Unsupervised Representation Transfer for Small Networks: I Believe I Can Distill On-the-Fly

Anonymous Author(s)
Affiliation
Address
email

Abstract

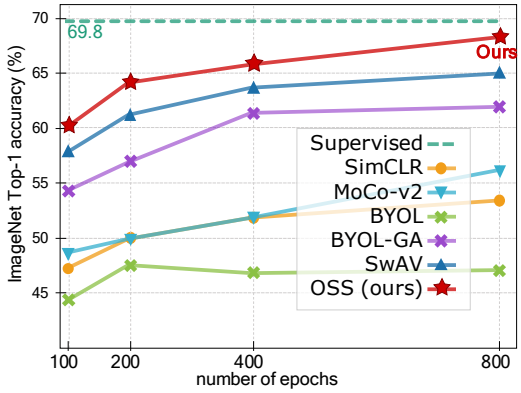
A current remarkable improvement of unsupervised visual representation learning is based on heavy networks with large-batch training. While recent methods greatly have reduced the gap between supervised and unsupervised performance of deep models such as ResNet-50, this development has been relatively limited for small models. In this work, we propose a novel unsupervised learning framework for small networks that combines deep self-supervised representation learning and knowledge distillation within one-phase training. In particular, a teacher model is trained to produce consistent cluster assignments between different views of the same image. Simultaneously, a student model is encouraged to mimic the prediction of on-the-fly self-supervised teacher. For effective knowledge transfer, we adopt an idea of domain classifier so that student training is guided by discriminative features invariant to representational space shift between the teacher and student. We also introduce a network driven multi-view generation paradigm to capture useful feature information contained in the network itself. Extensive experiments show that our student models surpass state-of-the-art offline distilled networks even from stronger self-supervised teachers as well as top-performing self-supervised models. Notably, our ResNet-18, trained with ResNet-50 teacher, achieves 68.3% ImageNet Top-1 accuracy on *frozen* feature linear evaluation, which is only 1.5% below the supervised baseline.

1 Introduction

Recently, there has been a growing attention in unsupervised and self-supervised learning where the goal is to effectively learn useful representation from a large amount of unlabeled data. Current self-supervised visual representation learning methods appear to approach and possibly even outperform the fully-supervised counterpart [8, 25, 29]. All top-performing self-supervised learning (SSL) algorithms for visual representation involve training big models on powerful computing resources. In particular, their smallest architecture is ResNet-50 [26], and the networks are trained with large batches (e.g., 4096 images) on multiple specialized hardware devices such as 128 TPU cores [6, 8, 25]. Yet, this heavy implementation is not a viable option in a resource-limited environment, and we empirically found that the SSL methods even do not work well on light models (Figure 1). Also, we need strong small networks that can operate on a system on a chip (SoC) for real-world applications.

Existing methods on SSL have demonstrated that deeper models learn general visual representation more effectively with unlabeled data [8, 39]. Moreover, it has been empirically shown that predictive performance of bigger networks transferred better into smaller ones [9]. Inspired by these, we are interested in pretraining light-weight models by transfering representational knowledge from a deep self-supervised network instead of directly conducting self-supervised training on small ones.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.



Method	Pretrained Epochs			
	100	200	400	800
Supervised	69.8			
SimCLR [8]	47.1	49.9	51.8	53.4
MoCo-v2 [11]	48.6	49.9	51.9	56.1
BYOL [25]	44.2	47.5	46.8	47.1
BYOL-GA [25]	54.2	56.9	61.4	61.9
SwAV [6]	57.7	61.2	63.7	64.9
OSS (ours)	**60.0**	**64.1**	**65.8**	**68.3**

Figure 1: ImageNet Top-1 accuracy of linear probe on ResNet-18 representations obtained from various self-supervised methods with 256-batch training. BYOL-GA denotes the implementation of BYOL with gradient accumulation (to 4096 batches). Green dotted line indicates supervised ResNet-18. Our method (red star) surpasses state-of-the-art self-supervised methods.

Most of previous distillation methods in unsupervised representation learning literature are offline in the sense that they leverage an already trained self-supervised teacher to transfer feature information into students [22, 31, 39, 56]. Often, their sequential training pipelines require extra pre/post-processing in order to boost performance [22, 56]. Unlike these approaches, we propose a novel unsupervised representation learning framework for small networks that combines deep self-supervised training and knowledge distillation [30] within one-phase training. To be specific, our teacher learns clusters and representation, and at the same time the student is encouraged to align its prediction to the clustering of **O**n-the-fly **S**elf-**S**upervised teacher. From this, we refer to our method as OSS. The main advantage over offline distillation is that our teacher better distills proper signals into the student at each training stage by solving the same task online. In distillation process, it is difficult for a low-capacity student to perfectly mimic a large teacher model accurately [36]. To deal with this, we incorporate an idea of domain classifier [23]. That is, we train a *feature classifier* not to be able to distinguish between the teacher and student embeddings, and this encourages the emergence of representational-space-invariant features in the course of joint training.

It is shown that increasing the number of different views during SSL improves the quality of resulting features [6, 8, 54]. Existing approaches to multi-view generation rely on random image transformation techniques. Different from these methods, we introduce a new network-driven paradigm in order to utilize rich feature information contained in the network itself. In particular, we apply random dropout and empirically show that it apparently improves label efficiency (Section 5).

