SimReg: Regression as a Simple Yet Effective Tool for Self-supervised Knowledge Distillation

K L Navaneet[1]
navanek1@umbc.edu
Soroush Abbasi Koohpayegani[1]
soroush@umbc.edu
Ajinkya Tejankar[1]
at6@umbc.edu
Hamed Pirsiavash[1, 2]
hpirsiav@ucdavis.edu

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

Abstract

Feature regression is a simple way to distill large neural network models to smaller ones. We show that with simple changes to the network architecture, regression can outperform more complex state-of-the-art approaches for knowledge distillation from self-supervised models. Surprisingly, the addition of a multi-layer perceptron head to the CNN backbone is beneficial even if used only during distillation and discarded in the downstream task. Deeper non-linear projections can thus be used to accurately mimic the teacher without changing inference architecture and time. Moreover, we utilize independent projection heads to simultaneously distill multiple teacher networks. We also find that using the same weakly augmented image as input for both teacher and student networks aids distillation. Experiments on ImageNet dataset demonstrate the efficacy of the proposed changes in various self-supervised distillation settings. Code is available at https://github.com/UCDvision/simreg

1 Introduction

There has been a tremendous improvement in deep learning methodologies and architectures in the last few years. While this has lead to significant improvements in performance on various computer vision tasks, it has also resulted in complex and deep networks that require high compute during inference [20, 25, 26, 52]. Various specialized architectures [24, 27, 46, 47] have been proposed to minimize the inference time and memory requirements of the model to be deployed. Knowledge distillation [1, 23] has been proposed as an effective technique to compress information from larger but effective models (teachers) to lighter ones (students).

With availability of large scale unlabeled datasets, self-supervised learning (SSL) has received great attention in recent times. Several SSL methods achieve close to supervised

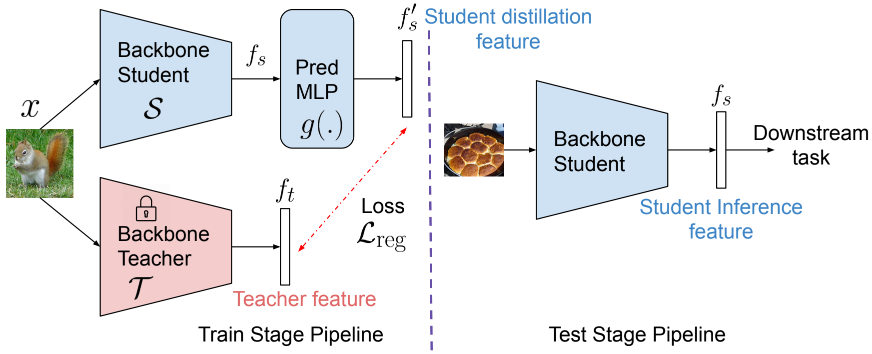

Figure 1: **Proposed distillation pipeline:** We propose a simple modification of using a MLP prediction module during distillation. The module is discarded during inference. Surprisingly, performance of backbone features f_s is better than those from MLP output, f'_s, though f'_s more closely matches the teacher. A deeper MLP helps improve distillation performance.

performance on the benchmark ImageNet object classification task [8, 19]. Unlike supervised models, the outputs of a self-supervised network are latent feature vectors and not class probabilities. An additional module is generally trained atop the pretrained SSL models using supervision to perform the downstream task. Conventional knowledge distillation methods proposed for supervised classification are thus not applicable for distillation from self-supervised networks. A simple way to handle this is to directly regress the teacher latent features. Recent works [16, 29] have proposed more complex solutions that try to capture the structure of the teacher latent space and are shown to outperform the regression baselines.

