

# ClusterFit: Improving Generalization of Visual Representations

Xueting Yan\* Ishan Misra\* Abhinav Gupta Deepti Ghadiyaram† Dhruv Mahajan†  
Facebook AI

## Abstract

Pre-training convolutional neural networks with weakly-supervised and self-supervised strategies is becoming increasingly popular for several computer vision tasks. However, due to the lack of strong discriminative signals, these learned representations may overfit to the pre-training objective (e.g., hashtag prediction) and not generalize well to downstream tasks. In this work, we present a simple strategy - ClusterFit (CF) to improve the robustness of the visual representations learned during pre-training. Given a dataset, we (a) cluster its features extracted from a pre-trained network using  $k$ -means and (b) re-train a new network from scratch on this dataset using cluster assignments as pseudo-labels. We empirically show that clustering helps reduce the pre-training task-specific information from the extracted features thereby minimizing overfitting to the same. Our approach is extensible to different pre-training frameworks – weak- and self-supervised, modalities – images and videos, and pre-training tasks – object and action classification. Through extensive transfer learning experiments on 11 different target datasets of varied vocabularies and granularities, we show that CF significantly improves the representation quality compared to the state-of-the-art large-scale (millions / billions) weakly-supervised image and video models and self-supervised image models.

## 1. Introduction

Weak and self-supervised pre-training approaches offer scalability by exploiting free annotation. But *there is no free lunch* – these methods often first optimize a *proxy objective* function, for example, predicting image hashtags [31] or color from grayscale images [34, 63]. Similar to supervised pre-training, the underlying assumption (hope) is that this proxy objective function is fairly well aligned with the subsequent transfer tasks, thus optimizing this function could potentially yield suitable pre-trained visual representations. While this assumption holds mostly true in case of fully-supervised pre-training, it may not extend to weak and

Pre-training method ( $N_{pre}$ )	$\Delta$ of CF ( $N_{cf}$ ) on transfer
<b>Fully-supervised Images</b> §3.2, Figure 3b ResNet-50, ImageNet-1K, 1K labels	+2.1% on ImageNet-9K [10]
<b>Weakly-supervised Images</b> §4.1.1, Table 4 ResNet-50, 1B Images, 1.5K hashtags [38]	+4.6% on ImageNet-9K [10] +5.8% on iNaturalist [55]
<b>Weakly-supervised Videos</b> §4.1.2, Table 5 R(2+1)D-34, 19M videos, 438 hashtags [20]	+3.2% on Kinetics [59] +4.3% on Sports1M [32]
<b>Self-supervised Images</b> §4.2, Tables 6 ResNet-50, 1M images Jigsaw [42] and RotNet [21], Multi-task	+7-9% on ImageNet-1K [47] +3-7% mAP on VOC07 [15] +3-5% on Places205 [65]

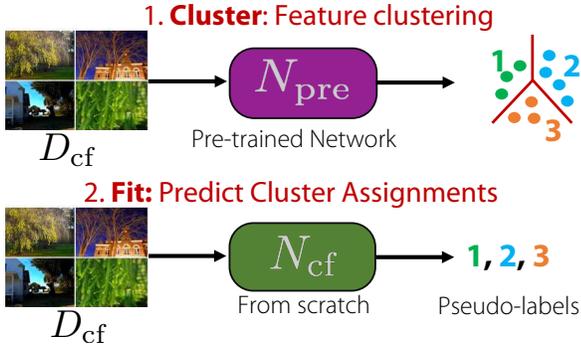
**Table 1: A summary of results:** We show that **ClusterFit (CF)** can be applied to a variety of different pre-training methods, modalities, and architectures. We report absolute gains in top-1 accuracy (except for VOC07 where we report mAP). In each setting, CF provides improvements with the *same* model architecture and *without* additional data or supervision.

self-supervision. In the latter pre-training cases, the lack of strong discriminative signals may result in an undesirable scenario where the visual representations overfit to the idiosyncrasies of the pre-training task and dataset instead, thereby rendering them unsuitable for transfer tasks. For instance, it was noted in [20, 38, 51] that factors such as label noise, polysemy (apple the fruit vs. Apple Inc.), linguistic ambiguity, lack of ‘visual’ness of tags (e.g. #love) significantly hampered the pre-training proxy objective from being well-aligned with the transfer tasks. Further, the authors of [23, 64] studied multiple self-supervised methods and observed that, compared to earlier layers, features from the last layer are more “aligned” with the proxy objective, and thus generalize poorly to target tasks.

In this work, we ask a simple question – is there a way to avoid such overfitting to the proxy objective during weak- and self-supervised pre-training? Can we overcome the ‘artifacts’ of proxy objectives so that the representation is generic and transferable? Our key insight is that *smoothing* the feature space learned via proxy objectives should help us remove these artifacts and avoid overfitting to the the proxy objective. But how do we smoothen the feature space? Should it be done while optimizing the proxy objective or in a post-hoc manner?

To this end, we propose a surprisingly simple yet effective framework called **ClusterFit (CF)**. Specifically, given a pre-trained network trained using a proxy objective and a new dataset, we first use the learned feature space to cluster that dataset. Next, we train a new network from scratch on this new dataset using the cluster memberships

\*† Equal Contribution



**Figure 1: ClusterFit (CF):** We start with a pre-trained network ( $N_{pre}$ ) that is trained on some pre-training task (not shown). We use this network to extract features and cluster a new dataset  $D_{cf}$  using k-means clustering. We show that training a new network  $N_{cf}$  from scratch on these cluster assignments as labels results in a more transferable feature representation.

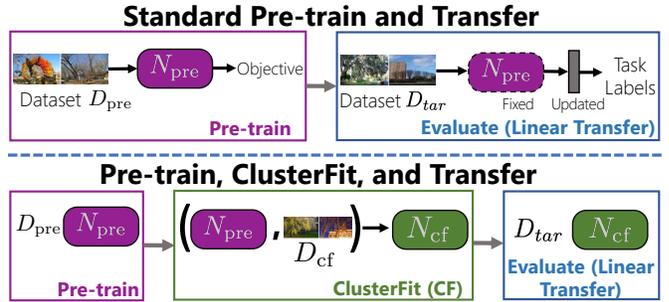
as pseudo labels (Figure 1). We demonstrate that clustering of the features helps retain only the essential invariances in them and eliminates proxy objective’s artifacts (essentially smoothing the feature space). Re-training on the cluster memberships yields a visually coherent pre-training feature space for downstream tasks. Our approach of feature space smoothing is guided through unsupervised k-means clustering, making it scalable to millions (billions) of videos and images in both weak- and self-supervised pre-training frameworks.