Our method is conceptually simple but surprisingly works well even with typical 256-batch training on a single 8-GPU machine. Extensive experiments show that our student models outperform state-of-the-art self-supervised models (Figure. 1) as well as students from top-performing distillation approaches even with stronger self-supervised teachers (Tables 1 and 5) on linear evaluation. We also validate our approach across different vision tasks (Tables 2 and 3) and network architectures (Table 4 and Figure 3) on various benchmark datasets and demonstrate significant performance gains. Overall, we make the following contributions.

- We propose a novel unsupervised representation learning framework for small networks that simultaneously conducts self-supervised teacher training and knowledge distillation.
- We introduce a network-driven view generation paradigm and adversarial distillation scheme to better capture representational information from network. This is the first work to show their effectiveness in the context of unsupervised representation learning.
- The proposed method significantly improves unsupervised learning performance of small models, and its validity is shown under multiple settings of network architectures, datasets and vision tasks.

2 Related Work

Unsupervised visual representation learning. In unsupervised or self-supervised visual representation learning, we learn rich feature information by solving various pretext tasks whose labels

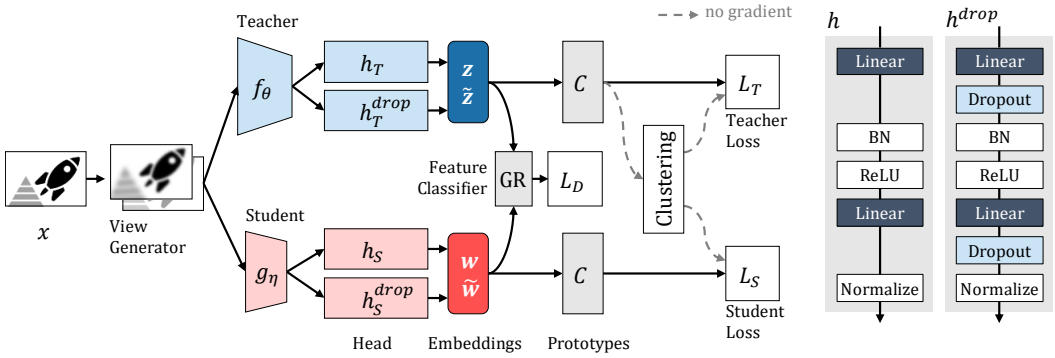


Figure 2: **Overview of proposed architecture.** OSS framework takes as input two randomly augmented views x_1 and x_2, and they are processed by teacher's and student's backbones f_θ and g_η. The processed outputs are then projected to lower dimensional embeddings z and w via teacher's and student's projection heads h_T and h_S. We add dropout to each linear layer in both projection heads (h^{drop}'s) for generating additional views of embeddings $\tilde{z}$ and $\tilde{w}$. We note that h and h^{drop} share weights. Network training is guided by cluster prediction mechanism, and the target cluster assignments are obtained by online clustering using the mapping of teacher's feature embeddings z to prototype vectors C. To distill useful discriminative feature information invariant to representational space shift between the teacher and student, we use a feature classifier connected to the last layer of each projection head with a gradient reversal layer (GR).

can be obtained from image data itself. These methods can be divided into several categories by types of pretext tasks used therein: construction-based methods [42, 61], prediction-based methods [20, 24, 25, 37, 38], cluster-based methods [3, 5, 6], and contrastive methods [8, 10, 11, 29]. Our work is mostly related to a clustering-based method of [6].

Multi-view generation. Most contrastive and clustering-based methods exploit random image transformation techniques to generate different views of each image. It is because increasing the number of views during training improves the performance of the resulting self-supervised model [6, 8, 54]. Our method uses dropout [49] for obtaining more views to enable diverse mining of feature information as well as applying existing data augmentation based strategies.

Knowledge distillation from self-supervised models. Most of knowledge distillation [30] techniques are proposed in the context of supervised learning, where a shallow network is trained via supervision signals from both labels and a deep network [1, 40, 43, 46, 51, 52, 57]. In unsupervised learning settings, recent work of [39] and [56] exploits clustering of teacher's embeddings as distillation signals in their sequential training pipeline, and CompRess [31] trains the student network by enforcing it to mimic the relative similarity between the datapoints in the teacher's embedding space. Also, SEED [22] transfers instance similarity distribution over a stored contrastive instance queue into students. Unlike aforementioned methods that rely on already trained static teachers, our framework injects distillation step to clustering-based self-supervised teacher learning. Only other online technique is concurrent work of [4], which combines contrastive and collaborative learning.

Domain adversarial learning. DANN [23] focuses on learning features that are discriminative for the main classification task and indiscriminative with respect to domain shift. This is achieved by minimizing the main classifier loss while adversarially maximizing a domain classifier loss via a gradient reversal layer that reverses a gradient. We adopt this classifier in our framework to transfer discriminative features invariant to representational space change between the teacher and student. Similar idea with a generative adversarial network [35] is also applied in the context of knowledge distillation in [12].