Use of a multi-layer perceptron (MLP) head atop CNN backbone model has been shown to help self-supervised models prevent overfitting to the SSL task and generalize better to downstream applications [10, 11, 12, 19]. Such modules are used only during SSL pre-training and are not part of inference network. In this work, we consider the task of distilling self-supervised models. We employ a similar *prediction head* atop the student backbone network to effectively mimic the teacher. As in SSL, the prediction module is discarded after distillation and thus, there is no change in the time and memory required during inference (refer Fig. 1). We empirically demonstrate that doing so does not hurt classification performance. Counter-intuitively, we observe that the features from the backbone network outperform those from the final layer of the prediction head though the final layer best matches the teacher. Unlike in SSL, overfitting to the training task (i.e, exactly mimicking the teacher) benefits distillation [5] and it is not clear why generalization could be better at intermediate layers where the similarity with teacher is reduced. Our finding suggests that we require a deeper analysis to understand how well the student models mimic the teacher in general and how knowledge distillation works. Crucially, it also enables us to use a deeper prediction head to achieve lower train and test error leading to better downstream performance without increasing the student capacity.

We empirically show that the above observation generalises to distillation with different teacher and student settings and to other self-supervised distillation technique. Our simple regression model with a MLP prediction head outperforms complex state-of-the-art approaches that require the use of memory banks and tuning of temperature parameter. Our work serves as an important benchmark for future self-supervised distillation works. The use of MLP heads also facilitates effective distillation from multiple SSL teacher networks.

Additionally, we demonstrate that using the same augmented image with weak augmentation for both student and teacher networks results in better student models. Since aggressive augmentation is necessary for effective self-supervised learning [10, 12] but hurts their ability to generalize [42], our approach could be used to learn better SSL models. To summarize, our contributions are simple changes to architecture and augmentation strategy of distillation networks that not only achieve state-of-the-art performance on SSL model distillation but also question our current understanding of knowledge distillation.

2 Related Works

Supervised knowledge distillation: Bucilua *et al.* [7] and Hinton *et al.* [23] pioneered the use of knowledge distillation for compressing information. The methods used the teacher prediction logits as soft-labels in addition to the supervised label to regularize the student model. [40] minimizes the divergence between the student and teacher probability distributions. Several works ([22, 28, 44]) utilize intermediate teacher outputs in distillation. FitNets [44] match both the final and intermediate teacher representations while [28] transfers knowledge from the attention maps of the teacher. RKD [38] transfers mutual relations instead of instance wise distillation. [50] proposes directly regressing the final teacher features with a modified loss function that strictly matches the direction of the features but allows flexibility in terms of feature magnitude.

Self-supervised representation learning (SSL): Earlier works on SSL ([8, 14, 18, 35, 36, 41]) learn effective representations by solving pretext tasks that do not require supervised labels. Recently, works based on contrastive learning ([4, 9, 10, 12, 21, 33, 48]) have gained focus. In contrastive learning, the distances between representations of positive pairs are minimized while those between negative pairs are maximized. The positive and negative pairs are generally constructed by utilizing multiple augmentations of each image. BYOL [19] is closer to knowledge distillation, where the distance between teacher and student representations are minimized. The inputs to the two networks must be different augmentations of the same image and the teacher network is obtained as a moving average of the student. Similar to our work, [19] employs MLP head atop the student network to predict the teacher features.

Distillation of self-supervised models: In [37], the student mimics the unsupervised cluster labels predicted by the teacher. CRD [49] maximizes a lower bound of the mutual information between the teacher and student networks. However, it additionally uses supervised loss for optimization. CompRess [29] and SEED [16] are specifically designed for compressing self-supervised models. In both these works, student mimics the relative distances of teacher over a set of anchor points. Thus, they require maintaining large memory banks of anchor features and tuning temperature parameters. As in regression, proposed prediction heads can also be used to improve CompRess and SEED.

3 Knowledge Distillation

We first consider the supervised model distillation formulation proposed in [23]. The teacher is trained on the task of object classification from images. Let X be the set of images, Y the set of corresponding class labels and c the total number of classes. Consider a teacher network $\mathcal{T}$ with $f_t = \mathcal{T}(x)$, $f_t \in \mathbb{R}^c$ as the output vector (logits) corresponding to input image x. The predicted class probabilities can be obtained by applying softmax operation $\sigma(.)$ atop the vector f_t.