We take inspiration from recent work in self-supervised learning which aims to learn a smooth visual feature space via clustering and trains representations on the clusters as classes [6, 7, 44]. While [6, 7] use clustering as the training objective itself, in our work, we investigate the value of post-hoc smoothing. ClusterFit can also be viewed as a variant of knowledge distillation [29] that distills via ‘lossy’ clustering, as opposed to the standard setup of using soft targets in original label space.

ClusterFit demonstrates significant performance gains on a total of 11 public, challenging image and video benchmark datasets. As summarized in Table 1, our approach, while extremely simple, consistently improves performance across different pre-training methods, input modalities, network architectures, and benchmark datasets.

## 2. Related Work

**Weakly Supervised Learning:** Training ConvNets on very large, weakly supervised images by defining the proxy tasks using the associated meta-data [11, 20, 25, 31, 36, 38, 48, 51–53, 56] has shown tremendous benefits. Proxy tasks include hashtags predictions [11, 20, 25, 38, 56], GPS [27, 58], search queries prediction [51], and word or n-grams predictions [31, 36]. Our approach builds upon these works and shows that even better representations can be trained by leveraging the features from such pre-training frameworks for clustering to mitigate the effect of noise.



**Figure 2: Full ClusterFit pipeline:** A typical transfer learning framework involves two stages: pre-training followed by transfer learning. ClusterFit introduces a step between these stages. We evaluate all representations by training a linear classifier on *fixed* ConvNet weights.

Yalniz *et al.* [62] propose a target task specific noise removal framework by ranking images for each class by their softmax values and retaining only top- $K$  images for re-training. However, their method is specific to a particular target task and discards most of the data during re-training. By contrast, our approach does not adhere to a particular target task and leverages all the data, since, they may contain complementary visual information beyond hashtags.

**Self-Supervised Learning:** Self-supervised approaches typically learn a feature representation by defining a ‘pre-text’ task on the visual domain. These pre-text tasks can either be domain agnostic [5, 6, 30, 45, 60, 61] or exploit domain-specific information like spatial structure in images [13, 21, 42–44], color [12, 34, 35, 63, 64], illumination [14], temporal structure [16, 26, 37, 39, 40] or a co-occurring modality like sound [2, 3, 9, 19, 46]. In this work, we use two diverse image-based self-supervision approaches - Jigsaw [42] and RotNet [21] that have shown competitive performance [7, 23, 33]. Since the difference between pretext tasks and semantic transfer learning tasks is huge, our method shows much larger improvement for self-supervised methods (§4.2).

Our work builds upon [6, 7], who use clustering and pseudo-labels for self-supervised learning and [44], who distill different self-supervised models to a common architecture. Compared to [6, 7], ClusterFit does not require any alternate optimization and thus is more stable and computationally efficient. As we show in §4, this property makes ClusterFit applicable to different modalities and large-scale data. In addition to the stark difference in the performance of ClusterFit and [44] (Table 1), there are other fundamental differences: (a) Unlike [44], our focus is not on distilling information to a common architecture, but instead to remove the pre-training task biases; (b) ClusterFit is applicable beyond the self-supervised framework (§4.2) and extends to fully supervised (§3.2) and weakly supervised (§4.1) frameworks. We highlight that this is not an obvious result since intuitively it is unclear that any approach can overcome task biases in strong and weak supervision paradigms.

Dataset	Label Type	# classes	Train/Eval	Metric
Weakly-supervised Images §4.1.1				
ImageNet-1K [47]	multi-class object	1000	1.3M/50K	top-1 acc
ImageNet-9K [10]	multi-class object	9000	10.5M/450K	top-1 acc
Places365 [65]	multi-class scene	365	1.8M/36.5K	top-1 acc
iNaturalist 2018 [55]	multi-class object	8142	438K/24K	top-1 acc
Weakly-supervised Videos §4.1.2				
Kinetics [59]	multi-class action	400	246K/20K	top-1 acc
SportsIM [32]	multi-class action	487	882K/204K	top-1 acc
Something-Something V1 [24]	multi-class action	174	86K/11.5K	top-1 acc
Self-supervised Images §4.2				
VOC07 [15]	multi-label object	20	5K/5K	mAP
ImageNet-1K [47]	multi-class object	1000	1.3M/50K	top-1 acc
Places205 [65]	multi-class scene	205	2.4M/21K	top-1 acc
iNaturalist 2018 [55]	multi-class object	8142	438K/24K	top-1 acc

**Table 2: Target tasks for Transfer Learning** used for evaluating feature representations.

**Model Distillation and Clustering:** Model distillation [1, 4, 18, 29] typically involves transferring knowledge from a ‘teacher’ model to a ‘student’ model by training the student on predictions of the teacher in addition to task labels. These methods are designed to transfer knowledge (not contained in the labels) about the task from the teacher to the student network. Since distillation retains more knowledge about the original task, it performs poorly in the case of weak-supervision (§4.1). Interestingly, the failure of standard knowledge distillation approaches in the context of self-supervised learning has also been shown in [44]. While Girard *et al.* [22] use a supervised clustering approach relying on a preexisting label hierarchy, our clustering framework is completely unsupervised and thus extensible to self-supervised training regime.

### 3. Approach

Our goal is to learn a generalizable feature space for a variety of target tasks that does not overfit to the pre-training proxy objective. We first describe the framework of ClusterFit (CF) in §3.1. Next, we report a control experiment on the ImageNet-1K dataset that sheds light on how CF combats the ‘bias’ introduced due to the proxy objective (§3.2).

#### 3.1. ClusterFit Framework

Our method starts with a ConvNet  $N_{pre}$  that is pre-trained on a dataset  $D_{pre}$  and labels  $L_{pre}$ . First, we use the penultimate layer of  $N_{pre}$  to extract features from each datapoint belonging to another dataset  $D_{cf}$ . Next, we **cluster** these features using k-means into  $K$  groups and treat these cluster assignments as the new categorical ‘labels’ ( $L_{cf}$ ) for  $D_{cf}$ . Finally, we **fit** a different network  $N_{cf}$  (initialized from scratch<sup>1</sup>) on  $D_{cf}$  that minimizes a cross-entropy objective on  $L_{cf}$ . We illustrate these steps in Figure 1. We highlight that re-learning  $N_{cf}$  from scratch on  $D_{cf}$  is completely unsupervised and thus allows leveraging large-scale datasets. **Intuition:** We hypothesize that ClusterFit (CF) leverages the underlying visual smoothness in the feature space to

<sup>1</sup>We empirically found that reusing weights from  $N_{pre}$  to initialize  $N_{cf}$  did not yield any benefit across any of the 11 transfer tasks.

create visually coherent clusters. We believe that ‘cluster’ followed by ‘fit’ weakens the underlying pre-training objective-specific bias. One may view ClusterFit from an information bottleneck [54] perspective wherein the ‘lossy’ clustering step introduces a bottleneck and removes any pre-training proxy objective bias.

