Beyond Supervised vs. Unsupervised: Representative Benchmarking and Analysis of Image Representation Learning

Matthew Gwilliam and Abhinav Shrivastava
University of Maryland, College Park

Abstract

By leveraging contrastive learning, clustering, and other pretext tasks, unsupervised methods for learning image representations have reached impressive results on standard benchmarks. The result has been a crowded field – many methods with substantially different implementations yield results that seem nearly identical on popular benchmarks, such as linear evaluation on ImageNet. However, a single result does not tell the whole story. In this paper, we compare methods using performance-based benchmarks such as linear evaluation, nearest neighbor classification, and clustering for several different datasets, demonstrating the lack of a clear front-runner within the current state-of-the-art. In contrast to prior work that performs only supervised vs. unsupervised comparison, we compare several different unsupervised methods against each other. To enrich this comparison, we analyze embeddings with measurements such as uniformity, tolerance, and centered kernel alignment (CKA), and propose two new metrics of our own: nearest neighbor graph similarity and linear prediction overlap. We reveal through our analysis that in isolation, single popular methods should not be treated as though they represent the field as a whole, and that future work ought to consider how to leverage the complimentary nature of these methods. We also leverage CKA to provide a framework to robustly quantify augmentation invariance, and provide a reminder that certain types of invariance will be undesirable for downstream tasks.

1. Introduction

Image features are critical components in many computer vision (CV) pipelines. In this paper, we define image features, also referred to as embeddings, encodings, or representations, as an \( n \)-dimensional vector that represents the content of an image. With the emergence of deep learning, classical approaches to computing image features have been supplanted by neural networks that use large amounts of data to generate powerful image representations. The most widespread method is straightforward: a neural work (e.g., a ResNet50 [30]) is trained to classify the images in some large dataset, typically ImageNet. The portion of the network that performs the classification, usually just the final layer, is then removed, and the outputs of the penultimate layer for a given image are considered the features for that image. This process relies on image classification, a supervised learning task, and thus requires the availability of large amounts of annotated, high-quality data.

Recent successes make unsupervised learning a viable alternative paradigm where image features are learned without the need for class labels. Within unsupervised learning, methods can be considered either generative or discriminative. Generative methods are typically designed for reconstruction or similar tasks [5, 19, 20, 35, 51]. Since we are more concerned with a potential transfer to downstream tasks such as image classification and object detection, we
choose to focus on discriminative methods.

There are many different ways to compare image representation learning algorithms. In this paper, we opt to focus on the role of the methods as feature extractors, where a model that is pre-trained for some task is expected to be able to generate useful features for unseen images. Thus, we only use benchmarks that keep the backbone (the portion of the neural network that generates the embedding) frozen. Prior works are often limited by their focus on a single benchmark, single method, or toy datasets. In contrast, we compare 6 SOTA unsupervised methods on ImageNet and 6 fine-grained visual categorization (FGVC) datasets using several different benchmarks. Figure 1 provides a sample of this angle of analysis.

Comparing these methods to each other is very important. Nevertheless, prior analysis works tend to lump unsupervised methods together, and often choose only a single representative such as MoCo or SimCLR for comparison against supervised representation learning [1,14,22,26,54]. This ignores the significant ways in which unsupervised methods differ from each other. In contrast to prior work, we extend existing methods and introduce novel methods to prove that unsupervised methods vary significantly in terms of how they learn to represent images, as shown in Figure 2.

State-of-the-art convolution-based unsupervised algorithms, whether they use contrastive learning, clustering, or some pretext task such as colorization, all attempt to learn invariance to some class of augmentations. In other words, they seek to learn a function $f$, such that $f(I) = f(I_A)$ for some image $I$ and some set of augmentations $A$ that are applied to that image. Xiao et al. speculate on the negative effect this may have on learned representations and performance on downstream tasks [56]. However, a careful reading reveals they don’t provide evidence for the existence of transform invariance in unsupervised models, only that their method seems to perform better than MoCo on tasks related to transform invariance. This inspires Section 4.4 – we take a closer look at the presence of augmentation invariance in representations learned by different unsupervised methods.

