Multiview Pseudo-Labeling for Semi-supervised Learning from Video

Bo Xiong[1] Haoqi Fan[1] Kristen Grauman[1,2] Christoph Feichtenhofer[1]

[1] Facebook AI Research [2] University of Texas at Austin

Abstract

We present a multiview pseudo-labeling approach to video learning, a novel framework that uses complementary views in the form of appearance and motion information for semi-supervised learning in video. The complementary views help obtain more reliable "pseudo-labels" on unlabeled video, to learn stronger video representations than from purely supervised data. Though our method capitalizes on multiple views, it nonetheless trains a model that is shared across appearance and motion input and thus, by design, incurs no additional computation overhead at inference time. On multiple video recognition datasets, our method substantially outperforms its supervised counterpart, and compares favorably to previous work on standard benchmarks in self-supervised video representation learning.

1. Introduction

3D convolutional neural networks (CNNs) [55, 8, 56, 17] have shown steady progress for video recognition, and particularly human action classification, over recent years. This progress also came with a shift from traditionally small-scale datasets to large amounts of labeled data [31, 6, 7] to learn strong spatiotemporal feature representations. Notably, as 3D CNNs are data hungry, their performance has never been able to reach the level of hand-crafted features [57] when trained '*from-scratch*' on smaller scale datasets [49].

However, collecting a large-scale annotated video dataset [21, 7] for the task at hand is expensive and tedious as it often involves designing and implementing annotation platforms at scale and hiring crowd workers to collect annotations. For example, a previous study [48] suggests it takes at least one dollar to annotate a single video with 157 human activities. Furthermore, the resource intensive annotation process needs to be repeated for each task of interest or when the label space needs to be expanded. Finally, another dilemma that emerges with datasets collected from the web is that the underlying content is vanishing over time, and therefore the datasets need to be replenished in a recurring fashion [50].

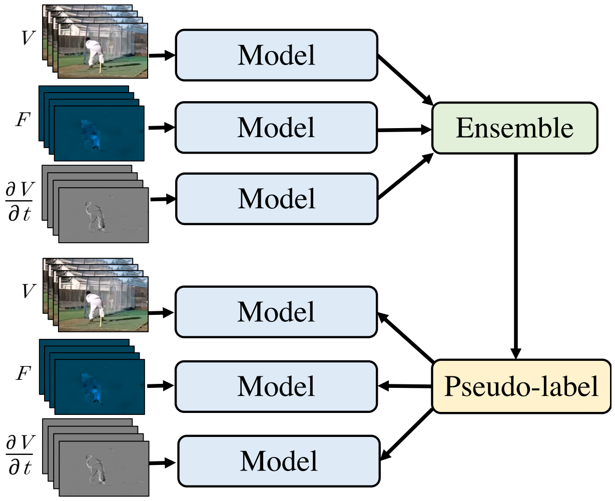
V
F
$\frac{\partial V}{\partial t}$
Model
Model
Model
Ensemble
V
F
$\frac{\partial V}{\partial t}$
Model
Model
Model
Pseudo-label

Figure 1: **Multiview pseudo-labeling (MvPL)** takes in multiple complementary views of a single unlabeled video clip, in the form of RGB (V), optical-flow (F), and temporal gradients ($\frac{\partial V}{\partial t}$) and uses a shared model to perform semi-supervised learning. After training, a single RGB view is used for inference.

The goal of this work is semi-supervised learning in video to learn from both labeled and *unlabeled* data, thereby reducing the amount of annotated data required for training. Scaling video models with unlabeled data is a setting of high practical interest, since collecting large amounts of unlabeled video data requires minimal human effort. Still, thus far this area has received far less attention than fully supervised learning from video.

Most prior advances in semi-supervised learning in computer vision focus on the problem of image recognition. "Pseudo-labeling" [38, 64, 61, 51] is a popular approach to utilize unlabeled images. The idea is to use the predictions from a model as target labels and gradually add the unlabeled images (with their inferred labels) to the training set. Compared to image recognition, semi-supervised video recognition presents its own challenges and opportunities. On the one hand, the temporal dimension introduces some ambiguity, *i.e*. given a video clip with an activity label, the activity may occur at any temporal location. On the other hand, video can also provide a valuable, complementary signal for recognition by the way objects move in space-time, *e.g*. the actions 'sit-down' *vs*. 'stand-up' cannot be discriminated without using the temporal signal. More specifically, video adds information about how actors, objects, and the environment *change* over time.

Therefore, directly applying semi-supervised learning algorithms designed for images to video could be sub-optimal (we will verify this point in our experiments), as image-based algorithms only consider appearance information and ignore the potentially rich dynamic structure captured by video.

To address the challenge discussed above, we introduce *multiview pseudo-labeling (MvPL)*, a novel framework for semi-supervised learning designed for video. Unlike traditional 3D CNNs that implicitly learn spatiotemporal features from appearance, our main idea is to explicitly force a *single model* to learn appearance and motion features by ingesting multiple complementary *views*[1] in the form of appearance, motion, and temporal gradients, so as to train the model from unlabeled data that augments labeled data.

We consider visual-only semi-supervised learning and all the views are computed from RGB frames. Therefore our method does not require any additional modalities nor does it require any change to the model architecture to accommodate the additional views. Our proposed multiview pseudo-labeling is general and can serve as a drop-in replacement for any pseudo-labeling based algorithm [38, 64, 61, 51] that currently operates only on appearance, namely by augmenting the model with multiple views and our ensembling approach to infer pseudo-labels.

Our method rests on two technical insights: 1) a single model that nonetheless benefits from multiview data; and 2) an ensemble approach to infer pseudo-labels.