3 Proposed Framework

Our goal is to learn visual features of small networks effectively without access to any ground-truth labels. In order to do this, we embed distillation step into the process of on-the-fly self-supervised teacher learning. Inspired by [56], which empirically shows clustering improves generalization power

of representation, our online framework is developed upon [6], the state-of-the-art clustering-based self-supervised method. We also incorporate an idea of domain classifier to facilitate learning of useful features invariant to representational space change between the teacher and student. In a nutshell, the overall training objective function of our network is given by

$$L = L_T + L_S + L_D, \tag{1}$$

where L_T, L_S, and L_D are designed for teacher network training (Sections 3.1 and 3.2), knowledge distillation for student network (Sections 3.1 and 3.2), and domain classifier (Section 3.3), respectively. Overall architecture is presented in Figure 2, and a pseudo code for training is available in A.1.

3.1 Teacher training and knowledge distillation

We now outline our teacher network (e.g. ResNet-50) training and knowledge distillation framework. Our architecture takes as input two randomly augmented views x_{n1} and x_{n2} from an image x_n. The two views are processed by teacher's backbone network f_θ and a projection head h_T that consists in multi-layer perceptron (MLP). The processed backbone features are then projected to the unit sphere. We denote teacher's backbone features as $f_{n1} = f_\theta(x_{n1})$ and $f_{n2} = f_\theta(x_{n2})$, and the projected outputs as $z_{n1} = \frac{h_T(f_{n1})}{\|h_T(f_{n1})\|_2}$ and $z_{n2} = \frac{h_T(f_{n2})}{\|h_T(f_{n2})\|_2}$. Also, the two images are feed-forwarded to student's backbone network g_η (e.g., ResNet-18). The backbone features are projected via MLP head h_S and then normalized as well. Similarly, we denote student's backbone features as $g_{n1} = g_\eta(x_{n1})$ and $g_{n2} = g_\eta(x_{n2})$, and their projected output vectors as $w_{n1} = \frac{h_S(g_{n1})}{\|h_S(g_{n1})\|_2}$ and $w_{n2} = \frac{h_S(g_{n2})}{\|h_S(g_{n2})\|_2}$.

We randomly initialize prototype vectors $C = \{c_1, \dots, c_K\}$, where K is the number of clusters. Code vectors, q_n's, for cluster assignments are computed from mapping teacher's embedding representation z_n's to the trainable prototype vectors C and then solving an optimization problem for online clustering. We then solve a swapped prediction problem [6] for the teacher and simultaneously distill teacher's knowledge into the student by enforcing a consensus in prediction.

Teacher training. A training objective for the teacher network is to solve a swapped cluster prediction problem [6] of predicting the code q_{n1} from the feature z_{n2} and q_{n2} from z_{n1}:

$$L_t(z_1, z_2) = \sum_n \left[L_c(z_{n1}, q_{n2}) + L_c(z_{n2}, q_{n1}) \right]. \tag{2}$$

Here, L_c is the cross entropy loss between the codes and the softmax probabilities (with a temperature parameter τ) of dot products of z_{ni} and all prototypes in C:

$$L_c(z_{n1}, q_{n2}) = -\sum_k q_{n2}^{(k)} \log p_{n1}^{(k)} \quad \text{where} \quad p_{n1}^{(k)} = \frac{\exp\left(\frac{1}{\tau} z_{n1}^\top c_k\right)}{\sum_{k'} \exp\left(\frac{1}{\tau} z_{n1}^\top c_{k'}\right)}. \tag{3}$$

The loss function (2) is jointly minimized with respect to the prototypes C and the parameters θ of the network implicitly contained in p_n's. The codes q_n's are computed using Sinkhorn-Knopp algorithm [18] under an equi-partition constraint for each augmented mini-batch with stored embeddings as done in [6].

Knowledge distillation. At the same time, the student is trained to follow teacher's code assignment. The training loss has exactly the same form of (2) with the features from the student and target from teacher's codes:

$$L_s(w_1, w_2) = \sum_n \left[L_c(w_{n1}, q_{n2}) + L_c(w_{n2}, q_{n1}) \right], \tag{4}$$

where the cross-entropy loss L_c is similarly calculated using student's embedding vectors w_n's and the prototype vectors C. Again, minimization of this loss is performed with respect to the prototypes C and the parameters η of the student backbone g_η contained in w_n's.

3.2 Network driven multi-view generation

In addition to applying multi-crop strategy [6], we propose a network driven multi-view generation method to utilize diverse feature information contained in the network. In particular, we add a dropout

layer after each linear layer in the projection heads to generate more feature embeddings. Additional loss terms from small-crop features $\{\hat{z}_{nv}\}_{v=1}^{V}$ and dropout features $\{\tilde{z}_{nv}\}_{v=1}^{V+2}$ are given by

$$L_{\mathrm{mv}}(\tilde{z}, \hat{z}) = \sum_{n} \sum_{i \in \{1,2\}} \sum_{v} \left[L_c(\tilde{z}_{nv}, q_{ni}) + L_c(\hat{z}_{nv}, q_{ni}) \right], \tag{5}$$

where we note that codes q_n's are computed from non-dropout full-crop features. To summarize, the teacher loss L_T in the training objective (1) is the sum of (2) and (5):

$$L_T(z_1, z_2, \tilde{z}, \hat{z}) = L_t(z_1, z_2) + L_{\mathrm{mv}}(\tilde{z}, \hat{z}), \tag{6}$$

and the corresponding loss L_S for the student in (1) is similarly computed by the sum of (4) and $L_{\mathrm{mv}}(\tilde{w}, \hat{w})$ obtained from student's small-crop and dropout generated features $(\tilde{w}, \hat{w})$ in place of teacher's $(\tilde{z}, \hat{z})$.

