**TimeBalance: Temporally-Invariant and Temporally-Distinctive Video Representations for Semi-Supervised Action Recognition**

Ishan Rajendrakumar Dave, Mamshad Nayeem Rizve, Chen Chen, Mubarak Shah  
Center for Research in Computer Vision, University of Central Florida, Orlando, USA  
{ishandave, nayeemrizve}@knights.ucf.edu, {chen.chen, shah}@crcv.ucf.edu

**Abstract**

Semi-Supervised Learning can be more beneficial for the video domain compared to images because of its higher annotation cost and dimensionality. Besides, any video understanding task requires reasoning over both spatial and temporal dimensions. In order to learn both the static and motion related features for the semi-supervised action recognition task, existing methods rely on hard input inductive biases like using two-modalities (RGB and Optical-flow) or two-stream of different playback rates. Instead of utilizing unlabeled videos through diverse input streams, we rely on self-supervised video representations, particularly, we utilize temporally-invariant and temporally-distinctive representations. We observe that these representations complement each other depending on the nature of the action. Based on this observation, we propose a student-teacher semi-supervised learning framework, TimeBalance, where we distill the knowledge from a temporally-invariant and a temporally-distinctive teacher. Depending on the nature of the unlabeled video, we dynamically combine the knowledge of these two teachers based on a novel temporal similarity-based reweighting scheme. Our method achieves state-of-the-art performance on three action recognition benchmarks: UCF101, HMDB51, and Kinetics400. Code: https://github.com/DAVEISHAN/TimeBalance.

1. Introduction

Recent development in action recognition have opened up a wide range of real-world applications: visual security systems [19, 50, 56], behavioral studies [35], sports analytics [46], elderly person fall detection systems [10, 49, 85], etc. Most of these developments are mainly courtesy of large-scale curated datasets like Kinetics [12], HVU [21], and HACS [87]. However, labeling such a massive video dataset requires an enormous amount of annotation time and human effort. At the same time, there is a vast amount of unlabeled videos available on the internet. The goal of semi-supervised action recognition is to use such large-scale unlabeled dataset to provide additional supervision along with the labeled supervision of the small-scale dataset.

Semi-supervised learning for image classification has seen tremendous progress in recent years [1, 57, 63, 68]. In semi-supervised action recognition, recent approaches have adapted these image-based methods by incorporating motion-related inductive biases into the setup. For instance, some methods [77, 79] use two different input modalities where the original RGB video promotes learning appearance-based features while optical flow/temporal gradients promotes learning of motion-centric features. Another set of methods uses input streams of different sampling rates to achieve this [64, 70]. Although these input-level inductive biases are simple-yet-very-effective to provide unlabeled supervision for action recognition, they are not suitable for large-scale datasets due to their multiplicative storage requirement and high preprocessing overhead.

Contrastive Self-supervised Learning (CSL) has emerged as a powerful technique to learn meaningful representations from unlabeled videos. Existing video CSL methods deal with mainly two different kinds of objectives: (1) Learning similarities across clips of the same video i.e temporally-invariant representations [29, 53, 55] (2) Learning differences across clips of the video i.e. temporally-distinctive representations [18, 36, 75]. Each objective has its own advantages, depending on the nature of the unlabeled videos being used.

Our experiments reveal a clear difference in the class-wise performance of both methods, as illustrated in Fig. 1. The right half of the figure shows the action classes where the temporally-invariant model is dominant. We can observe that all such action classes are atomic actions with high repetitions, e.g., Fencing, Knitting. Any two clips from such videos are highly similar, hence, increasing agreement between them i.e. learning temporal-invariant representation is more meaningful. The left half of the figure shows action classes where temporally-distinctive rep-
Classwise performance difference for action recognition task

(Temporally-Distinctive model) - (Temporally-Invariant model)

Temporally-Distinctive Representations perform better

Figure 1. Motivation for Temporally-Distinctive and Temporally-Invariant Representations. In order to leverage the unlabeled videos effectively, we consider two kinds of self-supervised video representation learning techniques with complementary goals: (1) Temporally Invariant Representations (Bottom Right) encourage learning the commonalities of the clips, hence it mainly focuses on learning features related to highly frequent repetitions and appearance. (2) Temporally Distinctive Representations (Bottom Left) encourage learning the dissimilarities between clips of the same video, hence it encourages learning features for sub-actions within the video. The plot shows the activity-wise UCF101 performance difference of finetuned models which were self-supervised pretrained with temporally-distinctive and temporally-invariant objectives. The plot shows extreme 25-25 classes after sorting the all classwise differences.

