lambda_2 \text{KL}(D_{\text{RGB}}|D'_{\text{OF}})\Big), \tag{6}$$

and

$$d_{C_{\text{OF}}}(F_{\text{OF}},P) \quad := \min_{D_{\text{OF}}\in U} \Big(\langle D_{\text{OF}}, C_{\text{OF}}\rangle - \lambda_1 h(D_{\text{OF}}) + \lambda_2 \text{KL}(D_{\text{OF}}|D'_{\text{RGB}})\Big). \tag{7}$$

Where $C_{\text{RGB}}$ and $C_{\text{OF}}$ are distance matrices from *RGB* and *OF* features to prototypes, $\text{KL}(\cdot|\cdot)$ represents the Kullback Leibler divergence between two assignment matrices, $\lambda_1$ and $\lambda_2$ are hyper-parameters. Note that we use the same prototypes $P$ for *OF* and *RGB* features. These two optimization problems can be solved via the following lemma:

**Lemma 1.** $D_{\text{RGB}}$ and $D_{\text{OF}}$ have a unique solution for $\lambda_1$ and $\lambda_2$ in the form of:

$$D_{\text{b}}^{\lambda_1,\lambda_2} = diag(u_{\text{b}})K_{\text{b}}diag(v_{\text{b}}) \qquad b \in \{RGB, OF\} \tag{8}$$

*Proof* We will prove only for $D_{\text{RGB}}$, but the same proof can be used for *OF* as well. Let $\mathcal{L}(D_{\text{RGB}}, \alpha, \beta)$ be the Lagrangian form of the Equation(6) with dual variables $\alpha \in R^M$ and $\beta \in R^N$ for the two equality constraints in $U$ (see Eq. (4)):

$$\begin{aligned}\mathcal{L}(D_{\text{RGB}}, \alpha, \beta) = &\sum_{i,j} \lambda_1 d_{\text{RGB}_{ij}} log(d_{\text{RGB}_{ij}}) \quad + \sum_{i,j} d_{\text{RGB}_{ij}} c_{\text{RGB}_{ij}} + \sum_{i,j} \lambda_2 d_{\text{RGB}_{ij}} log\left(\frac{d_{\text{RGB}_{ij}}}{d'_{\text{OF}_{ij}}}\right) \\ &+ \alpha(D_{RGB}1_M - \psi) + \beta(D_{RGB}^{\top}1_N - \omega)\end{aligned} \tag{9}$$

for all $(i, j)$ we set $(\partial\mathcal{L}/\partial d_{\text{RGB}_{ij}} = 0)$ and solve for $d_{\text{RGB}_{ij}}$

$$d_{\text{RGB}_{ij}} = \exp\left(-0.5 - \frac{\alpha_i}{\lambda_1 + \lambda_2}\right) \exp\left(\frac{-c_{\text{RGB}_{ij}} + \lambda_2 log(d'_{\text{OF}_{ij}})}{\lambda_1 + \lambda_2}\right) \exp\left(-0.5 - \frac{\beta_j}{\lambda_1 + \lambda_2}\right). \tag{10}$$

When we choose $K_{\text{RGB}} = exp(\frac{-c_{\text{RGB}_{ij}} + \lambda_2 log(d'_{\text{OF}_{ij}})}{\lambda_1 + \lambda_2})$, we can see that it is strictly positive since it is the element-wise exponential, therefore according the Sinkhorn's theorem [73], $D_{\text{RGB}}^{\lambda_1,\lambda_2}$ has unique solution for any given $\lambda_1$ and $\lambda_2$. Thus, $u_{\text{RGB}}$ and $v_{\text{RGB}}$ vectors in Eq. (8) can be computed with Sinkhorn algorithm. In our formulation, since the final cluster assignment relies on both views, formed clusters will be robust to noise and semantically abstract. The supplemental includes the details of the proof.

### 3.3 Prototype Regularization

It has been shown that SSL models suffer from feature collapse, i.e. when all features are mapped to the same representation [14,34,70,93]. Even though this problem was partially addressed in [11] by using equipartition constraint as in Eq. (3), in practice, $\lambda$ needs to be carefully tuned [9,16,17,69]. More specifically, we observe that a higher $\lambda$ leads to numerical issues, and lower values tend to cause feature collapse.

Alternatively, we introduce a regularization term that encourages prototypes to be maximally far apart. We achieve this by utilizing the idea of hyperspherical prototypes [58]. Formally, we divide the $\Phi$-dimensional hyperspherical space equally into N prototypes. For instance, for a 2-dimensional hyperspherical space (circle), this can be easily done by placing $N$ prototypes with a $\frac{2\pi}{N}$ angle difference. Even though this is easy to do for a 2-dimensional space, there is no exact solution for 3 or more dimensions [78]. We solve this problem by finding an approximate solution using gradient descent. Specifically, consider the $N$ prototypes that are represented with a linear layer, $W \in R^{N \times \Phi}$ in our network (Fig. 2). Instead of training $W$ with the rest of the network end-to-end, we train it separately, once before the main training phase, by minimizing the following loss under the following constraint:

$$\mathcal{L}_{reg} = \frac{1}{N}\sum_{i}^{N} max(\Omega_{i,.}), \quad \Omega = WW^{\top} - 2I \;\; s.t. \quad \forall_i \|\mathbf{w_i}\| = 1, \tag{11}$$

where $I$ and $w_i$ represents the identity matrix and $i$-th row in $W$, respectively. This loss minimizes the similarity of maximally similar prototypes. To apply the constraint, we continuously re-project our prototypes to the hypersphere during training via $l2$ normalization. This can be seen as a simple and quick initialization step, which takes only 5 minutes on a Nvidia GTX 1060 GPU.

### 3.4 Training procedure

After initializing (and fixing) the prototype layer as described in Eq. (11), we train the network by minimizing the following loss function:

$$\mathcal{L}_{final} = \mathcal{L}_p(F^t_{\text{RGB}}, F^s_{\text{RGB}}) + \mathcal{L}_p(F^t_{\text{OF}}, F^s_{\text{OF}}), \tag{12}$$

where

$$\mathcal{L}_p(F^t_{\text{b}}, F^s_{\text{b}}) = \sum_{i}^{M} l(d^t_{\text{b}_i}, g^s_{\text{b}_i}) + l(d^s_{\text{b}_i}, g^t_{\text{b}_i}) \tag{13}$$

Here $b \in \{RGB, OF\}$ and $d_{\text{b}_i}$ $i$-th row in $D_{\text{b}}$. $D_{\text{b}}$ represents optimal assignment matrix. We compute these matrices using *Lemma 1*, while our prototypes are obtained as described in Sec. 3.3. Note that $l$ and $g$ are defined in Eq. (2). Since *RGB* and *OF* features only interact when computing optimal assignment, this formulation encourages cluster similarity without enforcing *RGB* and *OF* features to be strictly similar, hence, not leading to information loss.