3.3 Feature classifier

In order to transfer good features that are discriminative for the main clustering task and indiscriminative with respect to representational space shift between the teacher and student, we adopt an idea of domain classifier [23], and we call it *feature classifier*. The feature classifier is adversarially trained not to be able to identify the teacher and student embeddings via a gradient reversal layer. To be specific, the feature classifier D consists in MLP with output size of 2. We compute the cross entropy loss between the feature label (0 for teacher and 1 for student) and the softmax probability of the feature classifier outputs given by

$$L_D(z, w) = -\sum_{n} \sum_{i \in \{1,2\}} \left(\log d_{T,ni}^{(0)} + \log d_{S,ni}^{(1)} \right) \quad \text{where} \quad d_{T,ni}^{(k)} = \frac{\exp\left(D^{(k)}(z_{ni})\right)}{\sum_{k' \in \{0,1\}} \exp\left(D^{(k')}(z_{ni})\right)} \tag{7}$$

and $d_{S,ni}^{(k)}$ is similarly computed as $d_{T,ni}^{(k)}$ with w_{ni} in place of z_{ni}. For both teacher and student, the feature classifier is connected to the last layer of the MLP projection head (h_T and h_S) via a gradient reversal layer that scales the gradient by a certain negative constant α during backpropagation training. Thus, minimizing L_D reduces discriminative power of the feature classifier.

4 Experiments

We perform extensive experiments across different combinations of teacher-student architecture pairs and various tasks and datasets to evaluate the performance of our student network. In this section, we focus on evaluating our ResNet-18 features trained with ResNet-50 teacher and a batch size of 256 for 800 epochs on ImageNet-1K [19] training set (see A.1 for details). For self-supervised ResNet-18 baselines, we use top-performing SSL models of SimCLR [8], MoCo-v2 [11], BYOL [25], and SwAV [6] that are trained with 256 batches for 800 epochs in `pytorch` [41] unless explicitly specified (see A.2). By default, the supervised performance comes from official `torchvision` models.

Table 1: Top-1 accuracy (%) of linear probe on ImageNet for ResNet-18 distilled from SwAV's ResNet-50. * indicates the final accuracy of online teacher.

Distillation Method	Teacher Top-1	Student Top-1
Supervised	-	69.8
CC [39]	75.6	60.8
CRD [51]	75.6	58.2
SEED [22]	75.3	62.6
CompRess-2q [31]	75.6	62.4
CompRess-1q [31]	75.6	65.6
OSS (ours)	73.0*	**68.3**

Table 2: Top-1 accuracy (%) of ResNet-18 fine-tuned on small subsets of ImageNet. Our method outperforms both supervised and all considered unsupervised methods.

Method	Top-1 Accuracy	
	1%	10%
Scratch	14.9	50.4
SimCLR [8]	28.8	54.2
MoCo-v2 [11]	25.2	54.1
BYOL [25]	25.6	52.3
SwAV [6]	39.7	60.4
CompRess-1q [31]	37.8	59.0
OSS (ours)	**47.8**	**63.6**

Table 3: Transfer performance (%) of ResNet-18 finetuned on small-scale datasets for varying tasks. Evaluation measure is AP_{50} (average precision with intersection over union threshold of 50%) for detection and instance segmentation and mIoU (mean intersection over union metric) for semantic segmentation. Overall, our model works best across all transfer tasks and datasets.

Method	$\mathrm{AP}^{\mathrm{bb}}_{50}$			$\mathrm{AP}^{\mathrm{inst}}_{50}$		mIoU
	VOC07	COCO-1K	COCO-10K	COCO-1K	COCO-10K	VOC12
Supervised	66.9	17.0	33.6	15.6	31.1	42.2
SimCLR [8]	66.1 (-0.8)	16.0 (-1.0)	33.0 (-0.6)	14.8 (-0.8)	30.6 (-0.5)	40.6 (-1.6)
MoCo-v2 [11]	65.4 (-1.5)	11.0 (-6.0)	28.8 (-4.8)	9.9 (-5.7)	26.9 (-4.2)	41.3 (-0.9)
BYOL [25]	64.9 (-2.0)	14.3 (-2.7)	30.4 (-3.2)	12.9 (-2.7)	28.0 (-3.1)	39.9 (-2.3)
SwAV [6]	68.1 (+1.2)	18.6 (+1.6)	36.8 (+3.2)	17.2 (+1.6)	34.4 (+3.3)	42.1 (-0.1)
CompRess-1q [31]	66.4 (-0.5)	14.8 (-2.2)	31.7 (-1.9)	13.5 (-2.1)	29.4 (-1.7)	42.0 (-0.2)
OSS (ours)	**68.3** (+1.4)	**20.0** (+3.0)	**37.9** (+4.3)	**18.4** (+2.8)	**35.4** (+4.3)	**43.2** (+1.0)