Based on our observation, we aim to leverage the strengths of both temporally-invariant and temporally-distinctive representations for semi-supervised action recognition. To achieve this, we propose a semi-supervised framework based on a student-teacher setup. The teacher supervision includes two models pre-trained using CSL with temporally-invariant and temporally-distinctive objectives. After pre-training, the teachers are fine-tuned with the labeled set to adapt to the semi-supervised training of the student. During semi-supervised training, we weigh each teacher model based on the nature of the unlabeled video instance. We determine the nature of the instance by computing its similarity score using the temporal self-similarity matrices of both teachers. This way, the student is trained using the labeled supervision from the labeled set and the unlabeled supervision from the weighted average of the teachers. It is worth noting that our framework doesn’t depend on complicated data-augmentation schemes like FixMatch [65].

The contributions of this work are summarized as follows:

• We propose a student-teacher-based semi-supervised learning framework that consists of two teachers with complementary self-supervised video representations: Temporally-Invariant and Temporally-Distinctive.

• In order to leverage the strength of each representation (invariant or distinctive), we weigh a suitable teacher according to an unlabeled video instance. We achieve it by the proposed temporal-similarity-based reweighting scheme.

• Our method outperforms existing approaches and achieves state-of-the-art results on popular action recognition benchmarks, including UCF101, HMDB51, and Kinetics400.

2. Prior Work

Semi-supervised learning in images Semi-supervised learning is one of the fundamental approaches to learning from limited labeled data [16, 26, 40, 41, 48, 54, 58]. There are two common approaches for semi-supervised learning are consistency regularization [43, 52, 61, 68, 78] and pseudo-labeling [1, 44, 57, 59, 63, 84]. The consistency regularization methods attempt to achieve perturbation invariant output space. To this end, these methods try to minimize a distance/divergence-based loss as a measure of
consistency between two differently perturbed/augmented versions of an image. Pseudo-labeling-based methods on the other hand promote entropy minimization to improve by performing self-training on the confident pseudo-labels generated from the network. Hybrid methods [7,8,65] combine consistency regularization and pseudo-labeling techniques to obtain further improvement. Some of the recent works [3, 16, 83] have also shown the effectiveness of self-supervised representations in solving this task. Our work is somewhat in the spirit of this last set of works as we also leverage self-supervised representation learning to obtain temporally distinct and invariant features.

**Semi-supervised learning in videos** Although there is a tremendous recent development in action recognition [2, 6, 9, 13, 20, 22, 32, 47, 51, 60, 72, 74, 86], semi-supervised learning in videos is not explored as in the image domain. CMPL [80] utilizes a FixMatch framework, where they study the effect of model capacity to provide complementary gains of unlabeled videos. They provide empirical evidence that the smaller model is responsible for learning the motion features whereas, the bigger model learns more appearance-biased features. However, defining smaller and bigger models is very relative, and the observation may not hold true for different architectures. Some prior work injects temporal dynamics to their semi-supervised framework by additional input modality to RGB videos like temporal gradients [77], optical-flow [79], or P-frames [69]. Another set of methods utilizes consistency loss between the slow and fast input streams of the video to leverage the unlabeled videos [64, 70]. We can see that the prior works rely on hard-inuctive biases like model architectures, input modality, or input sampling to learn temporal-invariance and distinctiveness. On the other hand, we do not have such a hard design choice; we leverage the unlabeled videos by the nature of the video instance using two complementary self-supervised teachers.

**Self-supervised learning (SSL)** In recent years, self-supervised learning has demonstrated learning powerful representations for images [11, 15, 30, 33, 34, 82] and videos [18, 23, 25, 36, 37, 55, 62, 71]. Although some works [16,83] have exploited self-supervised representation in semi-supervised image classification, Video SSL is not explored yet for semi-supervised action recognition.

Video SSL methods can be grouped mainly into two categories: the first set of methods focuses on learning temporal invariance [25,53,55]. These methods are a simple-yet-effective extension of instance-discrimination-based methods like SimCLR [15], MoCo [34], etc., where mutual information between two views of an instance is maximized. For videos, two views are two clips from different timestamps, hence it introduces temporal invariance in the learned representations. The second set of methods focuses on learning the temporal distinctiveness, where they try to learn different representations for different clips through contrastive loss [18,67,75,81] or through different temporal pretext transformations [5,14,17,36,37,76].

TCLR [18] introduces temporal contrastive losses for both temporally pooled and unpooled features to learn the temporal distinctiveness. In our method, we utilized these losses to learn the temporal distinctive teacher.