## 4 Experimental Results

In this section, we first describe datasets, metrics, and training details. We present our ablation results to show the importance of our method design choices in Sec. 4.1. Then, we compare our approach with SOTA in Sec. 4.2 for retrieval task. Then, we show our cluster analysis results in Sec. 4.3. Finally, we show classification results in Sec. 4.4.

**Datasets**: We conduct our experiments on four different datasets: Kinetics (K400) [45], UCF [74], HMDB [47], and Diving-48 [51]. We follow the same training protocols as other self-supervised learning approaches [6,37,68,80]. K400 training set contains 240k videos. UCF, HMDB, and Diving-48 contain 13k, 7k, and 17k videos respectively.
**Evaluation Metrics**: We evaluate our model performance on the action retrieval and action classification tasks. To evaluate action retrieval, we compute *Recall* at $K \in \{1, 5, 10, 20\}$, similar to earlier works [6,37,68,80]. More specifically, if the correct class is within the $K$ nearest neighbours we consider it a correct result. For action

classification, we consider top-1 accuracy for *Linear Classification* and *Fine Tuning* experimental setups (See Sec. 4.4). We report all numbers in terms of percentage.

**Data Augmentations**: Following [37,68,80], we use horizontal flipping, random cropping, Gaussian blurring, and color jittering for augmentation. We also use a *multi-temporal-resolution* idea which is analogous to multi-crop in SwAV [11]. In this augmentation strategy, we sample shorter length clips alongside longer ones. We observe this makes convergence faster, but does not affect the final accuracy.

**Backbones and Training details**: We conduct our experiments with two widely used backbones: S3D [85] and R(2+1)D+18 [81]. We use two datasets (K400 [45] and UCF [74]) for self-supervised training. Optical flow (*OF*) is computed using the TV-L1 algorithm [92] and pre-processed as in [12,37,80]. We follow earlier works [29,68,80] and use SGD+LARS [91] optimizer with a learning rate of $4.8$ that is increased during the first 10 warm-up epochs and then is decreased to $0.0048$ with cosine learning rate decay. All models are pre-trained on 64 V100 GPUs (10 samples per GPU) for 500 epochs as in [6,29,68,80]. Each clip contains 32 consecutive frames. During the test, we turn off all augmentations and use a standard 3 (spatial) $\times$ 10 (temporal). We always denote with a " $+$ " suffix the case of *RGB* and *OF* being used during testing. Following the earlier works [37,67,80] and we use 32 and 16 frames for S3D and R(2+1)D, respectively during the training of fine-tuning and linear-classification experiments.

### 4.1 Ablation study

In these experiments, we demonstrate the effectiveness of each proposed component in terms of recall values. All models pre-trained on the UCF training set and evaluated on the UCF and HMDB test sets with S3D backbone.

**Effect of view merging strategies** For a better understanding of the effectiveness of the proposed approach, we design 3 different baselines: **SView**, **Avg**, and **Sep**. **SView**: We train *OF* and *RGB* separately with identical but independent backbones, projection heads, and prototypes. This baseline can be considered a trivial extension of SwAV [11] to videos. **Avg**: We train *OF* and *RGB* jointly, by passing *OF* and *RGB* from different backbones and then feeding their average into the projection head and prototypes. **Sep**: We train the model jointly but this time *OF* and *RGB* information do not interact. We use separate backbones but share the projection head and prototypes. Our model differentiates from this only at the assignment computing stage. We train all our baselines by minimizing the loss described in Eq. (1). Finally, to better compare our method (**GOCA**) with the single-view (**SView**) baseline, we also evaluate it in single-view settings by only using one of the backbones during the test. To combine *RGB* and *OF* representations during the test, we simply take the average of the two.

Tab. 1 shows our results on the UCF and HMDB test sets. In single-view settings (first 6 rows), **SView** with *RGB* obtains very poor results, which can be attributed to an over-emphasis on the background scene and ignoring the motion, which is common for *RGB*-only models. Our results confirm that motion-only models can perform better than *RGB*, as observed in [37,80]. Furthermore, GOCA outperforms both baselines even in single-view testing, which confirms that the proposed joint training approach is beneficial for each individual backbone as well. We can also see that enforcing the similarity of the two views as in VICC does not improve *OF* results compared to single-

| | | | UCF | | HMDB | |
|---|---|---|---|---|---|---|
| Method | Train | Test | R@1 | R@5 | R@1 | R@5 |
| SView | R | R | 41.3 | 58.9 | 17.9 | 43.2 |
| SView | F | F | 62.0 | 75.8 | 28.2 | 53.3 |
| VICC [80] | R+F | R | 62.1 | 77.1 | 25.5 | 49.6 |
| VICC [80] | R+F | F | 59.7 | 77.3 | 27.7 | 53.3 |
| **GOCA** | R+F | R | 63.4 | 76.3 | 26.3 | 50.2 |
| **GOCA** | R+F | F | 67.9 | 79.5 | 31.8 | 56.3 |
| SView+ | - | R+F | 63.6 | 76.8 | 28.6 | 54.1 |
| Avg | R+F | R+F | 50.8 | 63.0 | 22.8 | 46.3 |
| Sep | R+F | R+F | 65.0 | 78.2 | 29.5 | 54.8 |
| **GOCA+** | R+F | R+F | 70.8 | 81.4 | 33.7 | 58.7 |

Table 1: Comparison of *RGB* (R) and *OF* (F) merging strategies

| | | $\lambda_1$ | | | |
|---|---|---|---|---|---|
| | | 0.01 | 0.02 | 0.03 | 0.04 |
| | 0.01 | 56.4 | 57.6 | 55.1 | 54.1 |
| $\lambda_2$ | 0.02 | 57.3 | 58.7 | **60.1** | 58.9 |
| | 0.03 | 58.1 | **60.9** | **60.5** | 57.5 |
| | 0.04 | 55.8 | 58.5 | **59.0** | 55.2 |

Table 2: The effect of lambda parameters on recall values on the UCF RGB-only val.

view training. In contrast, **GOCA** significantly improves the results for both views, by aligning the two representations in a smarter, more implicit manner.

