Constrained Mean Shift Using Distant Yet Related Neighbors for Representation Learning

K L Navaneet[1] *, Soroush Abbasi Koohpayegani[1] *, Ajinkya Tejankar[1] *, Kossar Pourahmadi[1], Akshayvarun Subramanya[2], and Hamed Pirsiavash[1]

[1] University of California, Davis
[2] University of Maryland, Baltimore County

Abstract. We are interested in representation learning in self-supervised, supervised, and semi-supervised settings. Some recent self-supervised learning methods like mean-shift (MSF) cluster images by pulling the embedding of a query image to be closer to its nearest neighbors (NNs). Since most NNs are close to the query by design, the averaging may not affect the embedding of the query much. On the other hand, far away NNs may not be semantically related to the query. We generalize the mean-shift idea by constraining the search space of NNs using another source of knowledge so that NNs are far from the query while still being semantically related. We show that our method (1) outperforms MSF in SSL setting when the constraint utilizes a different augmentation of an image from the previous epoch, and (2) outperforms PAWS in semi-supervised setting with less training resources when the constraint ensures that the NNs have the same pseudo-label as the query. Our code is available here: https://github.com/UCDvision/CMSF

1 Introduction

Recently, we have seen great progress in self-supervised learning (SSL) methods that learn rich representations from unlabeled data. Such methods are important since they do not rely on manual annotation of data, which can be costly, biased, or ambiguous. Hence, SSL representations may perform better than supervised ones in transferring to downstream visual recognition tasks.

Most recent SSL methods, *e.g.*, MoCo [29] and BYOL [27], pull the embedding of a query image to be closer to its own augmentation compared to some other random images. Follow-up works have focused on improving the positive pairs through generating better augmentations [62,51,41] and the negative set by increasing the set size [29] or mining effective samples [35,34,67], but have largely ignored possibility of utilizing additional positive images. More recently, [37,21,5] expand the positive set using nearest neighbors. Inspired by classic mean-shift algorithm, MSF [37] generalizes BYOL to group similar images together. MSF pulls a query image to be close to not only its augmentation, but also the top-k nearest neighbors (NNs) of its augmentation.

* equal contribution

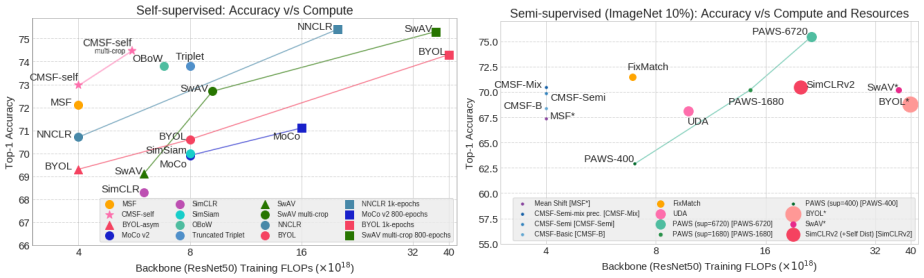


Fig. 1. Accuracy vs. training compute on ImageNet with ResNet50: We report the total training FLOPs for forward and backward passes through the CNN backbone. **(Left) Self-supervised:** All methods are for 200 epochs. $\text{CMSF}_{\text{self}}$ achieves competitive accuracy with considerably lower compute. **(Right) Semi-supervised:** Circle radius is proportional to the number of GPUs/TPUs used. The results are on ImageNet with 10% labels. In addition to being compute efficient, CMSF is trained with an order of magnitude lower resources, making it more practical and accessible. * methods use self-supervised pre-training and finetuning on the labeled set.

We argue that the top-k neighbors are close to the query image by construction, and thus may not provide a strong supervision signal. We are interested in choosing far away (non-top) neighbors that are still semantically related to the query image. This cannot be trivially achieved by increasing the number of NNs since the *purity* of retrieved neighbors decreases with increasing k (See Fig. 4 and Fig. 5). Purity is defined as the percentage of the NNs belonging to the same category as the query image.

We generalize MSF [37] method by simply limiting the NN search to a smaller subset that we believe is reasonably far from the query but still semantically related to it. We define this constraint to be (1) the nearest neighbors of another augmentation of the query in SSL setting and (2) images sharing the same label or pseudo-label as the query in supervised and semi-supervised settings. While we aim to obtain distant samples of the same category, note that we group only a few neighbors (k in our method) from the constrained subset instead of grouping the whole subset together. This is in contrast to cross-entropy supervised learning, where we pull all images of a category to form a cluster or be on the same side of a hyper-plane. Our method can benefit from this relaxation by preserving the latent structure of the categories and also being robust to noisy labels.

Our experiments show that the method outperforms the various baselines in all three settings with same or less amount of computation in training (refer Fig. 1). It outperforms MSF [37] in SSL, cross-entropy in supervised (with clean or noisy labels), and PAWS [4] in semi-supervised settings. Our main novelty is in developing a simple but effective method for searching for far away but semantically related NNs and in generalizing it to work across the board from self-supervised to semi-supervised and fully supervised settings. To summarize,

1. We propose constrained mean-shift (CMSF), a generalization of MSF [37], to utilize additional sources of knowledge to constrain the NN search space.

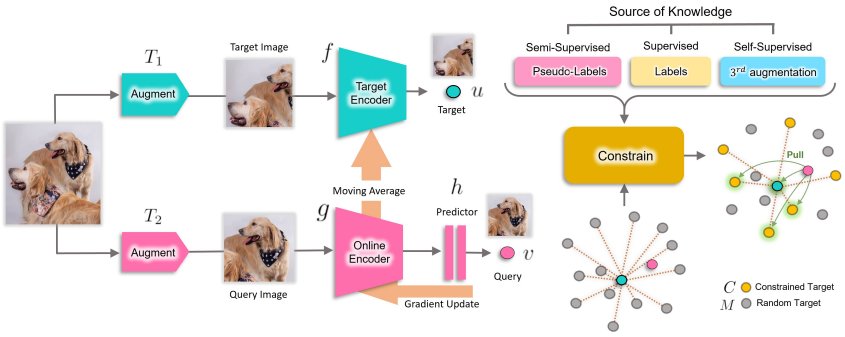


Fig. 2. Our method (CMSF): We augment an image twice and pass them through online and target encoders followed by ℓ_2 normalization to get u and v. Mean-shift [37] encourages v to be close to both u and its nearest neighbors (NN). To make NNs diverse, we constrain the NN search space based on additional knowledge in the form of NNs of the previous augmentation in self-supervised setting or the labels or pseudo-labels in semi or fully supervised settings. These constraints encourages the query to be pulled towards semantically related NNs that are farther away from the target embedding. See Fig 3 for constructing the constrained set.

2. We develop methods to select the constraint set in self-, semi- and fully supervised settings. The retrieved samples are empirically shown to be far away in the embedding space but semantically related to the query image, providing a stronger training signal compared to MSF.
3. CMSF achieves non-trivial gains in performance over self-supervised MSF and a direct extension of MSF to semi-supervised version. CMSF outperforms SOTA methods with comparable compute in self- and semi-supervised settings.