Prior work is constrained by some combination of limited metrics, use of toy datasets, and a tendency to consider a single unsupervised method as though it is representative of the field. In contrast to this, we contribute the following:

- We utilize multiple methods for measuring properties of learned embeddings, including 3 performance-based benchmarks, and an extension of prior work on uniformity-tolerance analysis to more unsupervised methods across more realistic datasets.
- We perform novel comparison by extending Linear Centered Kernel Alignment (CKA) analysis beyond toy datasets, and by developing two new metrics for comparing embeddings: nearest neighbor graph similarity and linear overlap.
- We propose a framework for measuring augmentation invariance, and demonstrate its results across several methods, augmentations, and datasets.

We conclude in Section 5 with key insights for future unsupervised methods for representation learning:

- Currently, there is no clear “best” method.
- Unsupervised models share properties that are circumstantially undesirable, e.g., color invariance.
- Unsupervised models have similar representations in most layers, but diverge substantially in the last layer.
2. Related Work

2.1. Unsupervised Learning Methods

Some of the first unsupervised methods of the deep learning era were fashioned after pretext tasks from natural language processing. A network would be trained to perform some auxiliary task before transfer for downstream tasks. These auxiliary pretext tasks included solving jigsaw puzzles [44], colorization from grayscale [39, 60], inpainting [46], relative patch prediction [17], predicting rotation angle [23], or a combination of tasks [18]. However, the introduction of Noise Contrastive Estimation (NCE) [27] triggered a paradigm shift within unsupervised learning [55], and subsequent methods which utilized contrastive learning [10, 12, 42] would surpass all of these.

**Contrastive Learning**, which implicitly performs instance discrimination, involves training a model to attract positive pairs (typically augmented views of a given image) and repel negative pairs (augmented views of different images) [27, 28]. Many papers have proposed successful methods using contrastive learning [4, 11, 12, 29, 31, 32, 42, 45, 48, 55, 58]. In this study, we include Barlow Twins [39], MoCov2 [12], and SimCLR [11].

**Clustering** has emerged as another important class of unsupervised methods [3, 6–8]. Popular methods such as DeepCluster [6] and SwAV [8], and even methods that don’t explicitly attempt representation learning, such as SCAN [49], share many traits with contrastive learning. Among these are the tendency to rely on a large batch size, the use of strong augmentations, which all of these have in common, and implementation details such as the use of projection heads, which are used by SwAV and DeepCluster [6, 8]. We take as our representatives from this category DeepCluster and SwAV.

**Other** methods, such as SimSiam [13], use neither negative pairs nor clustering objectives. Other methods highlight the potential of vision transformers within unsupervised regimes [9, 40, 57]. However, in this paper, we choose to compare methods that use ResNet-50 backbones. Furthermore, we believe the sample of methods we select are sufficient to support our main points.

2.2. Analysis of Unsupervised Learning

While each paper proposing a new algorithm uses some tasks to attempt to demonstrate their success compared to the prior art, other popular papers have entirely focused on benchmarking, evaluation, and comparison of specific methods. [47] studies augmentation invariance from the perspective of accuracy by using natural images that attempt to vary certain conditions, such as illumination. [54] and [53] address properties of the learned embeddings such as alignment, uniformity, and tolerance. Other works benchmark unsupervised performance on various tasks and conditions [14, 22, 33, 37]. [26] uses the centered kernel alignment (CKA) framework from [36] to compare supervised and unsupervised representations.