**How to evaluate CF?** As in prior efforts [20, 23, 38], we use transfer learning performance on downstream tasks to understand whether CF improves generalization of the feature representations. Specifically, to evaluate  $N_{pre}$  and  $N_{cf}$ , we train linear classifiers on *fixed* feature representations from the networks on the downstream task and report final performance on held-out data (see Table 2). Figure 2 illustrates ClusterFit’s setup. We stress that ClusterFit is simple to implement and makes minimal assumptions about input modalities, architectures *etc.* but provides a powerful way to improve the generalization of the feature space. We explore various design choices ClusterFit offers such as relative properties of  $N_{pre}$ ,  $N_{cf}$ ,  $D_{pre}$ , and  $D_{cf}$  in §5.

#### 3.2. Control Experiment using Synthetic Noise

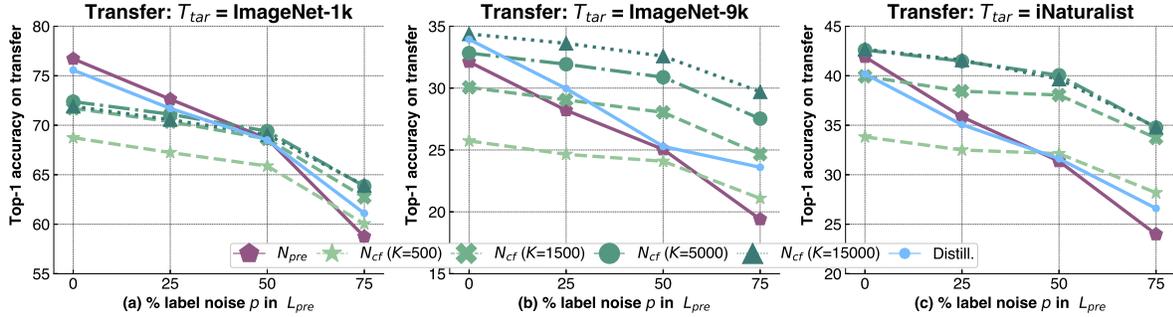
Here, our goal is to study the extent of generalization of features learned from a ‘proxy’ pre-training objective in a controlled setup. We start with a supervised pre-training dataset ImageNet-1K [47], and add synthetic label noise to it. Our motive behind this setup is to intentionally misalign the pre-training objective with downstream tasks. We acknowledge that the synthetic noise simulated in this experiment is an over simplification of the complex noise present in real world data. Nevertheless, it provides several key insights into ClusterFit as we show next.

**Control Experiment Setup:** To isolate the effect of CF, in this experiment, we fix  $D_{pre} = D_{cf} = \text{ImageNet-1K}$  and the network architectures  $N_{pre}$  and  $N_{cf}$  to ResNet-50 [28]. We start by adding varying amounts ( $p\%$ ) of uniform random label noise<sup>2</sup> to  $D_{pre}$ . Next, we train a separate  $N_{pre}$  for each fraction  $p$  of the noisy labels. We then apply CF (with different values of  $K$  in k-means) to each  $N_{pre}$  to obtain a corresponding  $N_{cf}$ . Finally, we evaluate the representations by training linear classifiers on fixed res5 features on three target image classification datasets - ImageNet-1K, ImageNet-9K, and iNaturalist. We use model distillation [29] as a baseline to better understand the behavior of ClusterFit.

Our motivation behind this setup is the following: when  $p = 0$ ,  $N_{pre}$  denotes the true, noise-free supervised task; as  $p$  increases, the proxy objective becomes a poorer approximation of the original pre-training objective and allows us to closely inspect ClusterFit.

**Results and Observations:** We report the transfer learning performance of  $N_{pre}$  (i.e., before CF) and  $N_{cf}$  (i.e., after

<sup>2</sup>We randomly replace a label ( $l$ ) in ImageNet-1K train split with one that is obtained by uniformly sampling from ImageNet-1K labels excluding  $l$ .



**Figure 3: Control Experiment:** We inject uniform label noise in the labels  $L_{pre}$  from  $D_{pre} = \text{ImageNet-1K}$  and train a separate ResNet-50 model ( $N_{pre}$ ) on these noisy labels. We apply ClusterFit on each of these pre-trained models ( $N_{pre}$ ) and vary the number of clusters  $K$  to train  $N_{cf}$ . We then study the transfer learning performance of the representations by training a linear classifier on fixed features from  $N_{pre}$  or  $N_{cf}$  on three target tasks - a noise free ImageNet-1K, ImageNet-9K, and iNaturalist. ClusterFit is able to learn more transferable features despite high amounts of label noise in pre-training. For finer-grained target tasks like ImageNet-9K, ClusterFit can even improve a fully supervised ResNet-50 model ( $p = 0$ ).

CF) in Figure 3 for different values of label noise  $p$ . Let us first consider  $p = 0$ , *i.e.*, a setting without any label noise. In this case,  $N_{pre}$  is trained on clean labels. On the target dataset ImageNet-1K,  $N_{pre}$  performs significantly better than  $N_{cf}$  for all values of  $K$  (Fig. 3 (a)). This is expected, since when  $D_{pre} = D_{tar} = \text{ImageNet-1K}$ , the pre-training and transfer tasks are exactly aligned. However,  $N_{cf}$  performs comparably or better than  $N_{pre}$  for other target datasets - ImageNet-9K and iNaturalist at higher values of  $K$ . This suggests that CF can **improve even fully-supervised** representations for more fine-grained downstream tasks. We note that model distillation also provides an improvement over  $N_{pre}$  on ImageNet-9K but is worse on iNaturalist.

Let us now consider scenarios where  $p > 0$ . Figure 3 indicates that increased label noise ( $p$ ) in  $D_{pre}$  translates to poor performance across all three target tasks. We highlight that the drop in the performance is more drastic for  $N_{pre}$  (*i.e.*, before CF), than for  $N_{cf}$  (*i.e.*, after CF). More importantly, the performance gap between  $N_{cf}$  and  $N_{pre}$  continues to increase with  $p$ . From Fig. 3 (b) and (c), we observe that  $N_{cf}$  consistently outperforms  $N_{pre}$  on two target tasks ImageNet-9K and iNaturalist. Notably, for  $D_{tar} = \text{ImageNet-1K}$  (Figure 3 (a)), when  $p \geq 50$ ,  $N_{cf}$  outperform  $N_{pre}$ , which is pre-trained on noisy ImageNet-1K. Model distillation provides some gains over  $N_{pre}$  but is consistently outperformed by ClusterFit.

These results suggest that as  $p$  increases, the proxy objective gets further away from the ‘true’ pre-training objective, and makes features from  $N_{pre}$  less transferable. In those very cases, CF captures useful visual invariances in the feature representations, thereby providing more noise-resilient pseudo-labels for learning transferable representations. Finally, we also note that larger number of clusters  $K$  generally leads to better transfer learning performance. The gains are larger for more challenging and fine-grained datasets like ImageNet-9K and iNaturalist. We study the effect of this hyper-parameter  $K$  in §5.