2 Method

Similar to MSF [37], given a query image, we are interested in pulling its embedding closer to the mean of the embeddings of its nearest neighbors (NNs). However, since top NNs are close to the target itself, they may not provide a strong supervision signal. On the other hand, far away (non-top) NNs may not be semantically similar to the target image. Hence, we constrain the NN search space to include mostly far away points with high purity. The purity is defined as the percentage of the selected NNs being from the same ground truth category as the query image. We use different constraint selection techniques to analyze our method in supervised, self- and semi-supervised settings.

Following MSF and BYOL, we use two embedding networks: a target encoder $f(.)$ with parameters θ_f and an online encoder $g(.)$ with parameters θ_g. The online encoder is directly updated using backpropagation while the target encoder is updated as a slowly moving average of the online encoder: $\theta_f \leftarrow m\theta_f + (1-m)\theta_g$ where m is close to 1. We add a predictor head $h(.)$ [27] to the end of the online encoder so that pulling the embeddings together encourages one embedding to be

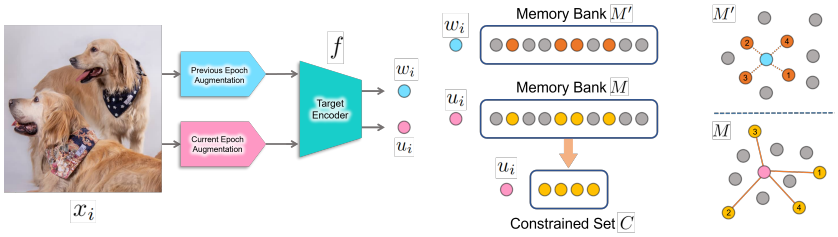


Fig. 3. CMSF$_{\text{self}}$: The indices of the NNs of the previous epoch's memory bank M' are used to construct the constrained set C from the current memory bank M.

predictable by the other one and not necessarily encouraging the two embeddings to be equal. In the experiments, we use a two-layer MLP for $h(.)$.

Given a query image x_i, we augment it twice with transformations $T_1(.)$ and $T_2(.)$, feed them to the two encoders, and normalize them with their ℓ_2 norm to get $u_i = \frac{f(T_1(x_i))}{||f(T_1(x_i))||_2}$ and $v_i = \frac{h(g(T_2(x_i)))}{||h(g(T_2(x_i)))||_2}$. We add u_i to the memory bank M and remove the oldest entries to maintain a fixed size M. We select the constraint set C_i as a subset of M. Constraint set selection is explained in detail in Sections 2.1, 2.2, and 2.3. We then find the set S_i of top-k nearest neighbors of u_i in C_i including u_i itself. Finally, we update $g(.)$ by minimizing:

$$L = \sum_{i=1}^{n} \frac{1}{|S_i|} \sum_{z \in S_i} v_i^T z$$

where n is the size of mini-batch and $|S_i|$ is the size of set S_i, *e.g.*, k in top-k. Finally, we update $f(.)$ with the momentum update. In the top-*all* variation of our method, number of neighbors k is set equal to the size of C_i, *i.e.*, $S_i = C_i$. Note that since u_i itself is included in the nearest neighbor search, the method will be identical to BYOL [27] when $k = 1$ and to self-supervised mean-shift [37] when the constraint is fully relaxed ($C_i = M$). Our method covers a larger spectrum of algorithms by defining the constrained set. Below we discuss the selection of constrained set in various settings.

2.1 Self-Supervised Setting

In addition to M, we maintain a second memory bank M' that is exactly the same as M but contains features from a different (3^{rd}) augmentation of the image x_i fed through target encoder $f(.)$. We assume $w_i \in M'$ and $u_i \in M$ are two embeddings corresponding to the same image x_i. Then, we find NNs of w_i in M' and use their indices to construct the search space C_i from M (See Fig. 3). Note that although the NNs of w_i in M' are already close to each other, their corresponding elements in M may not be close to each other since M contains different augmentations u_i of the same images. As a result, C_i will maintain good purity while containing distant NNs (refer to Table 1-Right and Fig. 5).

Since it is expensive to embed a 3rd augmentation of each image, we embed only two augmentations as in MSF and BYOL and cache the embeddings from

the previous epoch, keeping the most recent embedding for each image. The cached embedding will be still valid after one epoch since the target encoder is updated slowly using the momentum update rule (similar to MoCo). Since cache size is equal to the dataset size, we store it in the CPU memory and maintain the auxiliary memory bank M' by loading the corresponding part of it to the GPU memory for each mini-batch. Caching of features is not essential for CMSF to work and is only used to reduce computational cost. We performed experiments with an actual 3rd augmentation instead and found the results to be similar to our method except that it was nearly 30% slower due to forwarding an additional augmentation. Table 1-Right shows that in the intermediate stages of learning, the top elements of C_i are spread apart in M with higher median ranks, and get closer to the top elements of M as the learning progresses. Note that we use w_i instead of u_i in finding the NNs in M' since both w_i and M' use an older target model, so are more comparable.

Since CMSF adds farther NNs only for stronger supervision, we additionally employ MSF loss calculated on the unconstrained M. Then, in the self-supervised setting, the total loss is an equally weighted sum of MSF and CMSF losses.

Our method can be extended to cross-modal self-supervised setting where the constraint can use NNs in a different modality rather than the 3rd augmentation of the same modality. We report the details and some preliminary experiments on this setting in the supplementary.

2.2 Supervised Setting

While supervised setting is not our primary novelty or motivation, we study it to provide more insights into our constrained mean-shift framework. With access to the labels of each image, we can simply construct C_i as the subset of M that shares the same label as the query x_i. This guarantees 100% purity for NNs.

Note that most supervised methods, including cross-entropy loss, try to group all examples of a category together on the same side of a hyper-plane while remaining categories are on the other side. However, our method pulls the target to be close to only those examples of the same category that are already close to the target. This results in a supervised algorithm that may keep the latent structure of each category which can be useful for pre-training on coarse-grained labels. Moreover, as shown in the experiments (Fig. 6), our method is more robust to label noise since most mis-labeled images will be far from the target embedding, thus ignored in learning. This motivates applying our method to semi-supervised setting where the limited supervision provides noisy labels.

2.3 Semi-Supervised Setting

In this setting, we assume access to a dataset with a small labeled and a large unlabeled subset. We train a simple classifier using the current embeddings of the labeled data and use the classifier to pseudo-label the unlabeled data. Then, similar to the supervised setting, we construct C_i to be the elements of M that share the pseudo-label with the target embedding. Again, this method

increases the diversity of C_i while maintaining high purity. To keep the purity high, we enforce the constraint only when the pseudo-label is very confident (the probability is above a threshold.) For the samples with non-confident pseudo-label, we relax the constraint resulting in regular MSF loss (*i.e.,* $C_i = M$.) Moreover to reduce the computational overhead of pseudo-labeling, we cache the embeddings of labeled examples throughout the epoch and train a 2-layer MLP classifier using the frozen cached features and their groundtruth labels in the middle and end of each epoch.