### 3. Method

Let’s consider a small labeled set of videos $D_t = \{(v^{(i)}, y^{(i)})\}_{i=1}^{N_t}$, where $v^{(i)}$ and $y^{(i)}$ denote $i$th video instance and its associated action label and $N_t$ is the number of total instances in the dataset. We also have access to a unlabeled dataset $D_u = \{v^{(i)}\}_{i=1}^{N_u}$, where $N_u$ is the total number of unlabeled videos and $N_u \gg N_t$. The objective of the semi-supervised action recognition is to utilize both labeled and unlabeled sets ($D_t$ and $D_u$) to improve the action recognition performance.

A high-level schematic diagram of our framework is depicted in Fig. 2. Our semi-supervised learning framework, TimeBalance, is a teacher-student framework. To train on the unlabeled samples, we distill the knowledge of two teacher models: temporally-invariant teacher $f_I$ and temporally-distinctive teacher $f_D$, which are trained in self-
supervised manner, to a student model $f_S$. In the following, we explain the details of TimeBalance. To be particular, in Sec. 3.1, we present the self-supervised pretraining of teacher models. In Sec. 3.2 we explain the semi-supervised training of the student model and our Temporal Similarity based Teacher Reweighting (TSTR) scheme.

### 3.1. Self-supervised pretraining of teachers

From a video instance $v^{(i)}$, we sample $n$ consecutive clips $\mathcal{X}^{(i)} = \{x_i^{(i)}\}_{t=1}^{n}$, where $t$ represents clip location (timestamp). Each of these clips undergoes stochastic transformations (e.g. random crop, random color jittering, etc.). Next, we send these clips to the teacher model $f$ and a non-linear projection head $g$ respectively. The non-linear projection head, projects the clip-embedding from the teacher model to a lower-dimensional normalized representation $z$, s.t. $z \in \mathbb{R}^d$, where $d$ is the dimension of output vector $z$.

#### 3.1.1 Pretraining of Temporally-Invariant Teacher

The goal of temporal-invariant pretraining is to learn the shared information across the $n$ different clips $\{x_i^{(i)}\}_{t=1}^{n}$ of the same video instance $i$. To achieve this, we maximize the agreement between the projections, $z$, of two different clips from the same video instance $\{(z_1^{(i)}, z_2^{(i)}) \mid t_1, t_2 \in \{1...n\} \text{ and } t_1 \neq t_2\}$, while maximizing the disagreement between the projections of clips from different video instances $\{(z_1^{(i)}, z_1^{(j)}) \mid i, j \in B \text{ and } i \neq j\}$, where $B$ is the batch-size. This contrastive objective can be expressed as the following equation:

$$L^{(i)}_I = -\sum_{t_1, t_2 \neq t_1} \log \frac{\sum_{j=1}^B \mathbb{1}_{j \neq i} h(z_1^{(i)}, z_2^{(j)}) + h(z_1^{(i)}, z_2^{(j)})}{\sum_{j=1}^B \mathbb{1}_{j \neq i} h(z_1^{(i)}, z_2^{(j)}) + h(z_1^{(i)}, z_2^{(j)})},$$  

where $h(u_1, u_2) = \exp \left( \frac{u_1^T u_2}{||u_1|| ||u_2||} \right)$ is used to compute the similarity between $u_1$ and $u_2$ vectors with an adjustable temperature parameter, $\tau$. $\mathbb{1}_{j \neq i}$ is an indicator function which equals 1 iff $j \neq i$.

$$L^{(i)} = L^{(i)}_I + \omega L^{(i)}_{\text{unsup}},$$

where $\omega$ is the weight of the unsupervised loss.

#### 3.1.2 Pretraining of Temporally-Distinctive Teacher

Contrary to the goal of $L_I$, temporally-distinctive pretraining deals with learning the differences across the clips of the same video instance. To achieve this, we generate another set of clips $\tilde{\mathcal{X}}^{(i)} = \{\tilde{x}_i^{(i)}\}_{t=1}^{n}$, which are randomly-augmented versions of clips in $\mathcal{X}^{(i)}$. After that, we maximize the agreement between the projections of a pair of clips $\{(z_1^{(i)}, \tilde{z}_1^{(i)}) \mid t_1 \in \{1...n\}\}$ from the same timestamp and maximize the disagreement between the projections of pair of temporally misaligned clips. A mathematical expression for this contrastive objective can be written as:

$$L^{(i)}_{D1} = -\sum_{t_1=1}^{n} \log \frac{\sum_{t_2=1}^{n} h(z_1^{(i)}, \tilde{z}_1^{(i)}) + h(z_1^{(i)}, \tilde{z}_1^{(i)})}{\sum_{t_2=1}^{n} h(z_1^{(i)}, \tilde{z}_1^{(i)}) + h(z_1^{(i)}, \tilde{z}_1^{(i)})},$$