When we combine *RGB* and *OF* during testing (last 4 rows), we observe that naively merging features at training (**Avg**) performs better than *RGB*-only by 3.5% while significantly worse than *OF*-only training (-11.2%). This might be due to the fact that during training, *RGB* information dominates the gradients, and the *OF* backbone can not be fully trained. The performance of **Sep** indicates that joint training for *RGB* and *OF* can improve the results when we do not naively merge features. Finally, the proposed model, **GOCA**, further improves the results, which verifies the efficacy of the proposed guided cluster assignment idea.

**Effect of $\lambda_1$ and $\lambda_2$.** Our next ablation study is observing the impact of the $\lambda_1$ and $\lambda_2$ parameter values in Eq. (6) and Eq. (7). These parameters control the effect of the uniformity assumption and prior distribution. Tab. 2 shows how recall at 1 varies depending on the $\lambda_1$ and $\lambda_2$ parameters. We train GOCA for 200 epochs and evaluate it on the UCF dataset. We can observe that the proposed approach is robust to $\lambda_1$ and $\lambda_2$, especially in the range of $[0.02, 0.03]$. For all other experiments, we set $\lambda_1 = 0.02$ and $\lambda_2 = 0.03$.

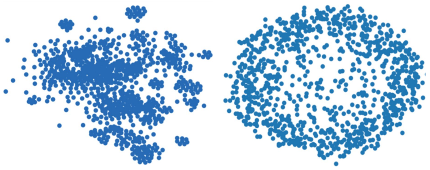

Fig. 3: The t-SNE plot of 1000 prototypes in 2D without regularization (left) and with regularization (right)

| | Use | | | UCF | | HMDB | |
|---|---|---|---|---|---|---|---|
| Method | ProtReg | Train | Test | R@1 | R@5 | R@1 | R@5 |
| SView | **No** | R | R | 40.9 | 57.4 | 17.1 | 42.7 |
| SView | **Yes** | R | R | 42.3 | 58.9 | 17.9 | 43.2 |
| **GOCA+** | **No** | R+F | R+F | 69.1 | 79.4 | 32.9 | 58.6 |
| **GOCA+** | **Yes** | R+F | R+F | 70.8 | 81.4 | 33.6 | 58.7 |

Table 3: Effect of prototype regularization on recall values

**Effect of prototype regularizer.** Our prototype regularizer idea guarantees that prototypes are maximally far apart from each other. Fig. 3 shows t-SNE plot of 1000 prototypes with and without regularization. As we can see, the proposed method locates the prototypes maximally far apart from each other, on the other hand, in the unregularized case, prototypes are quite closely grouped. As we discuss in Sec. 3.3, maximally far apart prototypes prevent the feature collapse and allow us stable training. Another benefit is that clusters formed around prototypes are also will be far apart from each other which is particularly important for retrieval tasks. We verify this benefit by comparing the recall accuracy with and without using prototype regularization. Tab. Tab. 3 shows the consistent effectiveness of the proposed method for both models on both datasets.

| | | | | | UCF | | | | HMDB | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Method | DS | Backbone | Res | Modl | R@1 | R@5 | R@10 | R@20 | R@1 | R@5 | R@10 | R@20 |
| Selavi [6] | K400 | R(2+1)D+18 | 112 | V+A | 52.0 | 68.6 | - | 84.5 | 24.8 | 47.6 | - | 75.5 |
| Rbst-xID [61] | K400 | R(2+1)D+18 | 112 | V+A | 60.9 | 79.4 | - | **90.8** | 30.8 | 55.8 | - | 79.7 |
| TCGL [55] | K400 | R(2+1)D+18 | 112 | V | 21.5 | 39.3 | 49.3 | 59.5 | 10.5 | 27.6 | 39.7 | 55.63 |
| MotionFit [33] | K400 | R(2+1)D+18 | 112 | V | 61.6 | 75.6 | - | 85.5 | 29.4 | 46.5 | - | 66.7 |
| ASCNet [40] | K400 | R3D-18 | 112 | V | 58.9 | 76.3 | 82.2 | 87.5 | - | - | - | - |
| Enhenced [67] | K400 | R3D-18 | 112 | V | 41.5 | 60.6 | 71.2 | 80.1 | 20.7 | 40.8 | 55.2 | 68.3 |
| MCN [52] | K400 | R(2+1)D-18 | 128 | V | 52.5 | 69.5 | 77.9 | 83.1 | 23.7 | 46.5 | 58.9 | 72.4 |
| Zhang [53] | K400 | R3D-18 | 112 | V | 46.7 | 63.1 | 69.7 | 78.0 | - | - | - | - |
| CoCLR [37] | K400 | S3D | 128 | V | 45.6 | 63.9 | 75.4 | 81.7 | - | - | - | - |
| **GOCA** | K400 | S3D | 128 | V | 67.3 | 79.1 | 84.9 | 89.9 | 32.7 | 55.1 | 68.5 | 79.5 |
| **GOCA+** | K400 | S3D | 128 | V | **68.6** | **80.7** | **86.6** | **90.1** | **33.2** | **56.3** | **68.5** | **80.2** |
| VCOP [86] | UCF | R(2+1)D+18 | 112 | V | 14.1 | 30.3 | 40.4 | 51.1 | 7.6 | 22.9 | 34.4 | 48.8 |
| Var. PSP [19] | UCF | R(2+1)D+18 | 112 | V | 24.6 | 41.9 | 51.3 | 62.7 | - | - | - | - |
| RTT [42] | UCF | R(2+1)D+18 | 112 | V | 26.1 | 48.5 | 59.1 | 69.6 | - | - | - | - |
| MemDPC [36] | UCF | R2D3D-18 | 112 | V | 20.2 | 40.4 | 52.4 | 64.7 | 7.7 | 25.7 | 40.6 | 57.7 |
| VCP [57] | UCF | R(2+1)D+18 | 112 | V | 19.9 | 33.7 | 42.0 | 50.5 | 6.7 | 21.3 | 32.7 | 49.2 |
| TCLR [23] | UCF | R(2+1)D+18 | 112 | V | 56.9 | 72.2 | 79.0 | 84.6 | 24.1 | 45.8 | 58.3 | 75.3 |
| Enhenced [67] | UCF | R3D-18 | 112 | V | 39.6 | 57.6 | 69.2 | 78.0 | 18.8 | 39.2 | 51.0 | 63.7 |
| **GOCA** | UCF | R(2+1)D+18 | 112 | V | 62.8 | 77.7 | 82.0 | **87.0** | 22.3 | 47.2 | 60.1 | 73.3 |
| **GOCA+** | UCF | R(2+1)D+18 | 112 | V | **63.4** | **78.6** | **82.5** | 86.2 | **28.5** | **54.4** | **66.2** | **76.6** |
| MCL+ [50] | UCF | S3D | 224 | V | 67.0 | 80.8 | 86.3 | 90.8 | 26.7 | 52.5 | 67.0 | 79.3 |
| Time-Equ* [41] | UCF | R3D-18 | 128 | V | 62.1 | - | - | - | 31.5 | - | - | - |
| CoCLR [37] | UCF | S3D | 128 | V | 53.3 | 69.4 | 76.6 | 82.0 | 23.2 | 43.2 | 53.5 | 65.5 |
| CoCLR+ [37] | UCF | S3D | 128 | V | 55.9 | 70.8 | 76.9 | 82.5 | 26.1 | 45.8 | 57.9 | 69.7 |
| ViCC [80] | UCF | S3D | 128 | V | 62.1 | 77.1 | 83.7 | 87.9 | 25.5 | 49.6 | 61.9 | 72.5 |
| ViCC+ [80] | UCF | S3D | 128 | V | 65.1 | 80.2 | **85.4** | **89.8** | 29.7 | 54.6 | 66.0 | 76.2 |
| **GOCA** | UCF | S3D | 128 | V | 63.4 | 76.3 | 81.3 | 86.5 | 26.3 | 50.2 | 62.6 | 77.0 |
| **GOCA+** | UCF | S3D | 128 | V | **70.8** | **81.4** | 85.3 | 89.5 | **33.7** | **58.5** | **70.0** | **80.6** |