This prior work operates under certain constraints. Many papers consider only a single unsupervised method, either MoCo [1, 22, 54] or SimCLR [14, 26], as though it represents the entire field; of those that consider other methods, none consider more than three [33, 57, 47]. We extend uniformity-tolerance analysis [53] to multiple unsupervised methods. We extend CKA analysis [15, 26, 36] beyond tiny datasets, and to multiple methods. We perform FGVC benchmarking [14, 33] for additional metrics and algorithms. We perform CKA analysis [36] to examine augmentation invariance without relying on linear classification as a confounding intermediate step [47]. We develop additional methods for comparing pairs or groups of methods for our first-of-its-kind comprehensive analysis of the similarities and complementary attributes of pretrained unsupervised methods for image representation learning. Our work follows in the spirit of Ericsson et al. in that we perform a comprehensive analysis using a sample of many unsupervised methods [21]. However, by using an entirely different set of tasks and datasets, we are able to both provide further evidence for one of their main conclusions (that no one method is the best), and uncover novel insights as well.

3. Methods

3.1. Performance-based Comparison

To measure the quality of the learned representations, we perform three performance-based measurements. For linear evaluation, we use the VISSL repository [24] to train a linear classifier on frozen features. For k-nearest neighbor classification, we also use the settings in VISSL, varying the number of neighbors for fine-grained datasets as described in the appendix. For k-means clustering, we try 10 initializations with the k-means++ method [2], and for ImageNet we use mini-batches of 16,384 images.

To obtain the accuracy for k-means cluster assignments when the number of classes is equal to the number of clusters, we use hungarian matching, mapping clusters to classes one-to-one in such a way that maximizes correspondence to ground truth classes. For overclustering, we greedily map each cluster to the ground truth class which has the most images in the cluster. Accuracy is then computed normally, using the cluster-to-class mapping result as the prediction. For each of linear evaluation, k-NN classification, and k-means clustering, models are trained on training data, and results come from the evaluation on test data.

3.2. Uniformity-Tolerance Tradeoff

To analyze how embeddings are distributed on the hypersphere, we borrow two key properties from prior work:
uniformity [54] and tolerance [53]. Uniformity, \( U \), which describes how closely the embeddings match a uniform distribution on a hypersphere, is defined in Equation 1, where \( t \) is a scaling hyperparameter that we set to 2. \( f \) represents the model, and \( x \) and \( y \) are any pair of images.

\[
U = \log \mathbb{E}_{x,y \sim \text{data}} \left( e^{-t\|f(x) - f(y)\|_2^2} \right) \tag{1}
\]

Tolerance, \( T \), is given in Equation 2, where \( f \), \( x \), and \( y \) are the same as in Equation 1, and \( I \) is the indicator function for the ground truth labels, returning 1 when \( x \) and \( y \) belong to the same class, and 0 otherwise.

\[
T = \mathbb{E}_{x,y \sim \text{data}} \left( \frac{\|f(x)\|_2^2 \cdot \|f(y)\|_2}{\|f(x) - f(y)\|_2} \cdot I_{l(x) = l(y)} \right) \tag{2}
\]

Whereas uniformity measures how equally spread out the features are, tolerance leverages ground truth labels to indicate how well the embeddings reflect the semantic relationships between the images.

3.3. Linear CKA for Comparing Representations

We follow procedures from prior work to compute Centered Kernel Alignment (CKA) values [26, 36], including using only a linear kernel [26]. To compute this, we first obtain the matrices containing the embeddings for two different methods, such as SimCLR and MoCo, which we represent \( X \) and \( Y \). We then compute the Gram matrices of the embedding matrices: \( K = XX^T \), \( L = YY^T \). The CKA value is given by the normalized Hilbert-Schmidt Independence Criterion (HSIC) [25] as follows:

\[
\text{CKA}(K, L) = \frac{\text{HSIC}(K, L)}{\sqrt{\text{HSIC}(K, K) \cdot \text{HSIC}(L, L)}} \tag{3}
\]

Prior work [26] performs these computations for tiny datasets, consisting of \( 32 \times 32 \) images. Since we extend this to \( 224 \times 224 \) images, we compensate for the increased memory requirements by taking a set random sample of 10,000 test images when working with ImageNet.