$$\hat{y}_t = \sigma(f_t;\tau_t) = \frac{e^{f_t/\tau_t}}{\sum_i e^{f_t^i/\tau_t}} \tag{1}$$

where f_t^i is the i^{th} dimensional output of the feature vector and τ_t is the temperature parameter. The teacher network $\mathcal{T}$ is trained using the image-label pairs in a supervised fashion with standard cross-entropy loss. The trained teacher network is to be distilled to a student network. Once trained, the teacher network parameters are frozen during the distillation process. Let $\mathcal{S}$ be the student network, $f_s = \mathcal{S}(x)$, $f_s \in \mathbb{R}^c$ the feature vector corresponding to input image x and $\hat{y}_s = \sigma(f_s;\tau_s)$ the predicted class probability vector. Knowledge distillation loss is given by

$$\mathcal{L}_{\text{KD}}(\hat{y}_t,\hat{y}_s) = \sum_{j=1}^{c} \hat{y}_t^j \log(\hat{y}_s^j) \tag{2}$$

The student is trained using a combined objective function involving supervised cross entropy loss on student features $\mathcal{L}_{\text{CE}}$ and distillation loss $\mathcal{L}_{\text{KD}}$:

$$\mathcal{L} = \lambda\mathcal{L}_{\text{CE}} + (1-\lambda)\tau_s^2\mathcal{L}_{\text{KD}} \tag{3}$$

where λ is a hyperparameter that determines the relative importance of each loss term. Since τ_t is generally set to 1, KD loss is multiplied by a factor of τ_s^2 to match the scale of gradients from both loss terms.

3.1 Distillation of Self-supervised Models

The value of λ in Eq. 3 can be set to 0 if the class labels are not available during student distillation. However, the formulation cannot be directly employed to distill from self-supervised teacher networks since the teacher outputs are latent representations and not logits or class probability vectors. Thus, to distill from such teachers, we simply regress the final feature vector of the teacher. Let $f_t = \mathcal{T}(x)$, $f_t \in \mathbb{R}^d$ and $f_s = \mathcal{S}(x)$, $f_s \in \mathbb{R}^m$. Since it is not necessary for the student and teacher representation dimensions to be the same, we use a linear projection of the student feature to match the dimensions.

$$f_s' = W^T f_s + b;\, W \in \mathbb{R}^m \times \mathbb{R}^d,\, b \in \mathbb{R}^d \tag{4}$$

The distillation objective is then given by $\mathcal{L} = \mathcal{L}_{\text{reg}} = \text{d}(f_t, f_s')$ where $\text{d}(.)$ is a distance metric. Here, we consider squared Euclidean distance of l_2 normalized features as the metric.

3.2 Prediction Heads for Regression Based Distillation

For a more effective matching of the teacher latent space, we propose a non-linear prediction head $g(.)$ atop the student backbone network $\mathcal{S}$ in place of the linear projection in Eq. 4. During training, the student feature is then obtained as $f_s' = g(f_s)$ where $g(.)$ is modeled using a multi-layer perceptron (MLP). Each layer in $g(.)$ is given by a linear layer with bias followed by batch-normalization and a non-linear activation function (we use ReLU non-linearity in all our experiments). The number of such layers is a hyperparameter to be optimized. The dimension of the last layer output matches that of the teacher. There is no non-linearity in the final layer to prevent constraining the output space of the student network. During inference, the prediction head $g(.)$ is removed and the output of the student network is obtained as $f_s = \mathcal{S}(x)$ (refer Fig. 1). Thus, there is no change in the architecture or the number of parameters of the model to be deployed. In our experiments, we demonstrate that the use of such MLP heads plays a crucial role in improving downstream performance. Surprisingly, we also observe that preserving the prediction heads during inference is not necessarily beneficial and might result in reduction in performance.

3.3 Multi-teacher Distillation

The prediction heads are particularly beneficial in distillation from multiple teacher networks. Independent deep non-linear projections of the student backbone features can be employed during distillation to match each of the teachers. Let f_t^k be the output vector of the k^{th} teacher and $f_s^k = g^k(f_s)$ that of the corresponding student prediction head $g^k(.)$. The multi-teacher distillation objective for K teachers is given by:

$$\mathcal{L} = \frac{1}{K}\sum_k \mathrm{d}(f_t^k, f_s^k) \tag{5}$$

The prediction heads are trained by the loss term from corresponding teachers while the backbone S is trained using the summation in Eq. 5.