Table 4: Video retrieval results on UCF [74] and HMDB51 [47] datasets. *DS*, *Res*, and *Modl* represent training dataset, input resolution, and input modality respectively. Rows above the double line are trained on Kinetics and the rest use UCF. Light grey colored methods (MCL [50]) use 2 times more input resolution (224). Time-Equ* [41] uses various additional loss functions that used in earlier works (speed [8,89], direction [66], order [86]). These loss functions can be combined with our loss functions as well.

**Comparison with the state-of-the-art** Given the large body of self-supervised video understanding works published thus far, we only selected recent (from 2019) publications. To the best of our ability, we conducted a fair comparison. However, we still observe small variations in terms of input resolution and fine-tuning details in the literature, which makes it extremely hard to perform perfectly fair comparisons. Furthermore, we encourage the readers to study the supplementary material for more in-depth experimental results. As noted in [29], we observe that during the *Fine-Tuning* experiment, backbone networks tend to overfit validation datasets (UCF and HMDB), therefore we perform extensive retrieval and cluster analysis experiments to show our contributions' influence (On these experiments there is no supervised training on validation datasets).

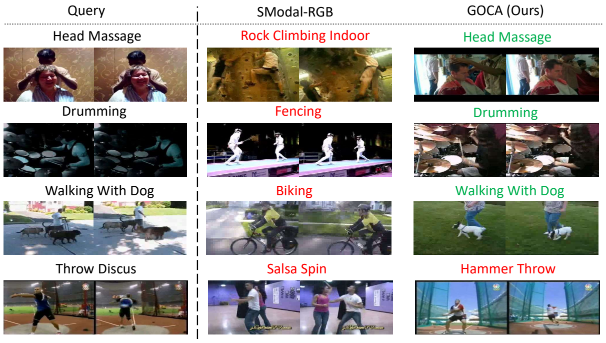

Fig. 4: Retrieval comparison between SView-RGB and GOCA representations on UCF dataset. Query videos (left) are selected from test set and top-1 nearest neighbour videos are retrieved from the training set. Ground truth labels are reported on top of each query.

| Method | R@1 | R@5 | R@10 | R@20 |
|---|---|---|---|---|
| CoCLR [37] | 6.7 | 23.6 | 37.4 | 55.2 |
| VICC [80] | 7.2 | 24.4 | 38.1 | 54.5 |
| **GOCA** | **8.1** | **24.8** | **38.3** | 55.1 |

Table 5: Retrieval results on Diving-48 [51] with S3D backbone(pre-trained on UCF)

| Method | Acc | NMI | F1 |
|---|---|---|---|
| SView-RGB | 42.1 | 65.1 | 40.3 |
| CoCLR [37] | 44.3 | 65.4 | 41.9 |
| CoCLR+ [37] | 46.1 | 68.6 | 43.1 |
| VICC [80] | 51.1 | 70.4 | 50.0 |
| VICC+ [80] | 51.9 | 72.9 | 53.7 |
| **GOCA** | 57.3 | 75.8 | 57.4 |
| **GOCA+** | **61.2** | **78.7** | **61.3** |

Table 6: Cluster quality results on UCF test set with S3D backbone. All models are trained on UCF training set.

### 4.2 Retrieval Results

Tab. 4 and Tab. 5 show the retrieval results for $Recall@K$ ($R@K$). In this experiment, we follow the standard protocol defined in [23,36,37,42,57,67,80] and perform the evaluation on the frozen features that were computed from a pre-trained model. At Tab. Tab. 4, the first part of the table (above the double line) includes models that are all pre-trained on K400 [45], while evaluated on the UCF and HMDB datasets. We can see a significant improvement (5% and 8% on UCF and HMDB respectively, averaged over all $K$ values). Surprisingly, the proposed approach even outperforms models [6,61] that use an additional view (Audio) by a large margin. Below the double line, we show models that are pre-trained on the UCF training set, using two different backbone architectures, S3D and $R(2+1)D$, separated by a horizontal line. On the S3D, we achieve an improvement of 5.7% on UCF and 4.0% on HMDB at $K = 1$. On the $R(2+1)D$, we can see a nearly 6% increase across all recall values for UCF and 5% for HMDB. Our *RGB*-only model also achieves state-of-the-art results and outperforms baselines by a large margin on both datasets for both backbones. These findings verify our intuition that effectively utilizing OF can result in strong spatio-temporal representations.

