

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

K L Navaneet<sup>1</sup>  
navanek1@umbc.edu

Soroush Abbasi Koohpayegani<sup>1</sup>  
soroush@umbc.edu

Ajinkya Tejankar<sup>1</sup>  
at6@umbc.edu

Hamed Pirsiavash<sup>1, 2</sup>  
hpirsiav@ucdavis.edu

<sup>1</sup> University of Maryland, Baltimore  
County  
Maryland, USA

<sup>2</sup> University of California, Davis  
California, USA

---

In this supplementary material, we present additional experimental results (Sec. 1) and details on experiment settings and implementation (Sec. 2). Additional results include those on the role of MLP head during training (Sec. 1.1) and self-distillation (Sec. 1.2). We publicly release the code<sup>1</sup>.

## 1 Additional Experimental Results

### 1.1 Role of MLP Head

In tables 1 and 2 of main we analyze how the depth of MLP head during training and inference affects classification performance. We present additional results here in table 1 with different teacher and student network settings. The student networks are trained with different prediction head configurations. The evaluation is always performed using features from backbone network for a fair comparison. In addition to the self-supervised (SSL) teacher models used in the main paper, we consider a supervised teacher network. The teacher is trained with cross-entropy loss using ground truth labels on the ImageNet dataset. As in SSL teachers, we use only the backbone network for distillation from a supervised teacher. Note that the supervised labels are absent during student training. In both the supervised and self-supervised settings, the student with 4 layer MLP head consistently outperforms others on all metrics. **Compared to Linear head, 4L-MLP achieves 5 (MoCo-v2, ResNet-18), 11.2 (MoCo-v2, MobileNet-v2) and 6 (Supervised, MobileNet-v2) percentage points improvement on linear evaluation.**

Teacher	Student Arch (Inference)	Prediction Head (Train)	1-NN	20-NN	Linear
MoCo-v2 ResNet-50	ResNet-18 Backbone	4L-MLP	<b>54.8</b>	<b>59.9</b>	<b>65.1</b>
		2L-MLP	52.7	58.5	63.6
		Linear	48.8	54.3	60.1
MoCo-v2 ResNet-50	MobileNet-v2 Backbone	4L-MLP	<b>55.46</b>	<b>59.73</b>	<b>69.1</b>
		2L-MLP	54.4	59.6	68.5
		Linear	48.7	54.2	57.9
Supervised ResNet-50	MobileNet-v2 Backbone	4L-MLP	63.77	67.87	<b>73.5</b>
		2L-MLP	<b>64.7</b>	<b>69.3</b>	<b>73.5</b>
		Linear	55.4	62.0	67.5

Table 1: **Effect of MLP Heads on ImageNet classification performance.** As in table 1 of main paper, we analyze the role of the prediction head used during training by varying the number of MLP layers. However, the evaluation here is performed using the features from the backbone network and the prediction head plays no role during inference. A linear prediction head corresponds to the architecture used in earlier works [5]. We observe that a deeper prediction module during training results in substantial boosts in performance. This observation is consistent across different teacher networks (both SSL and supervised) and student architectures. **Compared to Linear head, 4L-MLP achieves 5(MoCo-v2, ResNet-18), 11.2(MoCo-v2, MobileNet-v2) and 6(Supervised, MobileNet-v2) percentage points improvement on linear evaluation.**

## 1.2 Self-distillation

In all the previous experiments, a larger teacher network is distilled to a shallower student. In self-distillation, we consider the same backbone architecture for both teacher and student. Similar to other experiments, we use a prediction head (linear or MLP) atop student backbone during distillation and remove it during evaluation. As we observe in table 2, the student with a 4 layer MLP head outperforms the teacher in both ImageNet classification and transfer tasks. The improvement in transfer performance is particularly significant (+4 percentage points) and might be attributed to the use of prediction head and weaker augmentations during distillation.

## 1.3 Comparison with CompRes without MLP

In table 1 of the main paper, we observed that the use of MLP head during distillation benefits both the CompRes variants on the ImageNet classification task. Here, we show that similar boost in CompRes performance can be achieved on transfer tasks when distilled with MLP head. We use the officially provided pretrained models for vanilla CompRes-2q ResNet-18 and MobileNet-v2 architectures for our comparison and perform transfer analysis similar to that in table 5 of the main paper. Results in table 3 demonstrate that performance of vanilla CompRes models are significantly worse compared to both CompRes with MLP and proposed regression based distillation. Note that the MLP heads are not used during inference for fair comparison.