3.4. Proposed Metric: NN Graph Similarity

Even for pure contrastive learning, images that belong to the same ground truth class tend to be semantically similar. In contrast to this tolerance, which relies on ground truth labels and describes the semantic structure in terms of a single model, we propose an unsupervised way to compare the structure of the semantic relationships between two or more learned representations. Specifically, we consider models in terms of their nearest neighbor graphs for a given dataset. Each image is a node, and an image’s top-\( k \) neighbors are represented by directed edges. We can thus compute the similarity between two representations by comparing their nearest neighbor graphs. We choose to perform this computation in terms of neighbor overlap, where neighbor overlap refers to the average number of shared edges (neighbors) per node (image) for the graphs (unsupervised algorithms) considered. This neighbor overlap conveys the similarity in semantic structure learned by different algorithms. A score of 1.0 would indicate the structures are identical – the images have the same nearest \( k \) neighbors for both unsupervised algorithms. A score of 0.0 would indicate there are no shared neighbors. We use this method to compare pairs of models, as in Figure 2.

3.5. Proposed Metric: Linear Prediction Overlap

Unlike nearest neighbor graph similarity, this metric takes an indirect approach to compare two or more representations. For each image in a dataset, we get the predictions from the linear classifiers trained on each unsupervised backbone, described in Section 3.1. We then perform a few different calculations. For analysis that relies on ground truth labels, we calculate the portion of the dataset that all models classify correctly, or that no models classify correctly, or that only some subset of models from a group classifies correctly. For analysis that ignores the labels, we compute the percentage of the dataset for which some set of classifiers make the same prediction, regardless of the correctness of the prediction. For this measure, a score of 1.0 would indicate the set of classifiers make the same predictions for all images, while 0.0 would mean they do not have identical predictions for any image. We use linear overlap both to compare pairs of methods, as in Figure 2, as well as to compare sets of multiple models, as in Table 5.

3.6. Proposed Analysis: Augmentation Invariance

Popular unsupervised algorithms train models to be augmentation invariant. We develop a method for analyzing the prevalence of augmentation invariance in the representations learned by unsupervised models. Unlike prior work [47], we consider a broader set of unsupervised algorithms, and perform measurements on learned representations directly rather than relying on the performance of learned linear classifiers as a proxy. Instead, we use CKA to compare the similarity between embeddings of augmented and non-augmented images. We take \( \text{CKA}(K, L) \) on \( K = XX^T \), \( L = X_A X_A^T \) where \( X \) is the embedding matrix for the images for a given dataset, and \( X_A \) is the embedding matrix for the same images after some augmentation \( A \). We thus extend Linear CKA for unsupervised algorithms to overcome the limitations of previous methods and look directly at augmentation invariance for image representations.
4. Analysis

For our representative set of discriminative unsupervised methods, we consider contrastive methods SimCLR [11] and MoCo [12], clustering methods DeepClusterv2 [6] and SwAV [8], as well as Barlow Twins [59], which attempts redundancy reduction, and SimSiam [13], which uses neither negative pairs nor clustering. For SimCLR, DeepCluster, and SwAV, we use the 800 epoch checkpoints from the VISSL model zoo [24], unless otherwise specified. For Barlow Twins, we use VISSL’s 1000 epoch checkpoint. For MoCo, we use the 800 epoch checkpoint from the authors. For SimSiam, we use the author-provided 100 epoch checkpoint. While we could have opted to retrain the models, we believe that discrepancy in training time is not a legitimate confounding factor in our analysis, and any attempt to create some “fair” setting would inevitably favor whichever models perform best under that setting. These methods can be tuned on a variety of hyperparameters, and any given setting that is “fair” to one inevitably favors others, so we opt to use the settings for which models are available. Additionally, our ablation experiments (Figure 6) indicate training time would not impact any of our findings.

We perform the experiments here on some subset of the datasets in Table 1, which contain realistic images.

4.1. Performance Benchmarks

As explained in Section 3, we perform VISSL’s linear evaluation, where we train linear classifiers on frozen features from the first convolutional layer and for each of the 4 bottleneck blocks of the ResNet-50 network. We show results for the linear classifier trained on the outputs of the final block in Figure 3. We show results for k-NN classification in Table 2 and for k-means clustering in Table 3. As an additional benchmark for DeepCluster and SwAV, we also compare their clustering heads, which partition ImageNet into 3000 clusters, to k-means clustering in Table 4. We don’t consider other performance benchmarks such as full finetuning because we are trying to evaluate only the learned embeddings, and not the network initializations.