| Method | DS | Backbone | Res | UCF | HMDB |
|---|---|---|---|---|---|
| CVRL [68] | K400 | R3D-50 | 224 | 92.9 | 67.9 |
| MoCo [29] | K400 | R-50 | 224 | 93.2 | 70.6 |
| ASCNet[67] | K400 | R3D-18 | 112 | 80.5 | 52.3 |
| Enhanced[67] | K400 | R3D-18 | 112 | 79.1 | 47.6 |
| CoCLR [37] | K400 | S3D | 128 | 87.9 | 54.6 |
| CoCLR+ [37] | K400 | S3D | 128 | 90.6 | 62.9 |
| **GOCA** | K400 | S3D | 128 | 89.3 | 63.2 |
| **GOCA+** | K400 | S3D | 128 | **91.1** | **65.8** |
| MoCo* [29] | UCF | R(2+1)D+18 | 112 | 77.6 | 45.7 |
| CVRL* [68] | UCF | R3D+18 | 112 | 75.8 | 44.6 |
| TCLR [23] | UCF | R(2+1)D+18 | 112 | 82.4 | 52.9 |
| TCLR [23] | UCF | R(2+1)D+18 | 112 | 82.8 | 55.6 |
| Var. PSP [19] | UCF | R3D+18 | 112 | 74.8 | 36.8 |
| PacePred [83] | UCF | R(2+1)D+18 | 112 | 75.9 | 35.9 |
| VCP [57] | UCF | R(2+1)D+18 | 112 | 66.3 | 32.2 |
| PRP [89] | UCF1 | R(2+1)D+18 | 112 | 72.1 | 35.0 |
| RTT [42] | UCF | R(2+1)D+18 | 112 | 81.6 | 46.4 |
| Enhanced [67] | UCF | R3D+18 | 112 | 76.2 | 41.1 |
| Zhang [53] | UCF | R(2+1)D+18 | 112 | 79.0 | 45.4 |
| **GOCA** | UCF | R(2+1)D+18 | 112 | 82.1 | 54.7 |
| **GOCA+** | UCF | R(2+1)D+18 | 112 | **88.7** | **60.1** |
| CoCLR [37] | UCF | S3D | 128 | 81.4 | 52.1 |
| CoCLR+ [37] | UCF | S3D | 128 | 87.3 | 58.7 |
| ViCC [80] | UCF | S3D | 128 | 84.3 | 47.9 |
| ViCC+ [80] | UCF | S3D | 128 | **90.5** | 62.2 |
| **GOCA** | UCF | S3D | 128 | 83.4 | 53.5 |
| **GOCA+** | UCF | S3D | 128 | 90.2 | **64.8** |

Table 7: Fine Tuning. Light grey colored methods use much higher input resolution. MoCo* and CVRL* results obtained from [53] and [23], respectivelly.

| Method | DS | Backbone | Res | UCF | HMDB |
|---|---|---|---|---|---|
| MemDPC [36] | K400 | R-2D3D | 224 | 54.1 | 30.5 |
| CVRL [68] | K400 | R3D-50 | 224 | 89.8 | 58.3 |
| TCLR [23] | UCF | R3D+18 | 112 | 69.9 | 52.8 |
| Enhanced[67] | K400 | R3D+18 | 112 | 63.2 | 33.4 |
| CoCLR [37] | K400 | S3D | 128 | 74.5 | 46.1 |
| CoCLR+ [37] | K400 | S3D | 128 | 77.8 | 52.4 |
| **GOCA** | K400 | S3D | 128 | 78.9 | 50.5 |
| **GOCA+** | K400 | S3D | 128 | **82.8** | **58.7** |
| MCL+ [50] | UCF | S3D | 224 | 79.8 | - |
| Time-Equ [41] | UCF | R3D-18 | 128 | 74.1 | 47.5 |
| CoCLR [37] | UCF | S3D | 128 | 70.2 | 39.1 |
| CoCLR+ [37] | UCF | S3D | 128 | 72.1 | 40.2 |
| ViCC [80] | UCF | S3D | 128 | 72.2 | 38.5 |
| ViCC+ [80] | UCF | S3D | 128 | 78.0 | 47.9 |
| **GOCA** | UCF | S3D | 128 | 69.2 | 38.6 |
| **GOCA+** | UCF | S3D | 128 | **81.1** | **50.0** |

Table 8: Linear Classification.

We observe that Time-Equ* [41] method also performs well however, this work uses speed [8,89], direction [66], and order [86] as auxiliary loss as well. These loss functions can be combined with our method also. Time-Equ* [41] obtains 52.1% and 21.4% at at $K = 1$ on UCF and HMDB respectively, without these auxiliary loss function. We also evaluate our pre-trained model on motion centric dataset [51]. Tab. 5 show our result for Diving-48 [51]. We can see that proposed model improves the baselines for the motion centric dataset as well and obtain SOTA results. Fig. 4 shows our retrievals results. GOCA retrieves videos from the same semantic categories and it fails only for one case where it confuses *Disk Throwing* with *Hammer Throwing*. This is a quite hard example because these two activities have very similar motions and the only difference is the quite small object in the person's hand. Higher resolution input images would help to solve this problem. We can see that SModel-RGB fails to retrieve relevant videos and consistently retrieves based on the background.

### 4.3 Cluster Analysis

Ideally, video representations should form semantic clusters in order to facilitate activity recognition. To evaluate the clustering quality of our learned representations, we use two metrics that measure the correlation between clusters and semantic labels (ground truth). To this end, we perform K-means clustering on the our representations that are extracted from the UCF101 test set. Then we use the given labels of the test set to determine a label for each cluster via majority voting. We assign each cluster's label to all its members and compare those assigned labels to given labels. Specifically, we compute the accuracy of the assigned labels (Acc), as well as the $F_1$ score, which is the harmonic mean of recall and precision. We also compute the Normalized Mutual Information (NMI), which measures the mutual information between clusters and ground truth labels, divided by the sum of their entropy. Due to the randomness of K-means, we repeat experiments 50 times and take the average for each metric. As shown in Tab. 6, GOCA significantly improves the cluster quality in terms of all metrics, both with and without *OF*. This verifies the high quality semantic clustering ability of our method.

### 4.4 Classification Tasks

*Linear Classification* We follow the earlier works of [37,67,80] for the linear classification experiments. After the self-supervised training on the K400 dataset, we discard the projection head and prototypes and replace them with a linear layer. Then we train the linear layer on the training set of each downstream dataset (UCF and HMDB) with frozen backbone. The results are shown in the first section of Tab. 7, and demonstrate that the proposed model significantly outperforms CoCLR [37] on both datasets by 5% and 6%, respectively. Notably, when we combine *RGB* and *OF*, we achieve state-of-the-art results on the HMDB dataset, even though the other methods [36,68] benefit from a higher input resolution. For the case of UCF dataset, our model with the S3D marginally outperforms the other methods, achieving 80.1% on UCF and 50.0% on HMDB.