## 4. Experiments

We now examine the broad applicability of ClusterFit in three different pre-training scenarios for  $N_{pre}$ : (a) weakly-supervised pre-training for images (§4.1.1), (b) weakly-supervised pre-training for videos (§4.1.2), and (c) self-supervised pre-training for images (§4.2).

**Common CF Setting:** Throughout this section, we set  $D_{pre} = D_{cf}$  and  $N_{pre} = N_{cf}$  (architecture-wise). We train  $N_{pre}$  on  $D_{pre}$ ,  $N_{cf}$  on  $D_{cf}$  for equal number of epochs. Table 3 summarizes these settings. By keeping the data, architecture, and training schedule constant, we hope to measure the difference in performance between  $N_{cf}$  and  $N_{pre}$  solely due to ClusterFit.

**Evaluation:** As mentioned in §3.1, we evaluate ClusterFit via transfer learning on target tasks. Specifically, we train linear classifiers on the *fixed* features obtained from the penultimate layer of  $N_{pre}$  or  $N_{cf}$  on target datasets. The transfer learning tasks are summarized in Table 2.

**Baselines:** We use the following baselines:

- **$N_{pre}$ :** We use features from  $N_{pre}$  for transfer learning. Since ClusterFit (CF) is applied on  $N_{pre}$  to get  $N_{cf}$ , this baseline serves to show improvements through CF.
- **Distillation:** To empirically understand the importance of the clustering step in CF, we compare with model distillation [29]. Unlike CF, distillation transfers knowledge from  $N_{pre}$  *without clustering*, thus retaining more information about the learned features. We train a distilled model using a weighted average of 2 loss functions: (a) cross-entropy with soft targets computed using  $N_{pre}$  and temperature  $T$  and (b) cross-entropy with image/video labels in weakly-supervised setup. We also experimented with training a network to directly regress the features from  $N_{pre}$  but found consistently worse results.
- **Prototype:** ClusterFit uses unsupervised k-means to create pseudo-labels. To understand the effect of this unsupervised step, we add a baseline that uses *semantic information* during clustering. Under this prototype align-

Pre-training method	$D_{pre} = D_{cf}$	Arch. of $N_{pre}$ & $N_{cf}$
Weakly-Supervised Images §4.1.1	IG-ImageNet-1B	ResNet-50
Weakly-Supervised Videos §4.1.2	IG-Verb-19M	R(2+1)D-34
Self-supervised Images §4.2	ImageNet-1k	ResNet-50

**Table 3: Data and model architectures used in §4:** weakly supervised videos, weakly supervised images, and self supervised images. In each setting, we train  $N_{pre}$  and  $N_{cf}$  for equal number of epochs.

$D_{tar}$	$N_{pre}$	$N_{pre}2\times$	Distill.	Prototype	CF ( $N_{cf}$ ), $K \rightarrow$				
					1.5k	3.75k	7.5k	15k	30k
ImageNet-1K	78.0	<b>78.8</b>	73.8	76.9	75.3	76.1	<u>76.5</u>	<u>76.5</u>	76.2
ImageNet-9K	32.9	34.1	29.1	35.1	33.5	35.4	36.4	37.1	<b>37.5</b>
Places365	<u>51.2</u>	51.2	49.9	51.9	52.0	52.1	52.4	<b>52.6</b>	52.1
iNaturalist	43.9	45.3	35.9	49.0	43.8	46.4	47.9	<b>49.7</b>	49.5

**Table 4: Weakly-supervised Images:** Top-1 accuracy for various transfer learning datasets with  $D_{pre} = D_{cf} = \text{IG-ImageNet-1B}$  and the same architecture (ResNet-50) for  $N_{pre}$  and  $N_{cf}$ .

ment [49] baseline, unlike random cluster initialization as done in k-means, we use label information in  $D_{cf}$  to initialize cluster centers. Specifically, we first set  $K$  equal to the number of ‘classes’ in  $D_{cf}$ . Here, each cluster corresponds to a ‘prototype’ of that class. We then compute  $K$  prototypes by averaging image embeddings of all images belonging to each class. Finally, pseudo-labels are assigned to each data point by finding its nearest ‘prototype’ cluster center. Since this method uses explicit label information present in  $D_{cf}$ , it requires more ‘supervision’ than ClusterFit. We also note that this baseline is not applicable to self-supervised methods (suppl. material).

- **Longer pre-training:** Since  $N_{cf}$  is trained for the same number of epochs as  $N_{pre}$ , we also compare against a network trained on the pre-train task for  $2\times$  longer (denoted by  $N_{pre}2\times$ ). Specifically,  $N_{pre}2\times$  is trained for a combined number of epochs as  $N_{pre}$  and  $N_{cf}$ . By comparing  $N_{cf}$  against this baseline, we hope to isolate improvements due to longer pre-training.

## 4.1. Weakly-supervised pre-training

In this section, we study weakly-supervised pre-training on noisy web images and videos. These approaches predict the noisy hashtags associated with images/videos and thus minimize a *proxy objective* during pre-training.

### 4.1.1 Weakly-supervised image pre-training

**Data and Model:** As in [38], we collect IG-ImageNet-1B dataset of 1B public images associated with hashtags from a social media website. To construct this dataset, we consider images tagged with at least one hashtag that maps to any of the ImageNet-1K synsets. The architecture of  $N_{pre}$  and  $N_{cf}$  network is fixed to a ResNet-50 [28], while  $D_{pre} = D_{cf} = \text{IG-ImageNet-1B}$ .

**ClusterFit Details:** We extract features from the 2048 dimensional res5 layer from  $N_{pre}$  for clustering.  $N_{cf}$  is trained from scratch on  $D_{cf} = \text{IG-ImageNet-1B}$  on the cluster assignments as pseudo-labels. Details on the hyper pa-

rameters during pre-training and ClusterFit are provided in the supplementary material. We report results in Table 4, which we discuss next.

**Effect of longer pre-training:**  $N_{pre}$  pre-trained on  $D_{pre} = \text{IG-ImageNet-1B}$  already exhibits very strong performance on all target datasets. By construction, the label space of the target dataset ImageNet-1K matches with that of  $D_{pre}$ . As noted in [38], this translates to  $N_{pre}$  yielding an impressive top-1 accuracy of 78% on ImageNet-1K. Features from longer pre-training ( $N_{pre}2\times$ ) show improvements on ImageNet-1K, ImageNet-9K, and iNaturalist but not on Places365. As noted in [31, 38], Places365 is not well-aligned with ImageNet-1K (and by extension with IG-ImageNet-1B). Thus, (longer) pre-training yields no benefit. By contrast, the target dataset ImageNet-9K is well-aligned with  $D_{pre} = \text{IG-ImageNet-1B}$ , thus achieving improvements from longer pre-training.