The above contrastive loss imposes temporal distinctiveness at the clip-level i.e. temporally-average-pooled features. Similarly, we can also impose temporal distinctiveness ($L^{(i)}_{D2}$) on a more fine-grained level i.e. on the unpooled temporal feature slices [18]. More details in Supp. Sec. D. We combine these pooled and unpooled temporal-distinctiveness objectives to obtain $L^{(i)} = L^{(i)}_{D1} + L^{(i)}_{D2}$.

The primary objective of this work is semi-supervised action recognition. Therefore, even though the self-supervised teacher models lack any explicit notion of category-specific output space, we argue that to solve the downstream action recognition task their knowledge has to be distilled from the action category-specific output space. To this end, we finetune both of the self-supervised teacher...
models on the labeled set $D_l$ using cross-entropy loss.

### 3.2. Semi-supervised training of student model

We initialize the student model with weights from a video self-supervised model [18] trained on $D_u$. During the semi-supervised training, student model $f_S$ gets supervision from two sources: (i) ground-truth label (if available), and (ii) teacher supervision (Fig. 2). A visual aid is provided in Supp. Sec. D for loss computations over the labeled and unlabeled split in semi-supervised training. The supervision from the ground-truth label is utilized using standard cross-entropy loss, as depicted in the following equation:

$$
\mathcal{L}_{sup}^{(i)} = -\sum_{c=1}^{C} y_c^{(i)} \log p_c^{(i)}
$$

(4)

For instance $i$, the prediction vectors of the invariant and distinctive teacher are denoted as $p_I$ and $p_D$, respectively.

Next, we will discuss our TSTR scheme to distill knowledge from the temporally invariant and distinct teachers to train the student model on the unlabeled data, $D_u$ and videos of labeled data $D_l$.

**Temporal Similarity based Teacher Reweighting.** In order to combine supervision from $f_I$ and $f_D$ for a particular video instance $v^{(i)}$, we first compute temporal similarity scores. To this end, we compute the cosine similarity between each pair of clips to form a temporal similarity matrix, $C^{(i)}$, as depicted in Fig. 3. The temporal similarity matrix computation is described in the following,

$$
C^{(i)} = \left[ \text{Sim}(z_{t_1}^{(i)}, z_{t_2}^{(i)}) \right]_{t_1, t_2=1}^{n},
$$

(5)

where, $\text{Sim}(\cdot)$ is the cosine similarity function.

We compute the temporal similarity matrix for both invariant and distinctive teachers denoted as $C_{I}^{(i)}$, and $C_{D}^{(i)}$, respectively. Next, in order to get a similarity score, $s^{(i)}$, for an instance $i$, we take the average of non-diagonal elements of both $C_{I}^{(i)}$, and $C_{D}^{(i)}$ matrices.

$$
s^{(i)} = \frac{1}{2n(n-1)} \sum_{t_1, t_2=1}^{n} \left( C_{I}^{(i)} + C_{D}^{(i)} \right)
$$

(6)

We use this temporal similarity score, $s^{(i)}$, to aggregate the outputs of the teacher models. Let’s assume, for instance $i$, the prediction vectors of the invariant and distinctive teacher are denoted as $p_I^{(i)}$ and $p_D^{(i)}$, respectively. Now, we want to combine these teacher prediction vectors in such a way that the temporally-invariant prediction $p_I^{(i)}$ gets a higher weight if the temporal similarity score is high, and the weight of the temporally-distinctive prediction $p_D^{(i)}$ gets higher in the case of a lower temporal similarity score. This dynamic weighting scheme for obtaining the aggregated teacher prediction is provided below.

$$
p_T^{(i)} = s^{(i)} p_I^{(i)} + (1 - s^{(i)}) p_D^{(i)}
$$

(7)

We use the combined teacher prediction $p_T^{(i)}$ to provide supervision to the student model using $L_2$ loss. We compute this loss only on the unlabeled samples ($D_u$ and videos of $D_l$ without assigned labels), hence, we refer to this loss as the unsupervised loss. This loss term is defined below.

$$
L_{unsup}^{(i)} = \left( p_T^{(i)} - p_S^{(i)} \right)^2
$$

(8)

Finally, we sum both the supervised and unsupervised losses to train the student model. The overall objective function is defined below.