4 Experiments

We consider distillation of pretrained self-supervised models. We consider four such methods for teacher networks - MoCo-v2 [12], BYOL [19], SwAV [8] and SimCLR [10]. We use the official publicly released models for all the teacher networks (details in suppl.). We also use a ResNet-50 model trained with supervised labels (provided by PyTorch in [3]) as a teacher. All teacher training and student distillation is performed on the train set of ImageNet. We consider different teacher and student backbone network architectures. For the prediction head, we experiment with linear, 2 and 4 layer MLPs. Let the dimension of the student backbone output be m and that of teacher d. Similar to the prediction head in [19], the MLP dimensions are $(m, 2m, m, 2m, d)$.

Implementation details: We use SGD optimizer with cosine scheduling of learning rate and momentum of 0.9. Initial learning rate is set to 0.05. As in [29] the networks are trained for 130 epochs with batch size of 256. Cached teacher features are utilized for faster distillation in experiments with SimCLR, BYOL and SwAV teachers. We publicly release the code[1].

Datasets: We primarily evaluate the performance of distilled networks on ImageNet [45] classification task. Additionally, for transfer performance evaluation, we consider the following datasets: Food101 [6], CIFAR10 [31], CIFAR100 [31], SUN397 [51], Cars [30], Aircraft [32], DTD [13], Pets [39], Caltech-101 [17] and Flowers [34]. We train a single linear layer atop the frozen backbone network for transfer evaluation (refer suppl.)

Metrics: We use k-nearest neighbour (k-NN) and linear evaluation on all tasks. We also report mean squared error (MSE) between the student and teacher features over the test set. For k-NN evaluation, k=1 and 20 are considered and cosine similarity is used to calculate NNs. We employ FAISS [1] GPU library to perform fast k-NN evaluation. For linear evaluation, a single linear layer is trained atop the features from the network to be evaluated. As in [29], the inputs to the linear layer are normalized to unit l_2 norm and then each dimension is shifted and scaled to have unit mean and zero variance. The layer is trained for 40 epochs using SGD with learning rate of 0.01 and momentum of 0.9.

[1]Code is available at https://github.com/UCDvision/simreg

Train and Inference Arch	1-NN	20-NN	Linear	MSE
MobileNet-v2+4L-MLP	**54.5**	**58.7**	**68.5**	**0.090**
MobileNet-v2+2L-MLP	54.0	58.0	67.9	0.097
MobileNet-v2+Linear	50.8	55.1	58.3	0.149

Table 1: **Role of MLP Heads.** We train three models with varying number of layers in prediction head and use the features from *final* MLP layer of each prediction model for evaluation. Deeper models more closely match the teacher (lower MSE) and achieve better classification performance (1 and 20 Nearest Neighbour and Linear evaluation).

Train	Backbone(BB)+4L-MLP								
Inference	Backbone(BB)			BB+2L-MLP			BB+4L-MLP		
Metric	1-NN	Linear	MSE	1-NN	Linear	MSE	1-NN	Linear	MSE
ResNet-18	55.3	65.7	-	**56.0**	**66.4**	1.99	53.4	65.2	**0.1**

Table 2: **Effect of MLP Heads on inference.** We train a single model with 4 layer MLP head and perform evaluation using features from different layers (pre-MLP, intermediate MLP layer and MLP output). Since the final layer outputs are trained to mimic the teacher, MSE with teacher features is lowest at 4L-MLP while that at 2L-MLP is extremely high (dimension of BB and teacher are different, hence MSE is not reported). However, features from backbone and intermediate layer (+2L-MLP) outperform those from final layer (+4L-MLP) on classification, contrary to the notion that features with lower MSE generalize better.

4.1 Baseline Approaches

Regression: In addition to proposed MLP prediction head based regression (termed *SimReg-MLP*), we consider two additional regression baseline methods proposed in [29] termed 'Regress' and 'Regress-BN'. While Regress distills from unnormalized teacher features, Regress-BN uses batch-norm layer atop the final student and teacher features during distillation. Unlike Regress-MLP, both these approaches use a linear prediction head.