*Fine-Tuning* We follow the standard protocol from [37,67,80], where we train the full backbone on the downstream tasks. We summarize the results in the second section of Tab. 8. On the S3D backbone, our method improves CoCLR+ [37] by 0.5% and 2.9%, respectively. In addition, when using only *RGB* as input, we achieve improvements of 1.4% and 8.8% on both datasets. Moving to the UCF dataset, our approach obtains state-of-art results with R(2+1)D; while, with S3D, we are slightly worse than ViCC+ [80]. However, on the HMDB dataset, our results outperform ViCC+ [80] by a large margin of 6.6% and 2.6% when using *RGB* and *RGB+OF*. Note that CVRL [68] and $\rho$ BYOL [29] use higher input resolution (224) than ours (112), which leads to better performance, but needs significantly more computation. In fact, when we compare to CVRL at a resolution of 112, we significantly outperform CVRL [68].

## 5 Conclusion

In this paper, we presented a novel self-supervised learning approach for videos. We showed that the proposed guided online clustering idea and the prototype regularization approach both substantially improve the performance of our learned representations on both activity retrieval and action classification tasks. We believe that our work establishes a new direction for SSL research on multi-view and multi-modal data. Although we conduct our experiments using *RGB* and optical flow, the proposed idea can be applied to fuse other modalities such as *RGB+Audio* or *RGB+Text*, which we will explore in our future work. Furthermore, our method simplifies the training procedure for multi-modal SSL on videos (CoCLR and VICC require multi-stage training).

## References

1. Ahsan, U., Sun, C., Essa, I.: Discrimnet: Semi-supervised action recognition from videos using generative adversarial networks. arXiv preprint arXiv:1801.07230 (2018)
2. Akbari, H., Yuan, L., Qian, R., Chuang, W.H., Chang, S.F., Cui, Y., Gong, B.: Vatt: Transformers for multimodal self-supervised learning from raw video, audio and text. Advances in Neural Information Processing Systems (2021)
3. Alayrac, J.B., Recasens, A., Schneider, R., Arandjelovic, R., Ramapuram, J., De Fauw, J., Smaira, L., Dieleman, S., Zisserman, A.: Self-supervised multimodal versatile networks. NeurIPS **2**(6), 7 (2020)
4. Alwassel, H., Mahajan, D., Korbar, B., Torresani, L., Ghanem, B., Tran, D.: Self-supervised learning by cross-modal audio-video clustering. Advances in Neural Information Processing Systems **33** (2020)
5. Alwassel, H., Mahajan, D., Korbar, B., Torresani, L., Ghanem, B., Tran, D.: Self-supervised learning by cross-modal audio-video clustering. In: Advances in Neural Information Processing Systems (NeurIPS) (2020)
6. Asano, Y.M., Patrick, M., Rupprecht, C., Vedaldi, A.: Labelling unlabelled videos from scratch with multi-modal self-supervision. Advances in Neural Information Processing Systems (2020)
7. Asano, Y.M., Rupprecht, C., Vedaldi, A.: Self-labelling via simultaneous clustering and representation learning. In: International Conference on Learning Representations (ICLR) (2020)
8. Benaim, S., Ephrat, A., Lang, O., Mosseri, I., Freeman, W.T., Rubinstein, M., Irani, M., Dekel, T.: Speednet: Learning the speediness in videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9922–9931 (2020)
9. Cai, T., Gao, R., Lee, J.D., Lei, Q.: A theory of label propagation for subpopulation shift. arXiv preprint arXiv:2102.11203 (2021)
10. Caron, M., Bojanowski, P., Joulin, A., Douze, M.: Deep clustering for unsupervised learning of visual features. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 132–149 (2018)
11. Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A.: Unsupervised learning of visual features by contrasting cluster assignments. Advances in neural information processing systems (2020)
12. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6299–6308 (2017)
13. Chao, Y.W., Vijayanarasimhan, S., Seybold, B., Ross, D.A., Deng, J., Sukthankar, R.: Rethinking the faster r-cnn architecture for temporal action localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1130–1139 (2018)
14. Chen, S., Tian, Y., Wen, F., Xu, Y., Tang, X.: Easytoon: an easy and quick tool to personalize a cartoon storyboard using family photo album. In: El-Saddik, A., Vuong, S., Griwodz, C., Bimbo, A.D., Candan, K.S., Jaimes, A. (eds.) Proceedings of the 16th International Conference on Multimedia 2008, Vancouver, British Columbia, Canada, October 26-31, 2008. pp. 499–508. ACM (2008). https://doi.org/10.1145/1459359.1459426, `https://doi.org/10.1145/1459359.1459426`
15. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International conference on machine learning. pp. 1597–1607. PMLR (2020)
16. Chen, X., He, K.: Exploring simple siamese representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15750–15758 (2021)