Student Arch (Inference)	Prediction Head (Train)	ImageNet			Transfer
		1-NN	20-NN	Linear	Linear
MoCo-v2 Teacher	-	57.3	60.9	70.8	74.3
ResNet-50	4L-MLP	<b>58.2</b>	<b>62.2</b>	<b>72.0</b>	<b>78.3</b>
ResNet-50	Linear	56.4	60.6	69.7	71.8

Table 2: **ImageNet Classification and transfer results for self-distillation with prediction head.** In self-distillation, the teacher and student backbone architectures are the same (ResNet-50). We use a MoCo-v2 pretrained teacher and train student networks with linear and 4 layer MLP heads. All evaluations are performed using backbone features. The student with MLP prediction head outperforms the teacher on both ImageNet classification and transfer tasks. A boost of 4 percentage points on average transfer accuracy suggests that the use of prediction head and weaker augmentations during distillation are beneficial in learning a good generalizable model.

Arch Method	MobileNet-v2			ResNet-18		
	CompRess-2q plain	CompRess-2q -4L-MLP	SimReg -4L-MLP	CompRess-2q plain	CompRess-2q -4L-MLP	SimReg -4L-MLP
Food	61.4	71.4	<b>73.1</b>	57.6	61.7	<b>65.4</b>
CIFAR10	85.3	90.3	<b>91.2</b>	82.5	87.3	<b>88.6</b>
CIFAR100	65.1	73.9	<b>76.1</b>	62.5	68.4	<b>70.2</b>
SUN	53.9	58.0	<b>59.4</b>	52.2	54.3	<b>57.1</b>
Cars	35.0	60.3	<b>62.4</b>	30.0	37.2	<b>42.3</b>
Aircraft	42.1	57.7	<b>58.7</b>	38.0	42.3	<b>45.8</b>
DTD	70.4	71.7	<b>74.5</b>	67.4	69.3	<b>70.9</b>
Pets	82.9	<b>86.7</b>	85.6	81.6	<b>84.0</b>	83.9
Caltech	85.6	91.1	<b>91.7</b>	85.3	87.3	<b>89.2</b>
Flowers	87.6	94.3	<b>95.1</b>	83.0	86.4	<b>90.9</b>

Table 3: **Transfer learning performance of CompRess with and without MLP.** Since the teacher networks are self-supervised, generalization of learnt features to other datasets is important. Similar to ImageNet classification, CompRess with MLP significantly outperforms vanilla CompRess (CompRess-2q plain) on all datasets and metrics. MLP heads, if present, are only used during distillation and are not part of inference networks.

Eval Layer Student	Conv-1		ResBlk-1		ResBlk-2		ResBlk-3		ResBlk-4		2L-MLP		4L-MLP	
	1-NN	20-NN	1-NN	20-NN	1-NN	20-NN	1-NN	20-NN	1-NN	20-NN	1-NN	20-NN	1-NN	20-NN
ResNet-18	6.2	7.2	16.0	18.0	21.8	24.3	33.4	37.4	55.3	60.2	56.0	60.9	53.4	57.6

Table 4: **ImageNet classification using intermediate features.** We consider a single student network with 4 layer MLP head and perform k-NN evaluation using features from various intermediate layer features from the network. For fair comparison, we match the dimensions from the intermediate convolutional features (Conv-1 and residual block features) to that of the final backbone feature (ResBlk-4) by reducing their spatial dimensions. As expected, performance improves as we use features from deeper layers of the CNN. This changes in the MLP head where a drop in accuracy at the very last layer of the prediction head is observed.

## 1.4 Results with Intermediate Layers of CNN

In our results in table 2 of main paper, we analyze how the classification performance changes as we consider features from the earlier layers of the prediction head. Here, we analyze results from various intermediate layers including those from the CNN backbone. We train a single ResNet-18 student from a MoCo-v2 ResNet-50 teacher and perform k-NN evaluation using features from different layers. In table 4, Conv-1 refers to the output of the first convolutional layer while ResBlk-j refers to the output from the  $j^{th}$  residual block. The CNN features for evaluation are obtained by reducing their spatial dimension and then vectorizing. The spatial dimensions are reduced so that the feature lengths are roughly the same throughout the backbone for fair comparison. We observe that the performance increases as we go deeper into the backbone. The best performance is achieved at the intermediate layer of prediction head and there is a small drop in accuracy at the final prediction layer.