CompRess: CompRess [29] is designed to distill specifically from self-supervised models. Given a set of anchor points, the student is encouraged to have the same similarities with the anchors as that of the teacher. The anchor point features can either be common features from a teacher memory bank (CompRess-1q) or features from individual memory banks for teacher and student (CompRess-2q). We additionally implement CompRess with our MLP prediction head, termed CompRess-1q-MLP and CompRess-2q-MLP. SEED [16] proposes similarity based distillation similar to [29] but uses pre-trained teacher models with significantly lower number of training epochs and performance. Further, it requires access to the projection heads used atop teacher networks used only during SSL training and not inference, These parameters are generally not publicly released, making the setting less replicable. Thus we provide comparisons with only CompRess [29].

Contrastive Representation Distillation (CRD): CRD [49] uses a contrastive formulation to bring corresponding teacher and student features closer while pushing apart those from unrelated pairs. While the paper considered a supervised setup and loss term utilizing labels, we use the formulation with just the contrastive loss as proposed in [29].

Cluster Classification (CC): In CC [37], the student predicts unsupervised labels obtained by clustering samples using teacher features. We report metrics for CC and CRD from [29].

Teacher Student	MoCo-v2 ResNet-50 MobileNet-v2		MoCo-v2 ResNet-50 ResNet-18		SimCLR ResNet-50x4 ResNet-50	
Method	1-NN	Linear	1-NN	Linear	1-NN	Linear
Teacher*	57.3	70.8	57.3	70.8	64.5	75.6
Supervised*	64.9	71.9	63.0	69.8	71.4	76.2
Regress*	38.6	48.0	41.7	52.2	-	-
Regress-BN*	48.7	62.3	47.3	58.2	-	-
CC [37]*	50.2	59.2	51.1	61.1	55.6	68.9
CRD [49]*	36.0	54.1	43.7	58.4	-	-
CompRess-2q [29]*	54.4	63.0	53.4	61.7	63.0	71.0
CompRess-1q [29]*	54.8	65.8	53.5	62.6	63.3	71.9
CompRess-2q-4L-MLP	**56.3**	67.4	54.4	64.0	**62.5**	73.5
CompRess-1q-4L-MLP	55.5	67.1	**54.9**	64.6	60.9	72.9
SimReg-4L-MLP	55.5	**69.1**	54.8	**65.1**	60.3	**74.2**

Table 3: **Comparison of SSL distillation methods on ImageNet classification.** Our regression method with MLP head (SimReg-4L-MLP) is comparable to or better than the complex state-of-the-art approaches, especially on the linear evaluation metric. We also observe that CompRess-1q and 2q are improved when MLP heads are utilized. Interestingly, regression gets a significantly higher boost compared to CompRess upon addition of MLP layers. Note that the MLPs are used only during training and the inference network architecture remains the same for all approaches making the comparison fair. * metrics from CompRess [29].

5 Results

Role of Prediction Head: A deeper prediction head results in a student with higher representational capacity and thus a model that better matches the teacher representations. Table 1 shows results for models with a common MobileNet-v2 [46] backbone and different prediction head architectures. The prediction head is used during both student training and evaluation. We observe that a deeper model has lower MSE with teacher features and better classification performance. However, a deeper model also implies greater inference time and memory requirements. The student architecture is fixed based on deployment needs and thus requirement of larger model goes against the very essence of distillation. To analyze performance at different layers of the prediction head, we train a single ResNet-18 [20] student with all intermediate dimensions of MLP equal to that of the output. Surprisingly, a model trained with MLP prediction head performs well on downstream task even when the prediction head is discarded during inference (Table 2). The performance using features from backbone network is slightly better than that from the final layer outputs whenever a MLP head is used (more results in suppl.). More importantly, this observation enables us to use deeper prediction heads for distillation in place of linear layers without any concerns about altering the student architecture or increasing inference time.