4.1 Linear evaluation on ImageNet

We validate our method by linear evaluation of *frozen* features, following the protocol [6] described in A.3. We report 1-crop, Top-1 classification accuracy of ResNet-18 on ImageNet validation set. Figure 1 shows that our method significantly surpasses all the state-of-the-art SSL methods. Note that OSS requires shorter training epochs than the SSL algorithms to promise better performance. Indeed, our 200-epoch model (64.1%) is almost on par with 800-epoch SwAV's (64.9%). Moreover, our 800-epoch model achieves 68.3%, which is only 1.5% below the supervised model accuracy. Worse performance of SSL approaches here seems to be due to incapability of smaller networks to perform instance discrimination in a large-scale dataset [22]. A clustering-based SwAV [6] works better than contrastive methods of MoCo-v2 [11] and SimCLR [8]. We conjecture it is because clustering task is relatively easier for light networks. BYOL-GA indicates the implementation of BYOL [25] with gradient accumulation (GA) to 4096 batches (see Appendix G2 of [25]). For fair comparison, we will use its non-GA version (BYOL) throughout the following experiments as all other methods are trained using 256 batches without GA.

In Table 1, we also compare our approach with various offline distillation methods using SwAV's ResNet-50 teachers obtained in [31]. We note that all the offline compression models from stronger SwAV's ResNet-50 teacher do not still match our student's performance. A plausible reason for this is that large models easily overfit to the training data, so the static teachers provide less extra knowledge beyond hard annotations [2]. Summary of the offline techniques along with a extensive list of their performance is found in B.1. We also test general representational power of the pretrained frozen encoders on other classification datasets in B.2.

4.2 Data efficiency evaluation on ImageNet

We evaluate the performance on ImageNet classification task when finetuning our 800-epoch pretrained ResNet-18 model with small subsets of trainset using labels. We follow semi-supervised protocol [6] detailed in A.4, and use the 1% and 10% splits of [8]. We report Top-1 accuracy on ImageNet validation set in Table 2. Our method outperforms self-supervised approaches across all label fractions. We note that the gap in 1% data regime is larger than the gap in 10% counterpart. We also carry out the same data efficiency evaluation of ResNet-18 features obtained from the state-of-the-art offline knowledge distillation method of CompRess-1q [31] with SwAV [6]'s ResNet-50 teacher. We use authors' released model and find that our student significantly outperforms it by $+10.0\%$ in 1% and $+4.6\%$ in 10% settings. In fact, our network driven multi-view generation significantly improves label efficiency, and we will investigate this in ablations. Also, another strong low-shot learning performance on Places365 [62] dataset is available in B.3.

4.3 Transfer to other vision tasks

We now test transferability of ImageNet pretrained representation to different vision tasks. In order to better pin-point performance of each pretraining method, we chose small-scale datasets such as PASCAL VOC [21] and some subsets of COCO [33]. It is because a larger amount of data with longer finetuning schedule reduces the importance of the initial model. For COCO data, we randomly

Table 4: Top-1 accuracy (%) of linear evaluation on ImageNet for various shallow models. The second column (Params) is the number of parameters in each backbone network, and the unit M denotes 10^6. Params and supervised performance are obtained in the corresponding papers.

Model	Params	Method	Top-1
ResNet-18 [26]	11.5M	Supervised	69.8
		SwAV [6]	61.2
		OSS (ours)	64.1
RegNetY-600MF [44]	6.1M	Supervised	75.5
		SwAV [6]	67.0
		OSS (ours)	67.4
EffcientNet-B0 [50]	5.3M	Supervised	77.1
		SwAV [6]	59.3
		OSS (ours)	64.1
MobileNet-v2 [47]	3.4M	Supervised	72.0
		SwAV [6]	63.2
		OSS (ours)	66.1

Table 5: Top-1 accuracy (%) of linear probe on ImageNet for MobileNet-v2 distilled from MoCo-v2's ResNet-50 teacher. * indicates the final accuracy of online teacher. Our method outperforms all considered offline approaches.

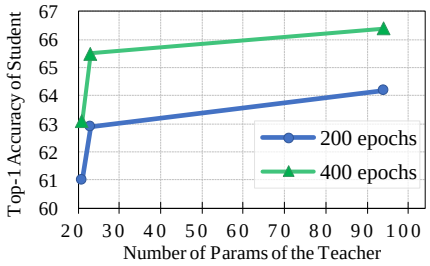
Distillation Method	Teacher Top-1	Student Top-1
Supervised	-	72.0
CC [39]	70.8	59.2
CRD [51]	70.8	54.1
Reg [59]	70.8	48.0
Reg-BN [31]	70.8	62.3
CompRess-1q [31]	70.8	63.0
CompRess-2q [31]	70.8	65.8
OSS (ours)	70.4*	**66.1**

sample five non-overlapping 1K and 10K train splits as in [28], and the results are averaged over the splits. We then transfer 800-epoch trained ResNet-18 backbones to object detection on VOC07 with Faster-RCNN [45], semantic segmentation on VOC12 with FCN [34], and object detection and instance segmentation on COCO splits with Mask R-CNN [27] (see A.5, A.6, and A.7 for details). Table 3 shows that our student produces better results than top-performing self-supervised and offline distilled models across all network architectures, datasets, and tasks. Notably, OSS is the only method that outperforms the supervised baselines on COCO-1K and VOC12. More downstream task performance is illustrated in B.4.