**Comparison with Model Distillation:** Training a student network via distillation, *i.e.*, soft targets provided by the teacher ( $N_{pre}$ ) and hashtags, performs worse than  $N_{pre}$  itself. In our case, the student and teacher network are of the same capacity (ResNet-50). We believe that the noisy label setting combined with the same capacity student and teacher networks are not ideal for model distillation.

**Comparison with Prototype:** Except on ImageNet-1K, the prototype baseline shows improvement over both  $N_{pre}$  and  $N_{pre}2\times$ . This shows that pseudo-labels derived based on label information can provide a better training objective than hashtags used for pre-training  $N_{pre}$ . However, similar to CF, prototype shows a reduction in performance on ImageNet-1K which we explain next.

**Gains of ClusterFit:**  $N_{cf}$  achieves substantial gains over the strong  $N_{pre}$  model especially on fine-grained datasets like ImageNet-9K (4.6 points) and iNaturalist (5.8 points), at higher values of  $K$ . This may be because  $N_{cf}$  captures a more diverse and finer-grained visual feature space that benefits fine-grained transfer tasks. We observe a small decrease in the performance on ImageNet-1K (1.5 points) which can be attributed again to the hand-crafted label alignment of the IG-ImageNet-1B with ImageNet-1K. This result is inline with observations from [38]. We believe the performance decrease of ‘prototype’ on ImageNet-1K is also due to this reason.  $N_{cf}$  shows improved performance than ‘prototype,’ yet does not use any additional supervision while generating pseudo-labels. Finally, we note that finding an optimal number of clusters  $K$  for each transfer learning task is procedurally easier than finding a pre-training task (or label space) that aligns with the target task.

### 4.1.2 Weakly-supervised video pre-training

**Data and Model:** Following [20], we collect IG-Verb-19M, a dataset of 19M public videos with hashtags from

$T_{tar}$	$N_{pre}$	$N_{pre}2\times$	Distill.	Prototype	CF ( $N_{cf}$ ), $K \rightarrow$				
					400	800	1600	3200	6400
Kinetics	68.8	69.2	63.6	70.3	70.1	71.2	71.2	71.5	<b>72.0</b>
Sports1M	52.9	53.1	48.4	55.1	55.8	56.6	57.1	<b>57.2</b>	57.2
Sth-Sth V1	16.9	16.4	15.6	20.3	20.2	20.0	<b>20.6</b>	19.3	19.7

**Table 5: Weakly-supervised videos:** Top-1 accuracy for various transfer learning datasets with  $D_{pre} = D_{cf} = \text{IG-Verb-19M}$  and the same architecture (R(2+1)D-34) for  $N_{pre}$  and  $N_{cf}$ .

a social media website. We consider videos tagged with at least one of the 438 verbs from Kinetics [59] and VerbNet [57]. We set  $D_{pre} = D_{cf} = \text{IG-Verb-19M}$ . We use the clip-based R(2+1)D-34 [8] architecture for  $N_{pre}$  and  $N_{cf}$ . Each video clip is generated by scaling its shortest edge to 128 followed by cropping a random patch of size  $112 \times 112$ . We use 32 consecutive frames per video clip, with temporal jittering applied to the input.

**ClusterFit details:** We uniformly sample 6 clips of 32 consecutive frames per video, extract video features per clip, and average pool them. We use the 512 dimensional res5 layer from  $N_{pre}$ . We direct the reader to the supplementary material for hyper-parameter details.

**Observations:** We present the transfer learning results in Table 5. Once again, the baseline  $N_{pre}$  exhibits strong performance on all target datasets. Longer pre-training ( $N_{pre}2\times$ ) provides limited benefit on Kinetics and Sports1M, and loses performance compared to  $N_{pre}$  on Sth-Sth V1. As observed in §4.1.1, model distillation performs worse than  $N_{pre}$  on all target datasets.

We observe that CF ( $N_{cf}$ ) provides significant improvements of **3.2 - 4.3%** across all the datasets over  $N_{pre}$ . The optimal number of clusters  $K$  vary depending on each dataset, but is typically an order of magnitude higher than the size of the original label space (i.e., 438 verbs in IG-Verb-19M). For example, performance does not saturate for Kinetics even at  $K = 6400$ . We study the effect of  $K$  in §5.2.

## 4.2. Self-Supervised pre-training for Images

We now apply ClusterFit framework to self-supervised methods. We study two popular and diverse self-supervised methods - Jigsaw [42] and RotNet [21]. These methods do not use semantic labels and instead create pre-training labels using a ‘pre-text’ task such as rotation. As mentioned in §2 and [44], distillation is not a valid baseline for these self-supervised methods (more in supplementary material). Also, as these methods do not use semantic label information, ‘prototype’ is also not a valid baseline.

**Data and Model:** We fix the network architectures of  $N_{pre}$  and  $N_{cf}$  to ResNet-50. We also fix  $D_{pre} = D_{cf} = \text{ImageNet-1K}$  to pre-train Jigsaw and RotNet models ( $N_{pre}$ ). We discard the semantic labels and use only images from both tasks. We use the models released by [23] for Jigsaw and train RotNet models following the approach in [21, 23].

**ClusterFit Details:** We set  $K = 16,000$ .  $N_{cf}$  is trained

Method	$T_{tar}$			
	ImageNet-1K	VOC07	Places205	iNaturalist
Jigsaw $N_{pre}$	46.0	66.1	39.9	22.1
Jigsaw $N_{pre}2\times$	45.1	65.4	38.7	21.8
Jigsaw $N_{cf}$ (Ours)	<b>55.2</b>	<b>69.5</b>	<b>45.0</b>	<b>29.8</b>
RotNet $N_{pre}$	48.9	63.9	41.4	23.0
RotNet $N_{pre}2\times$	50.0	64.9	42.9	25.3
RotNet $N_{cf}$ (Ours)	<b>56.1</b>	<b>70.9</b>	<b>44.8</b>	<b>28.4</b>

**Table 6: Self-supervised methods:** We apply ClusterFit to self-supervised methods and evaluate them following the setup in [23] on four datasets by training a linear classifier on fixed features. All methods use the ResNet-50 architecture for  $N_{pre}$  and  $N_{cf}$ . We report the performance of the best performing layer for each method and use the mean Average Precision (mAP) metric for the VOC07 dataset and top-1 accuracy for all other datasets.

for the same number of epochs as the pre-trained self-supervised network  $N_{pre}$ . We strictly follow the training hyper parameters and the transfer learning setup outlined in Goyal *et al.* [23]. We report additional results for different values of  $K$  in the supplemental material.