![Figure 3](image.png)

Figure 3. Linear Classifier Results on ImageNet and 6 FGVC datasets. Barlow Twins, DeepCluster, and SwAV tend to outperform Moco, SimCLR, and SimSiam, but there is no obvious winner.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#Cls</th>
<th>#Train</th>
<th>#Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>FGVC Aircraft</td>
<td>[41]</td>
<td>100</td>
<td>6,667</td>
</tr>
<tr>
<td>Stanford Cars</td>
<td>[38]</td>
<td>196</td>
<td>8,144</td>
</tr>
<tr>
<td>Caltech Birds</td>
<td>[52]</td>
<td>200</td>
<td>5,994</td>
</tr>
<tr>
<td>Stanford Dogs</td>
<td>[34]</td>
<td>120</td>
<td>12,000</td>
</tr>
<tr>
<td>Oxford Flowers</td>
<td>[43]</td>
<td>102</td>
<td>2,040</td>
</tr>
<tr>
<td>NABirds V1</td>
<td>[50]</td>
<td>555</td>
<td>23,929</td>
</tr>
<tr>
<td>ImageNet</td>
<td>[16]</td>
<td>1000</td>
<td>1.3mil</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>50,000</td>
</tr>
</tbody>
</table>

Figure 3 shows that contrary to claims in some prior work, unsupervised methods do not necessarily struggle on FGVC datasets [33]. This is possibly because that work used linear SVMs that were perhaps less accommodating to the ways the unsupervised embeddings tend to be distributed; since our linear evaluation protocol uses batch norm, it is able to better account for this. Nevertheless, we demonstrate while performance is worse for the two birds datasets and the dogs dataset, Barlow Twins, DeepCluster, and SwAV all outperform supervised pre-training for aircraft, cars, and flowers. We suggest that significant over-
Table 2. **k-NN Results.** Again, there is no obvious frontrunner.

<table>
<thead>
<tr>
<th>Method</th>
<th>ImageNet</th>
<th>Aircraft</th>
<th>Flowers</th>
<th>NABirds</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supervised</td>
<td>73.41</td>
<td>31.59</td>
<td>77.96</td>
<td>43.25</td>
</tr>
<tr>
<td>BTwins</td>
<td>62.90</td>
<td>31.83</td>
<td>86.18</td>
<td>22.29</td>
</tr>
<tr>
<td>DCv2</td>
<td>63.70</td>
<td><strong>32.70</strong></td>
<td>84.76</td>
<td>21.05</td>
</tr>
<tr>
<td>MoCo</td>
<td>58.59</td>
<td>21.39</td>
<td>74.53</td>
<td>15.40</td>
</tr>
<tr>
<td>SimCLR</td>
<td>54.57</td>
<td>21.21</td>
<td>74.78</td>
<td>14.03</td>
</tr>
<tr>
<td>SimSiam</td>
<td>53.66</td>
<td>27.39</td>
<td>80.01</td>
<td>15.18</td>
</tr>
<tr>
<td>SwAV</td>
<td>61.14</td>
<td>28.77</td>
<td>82.24</td>
<td>15.72</td>
</tr>
</tbody>
</table>

Table 3. **K-Means Results.** Supervised is best for most datasets.