5 Ablation Studies

5.1 Various teacher-student combinations

Smaller students. We validate our framework with various teacher-student combinations. Specifically, we consider a ResNet-50 teacher and three small students: MobileNet-v2 [47], RegNetY-600MF [44], and EfficientNet-B0 [50]. We pretrain the three pairs for 200 epochs on ImageNet [19] trainset and conduct linear probe on the validation set (see A.1 and A.3 for details). Table 4 shows that our method outperforms SwAV [6] for all student architectures. We see OSS is not equally effective over SwAV across different networks. One possible reason is because we use SwAV's hyperparameters, this may place SwAV at an *advantage*. We also compare our approach with other offline distillation methods using MobileNet-v2 provided in [31]. In Table 5, our MobileNet-v2 features obtain better results than those compressed from MoCo-v2 [11]'s ResNet-50 teacher even if our teacher's accuracy is lower during training. More comparison for EfficientNet-B0 encoder is found in B.1.



Teacher	Pretrain Epochs	Teacher Params	Top-1 Teacher	Top-1 Student
ResNet-34	200	21.5M	62.1	61.0
ResNet-50		23.9M	70.8	62.9
ResNet-50w2		94.0M	75.1	**64.2**
ResNet-34	400	21.5M	64.6	63.1
ResNet-50		23.9M	72.0	65.5
ResNet-50w2		94.0M	75.5	**66.4**

Figure 3: ImageNet Top-1 accuracy (%) of linear evaluation of ResNet-18 distilled from various teachers. The third column (Params) is the number of parameters in the teacher network, and the unit M is 10^6. Our student benefits more from larger teacher and longer training schedule.

Table 6: Impact of proposed dropout and distillation schemes on the classification accuracy (%) of ResNet-18 features on ImageNet. Dropout strategy combined with feature classifier improves label efficiency, and online distillation works better than offline version.

Method Name	Scheme Name	Dropout Applied	Feature Classifier	Top-1 Accuracy Frozen	Top-1 Accuracy 1%	Top-1 Accuracy 10%	Top-5 Accuracy Frozen	Top-5 Accuracy 1%	Top-5 Accuracy 10%
Ours	off.KD-Drop	√		66.0	33.9	61.5	86.9	59.4	84.2
	KD			68.2	42.5	63.2	88.2	69.0	85.4
	KD-Drop	√		**68.4**	47.2	63.3	**88.5**	73.3	85.6
	KD-Drop-Adv	√	√	68.3	**47.8**	**63.6**	88.4	**73.8**	**85.8**
SwAV [6]	Original			**64.9**	39.7	**60.4**	**86.3**	66.3	83.6
	Drop	√		64.2	**43.7**	60.3	85.9	**70.1**	**83.7**

Larger teachers. We perform an analysis with ResNet-18 student distilled from three different resnet teachers: ResNet-34, ResNet-50, and ResNet-50w2. ResNet-50w2 is a 50-layer resnet with $2\times$ wider channels. Here, we only consider resource-friendly pairs that are trainable on a single 8-GPU machine, so we do not explore beyond ResNet-50w2. We pretrain them using our framework on ImageNet trainset for 800 epochs and carry out a linear probe on the validation set at 200 and 400 epochs. Figure 3 shows our method benefits more from bigger teachers and longer train iterations. As in supervised knowledge distillation, student's performance is not necessarily getting better linearly when teacher's size becomes larger [36].

5.2 Framework design

Questions and experiment settings. A natural question that arises is how effective is "offline version of the proposed method" where a student is learning on SwAV's framework with dropout while the target code comes from SwAV's pretrained (frozen) teacher, not updating the prototypes. Also, one might be curious about performance gain from each proposed technique: network-driven multi-view generation and feature classifier. To answer these, we train ResNet-50 (teacher) and ResNet-18 (student) pair under the 4 settings in Table 6. The distillation epoch is 130 for the offline scheme (off.KD-Drop) from SwAV's 800-epoch teacher following [31] and 800 for the other online versions. We carry out linear probe and semi-supervised evaluation (see A.3 and A.4). For dropout experiment, we apply dropout at the two linear layers in the projection heads (see Figure 2) and compare it with non-dropout scheme. The dropout rates are 0.1 at Linear1 and 0.05 at Linear2.

Student's performance analysis. From Table 6, we observe that online schemes produce better results than offline version (off.KD-Drop). Our intuition is that online teachers better figure out and distill appropriate information in the course of joint training by solving the same task online. Yet, our offline method even works better than all offline techniques in Table 1 in linear evaluation, and this verifies effectiveness of our overall framework. On the other hand, strong performance of our method does not come free from (online) knowledge distillation. We argue that the proposed network driven strategy is effective by enabling diverse mining of rich knowledge contained in the network. Indeed, applying random dropout operation significantly improves data efficiency as demonstrated in 1%-label evaluation. We also see that our feature classifier generally introduces additional gain in label efficiency. From this empirical analysis, we find that the dropout strategy, combined with the feature classifier, benefits more from lower data regime.

Table 7: Classification accuracy (%) of ResNet-50 finetuned on small subsets of ImageNet. Results of OSS come from training with ResNet-18 student. The proposed framework produces data efficient teacher as well.

Method	Top-1 Accuracy 1%	Top-1 Accuracy 10%	Top-5 Accuracy 1%	Top-5 Accuracy 10%
Supervised	25.4	56.4	48.4	80.4
SimCLR	48.3	65.6	75.5	87.8
BYOL	53.2	68.8	78.4	89.0
SwAV	53.9	70.2	78.5	89.9
OSS (ours)	**57.5**	**71.1**	**81.4**	**90.5**

Table 8: Effects of dropout rates on low-shot learning classification accuracy on ImageNet for ResNet-18 student. Strong dropout improves data efficiency on 1% setup while performance is less sensitive at 10% setup.