First, we convert both optical flow and temporal difference to the same input format as RGB frames so that all the views can share the *same* 3D CNN model. The 3D CNN model takes only one view at a time and treats optical flow and temporal gradients as if they are RGB frames. The advantage is that we directly encode appearance and motion in the input space and distribute the information through multiple views, to the benefit of the 3D CNN.

Second, when predicting pseudo-labels for unlabeled data, we use an ensemble of all the views for prediction. We show that predicting pseudo-labels from all the views is more effective than predicting from a single view alone. Our method uses a single model that can seamlessly accommodate different views as input for video recognition. See Figure 1 for an overview of our approach.

In summary, this paper makes the following contributions:

- This work represents an exploration in semi-supervised learning for video understanding, an area that is heavily researched in image understanding [10, 22, 28, 51, 60]. Our evaluation establishes semi-supervised baselines on Kinetics-400 (1% and 10% label case), and UCF101 (similarly as the image domain which uses 1% and 10% of labels on ImageNet [10, 28, 51, 60]).
- Our technical contribution is a novel multiview pseudo-labeling framework for *general* application in semi-supervised learning from video, that delivers consistent improvement in accuracy on *multiple* pseudo-labeling algorithms.
- On several challenging video recognition benchmarks, our method substantially improves its single view counterpart. We obtain state-of-the-art performance on UCF101 [52] and HMDB-51 [35] when using Kinetics-400 [31] as unlabeled data, and outperform video self-supervised methods in this setting.

2. Related Work

Semi-supervised learning in images. Most prior advances in semi-supervised learning in computer vision focus on image recognition. Regularization on unlabeled data is a common strategy for semi-supervised learning. Entropy regularization [22] minimizes the conditional entropy of class probabilities for unlabeled data. Consistency regularization forces the model representations to be similar when augmentations are applied to unlabeled data [47]. VAT [42] uses adversarial perturbations while UDA [60] applies RandAugment [11] for augmentations. Pseudo-labeling [38, 64] or self-training [61] is another common strategy for semi-supervised learning, where predictions from a model are used as pseudo-labels for unlabeled data. Pseudo-labels can be generated using a consensus from previous model checkpoints [37] or an exponential moving average of model parameters [54]. FixMatch [51] predicts pseudo-labels from weak augmentation to guide learning for strong augmentation generated from RandAugment [11]. Unlike any of the prior work above, we consider video, and our method leverages multiple complementary views.

Semi-supervised learning in videos. Compared to images, semi-supervised learning for video has received much less attention. The work of [66] applies an encoder-decoder framework to minimize a reconstruction loss. The work in [30] combines pseudo-labeling and distillation [19] from a 2D image classifier to assist video recognition. However,

[1] We use the term *view* to refer to different input types (RGB frames, optical flow, or RGB temporal gradients), as opposed to camera viewpoints.

none of the prior semi-supervised work capitalizes on the rich views (appearance, motion, and temporal gradients) in videos. To the best of our knowledge, we are the first to explore multiple complementary views for semi-supervised video recognition. Co-training [4] is a seminal work for semi-supervised learning with two views, first introduced for the web page classification problem. Co-training learns separate models for each view, whereas we share a single model for all views. Our idea has the advantage that a single model can directly leverage the complementary sources of information from all the views. Our experiments demonstrate that our design outperforms co-training for this video learning setting.

Self-supervised learning. Another common direction to leverage unlabeled video data is self-supervised learning. Self-supervised learning first learns feature representations from a pretext task (e.g., audio video synchronization [34], clustering [1], clip order [63], and instance discrimination [46] etc.), where the labels are generated from the data itself, and then fine-tunes the model on downstream tasks with labeled data. Self-supervised learning in video can leverage modalities by learning the correspondence between visual and audio cues [45] or video and text [68, 41]. Appearance and motion [23] can be used to boost performance in a contrastive learning framework or address domain adaptation [44]. Self-supervised training learns task-agnostic features, whereas semi-supervised learning is task-specific. As suggested in [66], semi-supervised learning can also leverage a self-supervised task as pre-training, *i.e.* the two ideas are not exclusive, as we will also show in results.

Multi-modal video recognition. Supervised video recognition can benefit from multi-modal inputs. Two-stream networks [49, 18] leverage both appearance and motion. Temporal gradients [58, 67] can be used in parallel with appearance and motion to improve video recognition. Beyond visual cues, audio signals [59, 32, 2] can also assist video recognition. We consider visual-only semi-supervised learning for video recognition. Like [58, 67], we use appearance, motion, and temporal gradients, but unlike any of these prior models, our approach addresses semi-supervised learning.

3. Multiview Pseudo-Labeling (MvPL)

We focus on semi-supervised learning for videos and our objective is to train a model by using both labeled and unlabeled data.

Our main idea is to capitalize on the complementarity of appearance and motion views for semi-supervised learning from video. We first describe how we extract multiple views from video (§3.1), followed by how we use a single model to seamlessly accommodate all the views for multiview learning and how we obtain pseudo-labels with a multiview ensemble approach (§3.2). Subsequently, §3.3 outlines three concrete instantiations of our approach and §3.4 provides implementation specifics.

3.1. Multiple views of appearance and dynamics

Many video understanding methods only consider a single view (*i.e.*, RGB frames), thereby possibly failing to model the rich dynamics in videos. Our goal is to use three complementary views in the form of RGB frames, optical flow, and RGB temporal gradients to investigate this. Our motivation is that:

(i) RGB frames (V) record the static appearance at each time point but do not directly provide contextual information about object/scene motion.

(ii) Optical flow (F) explicitly captures motion by describing the instantaneous image velocities in both horizontal and vertical axes.

(iii) Temporal gradients ($\frac{\partial V}{\partial t}$) between two consecutive RGB frames encode appearance change and correspond to dynamic information that deviates from a purely static scene. Compared to optical flow, temporal gradients accentuate changes at boundaries of moving objects.