### 3.3. Algorithm

Let’s consider models $f_I$, $f_D$, and $f_S$ are parameterized by $\theta_I$, $\theta_D$, and $\theta_S$. All steps of our semi-supervised training are put together in Algorithm 1.

---

**Algorithm 1: TimeBalance training algorithm**

1. **Inputs:**
2. **Datasets:** $D_u$, $D_l$
3. \#Epochs: $\text{max ssl epoch}_I$, $\text{max ssl epoch}_D$, $\text{max epoch}_tune$, $\text{max epoch}$
4. **Learning Rates:** $\alpha_I$, $\alpha_D$, $\alpha_S$
5. **Output:** Student model $\theta_S$
6. Initialize $\theta_I$, $\theta_D$ randomly;
7. Initialize $\theta_S$ with any SSL [18, 55] method on $D_u$;
8. Temporally-Invariant Self-supervised Pretraining:
   9. for $e_0 \leftarrow 1$ to $\text{max ssl epoch}_I$ do
      10. \quad $\theta_I \leftarrow \theta_I - \alpha_I \nabla_{\theta_I} L_I(\theta_I)$
   11. end
12. Temporally-Distinctive Self-supervised Pretraining:
   13. for $e_0 \leftarrow 1$ to $\text{max ssl epoch}_D$ do
      14. \quad $\theta_D \leftarrow \theta_D - \alpha_D \nabla_{\theta_D} L_D(\theta_D)$
   15. end
16. Compute the similarity score $s^{(i)}$ using Eq. 6
17. Finetuning the teacher models on labeled set $D_l$:
   18. for $e_0 \leftarrow 1$ to $\text{max epoch}_tune$ do
      19. \quad $\theta_I^* = \arg\min_{\theta_I} L_{\text{sup}}(\theta_I)$
      20. \quad $\theta_D^* = \arg\min_{\theta_D} L_{\text{sup}}(\theta_D)$
   21. end
22. Semi-supervised training of student on $D_l + D_u$:
   23. for $e_0 \leftarrow 1$ to $\text{max epoch}$ do
      24. \quad $L = L_{\text{sup}}(\theta_S) + \omega L_{\text{unsup}}(\theta_S, \theta_I^*, \theta_D^*, s^{(i)})$
      25. \quad $\theta_S \leftarrow \theta_S - \alpha_S \nabla_{\theta_S} L$
   26. end
4. Experiments

4.1. Datasets

UCF101 [66] is a dataset for human action recognition collected from internet videos consisting of 101 action classes. We use split-1 for our experiments, which has 9,537 train videos and 3,783 test videos.

HMDB51 [42] is a relatively smaller dataset collected from movie videos. It has 51 human activity classes and has a high intra-class variance. We use split-1 in this paper, which has 9,537 train videos and 3,783 test videos.

Kinetics400 [12] is a large-scale dataset collected from YouTube videos. It has a standard split of 240k training videos and 20k validation videos which covers 400 actions.

4.2. Implementation Details

For our default experimental setup, we use an input clip resolution of $224 \times 224$. We use 3D-ResNet-50 [24] for both student and teacher models. We use most commonly used augmentations: geometry-based augmentations like random cropping, and flipping, and color-based augmentations like random greyscale, and color jittering.

Self-supervised pretraining The pretraining is performed with clips of 16 frames for 100 epochs for Kinetics400, 250 epochs for UCF101, and HMDB51 experiments. For contrastive losses, the default temperature is set to 0.1.

Semi-supervised training Semi-supervised training is performed with clips of 8 frames for 150 epochs.

Inference We follow standard protocol [73] of averaging the predictions of 10 uniformly spaced clips and 3 different spatial scales to get a video-level prediction. More implementation details in Supp. Sec. B.

4.3. Comparison with prior work

We compare our method with image-based baselines, video self-supervised baselines, and recent state-of-the-art methods for semi-supervised action recognition in Table 1. We present the results in two sections based on the backbone architecture used: (1) 3D-ResNet18 and (2) 3D-ResNet50. We report Top-1 classification accuracy as the performance measure and follow the standard protocol of reporting average performance over three independent runs.