<table>
<thead>
<tr>
<th>Method</th>
<th>ImageNet</th>
<th>Aircraft</th>
<th>Flowers</th>
<th>NABirds</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supervised</td>
<td>58.92</td>
<td><strong>15.69</strong></td>
<td>54.97</td>
<td><strong>25.95</strong></td>
</tr>
<tr>
<td>BTwins</td>
<td>34.88</td>
<td>13.20</td>
<td>63.70</td>
<td>11.87</td>
</tr>
<tr>
<td>DCv2</td>
<td>31.79</td>
<td>13.92</td>
<td>60.20</td>
<td>10.86</td>
</tr>
<tr>
<td>MoCo</td>
<td>38.30</td>
<td>9.84</td>
<td>43.34</td>
<td>10.75</td>
</tr>
<tr>
<td>SimCLR</td>
<td>29.78</td>
<td>11.16</td>
<td>43.99</td>
<td>9.08</td>
</tr>
<tr>
<td>SimSiam</td>
<td>26.20</td>
<td>12.66</td>
<td>54.51</td>
<td>9.53</td>
</tr>
<tr>
<td>SwAV</td>
<td>28.69</td>
<td>12.60</td>
<td>56.04</td>
<td>9.26</td>
</tr>
</tbody>
</table>

Table 4. **K-Means Overclustering Results.** The clustering heads of DeepCluster v2 and SwAV outperform k-means on the learned embeddings of DeepCluster v2 and SwAV.

<table>
<thead>
<tr>
<th>Method</th>
<th>ImageNet</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$k = 1000$</td>
</tr>
<tr>
<td>Supervised K-Means</td>
<td>58.92</td>
</tr>
<tr>
<td>DCv2 K-Means</td>
<td>31.79</td>
</tr>
<tr>
<td>DCv2 Clustering Head</td>
<td>n/a</td>
</tr>
<tr>
<td>SwAV K-Means</td>
<td>28.69</td>
</tr>
<tr>
<td>SwAV Clustering Head</td>
<td>n/a</td>
</tr>
</tbody>
</table>

lap with ImageNet contributes to part of the gap for performance on CUB and Dogs, which likely confounds those results; nevertheless, NABirds results confirm that unsupervised methods have substantial struggles on that dataset.

Table 2 echoes the results in Figure 3. Table 3 and Table 4, however, show that supervised pre-training dominates the k-means metric, except on Flowers. It seems clear that pre-training with labels gives supervised learning a strong advantage for k-NN classification and k-means clustering on ImageNet, to the extent that supervised representations even outperform the clustering heads of DeepCluster and SwAV in the overclustering regime on ImageNet.

We distill 3 key findings from this. First, from each of our benchmarks, unsupervised methods are comparable with supervised for generating embeddings for FGVC, and methods like Barlow Twins, DeepCluster, and SwAV seem particularly competitive. Second, setup matters — architectural decisions such as the design of the classification head can create subtle biases that favor certain methods, such as the SVM analysis from [33] favoring supervised representations. Finally, breadth is helpful; each of our benchmarking methods relies on some assumptions, and indirectly evaluates the robustness of the learned embeddings. Taken together, linear evaluation, k-NN classification, and k-means clustering give a more holistic view of how the representations compare.

### 4.2. Uniformity-Tolerance Tradeoff

High values for uniformity and tolerance are simultaneously desirable, as they indicate favorable distribution of the embeddings on the hypersphere. Nevertheless, our results in Figure 4 reinforce that in practice, these values have an inverse correlation. This is because as embeddings are more spread out on the hypersphere in general, they tend to also be more spread out with respect to each ground truth class. It is perhaps for this reason that supervised pre-training has, in general, the most tolerant and least uniform embeddings.

DeepCluster is similar to supervised for both these metrics. This is unsurprising when considering the DeepCluster objective: cross-entropy loss for pseudolabels. SwAV, with its own clustering objective, exhibits some of these same
tendencies, whereas the contrastive methods are the opposite. This strengthens support for our hypothesis – unsupervised methods are quite diverse in terms of the distribution of learned embeddings on the hypersphere.

4.3. Measuring Similarity of Representations

We consider three main metrics for measuring the similarity between representations: linear CKA, which compares embeddings for any given pair of models, nearest neighbor graph similarity, which compares nearest neighbors for a set of models, and overlap in linear predictions, which compares the predictions of linear classifiers trained on a set of models. See Figure 2 for linear CKA, neighbor similarity, and overlap in linear predictions for the outputs of each ResNet50’s final block for ImageNet. For linear CKA for the other layers we measure, see Figure 5. In Table 5, we perform an analysis of linear overlap for groups of models, and leverage ground truth labels to evaluate the uniqueness of the different linear classifiers.