Dropout rate Linear1	Dropout rate Linear2	Top-1 Accuracy 1%	Top-1 Accuracy 10%
0.00	0.05	37.5	60.0
0.10	0.05	43.5	59.6
0.10	0.10	43.7	59.5
0.20	0.05	44.0	59.1
0.20	0.10	44.0	59.4

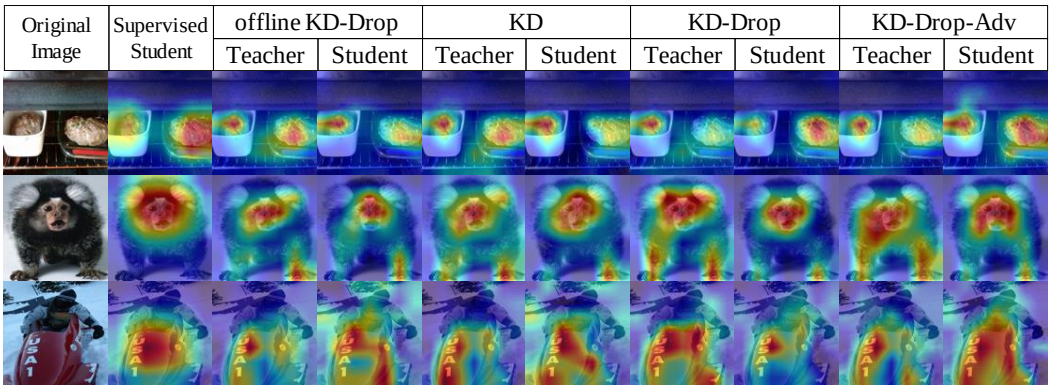


Figure 4: Grad-CAM [48] visualization of feature maps of different dropout and distillation schemes using some images in ImageNet validation set. Proposed method (KD-Drop-Adv) shows strongest activation on the target class region (Meatloaf, Marmoset and Bobsled) among all schemes.

Teacher's performance analysis. It is natural to ask whether the proposed network-driven view generation with feature classifier produces more data efficient teacher as the same setup is applied to the teacher model as well. We answer this question by performing the label efficiency evaluation of the teacher on ImageNet. We compare the results with other SSL ResNet-50's in Table 7. Our teacher outperforms the state-of-the-art self-supervised models on both 1% and 10% regimes. We note that the training objective of our ResNet-50 teacher is exactly the same as SwAV's ResNet-50 model except for dropout and feature classifier. This verifies that the performance gain over SwAV (+3.6% on 1% set and +0.9% on 10% set in Top-1) is sorely contributed by our proposed strategy.

Dropout rates. In order to provide more intuition on our network-driven generation paradigm, we run experiments over multiple choices of dropout rates in the proposed OSS framework (KD-Drop-Adv). We do not explore aggressive dropout rates because self-supervised training signal is already noisy so we might lose important feature information with higher rates. Table 8 summarizes low-shot classification results on ImageNet. We find strong dropout improves label efficiency at 1% regime. Yet, accuracy is less sensitive when relatively larger amount of data is available.

Qualitative analysis of feature distributions. We visualize the feature maps of some images in ImageNet validation set using Grad-CAM [48] in Figure 4. Grad-CAM identifies regions where the network focuses on. We show the results of teacher and student networks trained by our framework with different schemes in Table 6. As in Figure 4, our proposed method (KD-Drop-Adv) shows stronger activation on the target class region than other schemes. While the feature classifier seems to have little effect on the performance quantitatively over KD-Drop scheme, the feature map illustrates a clear improvement of the coverage of useful object region. Different from other setups, proposed student has high activation on both chunks of meat and wider part of the bobsled and monkey. From this, we see training with the feature classifier helps learning the true data distribution better. More Grad-CAM examples are available in B.6.

5.3 Dropout applied to SwAV

We also report the impact of applying our dropout strategy on the performance of SwAV [6] features. We train a dropout added version of SwAV for 800 epochs using a ResNet-18 model. We apply the same dropout scheme to the projection head and compare it with the original model. As shown in Table 6, dropout improves its label efficiency as well.

6 Conclusion

We introduce a unified unsupervised learning framework for small models that combines deep self-supervised learning with knowledge distillation. Our student outperforms state-of-the-art self-supervised shallow models as well as the networks obtained from offline compression techniques. We hope our work will attract the community's attention to the development of unsupervised learning methods for light-weight networks.

References

[1] Sungsoo Ahn, Shell Xu Hu, Andreas Damianou, Neil D. Lawrence, and Zhenwen Dai. Variational information distillation for knowledge transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[2] Rohan Anil, Gabriel Pereyra, Alexandre Passos, Róbert Ormándi, George E. Dahl, and Geoffrey E. Hinton. Large scale distributed neural network training through online distillation. In *International Conference on Learning Representations (ICLR)*, 2018.

[3] Yuki M. Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. In *International Conference on Learning Representations (ICLR)*, 2020.

[4] Prashant Bhat, Elahe Arani, and Bahram Zonooz. Distill on the Go: Online knowledge distillation in self-supervised learning, 2021. arXiv:2104.09866.