**Image-based baselines** We consider widely used semi-supervised image classification baselines like Pseudo-Label [45], MeanTeacher [68], UPS [57], S4L [83] and FixMatch [65]. From Table 1, we observe that the results of these image-based methods are significantly lower than the video-based methods across all benchmarks; which suggests that spatial information is not enough to excel in semi-

<table>
<thead>
<tr>
<th>Method</th>
<th>Backbone</th>
<th>Params (M)</th>
<th>Input #F</th>
<th>UCF101 1% 5% 10% 20% 50%</th>
<th>HMDB51 40% 50% 60% 1% 10%</th>
</tr>
</thead>
<tbody>
<tr>
<td>PL-ICM13 [45]</td>
<td>3D-ResNet18</td>
<td>13.5</td>
<td>V 16</td>
<td>- 17.6 24.7 37.0 47.5</td>
<td>27.3 32.4 33.5 - -</td>
</tr>
<tr>
<td>MT [68]</td>
<td>3D-ResNet18</td>
<td>13.5</td>
<td>V 16</td>
<td>- 17.5 25.6 36.3 45.8</td>
<td>27.2 30.4 32.2 - -</td>
</tr>
<tr>
<td>S4L [83]</td>
<td>3D-ResNet18</td>
<td>13.5</td>
<td>V 16</td>
<td>- 22.7 29.1 37.7 47.9</td>
<td>29.8 31.0 35.6 - -</td>
</tr>
<tr>
<td>UPS [57]</td>
<td>3D-ResNet18</td>
<td>13.5</td>
<td>V 16</td>
<td>- - - 39.4 50.2</td>
<td>- - - -</td>
</tr>
<tr>
<td>SD [28]</td>
<td>3D-ResNet18</td>
<td>13.5</td>
<td>V 16</td>
<td>- 31.2 40.7 45.4 53.9</td>
<td>32.6 35.1 36.3 - -</td>
</tr>
<tr>
<td>MT+SD [38]</td>
<td>3D-ResNet18</td>
<td>13.5</td>
<td>V 16</td>
<td>- 30.3 40.5 45.5 53.0</td>
<td>32.3 33.6 35.7 - -</td>
</tr>
<tr>
<td>3DRotNet [39]</td>
<td>3D-ResNet18</td>
<td>13.5</td>
<td>V 16</td>
<td>15.0 31.5 40.4 47.1 -</td>
<td>- - - -</td>
</tr>
<tr>
<td>VideoSemi [38]</td>
<td>3D-ResNet18</td>
<td>13.5</td>
<td>V 16</td>
<td>- 32.4 42.0 48.7 54.3</td>
<td>32.7 36.2 37.0 - -</td>
</tr>
<tr>
<td>TCL [64]</td>
<td>TSM-ResNet18</td>
<td>-</td>
<td>V 8</td>
<td>- - - - -</td>
<td>- 11.6 -</td>
</tr>
<tr>
<td>TG-FixMatch [77]</td>
<td>3D-ResNet18</td>
<td>13.5</td>
<td>V 8</td>
<td>- 44.8 62.4 76.1 79.3</td>
<td>46.5 48.4 49.7 9.8 43.8</td>
</tr>
<tr>
<td>MvPL [79]</td>
<td>3D-ResNet18</td>
<td>13.5</td>
<td>VFG 8</td>
<td>- 41.2 55.5 64.7 65.6</td>
<td>30.5 33.9 35.8 5.0 36.9</td>
</tr>
<tr>
<td>TCLR [18]</td>
<td>3D-ResNet18</td>
<td>13.5</td>
<td>V 16</td>
<td>26.9 - 66.1 73.4 76.7</td>
<td>- - - -</td>
</tr>
<tr>
<td>CMPL [80]</td>
<td>3D-ResNet18</td>
<td>13.5</td>
<td>V 8</td>
<td>23.8 - 67.6 - -</td>
<td>- - - 16.5 53.7</td>
</tr>
<tr>
<td>TACL [70]</td>
<td>3D-ResNet18</td>
<td>13.5</td>
<td>V 16</td>
<td>- 35.6 50.9 56.1 65.8</td>
<td>34.6 37.2 39.5 - -</td>
</tr>
<tr>
<td>TACL [70]</td>
<td>3D-ResNet18</td>
<td>13.5</td>
<td>V 16</td>
<td>- 43.7 55.6 59.2 67.2</td>
<td>38.7 40.2 41.7 - -</td>
</tr>
<tr>
<td>MemDPC [81]</td>
<td>3D-ResNet18</td>
<td>13.5</td>
<td>V 16</td>
<td>- - 44.2 50.9 62.3</td>
<td>- - - -</td>
</tr>
<tr>
<td>MotionFit [27]</td>
<td>3D-ResNet18</td>
<td>13.5</td>
<td>VFG 16</td>
<td>- - - 57.7 59.0</td>
<td>- - - -</td>
</tr>
<tr>
<td>Ours (TimeBalance) [31]</td>
<td>3D-ResNet18</td>
<td>13.5</td>
<td>V 8</td>
<td>29.1 47.9 69.8 79.1 83.3</td>
<td>49.8 51.4 53.1 17.1 54.9</td>
</tr>
<tr>
<td>ActorCM [81]</td>
<td>3D-ResNet18</td>
<td>13.5</td>
<td>V 8</td>
<td>- 27.0 40.2 51.7 59.9</td>
<td>32.9 38.2 38.9 - -</td>
</tr>
<tr>
<td>FixMatch [65]</td>
<td>3D-ResNet18</td>
<td>13.5</td>
<td>V 8</td>
<td>- 45.1 53.0 57.4 64.7</td>
<td>33.7 39.5 40.8 - -</td>
</tr>
<tr>
<td>MvPL [79]</td>
<td>3D-ResNet50</td>
<td>31.8</td>
<td>VFG 8</td>
<td>22.8 - 80.5 - -</td>
<td>- - - 17.0 58.2</td>
</tr>
<tr>
<td>CMPL [80]</td>
<td>3D-ResNet50</td>
<td>31.8</td>
<td>V 8</td>
<td>25.1 - 79.1 - -</td>
<td>- - - 17.6 58.4</td>
</tr>
<tr>
<td>Ours (TimeBalance) [31]</td>
<td>3D-ResNet50</td>
<td>31.8</td>
<td>V 8</td>
<td>30.1 53.5 81.1 83.3 85.0</td>
<td>52.6 53.9 54.5 19.6 61.2</td>
</tr>
</tbody>
</table>