Even though all three views are related to, and can be estimated from, each other by solving for optical-flow using the brightness constancy equation [27],

$$\nabla^\top V \cdot F + \frac{\partial V}{\partial t} = 0, \tag{1}$$

with $\nabla \equiv (\frac{\partial}{\partial x}, \frac{\partial}{\partial y})^\top$, F being the point-wise velocity vectors of the video brightness V and $\frac{\partial V}{\partial t}$ the temporal gradients at a single position in space, $\mathbf{x} = (x, y)^\top$, and time t, we find empirically that all three views expose complementary sources of information about appearance and motion that are useful for video recognition. This finding is related to the complementarity of hand-crafted space-time descriptors that have been successful in the past (*e.g.* histograms of space/time gradients [12, 33, 15] and optical flow [13, 57]).

3.2. Learning a single model from multiple views

One way to accommodate multiple views for learning is to train a separate model for each view and co-train their parameters [4]. However, each view only implicitly interacts with other views through predictions on unlabeled data. Another alternative is to use multiple network streams [49, 18]. However, here, the number of parameters of the model and the inference time grow roughly linearly with the number of streams, and during testing each stream has to be processed.

Instead, we propose to train a *single* model for all the complementary views by converting all the views to the same input format (*i.e.* we train a single model f, and it can take any view as input). By sharing the same model, the complementary views can serve as additional data augmentations to learn stronger representations. Compared to training separate models, our model can directly benefit from all the views instead of splitting knowledge between multiple models. Further, this technique does not incur any additional computation overhead after learning as only a single view is used for inference.

Formally, given a collection of labeled video data $\mathcal{X} = \{(x_i = [x_i^1, ..., x_i^M], y_i)\}$ for $i \in (1, \ldots, N_l)$, where y_i is the label for video instance x_i, N_l is the total number of labeled videos, and M is the total number of views, and a collection of unlabeled video data $\mathcal{U} = \{u_i = [u_i^1, ..., u_i^M]\}$ for $i \in (1, \ldots, N_u)$, our goal is to is to learn a classifier $f(x_i)$ by leveraging both labeled and unlabeled data.

We use a supervised cross entropy loss ℓ_s for labeled data and another cross entropy loss ℓ_u for unlabeled data. For our training batches, we assume $N_u = \mu N_l$ where μ is a hyper-parameter that balances the ratio of labeled and unlabeled samples N_l and N_u, respectively.

Supervised loss. For labeled data, we extend the supervised cross-entropy loss H to all the views on labeled data:

$$\ell_s = \frac{1}{N_l M} \sum_{i=1}^{N_l} \sum_{m=1}^{M} H(y_i, f(A(x_i^m))), \tag{2}$$

where y_i is the label, and A is a family of augmentations (*e.g.* cropping, resizing) applied to input x_i^m on view m.

Pseudo-label generation from multiple views. For the unlabeled data, we use an ensembling approach to obtain pseudo-labels. Given an unlabeled video with a view u_i^m, let s_i^m denote the pseudo-label class distribution, which is required because some of the instantiations we consider in the next section can filter out samples if the prediction is not confident. Then, the pseudo-label can be obtained by taking $\hat{s}_i^m = \arg\max(s_i^m)$. The model's class distribution prediction is $q_i^m = f(A(u_i^m))$, where A again corresponds to the family of augmentations applied to input u_i^m, and the class with the highest probability is $\hat{q}_i^m = \arg\max(q_i^m)$.

We explore the following variants to obtain pseudo-labels $\hat{s}_i^m$ given the class distribution prediction from all the views.

i **Self-supervision.** For each u_i^m, we directly use the most confident prediction $\hat{q}_i^m$ as its pseudo-label, that is, $\hat{s}_i^m = \hat{q}_i^m$. This is the most straightforward way to generate pseudo-labels. However, each view only supervises itself and does not benefit from the predictions from other views.

ii **Random-supervision.** For each u_i^m, we randomly pick another view $n \in (1, \ldots, M)$ and use the prediction on that view as the pseudo-label. Then we have $\hat{s}_i^m = \hat{q}_i^n$.

iii **Cross-supervision.** We first build a bijection function $b(m)$ for each view such that each view is deterministically mapped to another view and does not map to itself. Then for each u_i^m, we have $\hat{s}_i^m = \hat{q}_i^{b(m)}$. This is similar to co-training [4] in the two-view case.

iv **Aggregated-supervision.** For each unlabeled video, we obtain pseudo-labels by taking the weighted average of predictions from each view.

$$s_i^m = \frac{1}{\sum_{m=1}^{M} w_m} \sum_{m=1}^{M} w_m f(A(u_i^m)). \tag{3}$$

Then we obtain the pseudo-label by $\hat{s}_i^m = \arg\max(s_i^m)$. Note that in this case, all the views from the video u_i share the same pseudo-label. The advantage of this approach is that the pseudo-label contains information from all the views. We specify how to obtain the weight for each view in the implementation details.

Unsupervised loss. After obtaining the class distribution of each view $q_i^m = f(A(u_i^m))$ for an unlabeled video with M views ($u_i = [u_i^1, ..., u_i^M]$), we use one of the variants (i) – (iv) to obtain pseudo-label class distribution s_i^m and pseudo-label $\hat{s}_i^m$.

The pseudo-label $\hat{s}_i^m$ is then used as training signal for learning from the same, but differently augmented, data $\hat{A}(u_i^m)$, where $\hat{A}(u_i^m)$ denotes another family of transformations applied to the same unlabeled video u_i^m. Then our unsupervised loss is

$$\ell_u = \frac{1}{\mu N_l M} \sum_{i=1}^{\mu N_l} \sum_{m=1}^{M} \mathbb{1}(\max(s_i^m) \geq \tau) H(\hat{s}_i^m, f(\hat{A}(u_i^m))) \tag{4}$$

where τ is a threshold used to filter out unlabeled data if the prediction is not confident. The total loss is $\ell = \ell_l + \lambda_u \ell_u$, where λ_u controls the weight for the unlabeled data. Sec. 3.4 provides implementation details on the specific augmentations used.