[5] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[6] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[7] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark, 2019. arXiv:1906.07155.

[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, pages 1597–1607, 2020.

[9] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[10] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning, 2020. arXiv:2011.10566.

[11] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning, 2020. arXiv: 2003.04297.

[12] Inseop Chung, Seonguk Park, Jangho Kim, and Nojun Kwak. Feature-map-level online adversarial knowledge distillation. In *Proceedings of the 37th International Conference on Machine Learning*, pages 2006–2015, 2020.

[13] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi. Describing textures in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3606–3613, 2014.

[14] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 215–223, 2011. URL `http://proceedings.mlr.press/v15/coates11a.html`.

[15] MMSegmentation Contributors. MMSegmentation, an open source semantic segmentation toolbox. `https://github.com/open-mmlab/mmsegmentation`, 2020.

[16] OpenSelfSup Contributors. OpenSelfSup, an open source representation learning toolbox. `https://github.com/open-mmlab/OpenSelfSup`, 2020.

[17] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3213–3223, 2016.

[18] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transportation distances. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2013.

[19] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009.

[20] Carl Doersch, Abhinav Gupta, and Alexei A. Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1422–1430, 2015.

[21] Mark Everingham, Luc Gool, Christopher K. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes VOC challenge. *International Journal of Computer Vision*, 88 (2):303–338, 2010.

[22] Zhiyuan Fang, Jianfeng Wang, Lijuan Wang, Lei Zhang, Yezhou Yang, and Zicheng Liu. SEED: Self-supervised distillation for visual representation. In *International Conference on Learning Representations (ICLR)*, 2021. URL `https://openreview.net/forum?id=AHm3dbp7D1D`.

[23] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(59):1–35, 2016.

[24] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations (ICLR)*, 2018.

[25] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[26] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[27] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017.

[28] Kaiming He, Ross Girshick, and Piotr Dollar. Rethinking Imagenet pre-training. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 4917–4926, 2019.

[29] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9729–9738, 2020.

[30] Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015.

[31] Soroush Abbasi Koohpayegani, Ajinkya Tejankar, and Hamed Pirsiavash. CompRess: Self-supervised learning by compressing representations. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[32] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[33] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 740–755, Cham, 2014.

[34] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015.

[35] Xudong Mao, Qing Li, Haoran Xie, Raymond Y.K. Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2794–2802, 2017.

[36] Seyed-Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant: Bridging the gap between student and teacher. In *AAAI Conference on Artificial Intelligence*, pages 5191–5198, 2020.

[37] Ishan Misra, C. Lawrence Zitnick, and Martial Hebert. Shuffle and learn: Unsupervised learning using temporal order verification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 527–544, 2016.

[38] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 69–84, 2016.

[39] Mehdi Noroozi, Ananth Vinjimoor, Paolo Favaro, and Hamed Pirsiavash. Boosting self-supervised learning via knowledge transfer. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9359–9367, 2018.

[40] Nikolaos Passalis and Anastasios Tefas. Learning deep representations with probabilistic knowledge transfer. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[41] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 8024–8035. 2019. URL `http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf`.

[42] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2536–2544, 2016.

[43] Baoyun Peng, Xiao Jin, Jiaheng Liu, Dongsheng Li, Yichao Wu, Yu Liu, Shunfeng Zhou, and Zhaoning Zhang. Correlation congruence for knowledge distillation. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2019.

[44] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[45] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, pages 91–99, 2015.

[46] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In *International Conference on Learning Representations (ICLR)*, 2015.

[47] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen. MobilenetV2: Inverted residuals and linear bottlenecks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4510–4520, 2018.

[48] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 618–626, 2017.

[49] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.

[50] Mingxing Tan and Quoc Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6105–6114. PMLR, 09–15 Jun 2019.

[51] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. In *International Conference on Learning Representations (ICLR)*, 2020.

[52] Frederick Tung and Greg Mori. Similarity-preserving knowledge distillation. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 1365–1374, 2019.

[53] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019.

[54] Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3742–3733, 2018.

[55] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. SUN database: Large-scale scene recognition from abbey to zoo. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3485–3492, 2010.

[56] Xueting Yan, Ishan Misra, Abhinav Gupta, Deepti Ghadiyaram, and Dhruv Mahajan. ClusterFit: Improving generalization of visual representations. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6509–6518, 2020.

[57] Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[58] Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks, 2017. arXiv:1708.03888.

[59] L. Yu, V. O. Yazici, X. Liu, J. van de Weijer, Y. Cheng, and A. Ramisa. Learning metrics from teachers: Compact networks for image embedding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2902–2911, 2019.

[60] Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. S4L: Self-supervised semi-supervised learning. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1476–1485, 2019.

[61] Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful image colorization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 649–666, 2016.

[62] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1452–1464, 2018.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes]
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See Section 1.
   (b) Did you describe the limitations of your work? [Yes] See Section 5 and Appendix.
   (c) Did you discuss any potential negative societal impacts of your work? [No] Potential negative societal impacts are not identified in the proposed method.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See Appendix.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] We provide implementation details in Appendix.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Appendix.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] All references are given.
   (b) Did you mention the license of the assets? [Yes] See Appendix.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] Pseudo code is in Appendix.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] Public datasets are used.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]