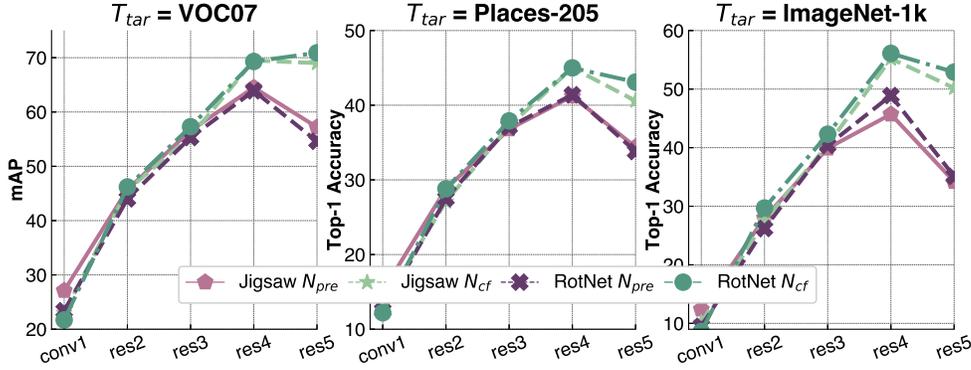
**Layer-wise transfer:** In Figure 4, we report the transfer learning performance of each layer of  $N_{pre}$  and compare with  $N_{cf}$  after applying ClusterFit. We see that for the pre-trained network  $N_{pre}$ , res5 features transfer poorly compared to res4 features. For example, on VOC07 dataset, linear classifiers trained on res4 perform  $\sim 3$ -10 points better than those trained on res5 for both Jigsaw and RotNet networks. As noted in [23, 64], this is because the final layer features overfit to the pre-training (‘pre-text’) task.

After applying ClusterFit, we see that features of  $N_{cf}$  transfer better across all the layers except for conv1– an improvement of **7 to 9 points** on ImageNet-1K– for both Jigsaw and RotNet methods. On VOC07, res5 features transfer better than res4: for  $N_{pre}$  the gap is  $-9$  points while for  $N_{cf}$  it is about  $+1$  points. On ImageNet-1K and Places205, the performance gap of  $N_{cf}$  when using res4 vs. res5 features is considerably reduced. This strongly suggests that ClusterFit reduces the overfitting of res5 features to the pre-text task, thus making them generalize better.

**Results:** We show additional transfer learning results in Table 6. Longer pre-training ( $N_{pre}2\times$ ) shows mixed results – a small drop in performance for Jigsaw and a small increase in performance for RotNet. ClusterFit provides consistent improvements on both Jigsaw and RotNet tasks, across all pre-training and target tasks. We achieve significant boosts of **3-5 points** on Places205 and **5-8 points** on iNaturalist.

**Easy multi-task Learning using ClusterFit:** In the supplementary material, we show that ClusterFit can be easily applied to combine multiple different self-supervised methods and provides impressive gains of more than **8 points** on ImageNet-1K in top-1 accuracy.

**Summary: Why does ClusterFit work?** We believe that grouping images / videos based on noisy hashtags (weak supervision) or on the angle of rotation (e.g., RotNet in



**Figure 4: Self-supervised Images (Layerwise):** We examine the layer-wise performance of self-supervised models before applying our technique ( $N_{pre}$ ) and after ( $N_{cf}$ ). We fix  $D_{pre} = D_{cf} = \text{ImageNet-1K}$  (without labels) and use the same architecture (ResNet-50) for  $N_{pre}$  and  $N_{cf}$ . The last layer (res5) features for  $N_{pre}$  transfer poorly compared to the lower res4 layer. After CF,  $N_{cf}$  shows an improved performance for all layers except for conv1 and reduces the gap in performance between res4 and res5.

SSL) is *less semantic* and leads to misalignment between pre-training and transfer tasks, thereby a less-generalizable feature space. ClusterFit combats this issue by first grouping images/videos using *more semantic* visual features in its clustering step; consequently training with cluster assignments as pseudo-labels yields higher gains on semantic tasks across different modalities and pre-training settings.

## 5. Analyzing ClusterFit

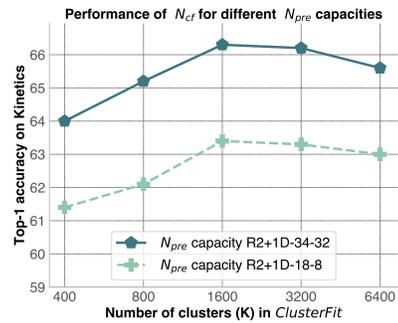
ClusterFit involves several aspects such as the relative model capacities of  $N_{pre}$  and  $N_{cf}$ , properties of  $D_{pre}$  and  $D_{cf}$ , size and granularity of the pre-training label space, and so on. In this section, we study the effect of these design choices on the transfer learning performance with videos as an example use case (Table 2).

**Experimental Setup:** Similar to IG-Verb-19M in Sec. 4.1.2, we construct IG-Verb-62M, a weakly-supervised dataset comprising 62M videos and use it as  $D_{cf}$ . For faster training of  $N_{cf}$ , we consider a computationally cheaper R(2+1)D-18 [8] architecture and process 8 frames per video clip. Unless specified otherwise,  $D_{pre} = \text{IG-Verb-19M}$  and  $N_{pre} = \text{R}(2+1)\text{D-34}$  [8] with 32 frames per video. All other settings are same as in Sec. 4.1.2.

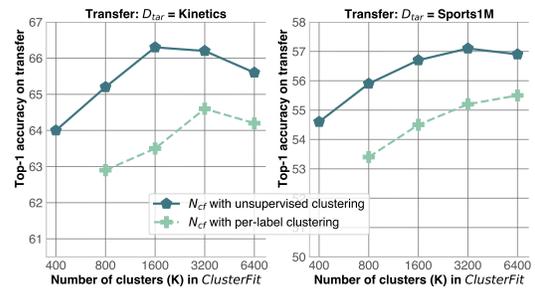
### 5.1. Relative model capacity of $N_{pre}$ and $N_{cf}$

The relative model capacities of  $N_{pre}$  and  $N_{cf}$  can impact the final transfer performance of  $N_{cf}$ . To study this behavior, we fix  $D_{pre} = \text{IG-Verb-19M}$  and  $D_{cf} = \text{IG-Verb-62M}$ , and  $N_{cf} = \text{R}(2+1)\text{D-18}$ . We vary the architecture of  $N_{pre}$  as follows: (a)  $N_{pre} = N_{cf} = \text{R}(2+1)\text{D-18}$ ; (b)  $N_{pre} > N_{cf}$ , where  $N_{pre} = \text{R}(2+1)\text{D-34}$  model (64M parameters) and thus higher capacity than  $N_{cf}$  (33M parameters).