3.3. MvPL instantiations

Our MvPL framework is generally applicable to multiple semi-supervised learning algorithms that are based on pseudo-labels. We instantiate our approach by unifying multiple methods in the same framework and analyze the commonality across methods. In this paper we concentrate on Pseudo-Label [38], FixMatch [51], and UDA [60]. On a high-level, these methods only differ in their utilization of unsupervised data in Eq. (4). We summarize our instantiations next.

Pseudo-Label. Pseudo-Label [38] uses the prediction from a sample itself as supervision. To apply our framework with Pseudo-Label [38], we simply use the same family of augmentations for obtaining pseudo-labels and learning from pseudo-labels, *i.e.* $\hat{A} = A$.

FixMatch. The main idea for FixMatch [51] is to predict pseudo-labels from weakly-augmented data and then use the

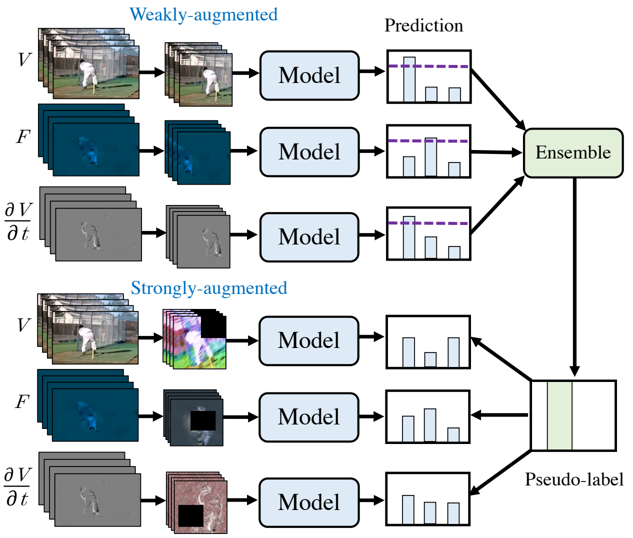

Figure 2: **Illustration of MvPL** applied to strongly-augmented data. Given an unlabeled video, we first obtain a weakly-augmented version of each view and then obtain predictions on them. Then we generate pseudo-labels by aggregating predictions from all the views. The pseudo-labels are used as a supervision signal for the strongly-augmented version of each view from the same video.

pseudo-label as the learning target for a strongly-augmented version of the same data. Given an unlabeled image, weakly-augmented data is obtained by applying standard data augmentation strategies, A, that include flipping and cropping. Strongly-augmented data is obtained by applying a family of augmentation operations $\hat{A}$, such as rotation, contrast, and sharpness *etc*., using RandAugment [11], that significantly alter the appearance of the unlabeled data.

Unsupervised Data Augmentation (UDA). Similar to FixMatch [51], UDA [60] also uses weak and strong augmentations by enforcing consistency between them in forms of predicted class distributions. To extend UDA with MvPL, we first sharpen the predicted class distribution s_i^m to obtain $\tilde{s}_i^m$. We then replace the hard label $\hat{s}_i^m$ in Eq. 4 with $\tilde{s}_i^m$. Strictly speaking, UDA [60] is not a pseudo-labeling algorithm per-se, because it uses soft labels (predicted class distribution with sharpening) as the learning signal.

We show an illustration of how to apply our method with strong augmentations in Figure 2.

3.4. Implementation Details

Model network architecture. As a backbone we use: R-50 [26] following the Slow pathway in [17] with clips of T=8 frames sampled with stride τ=8 from 64 raw-frames of video. This is a 3D ResNet-50 [26] without temporal pooling in the convolutional features. The input to the network is a clip of 8 frames with a sampling stride of 8, covering 64 frames of the raw video. The spatial input size is 224 × 224.

Inference. We follow the test protocol in [17]. The video model only takes RGB frames as input at inference time. For each video, we uniformly sample 10 clips along its temporal dimension. For each clip, we scale the shorter spatial side to 256 pixels and take 3 crops of 256×256. Finally, we obtain the prediction by averaging the softmax scores.

Converting optical flow and temporal gradients. We pre-compute (unsupervised) optical flow using the software package of [39] that implements a coarse-to-fine algorithm [5]. We convert both the raw optical flow and RGB temporal gradients into 3-channel inputs that are in the same range as RGB frames. For optical flow, the first two channels correspond to displacements in the horizontal and vertical directions, respectively. The third channel corresponds to the magnitude of the flow. All three channels are then normalized to the range of 0 and 255. We obtain temporal gradients by subtracting the next RGB frame from the current RGB frame. We then normalize them to the RGB range by adding 255 and dividing by 2.

Video augmentations. For weak augmentation, we use default video classification augmentations [17]. In particular, given a video clip, we first randomly flip it horizontally with a 50% probability, and then we crop 224×224 pixels from the video clip with a shorter side randomly sampled between 256 and 320 pixels.

As strong augmentations, we apply RandAugment [11] followed by Cutout [14] (we randomly cut a 128×128 patch from the same location across all frames in a video clip). RandAugment [11] includes a collection of image transformation operations (e.g., rotation, color inversion, translation, contrast adjustment, etc.). It randomly selects a small set of transformations to apply to data. RandAugment [11] contains a hyperparameter that controls the severity of all operations. We follow a random magnitude from 1 to 10 at each training step. When applying RandAugment to video clips, we keep the spatial transformations temporally consistent across all frames in a video clip.