Comparison with existing approaches: In all the remaining experiments, we use SimReg-4L-MLP with the prediction head used only during distillation. We compare the proposed regression method with other baselines and self-supervised distillation methods in tables 3 and 4. Surprisingly, our simple regression performs comparably or even outperforms the state-of-the-approaches on all settings and metrics. On linear evaluation, we outperform previous methods (without MLP) by **3.3**, **2.5** and **2.3** points respectively on MobileNet-v2, ResNet-

Teacher	BYOL ResNet-50			SwAV ResNet-50		
Method	1-NN	20-NN	Linear	1-NN	20-NN	Linear
Teacher	62.8	66.8	74.3	60.7	64.8	75.6
CompRess-2q-4L-MLP	56.0	60.6	65.2	53.2	58.1	63.9
CompRess-1q-4L-MLP	55.4	60.0	65.2	52.4	57.1	63.4
SimReg-4L-MLP	**56.7**	**61.6**	**66.8**	**54.0**	**59.3**	**65.8**

Table 4: **ImageNet Evaluation with different teacher networks.** We distill from two pretrained ResNet-50 SSL models, BYOL and SwAV to ResNet-18 students. When distilled from these stronger teacher networks, SimReg is significantly better than both CompRess variants on all metrics. Both SimReg and CompRess contain MLP head only during training.

Arch	ResNet-50	MobileNet-v2			ResNet-18		
Method	Teacher	Comp-2q -4L-MLP	Comp-1q -4L-MLP	SimReg -4L-MLP	Comp-2q -4L-MLP	Comp-1q -4L-MLP	SimReg -4L-MLP
Food	72.3	71.4	72.5	**73.1**	61.7	**65.9**	65.4
CIFAR10	92.2	90.3	90.4	**91.2**	87.3	**89.3**	88.6
CIFAR100	75.1	73.9	74.5	**76.1**	68.4	**71.9**	70.2
SUN	60.2	58.0	58.1	**59.4**	54.3	56.0	**57.1**
Cars	50.8	60.3	**63.1**	62.4	37.2	**44.1**	42.3
Aircraft	53.5	57.7	**59.7**	58.7	42.3	**47.8**	45.8
DTD	75.1	71.7	71.3	**74.5**	69.3	**71.2**	70.9
Pets	83.6	**86.7**	86.3	85.6	84.0	**84.4**	83.9
Caltech	89.3	91.1	91.5	**91.7**	87.3	**90.1**	89.2
Flowers	91.3	94.3	**95.4**	95.1	86.4	**91.3**	90.9
Mean	74.3	75.5	76.3	**76.8**	67.8	**71.2**	70.4

Table 5: **Transfer learning results on multiple classification tasks.** Since the teacher networks are self-supervised, generalization of learnt features to other datasets is important. SimReg is significantly better than CompRess-2q and comparable to CompRess-1q on most datasets. All methods employ 4 layer MLP heads only during distillation.

18 and ResNet-50 students. Our observation generalizes to similarity based distillation too. Use of MLP prediction head also consistently improves the classification performance of both the CompRess variants (table 3). Note that the linear metrics of our student model on MobileNet-v2 and ResNet-50 are just 1.7 and 1.4 points below the corresponding teacher accuracies. Our ResNet-50 model distilled from SimCLR teacher outperforms a ResNet-50 model trained from scratch using SimCLR (69.3% [10]) by **4.9** points.

Transfer Learning: Since an important goal of self-supervised learning is to learn models that generalize well to new datasets and tasks, we evaluate the transfer learning performance of our distilled networks. The results in table 5 suggest that the regression model transfers as well as or better than the state-of-the-approaches on most datasets. Among CompRess variants, CompRess-2q-MLP is generally better on ImageNet classification (table 3) but transfers poorly (table 5) compared to CompRess-1q-MLP. However, the same SimReg model performs comparably or outperforms them both in ImageNet and transfer tasks.

In addition to transfer learning on different datasets on the classification task, we consider

Method	AP_{50}	AP	AP_{75}
SimReg-4L-MLP	74.0	45.4	47.8
SimReg-2L-MLP	74.2	45.5	47.4
SimReg-Linear	73.6	45.1	47.9

Table 6: **Transfer learning for object detection on Pascal VOC dataset.** Student models with different MLP head architectures are used to perform distillation on ImageNet dataset and the backbone with R18-C4 architecture is fine-tuned on PASCAL VOC. Unlike in classification tasks, the performance of different distillation architectures is nearly identical.