From Figure 5, we observe a consistent improvement of 2% – 3% across different values of  $K$  when a higher capacity model was used as  $N_{pre}$ . This result is intuitive and indicates that a higher capacity  $N_{pre}$  yields richer visual features for clustering and thus improves the transfer learning

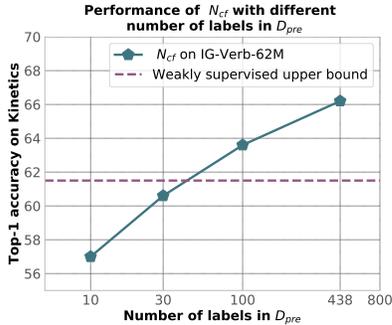


**Figure 5: Relative Model Capacity of  $N_{pre}$  and  $N_{cf}$  (§5.1):** We fix  $N_{cf} = \text{R}(2+1)\text{D-18}$ . We vary (a)  $N_{pre} = N_{cf} = \text{R}(2+1)\text{D-18}$  (light green) and (b)  $N_{pre} > N_{cf}$ , where  $N_{pre} = \text{R}(2+1)\text{D-34}$  (dark green). We report the transfer performance of the  $N_{cf}$  model for cases (a) and (b) on Kinetics. A higher capacity  $N_{pre}$  results in better transfer performance.



**Figure 6: Unsupervised vs. Per-Label Clustering (§5.2):** In per-label clustering, we retain the semantic information of the class labels and cluster videos belonging to each label. We note that for all values of  $K$ , unsupervised clustering used in ClusterFit yields better transfer learning performance on Kinetics and Sports1M.

performance. We note that in the aforementioned case (b), our framework can be viewed to be distilling knowledge from a higher capacity teacher model ( $N_{pre}$ ) to a lower-capacity student model ( $N_{cf}$ ).



**Figure 7: Effect of number of labels in  $N_{pre}$  (§5.3).** We design 4 different  $D_{pre}$ , each with  $19M$  videos but  $\#labels = \{10, 30, 100, 438\}$ ,  $K = 3200$  and  $D_{cf} = IG\text{-Verb-62M}$ . X-axis is in log-linear scale.

## 5.2. Unsupervised vs. Per-Label Clustering

As noted before, the clustering step in ClusterFit is ‘unsupervised’ because it discards the labels associated with  $D_{cf}$  and operates purely on the feature representations. But is there any advantage of using the semantic information of labels in  $D_{cf}$  for clustering? To address this question, we formulate a per-label clustering setup. Specifically, given each label  $l$ , we cluster videos belonging to it into  $k_l$  clusters. We treat  $K = \{k_l : \forall l\}$  as pseudo-labels to train  $N_{cf}$ . Each  $k_l$  is defined to be proportional to  $\sqrt{n_l}$ <sup>3</sup> where  $n_l$  denotes the number of videos associated with the label  $l$ .

Figure 6 compares the two clustering approaches on Kinetics and Sports1M. We observe that on both datasets, unsupervised clustering consistently outperforms per-label clustering across all values of  $K$ . We believe that by operating purely on video features, the unsupervised approach effectively captures the visual coherence in  $D_{cf}$ . Consequently, factors around label noise such as wrong / missing labels and lexical ambiguity are being automatically addressed in the unsupervised framework, leading to superior performance over per-label clustering.

## 5.3. Properties of $D_{pre}$

In this section, we address the following question: what constitutes a valuable pre-training label space ( $D_{pre}$ ) and how to construct one? Towards this end, we study two properties of  $D_{pre}$ : the nature of its labels and their cardinality. We refer the readers to the supplementary material for discussion on the nature of labels.

**Number of labels in  $D_{pre}$ :** We now study how varying the number of labels in  $D_{pre}$  effects ClusterFit. To study this, we fix the total number of unique videos in  $D_{pre}$  to  $19M$  and vary the number of pre-training labels. First, we consider IG-Verb-62M and rank its 438 weak verb labels by their frequency of occurrence. Next, we construct 4 different  $D_{pre}$  datasets by considering unique  $19M$  videos tagged

<sup>3</sup>We also experimented with  $k_l \approx n_l$  but this resulted in worse performance.

with top- $m$  verbs, where  $m = \{10, 30, 100, 438\}$ . Note that for a fixed number of videos in  $D_{pre}$ , reducing the number of labels implies reduced content diversity.

From Figure 7, we observe that the transfer learning performance increases log-linearly with the number of pre-training labels in  $D_{pre}$ . When we use just the top-10 verbs ( $m = 10$ ), accuracy drops by around 9% compared to  $m = 438$ . This indicates that label space diversity is essential to generate good quality clusters. However, when  $m = 100$ ,  $N_{cf}$  is within 2% of the accuracy obtained when using all 438 verbs, and it outperform its weakly supervised pre-trained counterpart which uses 62M videos and all 438 verbs. This experiment clearly demonstrates the utility of our approach in designing a generic pre-training label space with minimal effort. Contrary to [20, 38] which propose careful, manual label engineering, ClusterFit offers an easy way to construct a powerful, generalizable pre-training label space. Increasing the label space granularity is as simple as increasing the number of clusters in ClusterFit and requires no additional manual effort.

## 6. Discussion

In this work, we presented ClusterFit, a simple approach to significantly improve the generalizability of features learnt in weakly-supervised and self-supervised frameworks for images and videos. While models trained in these frameworks are prone to overfit to the pre-training objective, ClusterFit combats this issue by first clustering the original feature space and re-learning a new model on cluster assignments. Clustering in CF may be viewed as a *lossy* compression scheme that effectively captures the essential visual invariances in the feature space. Thus, predicting the cluster labels gives the ‘re-learned’ network an opportunity to learn features that are less sensitive to the original pre-training objective, making them more transferable.

While the clustering step in ClusterFit is unsupervised, in practice, domain knowledge from downstream target tasks can be used to guide clustering and possibly improve the transfer learning performance. Additionally, we found that in its current unsupervised form, iterative application of CF provides little improvements; incorporating domain knowledge could be a potential solution.

ClusterFit is highly scalable and imposes no restrictions on model architectures, modalities of data, and forms of supervision. Future research should take advantage of its flexibility and combine different types of pre-trained models for learning cluster assignments in a multi-task manner. Using evidence accumulation methods [17, 41, 50] for clustering is another worthwhile direction to explore.

**Acknowledgments:** We would like to thank Rob Fergus, and Zhenheng Yang for feedback on the manuscript; Filip Radenovic, and Vignesh Ramanathan for feedback on the experimental setup; Laurens van der Maaten, Larry Zitnick, Armand Joulin, and Xinlei Chen for helpful discussions.