Results in Figure 2 indicate that similar representations tend to have similar neighbors, and classifiers trained on more similar representations tend to make more similar predictions. The metrics reveal similarities between algorithms with related objectives, such as DeepCluster and SwAV. More surprisingly, we find more similarity between supervised and unsupervised representations such as DeepCluster than we do between some unsupervised representations, such as MoCo and Barlow Twins. Figure 5 simultaneously confirms these findings and those of [26], who found that supervised and unsupervised representations diverged the most in the final layer. We thus extend their hypothesis from SimCLR to additional unsupervised methods, and provide evidence that unsupervised algorithms differ enough from each other to have very different final representations. This highlights the idea that it is unreasonable to make the “supervised” vs. “unsupervised” comparisons that are so common in the literature where the “unsupervised” is represented by only a couple of algorithms.

We also compute CKA for unsupervised models under non-default settings, to validate our other findings. Figure 6, when compared to Figure 2, shows that training time, at least in the case of SimCLR, induces a comparatively small difference in learned representations, despite the large gaps in performance on benchmarks such as linear evaluation. While linear evaluation accuracies are different by several percentage points, SimCLR models trained 800 epochs are more similar to the other SimCLR checkpoints than to any other unsupervised algorithm. Our finding on training time stands in contrast to the results from other settings. We find details such as cropping strategy and batch size can have a massive impact on the similarity of neural representations, to the extent that SwAV with its full batch size and cropping strategy is more related to DeepCluster with the same settings than to SwAV methods that utilize small batch sizes. We suggest that future work should probe these effects fur-
Figure 6. Linear CKA for ablation settings for final residual outputs on the validation set of ImageNet. On the top, we compare supervised (SU) with SimCLR trained for 100, 200, 400, 800, and 1000 epochs. On the bottom, we compare SU with DeepCluster (DC) and SwAV (SW) trained for 400 epochs, as well as with SwAV and DeepCluster trained for 400 epochs with no crops (NC) and SwAV for 400 epochs with a smaller batch size (256).

4.4. Augmentation Invariance

We use linear CKA to test for augmentation invariance with respect to color jitter, blurring, jitter with blurring, horizontal and vertical flipping, and rotation. Figure 7 provides evidence that, contrary to the conclusions of [47], and confirming most other prior work, unsupervised algorithms learn representations that are invariant to their training augmentations. We note that the invariance is at least somewhat weaker for the clustering algorithms, SwAV and DeepCluster. Also, the unsupervised methods tend to be somewhat more invariant to augmentations not used at training time, rotation, and vertical flips. Nevertheless, these experiments suggest that for applications where color is critical, such as bird classification, methods that rely on learned color invariance are destined to underperform. Future works which seek to mitigate color invariance can leverage our framework as a way to evaluate success.

5. Key Takeaways and Conclusion

We highlight the following as key takeaways from our findings. First, there is no clear “best” method. Therefore, it is essential to avoid over-indexing on a given metric, and in the context of applications, representations should be selected to optimize for both downstream data and task. Second, unsupervised methods share properties that are situationally undesirable, such as robust color invariance. Thus, it is important for future work to develop methods that mitigate certain invariances when necessary, and our CKA-based framework can be utilized to validate their success. Finally, representations for all algorithms we considered are fairly similar until the last layer, where specialized loss functions or even training settings such as augmentation strategy and batch size can induce learning of substantially different representations. Therefore, it is critical to not assume that one method, such as MoCo or SimCLR, can act as a representative of the field. Additionally, taken in the context of our other findings, we suggest researchers continue to pursue meta-learning, distillation, ensembles, and other approaches that effectively combine different unsupervised algorithms to leverage their complementary nature.

Acknowledgements. This project was partially funded by the DARPA SAIL-ON (W911NF2020009) program, an independent grant from Facebook AI, and Amazon Research Award to AS.
References


[29] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised vi-