17. Chen, X., Xie, S., He, K.: An empirical study of training self-supervised vision transformers. arXiv preprint arXiv:2104.02057 (2021)
18. Cho, H., Kim, T., Chang, H.J., Hwang, W.: Self-supervised spatio-temporal representation learning using variable playback speed prediction. arXiv preprint arXiv:2003.02692 (2020)
19. Cho, H., Kim, T., Chang, H.J., Hwang, W.: Self-supervised visual learning by variable playback speeds prediction of a video. IEEE Access (2021)
20. Coskun, H., Zia, Z., Tekin, B., Bogo, F., Navab, N., Tombari, F., Sawhney, H.: Domain-specific priors and meta learning for low-shot first-person action recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence (2021)
21. Cuturi, M.: Sinkhorn distances: Lightspeed computation of optimal transport. Advances in neural information processing systems **26**, 2292–2300 (2013)
22. Damen, D., Doughty, H., Farinella, G.M., Fidler, S., Furnari, A., Kazakos, E., Moltisanti, D., Munro, J., Perrett, T., Price, W., et al.: Scaling egocentric vision: The epic-kitchens dataset. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 720–736 (2018)
23. Dave, I., Gupta, R., Rizve, M.N., Shah, M.: Tclr: Temporal contrastive learning for video representation. arXiv preprint arXiv:2101.07974 (2021)
24. Diba, A., Sharma, V., Gool, L.V., Stiefelhagen, R.: Dynamonet: Dynamic action and motion network. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6192–6201 (2019)
25. Doersch, C., Gupta, A., Efros, A.A.: Unsupervised visual representation learning by context prediction. In: Proceedings of the IEEE international conference on computer vision. pp. 1422–1430 (2015)
26. Dosovitskiy, A., Fischer, P., Springenberg, J.T., Riedmiller, M., Brox, T.: Discriminative unsupervised feature learning with exemplar convolutional neural networks. IEEE transactions on pattern analysis and machine intelligence **38**(9), 1734–1747 (2015)
27. Erlander, S., Stewart, N.F.: The gravity model in transportation analysis: theory and extensions, vol. 3. Vsp (1990)
28. Feichtenhofer, C., Fan, H., Malik, J., He, K.: Slowfast networks for video recognition. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 6202–6211 (2019)
29. Feichtenhofer, C., Fan, H., Xiong, B., Girshick, R., He, K.: A large-scale study on unsupervised spatiotemporal representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3299–3309 (2021)
30. Feichtenhofer, C., Pinz, A., Zisserman, A.: Convolutional two-stream network fusion for video action recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1933–1941 (2016)
31. Fernando, B., Bilen, H., Gavves, E., Gould, S.: Self-supervised video representation learning with odd-one-out networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3636–3645 (2017)
32. Gao, J., Chen, K., Nevatia, R.: Ctap: Complementary temporal action proposal generation. In: Proceedings of the European conference on computer vision (ECCV). pp. 68–83 (2018)
33. Gavrilyuk, K., Jain, M., Karmanov, I., Snoek, C.G.M.: Motion-augmented self-training for video recognition at smaller scale. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 10429–10438 (October 2021)
34. Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., Piot, B., kavukcuoglu, k., Munos, R., Valko, M.: Bootstrap your own latent - a new approach to self-supervised learning. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 21271–21284. Curran As-

sociates, Inc. (2020), https://proceedings.neurips.cc/paper/2020/file/f3ada80d5c4ee70142b17b8192b2958e-Paper.pdf

35. Gu, C., Sun, C., Ross, D.A., Vondrick, C., Pantofaru, C., Li, Y., Vijayanarasimhan, S., Toderici, G., Ricco, S., Sukthankar, R., et al.: Ava: A video dataset of spatio-temporally localized atomic visual actions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6047–6056 (2018)
36. Han, T., Xie, W., Zisserman, A.: Memory-augmented dense predictive coding for video representation learning. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16. pp. 312–329. Springer (2020)
37. Han, T., Xie, W., Zisserman, A.: Self-supervised co-training for video representation learning. In: Neurips (2020)
38. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9729–9738 (2020)
39. Hu, K., Shao, J., Liu, Y., Raj, B., Savvides, M., Shen, Z.: Contrast and order representations for video self-supervised learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 7939–7949 (October 2021)
40. Huang, D., Wu, W., Hu, W., Liu, X., He, D., Wu, Z., Wu, X., Tan, M., Ding, E.: Ascnet: Self-supervised video representation learning with appearance-speed consistency. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 8096–8105 (October 2021)
41. Jenni, S., Jin, H.: Time-equivariant contrastive video representation learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9970–9980 (2021)
42. Jenni, S., Meishvili, G., Favaro, P.: Video representation learning by recognizing temporal transformations. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVIII 16. pp. 425–442. Springer (2020)
43. Jiang, Y.G., Dai, Q., Liu, W., Xue, X., Ngo, C.W.: Human action recognition in unconstrained videos by explicit motion modeling. IEEE Transactions on Image Processing **24**(11), 3781–3795 (2015)
44. Johansson, G.: Visual perception of biological motion and a model for its analysis. Perception & psychophysics **14**(2), 201–211 (1973)
45. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al.: The kinetics human action video dataset. arXiv preprint arXiv:1705.06950 (2017)
46. Korbar, B., Tran, D., Torresani, L.: Cooperative learning of audio and video models from self-supervised synchronization. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 31. Curran Associates, Inc. (2018), https://proceedings.neurips.cc/paper/2018/file/c4616f5a24a66668f11ca4fa80525dc4-Paper.pdf
47. Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., Serre, T.: Hmdb: A large video database for human motion recognition. In: 2011 International Conference on Computer Vision. pp. 2556–2563 (2011). https://doi.org/10.1109/ICCV.2011.6126543
48. Larsson, G., Maire, M., Shakhnarovich, G.: Learning representations for automatic colorization. In: European conference on computer vision. pp. 577–593. Springer (2016)
49. Lee, H.Y., Huang, J.B., Singh, M., Yang, M.H.: Unsupervised representation learning by sorting sequences. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 667–676 (2017)
50. Li, R., Zhang, Y., Qiu, Z., Yao, T., Liu, D., Mei, T.: Motion-focused contrastive learning of video representations. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2105–2114 (2021)

51. Li, Y., Li, Y., Vasconcelos, N.: Resound: Towards action recognition without representation bias. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 513–528 (2018)
52. Lin, Y., Guo, X., Lu, Y.: Self-supervised video representation learning with meta-contrastive network. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8239–8249 (2021)
53. Lin, Z., Qi, S., Zhengyang, S., Changhu, W.: Inter-intra variant dual representations forself-supervised video recognition. BMVC (2021)
54. Liu, J., Chen, C., Zhu, Y., Liu, W., Metaxas, D.N.: Video classification via weakly supervised sequence modeling. Computer Vision and Image Understanding **152**, 79–87 (2016)
55. Liu, Y., Wang, K., Liu, L., Lan, H., Lin, L.: Tcgl: Temporal contrastive graph for self-supervised video representation learning. IEEE Transactions on Image Processing **31**, 1978–1993 (2022). https://doi.org/10.1109/TIP.2022.3147032
56. Liu, Y., Ma, L., Zhang, Y., Liu, W., Chang, S.F.: Multi-granularity generator for temporal action proposal. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3604–3613 (2019)
57. Luo, D., Liu, C., Zhou, Y., Yang, D., Ma, C., Ye, Q., Wang, W.: Video cloze procedure for self-supervised spatio-temporal learning. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 11701–11708 (2020)
58. Mettes, P., van der Pol, E., Snoek, C.: Hyperspherical prototype networks. Advances in Neural Information Processing Systems **32**, 1487–1497 (2019)
59. Miech, A., Alayrac, J.B., Smaira, L., Laptev, I., Sivic, J., Zisserman, A.: End-to-end learning of visual representations from uncurated instructional videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9879–9889 (2020)
60. Misra, I., Zitnick, C.L., Hebert, M.: Shuffle and learn: unsupervised learning using temporal order verification. In: European Conference on Computer Vision. pp. 527–544. Springer (2016)
61. Morgado, P., Misra, I., Vasconcelos, N.: Robust audio-visual instance discrimination. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12934–12945 (2021)
62. Noroozi, M., Favaro, P.: Unsupervised learning of visual representations by solving jigsaw puzzles. In: European conference on computer vision. pp. 69–84. Springer (2016)
63. Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018)
64. Pan, T., Song, Y., Yang, T., Jiang, W., Liu, W.: Videomoco: Contrastive video representation learning with temporally adversarial examples. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11205–11214 (2021)
65. Patrick, M., Asano, Y.M., Kuznetsova, P., Fong, R., Henriques, J.F., Zweig, G., Vedaldi, A.: Multi-modal self-supervision from generalized data transformations (2021)
66. Pickup, L.C., Pan, Z., Wei, D., Shih, Y., Zhang, C., Zisserman, A., Scholkopf, B., Freeman, W.T.: Seeing the arrow of time. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2035–2042 (2014)
67. Qian, R., Li, Y., Liu, H., See, J., Ding, S., Liu, X., Li, D., Lin, W.: Enhancing self-supervised video representation learning via multi-level feature optimization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 7990–8001 (October 2021)
68. Qian, R., Meng, T., Gong, B., Yang, M.H., Wang, H., Belongie, S., Cui, Y.: Spatiotemporal contrastive video representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6964–6974 (2021)