Curriculum learning. We find it useful to first warm up the training in the first few epochs with only the labeled data and then start training with both labeled and unlabeled data.

Training details. We implement our model with PySlowFast [16]. We adopt synchronized SGD training in 64 GPUs following the recipe in [20], and we found its accuracy is as good as typical training in one 8-GPU machine. We follow the learning rate schedule used in [17], which combines a half-period cosine schedule [40] of learning rate decaying and a linear warm-up strategy [20]. We use momentum of 0.9 and weight decay of 10^{-4}. Dropout [53] of 0.5 is used before the final classifier layer. Please see

method	Pseudo-Label [38]	UDA [60]	FixMatch [51]
base	30.4	47.0	48.5
MvPL	**70.0** (+39.6)	**77.8** (+30.8)	**79.1** (+30.6)

(a) **MvPL with different pseudo-labeling algorithms.** Our semi-supervised method consistently improves all three algorithms.

RGB	Flow	TG	**MvPL**
✓			48.5
✓	✓		76.5 **(+28.0)**
✓		✓	74.0 **(+25.5)**
✓	✓	✓	**79.1** **(+30.6)**

(b) **Complementarity of views.** Optical Flow and temporal gradients (TG) are complementary to RGB.

Supervision	Top 1
Self (i)	75.8
Random (ii)	75.5
Cross (iii) 1)	78.7
Cross (iii) 2)	76.8
Agg. (iv) (Exclusion)	78.1
Agg. (iv) (All)	**79.1**

(c) **Ways to generate pseudo-labels**. Aggregated-supervision incorporating (All) views obtains the best result.

Table 1: **Ablation study** on UCF101 split-1. We use *only 10%* of its training labels and the *entire* training set as unlabeled data. We report top-1 accuracy on the validation set. Backbone: R-50, Slow-pathway [17], $T \times \tau = 8{\times}8$.

supp. for additional details. For MvPL with FixMatch, we set the threshold τ to 0.3 (used for filtering training samples if the prediction is not confident). We set the ratio μ (a ratio to balance the number of labeled and unlabeled data) to 3 for Kinetics-400 [31] and set μ to 4 for UCF101 [52] and HMDB51 [36]. For aggregated-supervision (iv), we assign each view with the same weight w_m as we found this works well in practice. §A provides further specifics.

4. Experiments

We validate our approach for semi-supervised learning for video. First, we present ablation studies to validate our design choices in Sec. 4.1. Then, we show the main results by evaluating our proposed method on multiple video recognition datasets in Sec. 4.2. Finally, we compare our method with existing self-supervised methods in Sec. 4.3. Unless specified otherwise, we present results on our method used in conjunction with FixMatch [51] using aggregated-supervision to obtain pseudo-labels.

Datasets. We evaluate our approach on three standard video recognition datasets: Kinetics-400 [31] (K400), UCF101 [52] and HMDB51 [36]. K400 contains 400 action classes with roughly 240k training videos. Each video is around 10 seconds. UCF101 contains 101 action classes with roughly 10k training videos and HMDB51 contains 51 action classes with roughly 4k training videos. Both UCF101 and HMDB51 have 3 train-val splits.

4.1. Ablation Studies

We first investigate ablation studies to examine the effectiveness of MvPL. Our ablations are carried out on UCF101 split-1 and use *only 10%* of its training labels and the *entire* UCF101 training set as unlabeled data (evaluation is done on the validation set). For all ablation experiments, we train the network for 600 epochs from scratch with no warm-up and use *aggregated-supervision* (iv) from all the views to obtain pseudo-labels, unless specified otherwise.

MvPL generally improves pseudo-labeling techniques. Table 1a studies the effect of instantiating MvPL with various pseudo-labeling algorithms, as outlined in §3.3. MvPL consistently improves all three algorithms by a large margin with an average absolute gain of 33.7%. Pseudo-Label [38] receives a larger gain (+39.6), presumably as it only relies on weak augmentations, while UDA [60], and FixMatch [51] are using strong augmentations (RandAugment [11]) that lead to higher baseline performance for these methods.

The results show that the MvPL framework provides a general improvement for *multiple* baselines, instead of only improving *one* baseline. This suggests that MvPL is not tied to any particular pseudo-labeling algorithm and can be used to *generally* to enhance existing pseudo-labeling algorithms for video understanding. Since FixMatch provides slightly higher performance than UDA, we use it for all subsequent experiments.

Complementarity of views. We now study how the different views contribute to the performance. We report results in Table 1b for using MvPL on the FixMatch baseline from Table 1a, with different views added one-by-one. With RGB input alone, the FixMatch model fails to learn a strong representation (48.5%). Adding complementary views that encode motion information immediately boosts performance. We observe an absolute gain of **+28.0**% when additionally using Flow to RGB frames, and a **+25.5**% gain for using temporal gradients (TG). The last row in Table 1b shows that both optical flow and temporal gradients are complementary to RGB frames as adding both views can significantly improve performance by **+30.6**%. Here, it is important to note that *test-time computation cost of all these variants is identical*, since MvPL only uses the additional views during training.

Impact of pseudo-label variants. Table 1c explores different ways to generate pseudo-labels, namely, self-supervision (i), random-supervision (ii), cross-supervision (iii) and aggregated-supervision (iv).

We make the following observations:

Self-supervision (i) and random-supervision (ii) obtain relatively low performance. This could be because self-supervision only bootstraps from its own view and does not take full advantage of other complementary views, and random-supervision randomly picks a view to generate pseudo-labels, which we hypothesize might hinder the learning process, as the learning targets change stochastically.

For cross-supervision (iii), we show two variants with different bijections[2]:

1) RGB ⇐ Flow, Flow ⇐ TG, TG ⇐ RGB;

2) RGB ⇐ TG, Flow ⇐ RGB, TG ⇐ Flow.