# 2 Implementation Details

## 2.1 Teacher Networks

We use teacher networks trained using four different self-supervised representation learning approaches - MoCo-v2 [1], BYOL [2], SwAV [3] and SimCLR [4]. We use the official and publicly available pre-trained weights for these networks with ResNet-50x4 architecture pretrained model for SimCLR teacher and ResNet-50 models for the remaining methods. MoCo-v2 and SwAV have been trained for 800 epochs and BYOL and SimCLR for 1000 epochs. For distillation with BYOL, SwAV and SimCLR teachers we use cached features from the teacher. The cached features are obtained by passing the entire training data through the teacher network once and storing the features. Random image augmentation as would be used in non-cached version is employed to generate the inputs for caching.

## 2.2 Image Augmentations

We use two strategies for augmenting the input image during distillation - ‘weak’ and ‘strong’. ‘Strong’ augmentation refers to the setting used in MoCo-v2 [1]. In both augmentation settings, we apply a series of stochastic transformations on the input image. A random resized crop (scale is in range [0.2, 1.]), random horizontal flip with probability 0.5 and normalization to channel-wise zero mean and unit variance are common for both augmentation methods. In addition to these transformations, ‘strong’ augmentations use random color jittering (strength of 0.4 for brightness, contrast and saturation and 0.1 for hue) with probability

0.8, random grayscaling with probability 0.2 and Gaussian blur (standard deviation chosen uniformly from  $[0, 1]$ ).

## 2.3 Optimizer

In all our distillation experiments, we use SGD optimizer with cosine scheduling of learning rate, momentum of 0.9 and weight decay of 0.0001. Initial learning rate is set to 0.05. The networks are trained for 130 epochs with a batch size of 256 using PyTorch [1] framework.

## 2.4 Evaluation Metrics

We utilize k-NN and linear evaluation to evaluate classification performance on ImageNet and linear evaluation to evaluate transfer performance. For ImageNet linear evaluation, 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 [2]. The layer is trained for 40 epochs using SGD with initial learning rate of 0.01 and momentum of 0.9. The learning rate is scaled by 0.1 at epochs 15 and 30. For evaluation of transfer performance, we use the optimizer settings from [2]. The shorter side of the input image is resized to 256 and centre crop with length 224 is used. The input is channel-wise normalized using the statistics from ImageNet dataset. We use LBFGS optimizer with parameters `max_iter=20` and `history_size=10`. Learning rate and weight decay are optimized by performing a grid search using validation set. The best model is obtained by retraining with optimal parameters on the combined train and validation set. 10 different log spaced values in  $[-3, 0]$  are used for learning rate while 9 log values in  $[-10, -2]$  are used for weight decay.

## 2.5 MLP Architecture

For the proposed prediction head, we experiment with linear, 2 and 4 layer MLPs. Each MLP layer is composed of a linear projection followed by 1D batch normalization and ReLU activation. Let the dimension of the student backbone output be  $m$  and that of teacher  $d$ . For linear evaluation, a single layer with input and output dimensions of  $(m, d)$  is used. For a 2 layer MLP, following [2], we use the dimensions  $(m, 2m, d)$ . We extend this to a 4 layer MLP with the following intermediate feature dimensions:  $(m, 2m, m, 2m, d)$ . Batch normalization and ReLU activation are not employed at the end of layer 2 for 4 layer MLP head (equivalent to stacking two 2-layer MLP heads). For our ablation on the role of MLP head during inference (table 2 in main paper), we compare the performance of our method at different layers of the MLP head from a single trained network. For fair comparison, we require all the intermediate dimensions to be same as that of the output. Thus, for this experiment alone, we use an MLP such that the feature dimensions are  $(m, d, d, d, d)$ . The output dimension ( $m$ ) for ResNet-18, ResNet-50 and MobileNet-v2 are 512, 2048 and 1280 respectively. The teacher output dimensions are 2048 and 8192 respectively for ResNet-50 and ResNet-50x4 architectures. From table 2 (network with MLP feature dimensions (512, 2048, 2048, 2048, 2048)) and table 4 (network with MLP feature dimensions (512, 1024, 512, 1024, 2048)) results, we observe that higher MLP feature dimensions might help further boost performance (65.7 vs 65.1 on ImageNet linear). More ablations on this are necessary to optimize the MLP architecture.

## References

- [1] 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.
- [2] 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.
- [3] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.
- [4] 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.
- [5] Soroush Abbasi Koohpayegani, Ajinkya Tejankar, and Hamed Pirsiavash. Compress: Self-supervised learning by compressing representations. *NeurIPS*, 2020.
- [6] Soroush Abbasi Koohpayegani, Ajinkya Tejankar, and Hamed Pirsiavash. Mean shift for self-supervised learning. *arXiv preprint arXiv:2105.07269*, 2021.
- [7] 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 32*, pages 8024–8035. 2019.