69. Regatti, J.R., Deshmukh, A.A., Manavoglu, E., Dogan, U.: Consensus clustering with unsupervised representation learning. In: 2021 International Joint Conference on Neural Networks (IJCNN). pp. 1–9. IEEE (2021)
70. Richemond, P.H., Grill, J.B., Altché, F., Tallec, C., Strub, F., Brock, A., Smith, S.L., De, S., Pascanu, R., Piot, B., Valko, M.: Byol works even without batch statistics. CoRR **abs/2010.10241** (2020), `https://arxiv.org/abs/2010.10241`
71. Sevilla-Lara, L., Liao, Y., Güney, F., Jampani, V., Geiger, A., Black, M.J.: On the integration of optical flow and action recognition. In: German Conference on Pattern Recognition. pp. 281–297. Springer (2018)
72. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: Advances in Neural Information Processing Systems. pp. 568–576 (2014)
73. Sinkhorn, R.: Diagonal equivalence to matrices with prescribed row and column sums. The American Mathematical Monthly **74**(4), 402–405 (1967)
74. Soomro, K., Zamir, A.R., Shah, M.: UCF101: A dataset of 101 human actions classes from videos in the wild. CoRR **abs/1212.0402** (2012), `http://arxiv.org/abs/1212.0402`
75. Srivastava, N., Mansimov, E., Salakhudinov, R.: Unsupervised learning of video representations using lstms. In: International conference on machine learning. pp. 843–852. PMLR (2015)
76. Sun, C., Baradel, F., Murphy, K., Schmid, C.: Learning video representations using contrastive bidirectional transformer. arXiv preprint arXiv:1906.05743 (2019)
77. Sun, C., Myers, A., Vondrick, C., Murphy, K., Schmid, C.: Videobert: A joint model for video and language representation learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7464–7473 (2019)
78. Tammes, P.M.L.: On the origin of number and arrangement of the places of exit on the surface of pollen-grains. Recueil des travaux botaniques néerlandais **27**(1), 1–84 (1930)
79. Tian, Y., Krishnan, D., Isola, P.: Contrastive multiview coding. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16. pp. 776–794. Springer (2020)
80. Toering, M., Gatopoulos, I., Stol, M., Hu, V.T.: Self-supervised video representation learning with cross-stream prototypical contrasting. Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) (January 2022)
81. Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., Paluri, M.: A closer look at spatiotemporal convolutions for action recognition. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 6450–6459 (2018)
82. Vondrick, C., Pirsiavash, H., Torralba, A.: Generating videos with scene dynamics. Advances in neural information processing systems **29**, 613–621 (2016)
83. Wang, J., Jiao, J., Liu, Y.H.: Self-supervised video representation learning by pace prediction. In: European conference on computer vision. pp. 504–521. Springer (2020)
84. Wei, D., Lim, J.J., Zisserman, A., Freeman, W.T.: Learning and using the arrow of time. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8052–8060 (2018)
85. Xie, S., Sun, C., Huang, J., Tu, Z., Murphy, K.: Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In: Proceedings of the European conference on computer vision (ECCV). pp. 305–321 (2018)
86. Xu, D., Xiao, J., Zhao, Z., Shao, J., Xie, D., Zhuang, Y.: Self-supervised spatiotemporal learning via video clip order prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10334–10343 (2019)
87. Yang, J., Parikh, D., Batra, D.: Joint unsupervised learning of deep representations and image clusters. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5147–5156 (2016)

88. Yao, Y., Liu, C., Luo, D., Zhou, Y., Ye, Q.: Video playback rate perception for self-supervised spatio-temporal representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6548–6557 (2020)
89. Yao, Y., Liu, C., Luo, D., Zhou, Y., Ye, Q.: Video playback rate perception for self-supervised spatio-temporal representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2020)
90. YM., A., C., R., A., V.: Self-labelling via simultaneous clustering and representation learning. In: International Conference on Learning Representations (2020), `https://openreview.net/forum?id=Hyx-jyBFPr`
91. You, Y., Gitman, I., Ginsburg, B.: Large batch training of convolutional networks. arXiv preprint arXiv:1708.03888 (2017)
92. Zach, C., Pock, T., Bischof, H.: A duality based approach for realtime tv-l 1 optical flow. In: Joint pattern recognition symposium. pp. 214–223. Springer (2007)
93. Zbontar, J., Jing, L., Misra, I., LeCun, Y., Deny, S.: Barlow twins: Self-supervised learning via redundancy reduction. In: ICML (2021)
94. Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: European conference on computer vision. pp. 649–666. Springer (2016)
95. Zhao, Y., Xiong, Y., Wang, L., Wu, Z., Tang, X., Lin, D.: Temporal action detection with structured segment networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2914–2923 (2017)
96. Zhuang, C., Zhai, A.L., Yamins, D.: Local aggregation for unsupervised learning of visual embeddings. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6002–6012 (2019)