Both variants of cross-supervision obtain better performance than self-supervision because both optical flow and temporal gradients are complementary to RGB frames and boost overall model accuracy.

For aggregated-supervision (iv) we examine two variants:

(Exclusion): weighted average excluding self view;

(All): weighted average from all the views.

The All variant that uses aggregated-supervision from all the views obtains the best result with **79.1**%. Here, we hypothesize that predictions obtained by an ensemble of all the views are more reliable than the prediction from any single view which leads to more accurate models.

Curriculum warm-up schedule. Table 2 shows an extra ablation on UCF101 with 10% labeled data, described next.

Warm-up epochs	0	20	40	80	160
Top-1	79.1	79.5	80.4	**80.5**	80.3

Table 2: Accuracy on UCF101 with 10% labels used and a varying supervised warm-up duration. Supervised warm-up with 80 epochs obtains the best results.

Before semi-supervised training, we employ a supervised warm-up that performs training with only the labeled data. Here, we compare the performance for different warm-up durations in our 10% UCF-101 setting, *i.e.* the same setting as in Table 1 in which we train on UCF101 split-1, and use 10% of its labeled data and the entire UCF101 training set as unlabeled data.

Table 2 shows the results. Warm-up with 80 epochs obtains the best result of 80.5% accuracy, 1.3% better than not using supervised warm-up. We hypothesize that the warm-up allows the semi-supervised approach to learn with more accurate pseudo-label information in early stages of training. If the warm-up is longer, the accuracy slightly degrades, possibly because the model converges to the labeled data early, and therefore is not able to fully use the unlabeled data for semi-supervised training.

	Kinetics-400		UCF101	
Method	1%	10%	1%	10 %
Supervised	5.2	39.2	6.2	31.9
MvPL	**17.0** (+11.8)	**58.2** (+19.0)	**22.8** (+16.6)	**80.5** (+48.6)

Table 3: **Results on K400 and UCF101** when 1% and 10% of the labels are used for training. Our MvPL substantially outperforms the direct counterpart of supervised learning. Backbone: R-50, Slow-pathway [17], $T \times \tau = 8{\times}8$.

4.2. Results on Kinetics-400 and UCF-101

We next evaluate our approach for semi-supervised learning on Kinetics-400, in addition to UCF-101. We consider two settings where 1% or 10% of the labeled data are used. Again, the entire training dataset is used as unlabeled data. For K400, we form two balanced labeled subsets by sampling 6 and 60 videos per class. For UCF101, we use split 1 and sample 1 and 10 videos per class as labeled data. Evaluation is again performed on the validation sets of K400 and UCF101.

We compare our semi-supervised MvPL with the direct counterpart that uses supervised training on labeled data. Table 3 shows the results. We first look at the results in the supervised setting. With RGB input alone, the video model fails to learn a strong representation from limited labeled data: the model obtains 5.2% and 6.2% accuracy on K400 and UCF101 respectively when using 1% of the labeled data, and 39.2% and 31.9% on K400 and UCF101 when using 10% of the labels in the training data.

On Kinetics, compared to the fully supervised approach, our semi-supervised MvPL has an absolute gain of **+11.8**% and **+19.0**% when using 1% and 10% labels respectively. This substantial improvement comes without cost at test time, as, again, only RGB frames are used for MvPL inference. The gain in UCF101 is even more significant. Overall, the results show that MvPL can effectively learn a strong representation from unlabeled video data.

4.3. Comparison with self-supervised learning

In a final experiment, we consider comparisons with self-supervised learning. Here, we evaluate our approach by using UCF101 and HMDB51 as the labeled dataset and Kinetics-400 as unlabeled data. For both UCF101 and HMDB51, we train and test on all three splits and report the average performance. This setting is also common for self-supervised learning methods that are BXCpre-trained on K400 and fine-tuned on UCF101 or HMDB51. We compare with the state-of-the-art approaches [1, 43, 45, 3, 23, 65].

Table 4 shows the results. We experiment with two back-

[2] The notation A ⇐ B indicates view A uses the pseudo-labels predicted from view B.

Method	Data	Backbone	Param	T	Modalities	UCF-101	HMDB-51
XDC [1]	K400	R(2+1)D-18	15.4M	32	**V+A**	84.2	47.1
AVID [43]	K400	R(2+1)D-18	15.4M	32	**V+A**	87.5	60.8
GDT [45]	K400	R(2+1)D-18	15.4M	32	**V+A**	89.3	60.0
SpeedNet [3]	K400	S3D-G	9.1M	64	V	81.1	48.8
VTHCL [65]	K400	R-50, Slow pathway	31.8M	8	V	82.1	49.2
CoCLR [23]	K400	S3D-G	9.1M	32	V	87.9	54.6
CoCLR Two-Stream [23]	K400	2×S3D-G	2×9.1M	32	V	90.6	62.9
MvPL	K400	R-50, Slow pathway	31.8M	8	V	92.2	63.1
MvPL	K400	S3D-G	9.1M	32	V	**93.8**	**66.4**

Table 4: **Comparison to prior work on UCF101 and HMDB51.** All methods use K400 without labels. "param" indicates the number of parameters, T inference frames used, in the backbone. "Modalities" show modality used during training, where "V" is Visual and "A" is Audio input.

bones: (i) the R-50, Slow pathway [17] which we used in all previous experiments, and (ii) S3D-G [62], a commonly used backbone for self-supervised video representation learning with downstream evaluation on UCF101 and HMDB51.

When comparing to prior work, we observe that our **MvPL** obtains state-of-the-art performance on UCF101 and HMDB51 when using K400 as unlabeled data, outperforming the previous best approaches in self-supervised learning—both methods using visual (V) and audio (A) information. **MvPL** provides better performance than *e.g.* GDT [45] which is an audio-visual version of *SimCLR* [9].