Aug Type	Aug Strength		ImageNet			Transfer
	Teacher	Student	1-NN	20-NN	Linear	Linear
Same	Weak	Weak	**54.8**	**59.9**	**65.1**	**70.4**
Same	Strong	Strong	53.4	59.0	64.3	70.3
Different	Weak	Weak	54.7	59.8	64.6	70.0
Different	Weak	Strong	51.1	56.5	62.0	68.9
Different	Strong	Strong	50.3	56.0	61.4	68.7

Table 7: **Role of augmentation strength:** During distillation either the ‘same’ augmented image or two ‘different’ augmentations of a single image are used as inputs to the teacher and student networks. The augmentations strength is varied for both the settings. We find that the performance is best when the same image with weak augmentation is used. This is significant since using different and stronger augmentations improve classification performance of SSL models but decrease their generalizability [42].

the task of object detection. We distill a ResNet-50 teacher to multiple ResNet-18 student networks with different MLP heads. The MLP heads are not part of the model fine-tuned for detection. Following [21], we use Faster-RCNN [43] with `R18-C4` backbone architecture for the detection task. All methods are trained on the VOC `trainval07+12` and tested on `test07` subsets of the PASCAL VOC [15] dataset using the code from [2]. We report the standard AP_{50}, AP(COCO-style) and AP_{75} metrics in table 6. Unlike in classification tasks, we find that the use of deeper MLP heads during distillation does not aid detection performance. The performance of different distillation architectures is nearly identical on the detection task.

Effect of Data Augmentation: As shown in SEED [16] and CompRess [29], for a given student architecture, distillation from larger models trained using a particular method is better than directly training the student using the same method. Use of different and strong augmentations in contrastive SSL approaches has been shown to hurt generalization performance [42]. Here, we show that when distilling models, the best performance is obtained when the same augmented image with a weaker augmentation (details in suppl.) is used as input to both teacher and student networks (table 7). This suggests that our method can be used to improve generalizability of SSL models.

Multi-teacher Distillation: We train a single student model from multiple teacher networks trained with different SSL methods. Regression with a 4 layer MLP head significantly outperforms one with linear prediction (table 8).

Training			ResNet-18+Linear			ResNet-18+4L-MLP		
Inference			ResNet-18			ResNet-18		
MoCo-v2	BYOL	SwAV	1-NN	20-NN	Linear	1-NN	20-NN	Linear
✓	✓		50.9	56.5	62.6	56.4	61.3	66.3
✓		✓	49.9	55.6	61.8	55.1	60.4	65.4
	✓	✓	52.0	57.7	64.8	56.5	61.4	67.0
✓	✓	✓	51.1	56.7	63.2	56.5	61.6	66.9

Table 8: **Multi-teacher distillation on ImageNet.** We train a single student model (ResNet-18) from multiple SSL teacher networks (ResNet-50) using a common backbone network and a separate prediction head for each teacher. Networks with 4 layer prediction heads can better match each of the teachers and thus vastly outperform those with a linear head on both k-NN and linear evaluation metrics.

Teacher	Student Arch (Inference)	Prediction Head (Train)	1-NN	20-NN	Linear
Supervised ResNet-50	MobileNet-v2 Backbone	4L-MLP	63.77	67.87	**73.5**
		2L-MLP	**64.7**	**69.3**	**73.5**
		Linear	55.4	62.0	67.5

Table 9: **Distillation of supervised teacher.** Here, we analyze the role of MLP heads when distilling the features of a teacher trained using supervision. Note that the distillation remains unsupervised and only the backbone CNN features are regressed. Similar to distillation of SSL teachers, we observe that the use of a deep MLP head during training significantly improves classification performance on ImageNet classification task.

Distillation of Supervised Teacher: All the previous teacher networks were trained in a self-supervised manner. We additionally analyze the distillation from a teacher trained with supervision (table 9). Note that the distillation remains unsupervised and only the backbone CNN features of the teacher are regressed. Similar to distillation of self-supervised teachers, we observe that the use of a deep MLP head during training significantly improves performance on the ImageNet classification task.

6 Conclusion

Distilling knowledge with deeper student networks leads to better downstream performance. We surprisingly find that intermediate layer outputs of a distilled student model have better performance compared to the final layer, though final layer is trained to mimic the teacher representations. Thus, we use a prediction MLP head only for optimizing the distillation objective and achieve boosts in performance with just the backbone network during inference. We believe studying the reasoning for this effect is an interesting future work. Our work also serves as an improved benchmark for future self-supervised distillation works. Additionally, we show that using the same weakly augmented image for both teacher and student aids distillation.