Table 1. Comparison with state-of-the-art methods: Methods using pre-trained ImageNet weights are shown in Grey, V- Video (RGB), F- Optical Flow, G- Temporal Gradients. Best results are shown in Red, and Second-best in Blue.
supervised action recognition.

**Video Self-supervised Learning baselines** Video self-supervised learning methods are first pretrained on the full-training set without using any labels and then finetuned on the labeled set in a supervised manner. We compare with contrastive learning-based methods like TCLR [18], MemDPC [31], multimodal approach like MotionFit [27], and pretext-task based approach 3DRotNet [39]. From Table 1, we notice that the performance of these video self-supervised methods is significantly better than image-based baselines, and in some cases, it even performs favorably against the semi-supervised action recognition methods [77, 79, 80]. However, our proposed semi-supervised method outperforms all these video self-supervised baselines by a noticeable margin.

**Semi-supervised action recognition baselines** We also compare with prior semi-supervised action recognition works like TG-Fixmatch [77], VideoSemi [38], MvPL [79], TACL [70], and ActorCM [89]. Our method achieves significant improvement over these methods across all benchmarks. Remarkably, our method also outperforms the methods that use additional modalities like optical flow in MvPL [79] and additional data (ImageNet) in ActorCM [89] and TACL [70].

### 4.4. Ablations and Analysis

In the default setting of ablations, we consider UCF101 with 3D-ResNet50 as the teacher backbone. More ablations in Supp. Sec. C.

**Contribution of different training components** In Table 2, we analyze the effect of each model \((f_S, f_I, f_D)\) and our proposed teacher reweighting scheme with 5% and 20% labeled data on the UCF101 dataset. Row a–c, demonstrates the performance of teacher and student models individually. Results in Row f demonstrates that the student model performs the best when we average the predictions of both teachers and removing any of them (Row d, e) degrades the performance. This validates our hypothesis that for optimal performance, we need to distill the knowledge from both teachers. Finally, Row g, demonstrates the effectiveness of our proposed teacher reweighting scheme using temporal similarity. We found similar results with \(f_S\) from random initialization in Table 3.

<table>
<thead>
<tr>
<th>(f_S)</th>
<th>(f_I)</th>
<th>(f_D)</th>
<th>Teacher Reweighting</th>
<th>UCF101 % labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) ✓  ✓  ✓  ✓</td>
<td>48.66</td>
<td>79.70</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(b) ✗  ✓  ✓  ✓</td>
<td>43.79</td>
<td>74.55</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(c) ✗  ✗  ✓  ✓</td>
<td>44.08</td>
<td>75.13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(d) ✓  ✓  ✓  ✓</td>
<td>47.95</td>
<td>79.28</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(e) ✓  ✗  ✓  ✓</td>
<td>48.81</td>
<td>79.76</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(f) ✓  ✓  ✓  ✓</td>
<td>52.14</td>
<td>82.02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(g) ✓  ✓  ✓  ✓</td>
<td>53.48</td>
<td>83.26</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2. Ablation of different components of our framework. Student \((f_S)\) and teachers \((f_I\) and \(f_D)\) are 3D-ResNet50.