In comparison to the best published vision-only approach, CoCLR [23], which is a co-training variant of *MoCo* [24] that uses RGB and optical-flow input in training, **MvPL** provides a significant performance gain of **+5.9**% and **+11.4**% top-1 accuracy on UCF101 and HMDB51, using the *identical* backbone (S3D-G) and data, and even surpasses CoCLR [23] by +3.2% and +3.5% top-1 accuracy when CoCLR is using Two-Streams of S3D-G for inference.

Discussion. We believe this is a very encouraging result. In the image classification domain, semi-supervised and self-supervised approaches compare even-handed; *e.g.* see Table 7 in [9] where *a self-supervised approach* (SimCLR) *outperforms all semi-supervised approaches* (*e.g.* Pseudo-label, UDA, FixMatch). In contrast, our state-of-the-art result suggests that for video understanding semi-supervised learning is a promising avenue for future research. This is especially notable given the flurry of research activity in self-supervised learning from video in this setting [1, 43, 45, 3, 23, 65], compared to the relative lack of past research in semi-supervised learning from video.

5. Conclusion

This paper has presented a multiview pseudo-labeling framework that capitalizes on multiple complementary views for semi-supervised learning for video. On multiple video recognition datasets, our method substantially outperforms its supervised counterpart and its semi-supervised counterpart that only considers RGB views. We obtain state-of-the-art performance on UCF-101 and HMD-B51 when using Kinetics-400 as unlabeled data. In future work we plan to explore ways to automatically retrieve the most relevant unlabeled videos to assist semi-supervised video learning.

A. Additional implementation details

All our epoch measures in the paper are based only on the *labeled* data. Therefore, training 800 and 400 epochs on a 1% and a 10% fraction of K400 corresponds to the number of iterations that 24 and 120 epochs on 100% of K400 would take respectively. Similarly, training 1200 and 600 epochs on a 1% and a 10% fraction of UCF101 corresponds to the number of iterations that 48 and 240 epochs on 100% of UCF101 would take respectively (note we use $\mu = 3$ for K400 and $\mu = 4$ for UCF101, where μ is the ratio to balance the number of labeled and unlabeled data).

The learning rate is linearly annealed for the first 34 epochs [20]. We follow the learning rate schedule used in [17] with a half-period cosine schedule [40]. In particular, the learning rate at the n-th iteration is $\eta \cdot 0.5[\cos(\frac{n}{n_{\max}}\pi) + 1]$, where $n_{\max}$ is the maximum training iterations and the base learning rate η is 0.8. We use the initialization in [25].

We adopt synchronized SGD optimization in 64 GPUs following the recipe in [20]. We train with Batch Normalization (BN) [29], and the BN statistics are computed within the clips that are on the same GPU. We use momentum of 0.9 and SGD weight decay of 10^{-4}. Dropout [53] of 0.5 is used before the final classifier layer. The mini-batch size is 4 clips per GPU (4×64=256 overall) for labeled data and $4 \times \mu$ clips per GPU ($4 \times \mu \times 64 = 256 \times \mu$ overall) for unlabeled data.

In §4.2, we use curriculum warm up with the following schedule. For K400, we train the model for 400, with 200 warm-up, epochs and 800, with 80 warm-up, epochs for the 10% and 1% subsets respectively. For UCF101, we train the model for 600, with 80 warm-up, epochs and 1200 with no warm-up epochs for the 10% and 1% subsets respectively.

References

[1] Humam Alwassel, Dhruv Mahajan, Bruno Korbar, Lorenzo Torresani, Bernard Ghanem, and Du Tran. Self-supervised learning by cross-modal audio-video clustering. In *NeurIPS*, 2020. 3, 7, 8

[2] Yuki M. Asano, Mandela Patrick, Christian Rupprecht, and Andrea Vedaldi. Labelling unlabelled videos from scratch with multi-modal self-supervision. In *NeurIPS*, 2020. 3

[3] Sagie Benaim, Ariel Ephrat, Oran Lang, Inbar Mosseri, William T Freeman, Michael Rubinstein, Michal Irani, and Tali Dekel. Speednet: Learning the speediness in videos. In *CVPR*, 2020. 7, 8

[4] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, 1998. 3, 4

[5] Andrés Bruhn, Joachim Weickert, and Christoph Schnörr. Lucas/kanade meets horn/schunck: Combining local and global optic flow methods. *International journal of computer vision*, 2005. 5

[6] Joao Carreira, Eric Noland, Andras Banki-Horvath, Chloe Hillier, and Andrew Zisserman. A short note about kinetics-600. *arXiv preprint arXiv:1808.01340*, 2018. 1

[7] Joao Carreira, Eric Noland, Chloe Hillier, and Andrew Zisserman. A short note on the kinetics-700 human action dataset. *arXiv preprint arXiv:1907.06987*, 2019. 1

[8] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017. 1

[9] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020. 8

[10] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. In *NeurIPS*, 2020. 2

[11] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *CVPRW*, 2020. 2, 5, 6

[12] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 3

[13] Navneet Dalal, Bill Triggs, and Cordelia Schmid. Human detection using oriented histograms of flow and appearance. In *ECCV*, 2006. 3

[14] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. 5

[15] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *Second Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, In conjunction with the ICCV*, 2005. 3

[16] Haoqi Fan, Yanghao Li, Bo Xiong, Wan-Yen Lo, and Christoph Feichtenhofer. PySlowFast. https://github.com/facebookresearch/slowfast, 2020. 5

[17] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *ICCV*, 2019. 1, 5, 6, 7, 8

[18] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *CVPR*, 2016. 3

[19] Rohit Girdhar, Du Tran, Lorenzo Torresani, and Deva Ramanan. Distinit: Learning video representations without a single labeled video. In *ICCV*, 2019. 2