3 Experiments

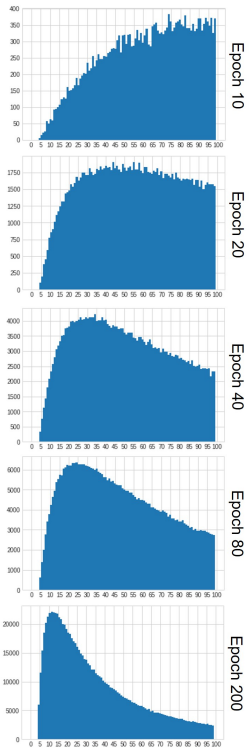
Implementation details: We use PyTorch for all our experiments. Unless specified, we use the same hyper-parameter values in self-, semi- and fully supervised settings. All models are trained on ImageNet-1k (IN-1k) for 200 epochs with ResNet-50 [30] backbone and SGD optimizer (learning rate=0.05, batch size=256, momentum=0.9, and weight decay=1e-4) with cosine scheduling for learning rate. While we focus on single crop setting in most of our experiments, we also report the results for multiple crop inputs in the SSL setting. Following SwAV [11], we use four additional crops of $96x96$ resolution as input. These are used as inputs only to the online encoder and not the target encoder. The momentum value of CMSF for the moving average key encoder is 0.99. The 2-layer MLP architecture for $\text{CMSF}_{\text{semi}}$ is as follows: (linear (2048x4096), batch norm, ReLU, linear (4096x512)). The default memory bank size is 128k. Top-k is set to 10 in the semi- and fully supervised settings and 5 in the self-supervised setting. Additional details are provided in the supplementary. Our main CMSF experiment with 200 epochs takes nearly 6 days on four NVIDIA-2080TI GPUs. The overhead in training time due to NN search is negligible compared to the forward and backward passes through the network (that is also done in BYOL): the increase in time is 0.7% for MSF [37] and 2.1% for $\text{CMSF}_{\text{self}}$.

Recent SSL methods are usually computationally expensive leading to worse environmental impact and exclusion of smaller research labs. While our experiments are more efficient and accessible than most SOTA methods, *e.g.*, PAWS, we limit our training length to 200 epochs due to resource constraints. We do not empirically verify whether the improvements observed over SOTA approaches at lower epochs (200) are persistent with longer training (*e.g.*, 800 or 1000 epochs).

Evaluation: We evaluate the pre-trained models using linear evaluation (*Linear IN-1k*) in both ImageNet classification and transfer settings. The model backbone parameters are fixed and a single linear layer is trained atop them following the setting in CompRess [2]. Additionally, we report k-nearest neighbor ($k = 1, 20$) evaluation for the SSL setting as in [2]. The transfer performance is evaluated on the following datasets: Food101 [8], SUN397 [73], CIFAR10 [39], CIFAR100 [39], Cars196 [38], Aircraft [43], Flowers (Flwrs102) [46], Pets [49], Caltech-101 (Calt101) [22], and DTD [18] (additional details in supplementary material.)

Table 1. Left: Evaluation on full ImageNet: We compare our model with other SOTA methods in Linear (Top-1 Linear) and Nearest Neighbor (1-NN,20-NN) evaluation. We use a memory bank of size 128K for CMSF and provide comparison with both 256K and 1M memory bank versions of MSF. Since $\text{CMSF}_{\text{self}}$ uses NNs from two memory banks, it is comparable to MSF (256K) in memory and computation overhead. Both single crop and multi-crop versions of our method outperform other SOTA methods, including MSF, with similar compute. **Right: Histogram of constrained sample ranks:** We consider the 5^{th} NN in the constrained set C and obtain its rank in the unconstrained memory bank M. The histogram of these ranks are shown up to rank 100 for different train stages of $\text{CMSF}_{\text{self}}$. Also, the median of these ranks are shown in Figure 5. A large number of distant neighbors are included in the constrained set in the early stages of training while there is a higher overlap between constrained and unconstrained NN sets towards the end of training.

Method	Ref.	Batch Size	Epochs	Sym. Loss 2x FLOPS	Multi-Crop Training	Top-1 Linear	NN	20-NN
Supervised	[1]	256	100	-	-	76.2	71.4	74.8
Random-init	-	-	-	-	-	5.1	1.5	2.0
SeLa-v2 [76]	[11]	4096	400	✓	✗	67.2	-	-
SimCLR[13]	[13]	4096	1000	✓	✗	69.3	-	-
SwAV [11]	[11]	4096	400	✓	✗	70.1	-	-
DeepCluster-v2 [10]	[11]	4096	400	✓	✗	70.2	-	-
SimSiam [16]	[16]	256	400	✓	✗	70.8	-	-
MoCo v2 [29]	[15]	256	800	✗	✗	71.1	57.3	61.0
CompRess [2]	[2]	256	1K+130	✗	✗	71.9	63.3	66.8
InvP	[66]	256	800	✗	✗	71.3	-	-
BYOL [27]	[27]	4096	1000	✓	✗	74.3	62.8	66.9
SwAV [11]	[11]	4096	800	✓	✓	75.3	-	-
NNCLR	[21]	4096	1000	✗	✗	**75.4**	-	-
SimCLR[13]	[16]	4096	200	✓	✗	68.3	-	-
SwAV [11]	[16]	4096	200	✓	✗	69.1	-	-
MoCo v2 [29]	[16]	256	200	✓	✗	69.9	-	-
SimSiam [16]	[16]	256	200	✓	✗	70.0	-	-
NNCLR[21]	[21]	4096	200	✗	✗	70.7	-	-
BYOL [27]	[16]	4096	200	✓	✗	70.6	-	-
SwAV [11]	[16]	256	200	✓	✓	72.7	-	-
Truncated Triplet [67]	[67]	832	200	✓	✗	73.8	-	-
OBoW [24]	[24]	256	200	✗	✓	73.8	-	-
$\text{CMSF}_{\text{self}}$ (128K)	-	256	200	✗	✓	**74.4**	**62.3**	**66.2**
MoCo v2 [29]	[15]	256	200	✗	✗	67.5	50.9	54.3
CO2 [69]	[69]	256	200	✗	✗	68.0	-	-
BYOL-asym [27]	[37]	256	200	✗	✗	69.3	55.0	59.2
ISD [60]	[60]	256	200	✗	✗	69.8	59.2	62.0
MSF (1M) [37]	[37]	256	200	✗	✗	72.4	62.0	64.9
MSF (256K)[37]	[37]	256	200	✗	✗	72.2	62.1	65.1
$\text{CMSF}_{\text{self}}$ (128K)	-	256	200	✗	✗	**73.0**	**63.2**	**66.4**



3.1 Self-Supervised Learning ($\text{CMSF}_{\text{self}}$)

To reduce the GPU memory footprint, we cache the previous augmentation embedding of each sample in the dataset in the CPU. The cached features corresponding to the current mini-batch are retrieved from CPU memory to maintain memory bank M' with previous augmentations. This cache is updated using the oldest features in M that we remove from M after each iteration.

Results on ImageNet: Results of $\text{CMSF}_{\text{self}}$ are shown in Table 1. $\text{CMSF}_{\text{self}}$ outperforms MSF baseline with a larger memory bank, which we believe is due to

Fig. 4. Nearest neighbor selection on constrained memory bank: First row shows top-5 NNs of target in constrained set C and their corresponding rank in the unconstrained memory bank M obtained using an intermediate checkpoint (epoch 100). While they are not the closest samples to the target (higher rank index), they are semantically similar to the target. This shows that the constraint can capture far away samples with similar semantic as the target. The second row depicts images from memory bank with one rank lower than the corresponding image in the first row. These images contain incorrect category retrievals. Distant neighbors cannot be trivially obtained by increasing the number of NNs. Examples are chosen randomly.

pulling together far yet semantically similar samples (Fig. 4). We use MSF with $2x$ larger memory bank for fair comparison. $\text{CMSF}_{\text{self}}$ also achieves state-of-the-art performance on both NN and Linear metrics when compared with approaches with similar computational budget. We compare our method to other state-of-the-art approaches with 200 epochs of training in Fig. 1. We observe a good trade-off in terms of accuracy and compute for $\text{CMSF}_{\text{self}}$. Our best performance is obtained with the multi-crop version but at the cost of increased compute.