**Visualization of Temporal Similarity Matrix** We sample four consecutive frames from unlabeled videos and compute the temporal similarity matrix for both \(f_I\) and \(f_D\). Visualization is shown Fig. 4. Firstly, we observe that \(f_I\) gives higher similarity values in both video instances than \(f_D\). Secondly, we can observe that the final instance similarity score aligns with our goal i.e. providing more weight to \(f_I\) in atomic actions and providing more weight to \(f_D\) in complex actions.

**Different types of teacher pairs** In order to further investigate the importance of having teachers with complemen-
we use (Inv1, Dist1) as teacher pair.

<table>
<thead>
<tr>
<th>Teacher-1</th>
<th>Teacher-2</th>
<th>UCF101 % labels</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>5%</td>
</tr>
<tr>
<td>(a)</td>
<td>Inv1</td>
<td>48.33</td>
</tr>
<tr>
<td>(b)</td>
<td>Dist1</td>
<td>49.15</td>
</tr>
<tr>
<td>(c)</td>
<td>Inv1</td>
<td>52.14</td>
</tr>
<tr>
<td>(d)</td>
<td>Inv1</td>
<td>51.78</td>
</tr>
</tbody>
</table>

Table 4. Ablation of different teacher combinations.

Student initialization We also study various video self-supervised methods to initialize the student model and report the results in Fig 5. We conduct these experiments on the UCF101 dataset with 3D-ResNet18 as the student backbone. We observe that even across a diverse set of video SSL-based initialization techniques, our proposed semi-supervised framework can achieve better/competitive performance against the current state-of-the-art. This further validates the effectiveness of our idea of leveraging temporally distinct and invariant teachers and our proposed teacher prediction reweighting scheme.

Number of clips in Teacher SSL pretraining We analyze the effect of using a different number of clips to impose temporal invariance/distinctiveness in teacher pretraining in Table 5. We perform pretraining on UCF101 with the 3D-ResNet50 model and report the performance of the teacher model after finetuning on the labeled set. In the default setting, we use n=4 clips. We observe that while reducing the clips from 4 to 2, the performance drop is significant for temporally-distinctive teachers, whereas, it is not severe for the temporally-invariant teacher. Since the distinctiveness contrastive loss $L_D$ treats other clips as negatives, its contrastive objective becomes very easy to solve if we reduce the number of clips (Eq. 3). Whereas, temporal-invariance contrastive loss $L_I$ takes negatives from the other instances of the batch, and decreasing the number of clips does not change the difficulty of the loss.

<table>
<thead>
<tr>
<th>Pretraining</th>
<th>Number of clips (n)</th>
<th>UCF101 % labels</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>5%</td>
</tr>
<tr>
<td>Invariant Teacher</td>
<td>n=4</td>
<td>43.79</td>
</tr>
<tr>
<td>Distinctive Teacher</td>
<td>n=2</td>
<td>44.08</td>
</tr>
</tbody>
</table>

Table 5. Number of clips per video in self-supervised pretraining.

5. Conclusion and Future Work

In this work, we have proposed TimeBalance, a teacher-student framework for semi-supervised action recognition. We utilize the complementary strengths of temporally-invariant and temporally-distinctive representations to leverage unlabeled videos. Our extensive experimentation has empirically validated the effectiveness of different components of our framework and results on multiple benchmark datasets established TimeBalance as the new state-of-the-art for semi-supervised action recognition.

Our findings regarding the complementary strengths of temporally-invariant and temporally-distinctive video representations could be applied to other data-efficient video understanding problems, such as few-shot action recognition and spatio-temporal action detection. It would also be interesting to explore the invariance and distinctiveness properties of video using recent masked-image modeling techniques [23, 71] and multi-modal settings [4], such as audio + video and text + video.

Acknowledgments

We thank Tristan de Blegiers for his help with visualization.
References


[23] Christoph Feichtenhofer, Haoqi Fan, Yanghao Li, and Kaiming He. Masked autoencoders as spatiotemporal learners. In Advances in Neural Information Processing Systems, 2021. 3


video representation learning by decoupling the scene and the motion. In The AAAI Conference on Artificial Intelligence (AAAI), 2021.