Acknowledgment: This material is based upon work partially supported by the United States Air Force under Contract No. FA8750-19-C-0098, funding from SAP SE, and also NSF grant numbers 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] A library for efficient similarity search and clustering of dense vectors. https://github.com/facebookresearch/faiss.

[2] Official pytorch code base for moco. https://github.com/facebookresearch/moco, .

[3] Official pytorch resnet-50 pretrained model. https://pytorch.org/vision/stable/models.html, .

[4] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. *NeurIPS*, 2019.

[5] Lucas Beyer, Xiaohua Zhai, Amélie Royer, Larisa Markeeva, Rohan Anil, and Alexander Kolesnikov. Knowledge distillation: A good teacher is patient and consistent. *arXiv preprint arXiv:2106.05237*, 2021.

[6] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *European conference on computer vision*, pages 446–461. Springer, 2014.

[7] Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541, 2006.

[8] 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)*, pages 132–149, 2018.

[9] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *NeurIPS*, 2020.

[10] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[11] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021.

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

[13] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3606–3613, 2014.

[14] 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*, pages 1422–1430, 2015.

[15] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[16] Zhiyuan Fang, Jianfeng Wang, Lijuan Wang, Lei Zhang, Yezhou Yang, and Zicheng Liu. Seed: Self-supervised distillation for visual representation. *ICLR*, 2021.

[17] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 conference on computer vision and pattern recognition workshop*, pages 178–178. IEEE, 2004.

[18] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *ICLR*, 2018.

[19] 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, et al. Bootstrap your own latent: A new approach to self-supervised learning. *NeurIPS*, 2020.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[21] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

[22] Byeongho Heo, Minsik Lee, Sangdoo Yun, and Jin Young Choi. Knowledge transfer via distillation of activation boundaries formed by hidden neurons. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3779–3787, 2019.

[23] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *NeurIPS Workshop*, 2014.

[24] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[25] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[26] Yanping Huang, Youlong Cheng, Ankur Bapna, Orhan Firat, Dehao Chen, Mia Chen, HyoukJoong Lee, Jiquan Ngiam, Quoc V Le, Yonghui Wu, et al. Gpipe: Efficient training of giant neural networks using pipeline parallelism. *Advances in neural information processing systems*, 32:103–112, 2019.

[27] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

[28] Nikos Komodakis and Sergey Zagoruyko. Paying more attention to attention: improving the performance of convolutional neural networks via attention transfer. In *ICLR*, 2017.

[29] Soroush Abbasi Koohpayegani, Ajinkya Tejankar, and Hamed Pirsiavash. Compress: Self-supervised learning by compressing representations. *NeurIPS*, 2020.

[30] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013.

[31] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[32] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

[33] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6707–6717, 2020.

[34] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729. IEEE, 2008.

[35] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pages 69–84. Springer, 2016.

[36] Mehdi Noroozi, Hamed Pirsiavash, and Paolo Favaro. Representation learning by learning to count. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5898–5906, 2017.

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

[38] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3967–3976, 2019.

[39] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3498–3505. IEEE, 2012.

[40] Nikolaos Passalis and Anastasios Tefas. Learning deep representations with probabilistic knowledge transfer. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 268–284, 2018.

[41] 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*, pages 2536–2544, 2016.

[42] Senthil Purushwalkam Shiva Prakash and Abhinav Gupta. Demystifying contrastive self-supervised learning: Invariances, augmentations and dataset biases. *Advances in Neural Information Processing Systems*, 33, 2020.

[43] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28:91–99, 2015.

[44] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *ICLR*, 2015.

[45] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115 (3):211–252, 2015.

[46] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[47] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019.

[48] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019.

[49] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. *ICLR*, 2020.

[50] Guo-Hua Wang, Yifan Ge, and Jianxin Wu. In defense of feature mimicking for knowledge distillation. *arXiv preprint arXiv:2011.01424*, 2020.

[51] Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3485–3492, 2010. doi: 10.1109/CVPR.2010.5539970.

[52] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *British Machine Vision Conference 2016*. British Machine Vision Association, 2016.