Evaluation on ImageNet subsets: Following [31,13], we evaluate the pre-trained models on the ImageNet classification task with limited labels. We report results with 1% and 10% labeled subsets of ImageNet (Table 4). $\text{CMSF}_{\text{self}}$ outperforms MSF on top-1 accuracy in both 1% and 10% settings and is comparable to existing approaches that require significantly higher training time.

Transfer learning: We follow the procedure in [27,13] for transfer evaluation (refer to Table 2). Hyperparameters for each dataset are tuned independently based on the validation set accuracy and final accuracy is reported on the held-out test set (more details in supplementary). $\text{CMSF}_{\text{self}}$ achieves SOTA average performance among methods trained for 200 epochs.

Purity of constrained samples: In $\text{CMSF}_{\text{self}}$, we depend on information from previous augmentations to constrain NN search in the current memory bank. Our goal is to improve learning by using distant samples with a good purity. We observe that the top-k samples from constrained memory bank C have higher rank in M, so are far neighbors of the target (see Table 1-Right and Fig. 5). Also, as shown in Fig. 5, those samples maintain almost the same purity as the top-k samples from unconstrained memory bank M. As a result, C maintains good purity while being diverse.

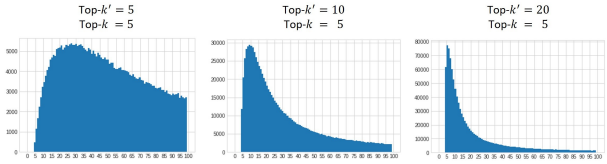
Effect of k': In $\text{CMSF}_{\text{self}}$, we first calculate top-k' samples (the first k' NNs of the target) from the secondary memory bank M'. We then use those indices to constrain NN search space in the primary memory bank M and select top-k for optimization. We varied the value of k' in $\text{CMSF}_{\text{self}}$ to explore its effect, keeping k fixed to 5. We observe that increasing k' (relaxing the constraint) will decrease the accuracy of the model. As observed in Table 3-right, the overlap between

Table 2. Transfer learning evaluation: Our supervised CMSF model at just 200 epochs outperforms all supervised baselines on transfer learning evaluation. Our SSL model outperforms MSF, the comparable state-of-the-art approach, by 1.2 points on average over 10 datasets. We get the results for MoCo v2, MSF, and BYOL-asym from [37], SimCLR and Xent (1000 epoch) from [13], and BYOL from [27].

Method	Epoch	Food 101	CIFAR 10	CIFAR 100	SUN 397	Cars 196	Air-craft	DTD	Pets	Calt. 101	Flwr 102	**Mean Trans**	**Linear IN-1k**
Supervised Models													
Xent	200	67.7	89.8	72.5	57.5	43.7	39.8	67.9	91.8	91.1	88.0	71.0	77.2
Xent	90	72.8	91.0	74.0	59.5	56.8	48.4	70.7	92.0	90.8	93.0	74.9	76.2
ProtoNW	200	73.3	93.2	78.3	61.5	65.0	57.6	73.7	92.2	94.3	93.7	78.3	76.0
SupCon	200	72.5	93.8	77.7	61.5	64.8	58.6	74.6	**92.5**	93.6	94.1	78.4	**77.5**
Xent	1000	72.3	93.6	78.3	61.9	66.7	61.0	**74.9**	91.5	94.5	94.7	78.9	76.3
CMSF_{sup} top-*all*	200	73.7	94.2	**78.7**	62.1	**71.7**	**64.1**	73.4	**92.5**	94.5	**95.8**	**80.1**	75.7
CMSF_{sup} top-10	200	**74.9**	**94.4**	**78.7**	**62.7**	70.8	63.4	73.8	92.2	**94.9**	95.6	**80.1**	76.4
Self-Supervised Models													
SimCLR	1000	72.8	90.5	74.4	60.6	49.3	49.8	**75.7**	84.6	89.3	92.6	74.0	69.3
MoCo v2	800	72.5	**92.2**	74.6	59.6	50.5	53.2	74.4	84.6	90.0	90.5	74.2	71.1
BYOL	1000	**75.3**	91.3	**78.4**	**62.2**	**67.8**	**60.6**	75.5	**90.4**	**94.2**	**96.1**	**79.2**	**74.3**
MoCo v2	200	70.4	91.0	73.5	57.5	47.7	51.2	73.9	81.3	88.7	91.1	72.6	67.5
BYOL-asym	200	70.2	91.5	74.2	59.0	54.0	52.1	73.4	86.2	90.4	92.1	74.3	69.3
MSF	200	72.3	**92.7**	76.3	60.2	59.4	56.3	71.7	89.8	90.9	93.7	76.3	72.1
$\text{CMSF}_{\text{self}}$	200	**73.0**	92.2	**77.2**	**61.0**	**60.6**	**58.4**	**74.1**	**91.1**	**92.0**	**94.5**	**77.4**	**73.0**

Table 3. Effect of k' in sampling NN from M': In $\text{CMSF}_{\text{self}}$, we constrain top-$k$ NN search space in M with top-k' samples from M'. (**Left**) Increasing k' results in a drop in accuracy. The k in top-k is set to 5 for all values of k'. (**Right**) Histogram of the constrained sample ranks at epoch 50. The histogram shifts left, *i.e.*, overlap between constrained and unconstrained NN sets increases with increasing value of k'.

k'	5	10	20	40	80
NN	63.2	62.9	62.7	62.3	61.7
20-NN	66.4	66.1	65.9	65.6	65.0



constrained and unconstrained NN set increases with increasing value of k'. Note that in a case where $k' = \infty$, $\text{CMSF}_{\text{self}}$ will be identical to the MSF baseline.

3.2 Supervised Learning

Evaluation: Unlike cross-entropy (Xent [7,42,52]) baseline, SupCon [36], ProtoNW [55] and CMSF do not train a linear classifier during the pre-training stage. Thus, we use the pre-training dataset ImageNet-1k (IN-1k) for linear evaluation of the frozen features as done in SSL. For Xent, we use the linear classifier trained during pre-training. We use the same settings and datasets as self-supervised for transfer learning evaluation.