[20] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017. 5, 8

[21] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense. In *ICCV*, 2017. 1

[22] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *Neurips*, 2005. 2

[23] Tengda Han, Weidi Xie, and Andrew Zisserman. Self-supervised co-training for video representation learning. In *Neurips*, 2020. 3, 7, 8

[24] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 8

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *CVPR*, 2015. 8

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5

[27] Berthold KP Horn and Brian G Schunck. Determining optical flow. In *Techniques and Applications of Image Understanding*, volume 281, pages 319–331. International Society for Optics and Photonics, 1981. 3

[28] Olivier J. Hénaff, Aravind Srinivas, Jeffrey De Fauw, Ali Razavi, Carl Doersch, S. M. Ali Eslami, and Aaron van den Oord. Data-efficient image recognition with contrastive predictive coding. In *ICML*, 2020. 2

[29] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 8

[30] Longlong Jing, Toufiq Parag, Zhe Wu, Yingli Tian, and Hongcheng Wang. Videossl: Semi-supervised learning for video classification. *arXiv:2003.00197*, 2020. 2

[31] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv:1705.06950*, 2017. 1, 2, 6

[32] Evangelos Kazakos, Arsha Nagrani, Andrew Zisserman, and Dima Damen. Epic-fusion: Audio-visual temporal binding for egocentric action recognition. In *ICCV*, 2019. 3

[33] Alexander Kläser, Marcin Marszałek, and Cordelia Schmid. A spatio-temporal descriptor based on 3d-gradients. In *BMVC*, 2008. 3

[34] Bruno Korbar, Du Tran, and Lorenzo Torresani. Cooperative learning of audio and video models from self-supervised synchronization. In *Neurips*, 2018. 3

[35] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: A large video database for human motion recognition. In *ICCV*, 2011. 2

[36] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *ICCV*, 2011. 6

[37] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *ICLR*, 2017. 2

[38] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, 2013. 1, 2, 4, 6

[39] Ce Liu et al. *Beyond pixels: exploring new representations and applications for motion analysis*. PhD thesis, Massachusetts Institute of Technology, 2009. 5

[40] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 5, 8

[41] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *CVPR*, 2020. 3

[42] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 2018. 2

[43] Pedro Morgado, Vasconcelos Nuno, and Misra Ishan. Audio-visual instance discrimination with cross-modal agreement. *arXiv preprint arXiv:2004.12943*, 2020. 7, 8

[44] Jonathan Munro and Dima Damen. Multi-modal Domain Adaptation for Fine-grained Action Recognition. In *CVPR*, 2020. 3

[45] Mandela Patrick, Yuki M Asano, Ruth Fong, João F Henriques, Geoffrey Zweig, and Andrea Vedaldi. Multi-modal self-supervision from generalized data transformations. *arXiv preprint arXiv:2003.04298*, 2020. 3, 7, 8

[46] Rui Qian, Tianjian Meng, Boqing Gong, Ming-Hsuan Yang, Huisheng Wang, Serge Belongie, and Yin Cui. Spatiotemporal contrastive video representation learning. *arXiv preprint arXiv:2008.03800*, 2020. 3

[47] Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *Neurips*, 2016. 2

[48] Gunnar A Sigurdsson, Olga Russakovsky, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Much ado about time: Exhaustive annotation of temporal data. *arXiv preprint arXiv:1607.07429*, 2016. 1

[49] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Neurips*, 2014. 1, 3

[50] Lucas Smaira, João Carreira, Eric Noland, Ellen Clancy, Amy Wu, and Andrew Zisserman. A short note on the kinetics-700-2020 human action dataset. *arXiv preprint arXiv:2010.10864*, 2020. 1

[51] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv preprint arXiv:2001.07685*, 2020. 1, 2, 4, 5, 6

[52] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 2, 6

[53] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 2014. 5, 8

[54] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Neurips*, 2017. 2

[55] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015. 1

[56] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, 2018. 1

[57] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *ICCV*, 2013. 1, 3

[58] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, 2016. 3

[59] Fanyi Xiao, Yong Jae Lee, Kristen Grauman, Jitendra Malik, and Christoph Feichtenhofer. Audiovisual slowfast networks for video recognition. *arXiv preprint arXiv:2001.08740*, 2020. 3

[60] Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. Unsupervised data augmentation for consistency training. *arXiv preprint arXiv:1904.12848*, 2019. 2, 4, 5, 6

[61] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V. Le. Self-training with noisy student improves imagenet classification. In *CVPR*, 2020. 1, 2

[62] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning for video understanding. In *ECCV*, 2018. 8

[63] Dejing Xu, Jun Xiao, Zhou Zhao, Jian Shao, Di Xie, and Yueting Zhuang. Self-supervised spatiotemporal learning via video clip order prediction. In *CVPR*, 2019. 3

[64] I Zeki Yalniz, Hervé Jégou, Kan Chen, Manohar Paluri, and Dhruv Mahajan. Billion-scale semi-supervised learning for image classification. *arXiv preprint arXiv:1905.00546*, 2019. 1, 2

[65] Ceyuan Yang, Yinghao Xu, Bo Dai, and Bolei Zhou. Video representation learning with visual tempo consistency. *arXiv preprint arXiv:2006.15489*, 2020. 7, 8

[66] Ming Zeng, Tong Yu, Xiao Wang, Le T Nguyen, Ole J Mengshoel, and Ian Lane. Semi-supervised convolutional neural networks for human activity recognition. In *IEEE International Conference on Big Data*, 2017. 2, 3

[67] Yue Zhao, Yuanjun Xiong, and Dahua Lin. Recognize actions by disentangling components of dynamics. In *CVPR*, 2018. 3

[68] Linchao Zhu and Yi Yang. Actbert: Learning global-local video-text representations. In *CVPR*, 2020. 3