## References

- [1] Rohan Anil, Gabriel Pereyra, Alexandre Passos, Robert Ormandi, George E Dahl, and Geoffrey E Hinton. Large scale distributed neural network training through online distillation. *arXiv preprint arXiv:1804.03235*, 2018.
- [2] Relja Arandjelovic and Andrew Zisserman. Look, listen and learn. In *ICCV*, 2017.
- [3] Relja Arandjelovic and Andrew Zisserman. Objects that sound. In *ECCV*, 2018.
- [4] Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? In *Advances in neural information processing systems*, pages 2654–2662, 2014.
- [5] Piotr Bojanowski and Armand Joulin. Unsupervised learning by predicting noise. In *ICML*, 2017.
- [6] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, 2018.
- [7] Mathilde Caron, Piotr Bojanowski, Julien Mairal, and Armand Joulin. Unsupervised pre-training of image features on non-curated data. In *ICCV*, 2019.
- [8] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri. A closer look at spatiotemporal convolutions for action recognition. *CVPR*, 2018.
- [9] Virginia R de Sa. Learning classification with unlabeled data. In *NIPS*, 1994.
- [10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. 2009.
- [11] E. Denton, J. Weston, M. Paluri, L. Bourdev, and R. Fergus. User conditional hashtag prediction for images. In *Proc. KDD*, pages 1731–1740, 2015.
- [12] Aditya Deshpande, Jason Rock, and David Forsyth. Learning large-scale automatic image colorization. In *ICCV*, 2015.
- [13] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, pages 1422–1430, 2015.
- [14] Alexey Dosovitskiy, Philipp Fischer, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *TPAMI*, 38(9):1734–1747, 2016.
- [15] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, Jan. 2015.
- [16] Basura Fernando, Hakan Bilen, Efstratios Gavves, and Stephen Gould. Self-supervised video representation learning with odd-one-out networks. In *CVPR*, 2017.
- [17] Ana LN Fred and Anil K Jain. Data clustering using evidence accumulation. In *Object recognition supported by user interaction for service robots*, volume 4, pages 276–280. IEEE, 2002.
- [18] Tommaso Furlanello, Zachary C Lipton, Michael Tschanen, Laurent Itti, and Anima Anandkumar. Born again neural networks. *arXiv preprint arXiv:1805.04770*, 2018.
- [19] Ruohan Gao, Rogerio Feris, and Kristen Grauman. Learning to separate object sounds by watching unlabeled video. In *ECCV*, 2018.
- [20] Deepti Ghadiyaram, Matt Feiszli, Du Tran, Xueting Yan, Heng Wang, and Dhruv Mahajan. Large-scale weakly-supervised pre-training for video action recognition. *arXiv preprint arXiv:1905.00561*, 2019.
- [21] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Un-supervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.
- [22] Julien Girard, Youssef Tamaazousti, Hervé Le Borgne, and Céline Hudelot. Learning finer-class networks for universal representations. *arXiv preprint arXiv:1810.02126*, 2018.
- [23] Priya Goyal, Dhruv Mahajan, Abhinav Gupta, and Ishan Misra. Scaling and benchmarking self-supervised visual representation learning. *arXiv preprint arXiv:1905.01235*, 2019.
- [24] 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.
- [25] Sam Gross, Marc’Aurelio Ranzato, and Arthur Szlam. Hard mixtures of experts for large scale weakly supervised vision. In *CVPR*, 2017.
- [26] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, 2006.
- [27] James Hays and Alexei A Efros. Im2gps: estimating geographic information from a single image. In *2008 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2008.
- [28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.
- [29] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.
- [30] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.
- [31] Armand Joulin, Laurens van der Maaten, Allan Jabri, and Nicolas Vasilache. Learning visual features from large weakly supervised data. In *ECCV*, 2016.
- [32] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.
- [33] Alexander Kolesnikov, Xiaohua Zhai, and Lucas Beyer. Re-visiting self-supervised visual representation learning. *arXiv preprint arXiv:1901.09005*, 2019.
- [34] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Learning representations for automatic colorization. In *ECCV*, 2016.
- [35] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Colorization as a proxy task for visual understanding. In *CVPR*, 2017.
- [36] A. Li, A. Jabri, A. Joulin, and L.J.P. van der Maaten. Learning visual n-grams from web data. In *Proc. ICCV*, 2017.

- [37] Pauline Luc, Natalia Neverova, Camille Couprie, Jakob Verbeek, and Yann LeCun. Predicting deeper into the future of semantic segmentation. In *ICCV*, 2017.
- [38] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In *ECCV*, 2018.
- [39] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *ECCV*, 2016.
- [40] Hossein Mobahi, Ronan Collobert, and Jason Weston. Deep learning from temporal coherence in video. In *ICML*, 2009.
- [41] Nam Nguyen and Rich Caruana. Consensus clusterings. In *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, pages 607–612. IEEE, 2007.
- [42] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2016.
- [43] Mehdi Noroozi, Hamed Pirsiavash, and Paolo Favaro. Representation learning by learning to count. In *ICCV*, 2017.
- [44] Mehdi Noroozi, Ananth Vinjimoor, Paolo Favaro, and Hamed Pirsiavash. Boosting self-supervised learning via knowledge transfer. In *CVPR*, 2018.
- [45] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.
- [46] Andrew Owens, Jiajun Wu, Josh H McDermott, William T Freeman, and Antonio Torralba. Ambient sound provides supervision for visual learning. In *ECCV*, 2016.
- [47] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115, 2015.
- [48] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015.
- [49] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4077–4087, 2017.
- [50] Alexander Strehl and Joydeep Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research*, 3(Dec):583–617, 2002.
- [51] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *ICCV*, 2017.
- [52] Y. Taigman, M. Yang, M.A. Ranzato, and L. Wolf. Web-scale training for face identification. In *CVPR*, 2015.
- [53] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73, 2016.
- [54] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop (ITW)*, pages 1–5. IEEE, 2015.
- [55] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *CVPR*, pages 8769–8778, 2018.
- [56] A. Veit, M. Nickel, S. Belongie, and L.J.P. van der Maaten. Separating self-expression and visual content in hashtag supervision. In *arXiv 1711.09825*, 2017.
- [57] VerbNet. VerbNet : A Computational Lexical Resource for Verbs. [Online] Available <https://verbs.colorado.edu/verbnnet/>.
- [58] Nam Vo, Nathan Jacobs, and James Hays. Revisiting im2gps in the deep learning era. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2621–2630, 2017.
- [59] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman. The kinetics human action video dataset. *arXiv:1705.06950*, 2017.
- [60] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, 2018.
- [61] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *ICML*, pages 478–487, 2016.
- [62] I. Zeki Yalniz, Hervé Jégou, Kan Chen, Manohar Paluri, and Dhruv Mahajan. Billion-scale semi-supervised learning for image classification. *CoRR*, abs/1905.00546, 2019.
- [63] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *ECCV*, 2016.
- [64] Richard Zhang, Phillip Isola, and Alexei A Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *CVPR*, 2017.
- [65] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *NIPS*, 2014.