Results: Results on IN-1k dataset are shown in Table 2. In top-*all* variation of our method, k is equal to the total size of C. SSL inspired methods like CMSF

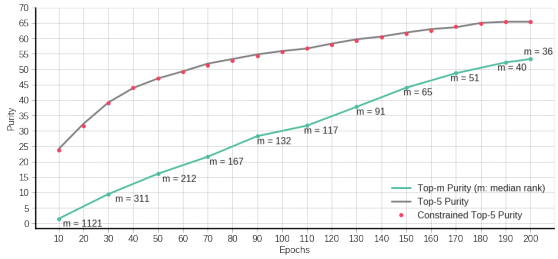


Fig. 5. Purity of constrained samples: During training of $\text{CMSF}_{\text{self}}$, we plot purity of the top-5 samples in unconstrained set M (in black) and that of the top-5 samples in constrained set C (in red). The red curve is not significantly below the black one suggesting that the purity is not dropped by increasing the distance of the NNs. To show that elements in C may be far from the target u, we choose the 5^{th} element in C and find its rank in the set M. We calculate the median of this rank as m. The purity of the top-m elements of set M (green curve) is consistently lower than that of top-5 elements of the constrained set C (red curve). This suggests that one cannot maintain high purity by simply considering more NNs using a larger k.

and SupCon significantly outperform Xent when trained for similar number of epochs. We observe that improvements in ImageNet performance do not always translate to transfer performance. Interestingly, CMSF performs the best on transfer evaluation, particularly on fine-grained datasets like Cars196 and Aircraft. We believe that the absence of explicit cross-entropy based optimization using the supervised labels preserves the multi-modal distribution of categories improving fine-grained performance. Supervised CMSF uses class labels only as a constraint for MSF during pre-training and does not explicitly optimize on the classification task. Superior performance of CMSF_{sup} top-10 demonstrates the importance of using distant yet semantically related neighbors as positives.

Noisy Labels: In the noisy setting, we use random i.i.d. noise to corrupt the labels (change the label randomly) of a percentage of images. We consider, 5%, 10%, 25%, and 50% label corruption (noise) rates. For faster experiments, we report results on the ImageNet-100 dataset [61] (Fig. 6). We observe a significantly higher degradation in performance of Xent baseline and CMSF_{sup} top-*all* compared to CMSF_{sup} top-10 at high noise levels. The gap between the approaches is larger on transfer learning. These observations indicate that NN based methods like CMSF are better suited for noisy constraint settings compared to approaches utilizing all samples of a class as positives. This robustness to label noise motivates our application of CMSF to self- and semi-supervised settings where pseudo-labels or the NNs of previous augmentations may be noisy.

Coarse-grained ImageNet: CMSF groups together only top-k neighbors and thus can help in preserving the latent structure of the data compared to top-*all*. To verify this, we consider a dataset with coarse-grained labels where this difference is pronounced. Based on the WordNet hierarchy, we merge each category in the ImageNet dataset to its parent class. We further ensure that no two classes are in the same path in the graph by merging the descendant into the ancestor

Table 4. Evaluation on small labeled ImageNet : We compare our model to MSF and other baselines on ImageNet 1% and 10% linear evaluation benchmarks. "Fine-tuned" refers to fine-tuning the entire backbone network instead of a single linear layer. $\text{CMSF}_{\text{self}}$ outperforms MSF on top-1 metric in both 1% and 10% settings.

Method	Fine-tuned	Epochs	Top-1		Top-5	
			1%	10%	1%	10%
Supervised	✓		25.4	56.4	48.4	80.4
PIRL [44]	✓	800	-	-	57.2	83.8
CO2 [69]	✓	200	-	-	71.0	85.7
SimCLR [13]	✓	1000	48.3	65.6	75.5	87.8
InvP [66]	✓	800	-	-	78.2	88.7
BYOL [27]	✓	1000	53.2	68.8	78.4	89.0
SwAV [11]	✓	800	**53.9**	**70.2**	**78.5**	**89.9**
MoCo v2 [15]	✗	800	51.5	63.6	77.6	86.1
BYOL [27]	✗	1000	55.7	**68.6**	80.0	**88.6**
CompRess [2]	✗	1K+130	**59.7**	67.0	**82.3**	87.5
MoCo v2 [15]	✗	200	43.6	58.4	71.2	82.9
BYOL-asym	✗	200	47.9	61.3	74.6	84.7
ISD [60]	✗	200	53.4	63.0	78.8	85.9
MSF [37]	✗	200	55.5	66.5	**79.9**	87.6
$\text{CMSF}_{\text{self}}$	✗	200	**56.4**	**67.5**	79.8	**87.7**

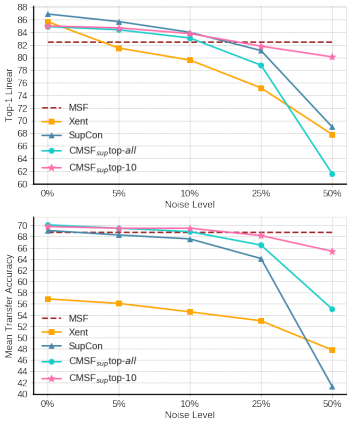


Fig. 6. Noisy supervised setting on ImageNet-100: Our method is more robust to noisy annotation compared to Xent and SupCon. Also, using top-*all* degrades the results since all images from a single category are not guaranteed to be semantically related due to noisy labels. Mean Transfer Accuracy is the average over 10 transfer datasets.

Table 5. Supervised learning on coarse grained ImageNet: We train on the coarse grained version of ImageNet (93 super categories) and perform linear evaluation on the original ImageNet-1k validation set with fine-grained labels (1000 categories).

Train Dataset	ImageNet-1k Validation Set			
	Xent	SupCon	CMSF_{sup} top-*all*	CMSF_{sup} top-10
ImageNet-1k	77.2	**77.5**	75.7	76.4
ImageNet-coarse	61.4	58.7	67.0	**74.2**

class. The total number of classes is thus reduced from 1000 in ImageNet-1k to 93 in our ImageNet-coarse. We train CMSF and the baseline approaches in a supervised manner using the coarse labels and then evaluate on the fine-grained / original labels on ImageNet-1k validation set. In Table 5 we compare the top-*all* and top-k variants on the coarse grained version of ImageNet. CMSF_{sup} top-k sees a minor drop in performance compared to training on ImageNet-1k. However, methods in which all samples in a class are explicitly brought closer - CMSF_{sup} top-*all*, cross-entropy and supervised contrastive - see a huge drop in accuracy. More details on coarse-grained ImageNet are in the supplementary.

3.3 Semi-Supervised Learning

Implementation Details: We train a 2-layer MLP atop the cached target features of supervised set for pseudo-labeling. The pseudo-label training is performed

Table 6. Semi-supervised learning on ImageNet dataset with 10% labels: FLOPs denotes the total number of FLOPS for forward and backward passes through ResNet-50 backbone while batch size denotes the sum of labeled and unlabeled samples in a batch. $CMSF_{semi}$-mix precision is compute and resource efficient, achieving SOTA performance at comparable compute. PAWS requires large number of GPUs to be compute efficient and its performance drastically drops with 4/8 GPUs. † Trained with stronger augmentations like RandAugment [19]. * TPUs are used.

Method	Epochs	Batch Size	GPUs	FLOPs ($x10^{18}$)	Top-1
Self-supervised Pre-training					
Mean Shift [37]	200	256	4	4	67.4
BYOL [27]	1000	4096	512*	40	68.8
SwAV [11]	800	4096	64	37	70.2
SimCLRv2 [14]	800	4096	128*	16	68.4
Semi-supervised Pre-training					
SimCLRv2 (+Self Dist) [14]	1200	4096	128*	20	70.5
UDA† [74]	800	15872	64*	10	68.1
FixMatch† [57]	300	6144	32*	7	71.5
MPL† [50]	800	2048	-	30	73.9
PAWS (support=6720) [4]	300	4096	64	21	75.5
PAWS (support=1680) [4]	100	256	8	15	70.2
PAWS (support=400) [4]	100	256	4	7	62.9
$CMSF_{semi}$-basic	200	256	4	4	68.6
$CMSF_{semi}$	200	256	4	4	69.9
$CMSF_{semi}$-mix precision	200	768	4	4	70.5

twice per epoch (takes 40 seconds per training) and the label assignment is done in an online fashion for each mini-batch. The confidence threshold for pseudo-labeling is set to 0.85. We use the same optimizer settings as in self-supervised CMSF for the pre-training stage. Similar to S4L [78], we perform two stages of fine-tuning with supervised and pseudo-labels. We fine-tune the backbone network with two MLPs (as in PAWS [4]) on the 10% labeled set for 20 epochs and pseudo-label the train set. Samples above confidence threshold (nearly 30% of dataset) are combined with supervised set to fine-tune again for 20 epochs (more details in suppl.). The second fine-tuning is equivalent to 5 epochs with full data and is a small increase in our total compute. This is needed since we do not directly optimize cross-entropy loss in pre-training as in [57,74,50].

Evaluation: The final epoch parameters are used to perform evaluation. We report top-1 accuracy on the ImageNet validation set. We additionally report the total number of FLOPs for forward and backward passes (backward is 2× forward) through ResNet-50 backbone and the number of GPUs/TPUs used by each method in the pre-training stage (more details in suppl.).

Baselines: We compare the proposed approach ($CMSF_{semi}$) with self- and semi-supervised approaches. *$CMSF_{semi}$-basic* minimizes unconstrained MSF loss on the unlabeled examples (no pseudo-labeling) and CMSF loss on the labeled examples only. We provide comparison of PAWS method with different support set sizes. We train PAWS on 4x 16GB GPUs with maximum possible support set size (200 classes, 2 images/class) using code provided by the authors. We also

report results using mixed precision training (*CMSF$_{semi}$-mix precision*) as in PAWS [4] with a higher batch size of 768 since it has lower memory requirement. **Results:** CMSF$_{\text{semi}}$-mix precision achieves comparable performance to most methods with significantly less training and without the use of stronger augmentation schemes like RandAugment [19] (Table 6, Fig. 1). PAWS with a support set size of 6720 outperforms other approaches. However, this requires significantly higher compute (4.8× FLOPs) and resources (64 GPUs) compared to CMSF$_{\text{semi}}$-mix precision (4 GPUs). Since PAWS requires a large support set, it does not scale well to lower resource (4/8 GPUs) settings even if the total compute remains the same. When trained on only 4 GPUs, CMSF outperforms PAWS by **7.6%** points. Additional ablations and results on ImageNet-100 dataset are in supplementary.

4 Related Work

Self-supervised learning (SSL): Earlier works on SSL focused on solving a pretext task that does not require additional labeling. Examples of pretext tasks include colorization [80], jigsaw puzzle [47], counting [48], and rotation prediction [25]. Another class of SSL methods is based on instance discrimination [20]. The idea is to classify each image as its own class. Some methods adopt the idea of contrastive learning for instance discrimination [29,13,10,11,12]. BYOL [27] proposes a non-contrastive approach by removing the negative set and simply regressing one view of an image from another.

Several recent works aim to find a larger positive sample set to improve learning. In LA [82], samples are clustered using k-means and samples within a cluster are brought closer together compared to cross-cluster samples. MSF [37] and MYOW [5] generalize BYOL by regressing target view and its NNs. NNCLR [21] extends SimCLR to use NNs as positives. CLD [68] integrates grouping using instance-group discrimination. Affinity diffusion [33] uses strongly connected nodes in a graph constructed using embeddings to find positive samples. Unlike these methods, we focus on grouping together far away neighbors that are semantically similar. We show quantitatively and qualitatively the diversity and purity of retrieved neighbors and improved performance over MSF. We generalize the idea in MSF [37] to use an additional source of knowledge to constrain the NN search space for the target view. CoCLR [28] and Cl-InfoNCE [64] also use additional information sources in the form of additional modality and auxiliary labels respectively to improve performance. However, we focus on self- and semi-supervised classification settings and design methods to obtain and use the additional information as a constraint in NN search space.

Supervised learning: A drawback of Cross-entropy is its lack of robustness to noisy labels [81,58]. [59,45,63,75] address the issue of hard labeling, *e.g.*, (one-hot labels) with label smoothing, [32,6,23] replace hard labels with prediction of pre-trained teacher, and [79,77] propose an augmentation strategy to train on combination of instances and their labels. Another line of work [26,53] is to learn representations with good kNN performance. SupCon [36] and [72] improve upon [26] by changing the distance to inner product on ℓ_2 normalized embeddings. We

include the supervised setting to better understand the effect of using constrained NNs, particularly in the noisy label setting.

Semi-supervised learning: Several methods combine self-supervised and supervised learning to form semi-supervised methods. S4L [78] uses rotation prediction based loss on the unlabeled set along with cross-entropy loss on the labeled set. Similarly, SuNCEt [3] combines SimCLR [13] and SwAV [11] methods with supervised contrastive loss. Pseudo-labeling is frequently used in semi-supervised learning. In Pseudo-Label [40], the network is trained with cross-entropy loss using supervised data on the labeled examples and pseudo-labels on the unlabeled ones. In SimCLR-v2 [14], a teacher network is pre-trained using SimCLR [13] and fine-tuned with supervised labels. The teacher is then distilled to a student network using pseudo-labels on the unlabeled set. FixMatch [57] uses pseudo-labels obtained using a weakly augmented image to train a strongly augmented version of the same image. UDA [74] leverages strong data augmentation techniques in enforcing this consistency in pseudo-labels across augmentations. MPL [50] optimizes a student network using pseudo-labels from a teacher network, while the teacher is optimized to maximize the student's performance on the labeled set. PAWS [4] uses consistency based loss on soft pseudo-labels obtained in a non-parametric manner. Our method too uses pseudo-labels to train the unlabeled samples. However, we use the labels as a constraint in MSF [37] and do not directly optimize samples using cross-entropy loss.

Metric learning: The goal of metric learning is to train a representation that puts two instances close in the embedding space if they are semantically close. Two important methods in metric learning are: triplet loss [17,70,54] and contrastive loss [56,9]. Metric learning methods perform well on tasks like image retrieval [71] and few-shot learning [65,55]. Prototypical networks [55] is similar to a contrastive version of our method with top-*all*.

5 Conclusion

MSF is a recent SSL method that pulls an image towards its nearest neighbors. We argue that the model can benefit from more diverse yet pure neighbors. Hence, we generalize MSF method by constraining the NN search. This opens the door to using the mean-shift idea to various settings of self-supervised, supervised, and semi-supervised. To construct the constraint, our SSL method uses cached augmentations from the previous epoch while the supervised and semi-supervised settings use labels or pseudo-labels. We show that our method outperforms SOTA approaches like MSF in SSL, PAWS in semi-supervised, and supervised contrastive in transfer-learning evaluation of supervised settings.

Acknowledgments: This material is based upon work partially supported by DARPA under Contract No. HR00112190135, the United States Air Force under Contract No. FA8750-19-C-0098, funding from SAP SE, and NSF grants 1845216 and 1920079. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the United States Air Force, DARPA, or other funding agencies.

References

1. Torchvision models. https://pytorch.org/docs/stable/torchvision/models.html 7
2. Abbasi Koohpayegani, S., Tejankar, A., Pirsiavash, H.: Compress: Self-supervised learning by compressing representations. Advances in Neural Information Processing Systems **33** (2020) 6, 7, 11
3. Assran, M., Ballas, N., Castrejon, L., Rabbat, M.: Supervision accelerates pre-training in contrastive semi-supervised learning of visual representations. arXiv preprint arXiv:2006.10803 (2020) 14
4. Assran, M., Caron, M., Misra, I., Bojanowski, P., Joulin, A., Ballas, N., Rabbat, M.: Semi-supervised learning of visual features by non-parametrically predicting view assignments with support samples. ICCV (2021) 2, 12, 13, 14
5. Azabou, M., Azar, M.G., Liu, R., Lin, C.H., Johnson, E.C., Bhaskaran-Nair, K., Dabagia, M., Avila-Pires, B., Kitchell, L., Hengen, K.B., et al.: Mine your own view: Self-supervised learning through across-sample prediction. arXiv preprint arXiv:2102.10106 (2021) 1, 13
6. Bagherinezhad, H., Horton, M., Rastegari, M., Farhadi, A.: Label refinery: Improving imagenet classification through label progression. arXiv preprint arXiv:1805.02641 (2018) 13
7. Baum, E., Wilczek, F.: Supervised learning of probability distributions by neural networks. In: Anderson, D. (ed.) Neural Information Processing Systems. American Institute of Physics (1988), https://proceedings.neurips.cc/paper/1987/file/eccbc87e4b5ce2fe28308fd9f2a7baf3-Paper.pdf 9
8. Bossard, L., Guillaumin, M., Van Gool, L.: Food-101 – mining discriminative components with random forests. In: European Conference on Computer Vision (2014) 6
9. Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., Shah, R.: Signature verification using a" siamese" time delay neural network. Advances in neural information processing systems **6**, 737–744 (1993) 14
10. Caron, M., Bojanowski, P., Joulin, A., Douze, M.: Deep clustering for unsupervised learning of visual features. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 132–149 (2018) 7, 13
11. Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A.: Unsupervised learning of visual features by contrasting cluster assignments. In: Advances in Neural Information Processing Systems. pp. 9912–9924. Curran Associates, Inc. (2020), https://proceedings.neurips.cc/paper/2020/file/70feb62b69f16e0238f741fab228fec2-Paper.pdf 6, 7, 11, 12, 13, 14
12. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers (2021) 13
13. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International conference on machine learning. pp. 1597–1607. PMLR (2020) 7, 8, 9, 11, 13, 14
14. Chen, T., Kornblith, S., Swersky, K., Norouzi, M., Hinton, G.E.: Big self-supervised models are strong semi-supervised learners. Advances in Neural Information Processing Systems **33**, 22243–22255 (2020) 12, 14
15. Chen, X., Fan, H., Girshick, R., He, K.: Improved baselines with momentum contrastive learning. arXiv preprint arXiv:2003.04297 (2020) 7, 11
16. Chen, X., He, K.: Exploring simple siamese representation learning. arXiv preprint arXiv:2011.10566 (2020) 7

17. Chopra, S., Hadsell, R., LeCun, Y.: Learning a similarity metric discriminatively, with application to face verification. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). vol. 1, pp. 539–546. IEEE (2005) 14
18. Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., Vedaldi, A.: Describing textures in the wild. In: Computer Vision and Pattern Recognition (2014) 6
19. Cubuk, E.D., Zoph, B., Shlens, J., Le, Q.: Randaugment: Practical automated data augmentation with a reduced search space. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 18613–18624. Curran Associates, Inc. (2020), https://proceedings.neurips.cc/paper/2020/file/d85b63ef0ccb114d0a3bb7b7d808028f-Paper.pdf 12, 13
20. Dosovitskiy, A., Springenberg, J.T., Riedmiller, M., Brox, T.: Discriminative unsupervised feature learning with convolutional neural networks. In: Advances in neural information processing systems. pp. 766–774 (2014) 13
21. Dwibedi, D., Aytar, Y., Tompson, J., Sermanet, P., Zisserman, A.: With a little help from my friends: Nearest-neighbor contrastive learning of visual representations (2021) 1, 7, 13
22. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. Computer Vision and Pattern Recognition Workshop (2004) 6
23. Furlanello, T., Lipton, Z.C., Tschannen, M., Itti, L., Anandkumar, A.: Born again neural networks (2018) 13
24. Gidaris, S., Bursuc, A., Puy, G., Komodakis, N., Cord, M., Perez, P.: Obow: Online bag-of-visual-words generation for self-supervised learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6830–6840 (June 2021) 7
25. Gidaris, S., Singh, P., Komodakis, N.: Unsupervised representation learning by predicting image rotations. In: International Conference on Learning Representations (2018), https://openreview.net/forum?id=S1v4N2l0- 13
26. Goldberger, J., Hinton, G.E., Roweis, S., Salakhutdinov, R.R.: Neighbourhood components analysis. Advances in neural information processing systems **17**, 513–520 (2004) 13
27. Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P.H., Buchatskaya, E., Doersch, C., Pires, B.A., Guo, Z.D., Azar, M.G., et al.: Bootstrap your own latent: A new approach to self-supervised learning. arXiv preprint arXiv:2006.07733 (2020) 1, 3, 4, 7, 8, 9, 11, 12, 13
28. Han, T., Xie, W., Zisserman, A.: Self-supervised co-training for video representation learning (2021) 13
29. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9729–9738 (2020) 1, 7, 13
30. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016) 6
31. Hénaff, O.J., Srinivas, A., De Fauw, J., Razavi, A., Doersch, C., Eslami, S., Oord, A.v.d.: Data-efficient image recognition with contrastive predictive coding. arXiv preprint arXiv:1905.09272 (2019) 8
32. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015) 13

33. Huang, J., Dong, Q., Gong, S., Zhu, X.: Unsupervised deep learning via affinity diffusion. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 11029–11036 (2020) 13
34. Huynh, T., Kornblith, S., Walter, M.R., Maire, M., Khademi, M.: Boosting contrastive self-supervised learning with false negative cancellation. arXiv preprint arXiv:2011.11765 (2020) 1
35. Kalantidis, Y., Sariyildiz, M.B., Pion, N., Weinzaepfel, P., Larlus, D.: Hard negative mixing for contrastive learning. Advances in Neural Information Processing Systems (2020) 1
36. Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., Krishnan, D.: Supervised contrastive learning. Advances in Neural Information Processing Systems **33** (2020) 9, 13
37. Koohpayegani, S.A., Tejankar, A., Pirsiavash, H.: Mean shift for self-supervised learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 10326–10335 (October 2021) 1, 2, 3, 4, 6, 7, 9, 11, 12, 13, 14
38. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3D object representations for fine-grained categorization. In: Workshop on 3D Representation and Recognition. Sydney, Australia (2013) 6
39. Krizhevsky, A.: Learning multiple layers of features from tiny images. Tech. rep., University of Toronto (2009) 6
40. Lee, D.H., et al.: Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In: Workshop on challenges in representation learning, ICML. vol. 3, p. 896 (2013) 14
41. Lee, K., Zhu, Y., Sohn, K., Li, C.L., Shin, J., Lee, H.: *i*-mix: A domain-agnostic strategy for contrastive representation learning. In: International Conference on Learning Representations (2020) 1
42. Levin, E., Fleisher, M.: Accelerated learning in layered neural networks. Complex systems **2**(625-640), 3 (1988) 9
43. Maji, S., Rahtu, E., Kannala, J., Blaschko, M.B., Vedaldi, A.: Fine-grained visual classification of aircraft. arXiv preprint arXiv:1306.5151 (2013) 6
44. Misra, I., van der Maaten, L.: Self-supervised learning of pretext-invariant representations. arXiv preprint arXiv:1912.01991 (2019) 11
45. Müller, R., Kornblith, S., Hinton, G.: When does label smoothing help? (2020) 13
46. Nilsback, M.E., Zisserman, A.: Automated flower classification over a large number of classes. In: Indian Conference on Computer Vision, Graphics and Image Processing (2008) 6
47. Noroozi, M., Favaro, P.: Unsupervised learning of visual representations by solving jigsaw puzzles. In: European Conference on Computer Vision. pp. 69–84. Springer (2016) 13
48. Noroozi, M., Pirsiavash, H., Favaro, P.: Representation learning by learning to count. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 5898–5906 (2017) 13
49. Parkhi, O.M., Vedaldi, A., Zisserman, A., Jawahar, C.V.: Cats and dogs. In: Computer Vision and Pattern Recognition (2012) 6
50. Pham, H., Dai, Z., Xie, Q., Le, Q.V.: Meta pseudo labels. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11557–11568 (2021) 12, 14
51. Reed, C.J., Metzger, S., Srinivas, A., Darrell, T., Keutzer, K.: Selfaugment: Automatic augmentation policies for self-supervised learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2674–2683 (2021) 1

52. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. nature **323**(6088), 533–536 (1986) 9
53. Salakhutdinov, R., Hinton, G.: Learning a nonlinear embedding by preserving class neighbourhood structure. In: Artificial Intelligence and Statistics. pp. 412–419. PMLR (2007) 13
54. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 815–823 (2015) 14
55. Snell, J., Swersky, K., Zemel, R.S.: Prototypical networks for few-shot learning. arXiv preprint arXiv:1703.05175 (2017) 9, 14
56. Sohn, K.: Improved deep metric learning with multi-class n-pair loss objective. In: Proceedings of the 30th International Conference on Neural Information Processing Systems. pp. 1857–1865 (2016) 14
57. Sohn, K., Berthelot, D., Carlini, N., Zhang, Z., Zhang, H., Raffel, C.A., Cubuk, E.D., Kurakin, A., Li, C.L.: Fixmatch: Simplifying semi-supervised learning with consistency and confidence. Advances in Neural Information Processing Systems **33** (2020) 12, 14
58. Sukhbaatar, S., Bruna, J., Paluri, M., Bourdev, L., Fergus, R.: Training convolutional networks with noisy labels (2015) 13
59. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision (2015) 13
60. Tejankar, A., Koohpayegani, S.A., Pillai, V., Favaro, P., Pirsiavash, H.: Isd: Self-supervised learning by iterative similarity distillation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 9609–9618 (October 2021) 7, 11
61. Tian, Y., Krishnan, D., Isola, P.: Contrastive multiview coding. arXiv preprint arXiv:1906.05849 (2019) 10
62. Tian, Y., Sun, C., Poole, B., Krishnan, D., Schmid, C., Isola, P.: What makes for good views for contrastive learning? In: Advances in Neural Information Processing Systems. vol. 33, pp. 6827–6839. Curran Associates, Inc. (2020), https://proceedings.neurips.cc/paper/2020/file/4c2e5eaae9152079b9e95845750bb9ab-Paper.pdf 1
63. Touvron, H., Sablayrolles, A., Douze, M., Cord, M., Jégou, H.: Grafit: Learning fine-grained image representations with coarse labels (2020) 13
64. Tsai, Y.H.H., Li, T., Liu, W., Liao, P., Salakhutdinov, R., Morency, L.P.: Integrating auxiliary information in self-supervised learning (2021) 13
65. Vinyals, O., Blundell, C., Lillicrap, T., Kavukcuoglu, K., Wierstra, D.: Matching networks for one shot learning (2017) 14
66. Wang, F., Liu, H., Guo, D., Fuchun, S.: Unsupervised representation learning by invariance propagation. In: Advances in Neural Information Processing Systems. vol. 33, pp. 3510–3520. Curran Associates, Inc. (2020), https://proceedings.neurips.cc/paper/2020/file/23af4b45f1e166141a790d1a3126e77a-Paper.pdf 7, 11
67. Wang, G., Wang, K., Wang, G., Torr, P.H.S., Lin, L.: Solving inefficiency of self-supervised representation learning (2021) 1, 7
68. Wang, X., Liu, Z., Yu, S.X.: Unsupervised feature learning by cross-level instance-group discrimination. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 12586–12595 (June 2021) 13
69. Wei, C., Wang, H., Shen, W., Yuille, A.: Co2: Consistent contrast for unsupervised visual representation learning. arXiv preprint arXiv:2010.02217 (2020) 7, 11

70. Weinberger, K.Q., Blitzer, J., Saul, L.K.: Distance metric learning for large margin nearest neighbor classification. In: Advances in neural information processing systems. pp. 1473–1480 (2006) 14
71. Wu, C.Y., Manmatha, R., Smola, A.J., Krahenbuhl, P.: Sampling matters in deep embedding learning. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (Oct 2017) 14
72. Wu, Z., Efros, A.A., Yu, S.X.: Improving generalization via scalable neighborhood component analysis (2018) 13
73. Xiao, J., Hays, J., Ehinger, K.A., Oliva, A., Torralba, A.: Sun database: Large-scale scene recognition from abbey to zoo. In: Computer Vision and Pattern Recognition (2010) 6
74. Xie, Q., Dai, Z., Hovy, E., Luong, M.T., Le, Q.V.: Unsupervised data augmentation for consistency training. NeurIPS (2020) 12, 14
75. Xu, Y., Qian, Q., Li, H., Jin, R., Hu, J.: Weakly supervised representation learning with coarse labels (2021) 13
76. YM., A., C., R., A., V.: Self-labelling via simultaneous clustering and representation learning. In: International Conference on Learning Representations (2020), https://openreview.net/forum?id=Hyx-jyBFPr 7
77. Yun, S., Han, D., Oh, S.J., Chun, S., Choe, J., Yoo, Y.: Cutmix: Regularization strategy to train strong classifiers with localizable features (2019) 13
78. Zhai, X., Oliver, A., Kolesnikov, A., Beyer, L.: S4l: Self-supervised semi-supervised learning. In: The IEEE International Conference on Computer Vision (ICCV) (October 2019) 12, 14
79. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization (2018) 13
80. Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: European conference on computer vision. pp. 649–666. Springer (2016) 13
81. Zhang, Z., Sabuncu, M.R.: Generalized cross entropy loss for training deep neural networks with noisy labels. arXiv preprint arXiv:1805.07836 (2018) 13
82. Zhuang, C., Zhai, A.L., Yamins, D.: Local aggregation for unsupervised learning of visual embeddings. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6002–6012 (2019) 13